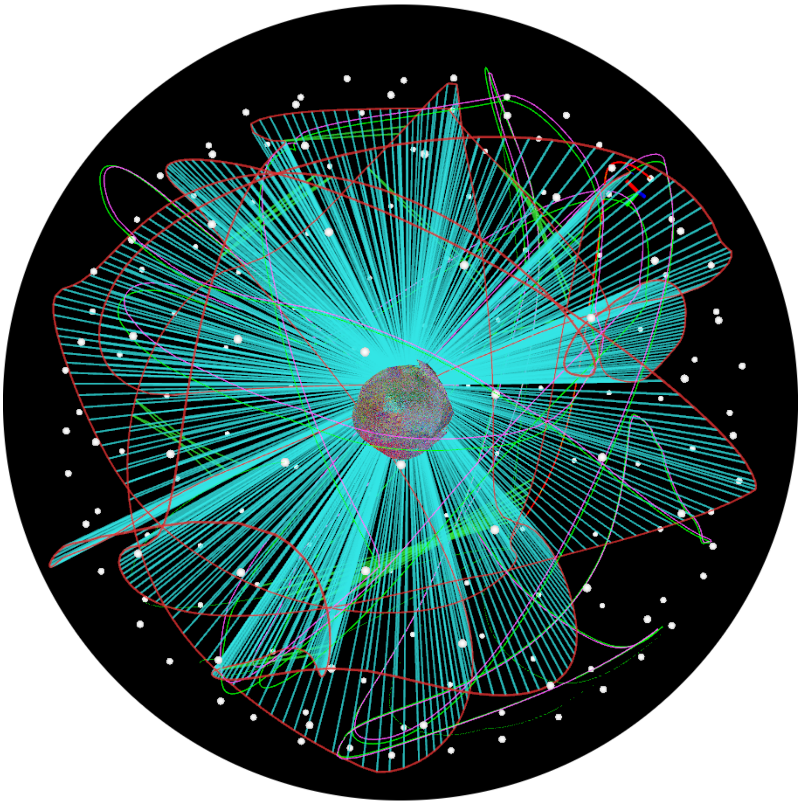
Active Perception for Autonomous Systems

In a Deep Space Navigation Scenario



David Nakath

Januar 2019

Kumulative Dissertation zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften
– Dr.-Ing. –

Universität Bremen

Universität Bremen – Fachbereich 3 Mathematik/Informatik

Title Figure: Active Asteroid Graph SLAM in the vicinity of the asteroid Itokawa displayed in the ROS-visualization tool RViz. In the center, the scan map of the asteroid can be seen which is comprised of randomly color coded point clouds. The white dots are discrete points on a tessellated sphere used to evaluate trajectories. The ground truth trajectory is displayed as a green line, while the corresponding trajectory estimate is shown as a pink line. The graph structure is comprised of red edges denoting pose constraints and cyan edges respectively denoting attitude constraints. In addition, the green edges within the graph structure denote pose constraints obtained by loop closing. Figure adopted from Nakath et al. [2019].

Active Perception for Autonomous Systems

In a Deep Space Navigation Scenario

David Nakath

Date of Submission
January 24, 2019

Date of Defense
April 24, 2019

Reviewers
Prof. Dr. Kerstin Schill, University of Bremen, Germany
Prof. Dr. Udo Frese, University of Bremen, Germany

Abstract

Autonomous systems typically pursue certain goals for an extended amount of time in a self-sustainable fashion. To this end, they are equipped with a set of sensors and actuators to perceive certain aspects of the world and thereupon manipulate it in accordance with some given goals. This kind of interaction can be thought of as a closed loop in which a perceive-reason-act process takes place. The bi-directional interface between an autonomous system and the outer world is then given by a sequence of imperfect observations of the world and corresponding controls which are as well imperfectly actuated. To be able to reason in such a setting, it is customary for an autonomous system to maintain a probabilistic state estimate. The quality of the estimate – or its uncertainty – is, in turn, dependent on the information acquired within the perceive-reason-act loop described above. Hence, this thesis strives to investigate the question of how to actively steer such a process in order to maximize the quality of the state estimate.

The question will be approached by introducing different probabilistic state estimation schemes jointly working on a manifold-based encapsuled state representation. On top of the resultant state estimate different active perception approaches are introduced, which determine optimal actions with respect to uncertainty minimization. The informational value of the particular actions is given by the expected impact of measurements on the uncertainty. The latter can be obtained by different direct and indirect measures, which will be introduced and discussed.

The active perception schemes for autonomous systems will be investigated with a focus on two specific deep space navigation scenarios deduced from a potential mining mission to the main asteroid belt. In the first scenario, active perception strategies are proposed, which foster the correctional value of the sensor information acquired within a heliocentric navigation approach. Here, the expected impact of measurements is directly estimated, thus omitting counterfactual updates of the state based on hypothetical actions. Numerical evaluations of this scenario show that active perception is beneficial, i.e., the quality of the state estimate is increased. In addition, it is shown that the more uncertain a state estimate is, the more the value of active perception increases. In the second scenario, active autonomous deep space navigation in the vicinity of asteroids is investigated. A trajectory and a map are jointly estimated by a Graph SLAM algorithm based on measurements of a 3D Flash-LiDAR. The active perception strategy seeks to trade-off the exploration of the asteroid against the localization performance. To this end, trajectories are generated as well as evaluated in a novel twofold approach specifically tailored to the scenario. Finally, the position uncertainty can be extracted from the graph structure and subsequently be used to dynamically control the trade-off between localization and exploration. In a numerical evaluation, it is shown that the localization performance of the Graph SLAM approach to navigation in the vicinity of asteroids is generally high. Furthermore, the active perception strategy is able to trade-off between localization performance and the degree of exploration of the asteroid. Finally, when the latter process is dynamically controlled, based on the current localization uncertainty, a joint improvement of localization as well as exploration performance can be achieved.

In addition, this thesis comprises an excursion into active sensorimotor object recognition. A sensorimotor feature is derived from biological principles of the human perceptual system. This feature is then employed in different probabilistic classification schemes. Furthermore, it enables the implementation of an active perception strategy, which can be thought of as a feature selection process in a classification scheme. It is shown that those strategies might be driven by top-down factors, i.e., based on previously learned information, or by bottom-up factors, i.e., based on saliency detected in the currently considered data. Evaluations are conducted based on real data acquired by a camera mounted on a robotic arm as well as on datasets. It is shown that the integrated representation of perception and action fosters classification performance and that the application of an active perception strategy accelerates the classification process.

Kurzfassung

Die vorliegende Dissertation befasst sich mit der aktiven Wahrnehmung von autonomen Systemen. Letztere können über längere Zeiträume hinweg autark bestimmte Ziele verfolgen. Um ein solches Verhalten zu ermöglichen, sind autonome Systeme typischerweise mit Sensoren und Aktuatoren ausgestattet. So können sie die Welt wahrnehmen und diese dann hinsichtlich bestimmter Ziele manipulieren. Ein solcher perceive-reason-act Vorgang kann als closed-loop Szenario modelliert werden, bei dem eine bi-direktionale Schnittstelle die Außenwelt und das autonome System verbindet. Diese besteht zum einen aus einer Sequenz von unvollkommenen Beobachtungen der Außenwelt, da diese lediglich imperfekt, also ausschnittsweise und mit Hilfe von verrauschten Sensordaten, beschrieben werden kann. Zum anderen besteht sie aus Steuerbefehlen, die von den Aktuatoren ebenfalls nur ungenau umgesetzt werden. Damit ein autonomes System in einem solchen Szenario erfolgreich Ziele verfolgen kann, wird typischerweise eine probabilistische Zustandsschätzung durchgeführt. Die Qualität einer solchen Schätzung – also ihre Unsicherheit – ist wiederum abhängig von den oben erwähnten sequentiellen Beobachtungen der Außenwelt. Daher befasst sich diese Dissertation mit der Frage, wie der Beobachtungsprozess gesteuert werden sollte, sodass die Qualität der Zustandsschätzung maximiert wird.

Diese Frage wird angegangen, indem zunächst verschiedene Methoden der probabilistischen Zustandsschätzung vorgestellt werden. Diese haben gemeinsam, dass sie mit Zuständen arbeiten, die gekapselt auf einer Mannigfaltigkeit repräsentiert werden. Weiterhin werden Ansätze zur aktiven Wahrnehmung vorgestellt, deren Bestreben es ist die Unsicherheit in der Zustandsschätzung durch bestimmte Aktionen zu minimieren. Der Informationswert einzelner Aktionen ist durch ihren erwarteten Einfluss auf die Unsicherheit gegeben. Um diesen messbar zu machen, werden verschiedene direkte als auch indirekte Methoden vorgestellt und diskutiert.

Die Ansätze zur aktiven Wahrnehmung für autonome Systeme werden mit einem besonderen Fokus auf autonome Navigation im Weltraum untersucht, wobei zwei konkrete Szenarien in Betracht gezogen werden. Zum einen die heliozentrische Navigation relativ zur Sonne und zum anderen die asteroidenzentrische Navigation, am Beispiel des Asteroiden Itokawa. Im ersten Szenario werden Strategien zur aktiven Wahrnehmung innerhalb eines autonomen optischen Navigationsverfahrens vorgestellt, die das korrektive Potential der gesammelten Sensordaten fördern. Dazu wird der erwartete Einfluss von Sensormessungen auf die Unsicherheit direkt geschätzt. Somit kann es vermieden werden, verschiedene Verteilungen zu vergleichen die aufgrund von kontrafaktischen, also aus hypothetischen Aktionen resultierenden, Sensormessungen korrigiert wurden. Numerische Evaluationen zeigen, dass die Strategien zur aktiven Wahrnehmung die Qualität der Zustandsschätzung signifikant erhöhen. Zusätzlich kann gezeigt werden, dass der Effekt der aktiven Wahrnehmung größer wird, je mehr Unsicherheit in den Schätzungen vorhanden ist. Im zweiten Szenario werden eine Trajektorie wie auch eine Karte gemeinsam geschätzt. Dazu wird ein Graph SLAM-

Ansatz genutzt, der sich im wesentlichen auf Punktwolken stützt, die durch einen 3D Flash-LiDAR erzeugt werden. Die Strategie zur aktiven Wahrnehmung versucht den Zielkonflikt zwischen der Exploration des Asteroiden und der Lokalisierungsgenauigkeit messbar und damit steuerbar zu machen. Um dies zu erreichen, werden zunächst approximierte und dann optimale Trajektorien in einer Evaluationsstruktur bewertet, die auf das spezifische Szenario zugeschnitten ist. Zusätzlich ist es möglich, diesen Prozess aufgrund der in der Graphstruktur gemessenen Unsicherheit dynamisch zu kontrollieren. Numerische Evaluationen dieses Ansatzes zeigen eine gute Lokalisierungsqualität des Graph SLAM-Ansatzes in der Nähe des Asteroiden Itokawa. Weiterhin ist es möglich, gezielt Trajektorien zu generieren und daraufhin solche zu wählen, die entweder die Lokalisierungsgenauigkeit des Raumschiffes fördern oder aber die Exploration des Asteroiden. Wenn dieser Prozess dynamisch, auf Grundlage der im Graphen gemessenen Unsicherheit, gesteuert wird, wirkt sich das Vorteilhaft sowohl auf die Lokalisierungsgenauigkeit wie auch auf die Exploration des Asteroiden aus.

Im Rahmen eines Exkurses wird ein Ansatz zur aktiven sensormotorischen Objekterkennung präsentiert. Dazu wird zunächst ein sensomotorisches Feature vorgestellt, welches eine integrierte Repräsentation von Wahrnehmung und Aktion darstellt. Es wurde auf Basis von biologisch-plausiblen Prinzipien, das menschliche visuelle System beschreibend, entwickelt. Dieses Feature wird dann zur aktiven Objekterkennung mit Hilfe von verschiedenen probabilistischen Klassifikatoren eingesetzt. Zusätzlich ermöglicht es, aktive Wahrnehmungsstrategien zu implementieren, die bestimmte Features zur Klassifikation auswählen. Die Auswahl kann entweder top-down – also aus der Erfahrung heraus – oder bottom-up – also aus den aktuellen Daten heraus – getrieben sein. Zwei wesentliche Ergebnisse konnten in Evaluationen erzielt werden, die mit Bilddatensätzen durchgeführt wurden, als auch mit einem Robotarm, der eine Kamera führt. Die integrierte Repräsentation von Wahrnehmung und Aktion verbessert die Klassifikationsleistung als solche. Zusätzlich kann diese Repräsentation als Grundlage für Strategien zur aktiven Wahrnehmung genutzt werden, was den Klassifikationsprozess wiederum beschleunigt.

Acknowledgements

I would like to thank Tobias Kluth and Thomas Reineking for the fruitful discussions and guidance throughout my first steps in science. Utmost respect is due to my co-workers Joachim Clemens and Carsten Rachuy for their tireless support and cooperation. Joachim Clemens additionally has to be acknowledged for providing the initial template for this thesis, for some of the images used, for proofreading, and for providing helpful comments.

I would like to thank my co-authors Christof Büskens, Mitja Echim, Bernd Eisfeller, Graciela González Peytaví, Konrad Gadzicki, Patrick Lange, Alena Probst, Anne Schattel, Verena Schwarting, Roger Förstner, Abhishek Srinivas, Gabriel Zachmann, and Christoph Zetzsche. Finally, I would like to thank my thesis advisor Kerstin Schill. I furthermore gratefully acknowledge the support and advise of Udo Frese – especially in the final phase of this thesis.

Mireia Bolíbar Planas and Daniel Martos Nebrera have to be praised for their unequaled hospitality. I would like to thank Dorothea Brückner for the guidance and support while finalizing my thesis. In addition, I am wholeheartedly thankful for proofreading and helpful comments by Nicoletta Momtahen, Hendrik Heuer, and Sylvain Roy. Finally, I would like to thank my parents Ute Nakath and Francis Jipps for their truthful and virtually everlasting support.

Contents

List of Figures

List of Acronyms

ADAS advanced driver assistance systems

au astronomical unit

CCD charge-coupled device

CDF cumulative distribution function

CNN Convolutional Neural Network

COM center of mass

DBN dynamic Bayesian network

DCM direction cosine matrix

DEPTHX Deep Phreatic Thermal Explorer

DLR German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)

DOF degrees of freedom

DSI Deep Space Industries

DSN Deep Space Network

EKF extended Kalman filter

EnEx Enceladus Explorer

ESA European Space Agency

ESTRACK European Space Tracking

FOV field of view

gt ground truth

Hipparcos High Precision Parallax Collecting Satellite

I2D Intrinsic 2-Dimensional Features

ICP Iterative Closest Point

iid independent and identically distributed

IMU inertial measurement unit

ISP specific impulse

ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency
KaNaRiA	Cognition-based, Autonomous Navigation for Deep-Space Resource Mining
KDE	kernel density estimation
KF	Kalman filter
KL	Kullback-Leibler
LEO	low Earth orbit
LiDAR	light detection and ranging
LOS	line of sight
LSTM	Long Short-Term Memory
MAP	maximum a posteriori
MDP	Markov decision process
MPC	model predictive control
MRPT	Mobile Robotics Programming Toolkit
NASA	National Aeronautics and Space Administration
NBNN	Naive Bayes Nearest Neighbor
NDT	Normal Distribution Transform
OpenMP	Open Multi-Processing
OSIRIS-REx	Origins Spectral Interpretation Resource Identification Security – Regolith Explorer
PCA	principal component analysis
PCL	Point Cloud Library
PDF	probability density function
PF	particle filter
PGMs	platinum-group metals
PO	parking orbit
POMDP	partially observable Markov decision process

PTCM Potential Target Characterization Module

RBPF Rao-Blackwellized particle filter

ROS Robot Operating System

RREs rare Earth elements

RViz ROS Robot Visualizer

S/C spacecraft

SIFT scale-invariant feature transform

SLAM Simultaneous Localization and Mapping

SMF sensorimotor feature

SMR sensorimotor representation

SMX Sensorimotor Explorer

SpaceX Space Exploration Technologies Corporation

SURF Speeded Up Robust Features

SVD singular value decomposition

UKF unscented Kalman filter

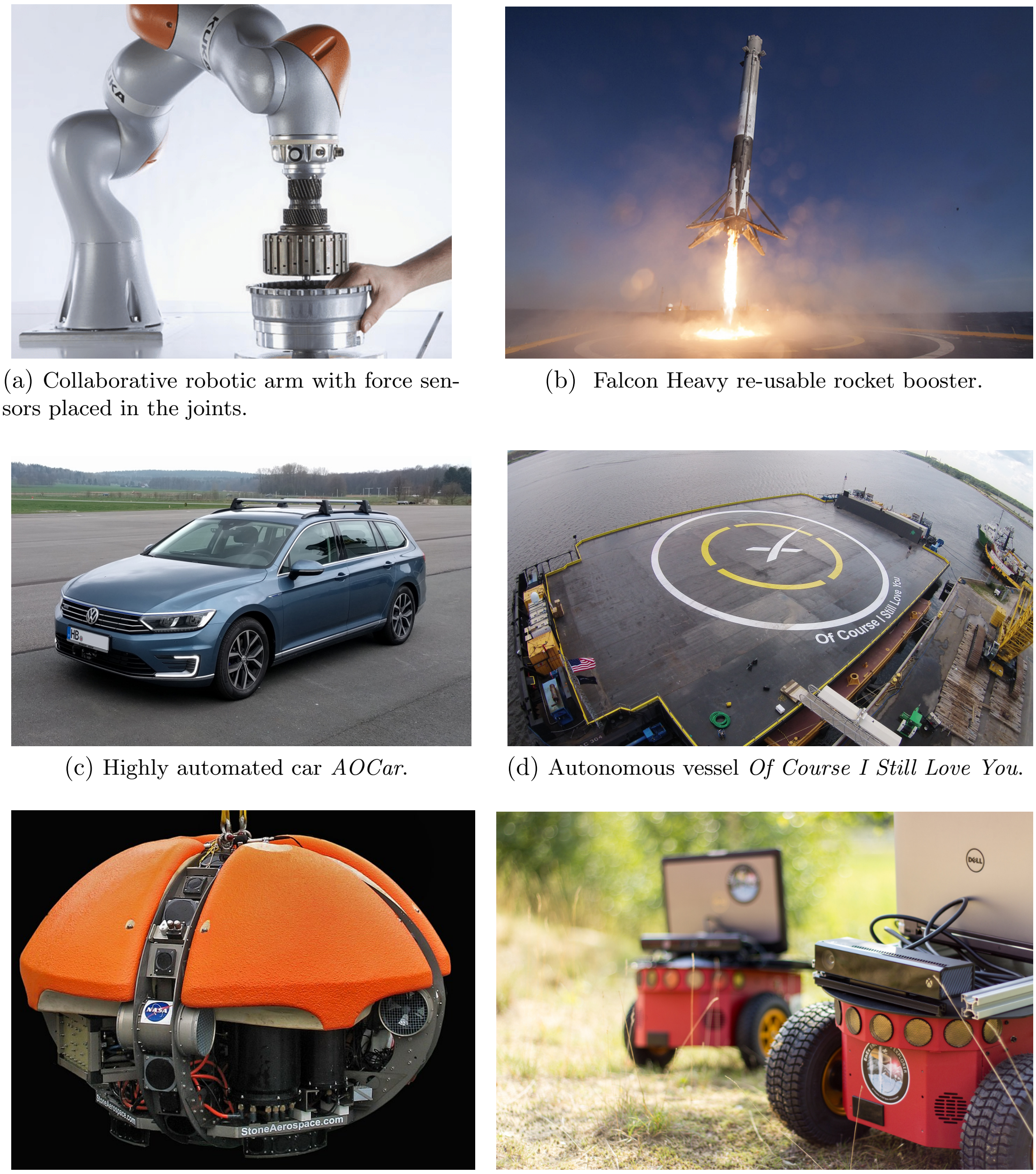
WORHP ESA-NLP solver *We Optimize Really Huge Problems*

1
Introduction

Autonomous systems are able to pursue goals over an extended range of time in a self-sustainable fashion. In general, this category of technical systems has been constantly gaining importance over the last decades. This success rests on a twofold steady development in terms of hardware and software. In the former area, precision, speed, and working ranges of sensors are constantly improving, while their size and weight are decreasing. The same holds for actuators, where smaller and lighter devices are able to execute control commands with higher precision and force. And, of course, the computational power is still increasing at a high rate. The latter, in turn, enables real-time on-board execution of complex algorithms relying on extensive computational capabilities. But also the methodology of the algorithms themselves has improved a lot, benefiting from huge theoretical and practical advances in the fields of sensor fusion, state estimation, high-level reasoning, pattern recognition, and control. The joint evolution of hardware and software thus allows for the implementation of perceive-reason-act loops with ever growing complexity.

Hence, autonomous systems will keep gaining importance throughout their various fields of applications. A prominent example from the industrial field are robotic arms with sensors in the joints, which report the particular acting forces (see Fig. 1.1a) [Haag, 2015]. This capability – inter alia – allows for direct human-robot collaboration in working spaces where no safety-cages are needed anymore. A lot of research is conducted in the field of automotive engineering, ranging from advanced driver assistance systems (ADAS) to actual autonomous driving vehicles. Fig. 1.1c shows a highly automated car, while Fig. 1.1f shows the Pioneer research robots 3-DX and 3-AT, both equipped with ultrasonic range finders, a Kinect RGB-D camera, and a laptop for the on-board computations. The autonomous submarine Deep Phreatic Thermal Explorer (DEPTHX) [Gary et al., 2008], which was designed to explore and map [Fairfield et al., 2007] underwater sinkholes in northern Mexico as well as to autonomously collect samples from their walls is depicted in Fig. 1.1e. A recent example for collaborating autonomous systems are the re-usable rocket boosters of the Falcon Heavy and the landing droneship, going by the name of *Of course I still love you* (see Fig. 1.1d). Both are operated by the commercial space transportation company Space Exploration Technologies Corporation (SpaceX) [Seedhouse, 2013]. The corresponding Fig. 1.1b shows a re-usable part of the *Falcon Heavy* in the final descend over the droneship, after having deployed its payload in space.

Not only on the ground but also for operations in deep space itself the need for au-



(a) Collaborative robotic arm with force sensors placed in the joints.

(b) Falcon Heavy re-usable rocket booster.

(c) Highly automated car *AOCar*.

(d) Autonomous vessel *Of Course I Still Love You*.

(e) Deep Phreatic Thermal Explorer.

(f) Pioneer 3-DX and 3-AT reserach robots.

Figure 1.1.: Examples of autonomous systems and their applications.
Figure (a) reprinted from [Haag, 2015],
Figures (b) and (d) photos by SpaceX, licensed under CC0 1.0 (`https://creativecommons.org/publicdomain/zero/1.0/`),
Figures (c) and (f) courtesy of Joachim Clemens,
Figure (e) photo by Marc Airhart, licensed under CC BY-SA 4.0 (`https://creativecommons.org/licenses/by-sa/4.0/`).

tonomous behavior is constantly increasing. It may be induced by cost considerations, details of the mission design, usage of complex maneuvers, or technical implications. Especially demanding are small-body sample-return missions, as they have to extract samples in deep space and subsequently return them to Earth. Noteworthy missions of that type are Hayabusa I and II, conducted by the Japan Aerospace Exploration Agency (JAXA) in close cooperation with the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt) (DLR), investigating the asteroids Itokawa and Ryugu respectively [Yoshikawa et al., 2006; Tsuda et al., 2013]. In addition, the National Aeronautics and Space Administration (NASA) is currently flying the mission Origins Spectral Interpretation Resource Identification Security – Regolith Explorer (OSIRIS-REx) in an attempt to capture a sample of the asteroid Bennu in a touch-and-go maneuver [Berry et al., 2013].

Autonomous systems may come in the form of, e.g., flying aircraft, driving vehicles, swimming vessels, or diving submarines. Alternatively, they can also take the form of stationary robotic manipulators in factories. While having diverse fields of applications, autonomous systems typically operate on time-series data. Thus, depending on the application, they are equipped with an appropriate set of sensors, actuators, and computational capabilities to acquire and process the latter. This allows them to *perceive* the world in order to establish some imperfect estimation about its state, subsequently to *reason* about this state, and finally to *act* in accordance with the given goals. This perceive-reason-act process between the autonomous system and the outer world can be thought of as a closed loop with a bidirectional interface between them. The latter conveys a sequence of imperfect observations of the world in one direction and a corresponding sequence of imperfectly actuated controls in the opposite direction. To enable reasoning in such a setting, it is customary to maintain some kind of probabilistic state estimate. The quality of such an imperfect estimate, or its uncertainty, is dependent on the information acquired by the sensors in the loop described above. Typically, those sensors only operate in a certain modality, restricted by a field of view (FOV) and are additionally impaired by noise such that only certain aspects of the world can be observed with limited precision at a given time. Thus, depending on how an autonomous system acts, the type and quality of acquired information varies. Naturally, the following question arises: How should the process of active information intake be guided in order to perceive the most informative aspects of the world?

Hence, this thesis seeks to contribute to the field of active perception for autonomous systems by investigating the generation of optimal actions or sequences of actions with respect to the current probabilistic state estimate. The proposed approaches have a special focus on sensor fusion and state estimation within a deep space navigation scenario.

1.1. Outline and Contribution

This introduction will be followed by a description of active perception approaches for autonomous systems from a methodological perspective in Chap. 2. To this end, the class of autonomous systems to be investigated in this thesis is initially defined. Subsequently, the underlying approaches to probabilistic state estimation are described with a special focus on recursive state estimation and Simultaneous Localization and Mapping (SLAM). Furthermore, all state estimation approaches operate on a manifold state representation, which is also briefly introduced. Finally, active perception approaches operating on top of the uncertain state estimates are presented and discussed.

The following Chaps. 3 and 4 seek to convey the underlying ideas of the particular approaches to active perception proposed in this thesis devised in their respective scientific context. To this end, they follow a threefold common structure. First, general related concepts and relevant scientific developments will be discussed. Second, the related work is presented to jointly illustrate the findings which served as a basis for this thesis. And finally, the central ideas of the particular approaches will be presented alongside a brief description of the empirical results. This rather informal proceeding seeks to offer the reader opportunities to follow the given citations leading to the papers attached in Appx. A. The reader should then be prepared to assimilate the more formal and brief descriptions given in the particular papers.

In particular, in Chap. 3 novel active autonomous deep space navigation approaches will be proposed. Initially, a general introduction to deep space missions will be given, with a special focus on scientific missions as well as commercial missions. The main scenario investigated in this thesis is a potential commercial asteroid mining mission, where a prospective spacecraft (S/C) examines a sequence of asteroids in the main belt. To fulfill its task, the S/C has to cope with two particular scenarios which are (i) absolute heliocentric navigation and (ii) relative asteroid-centric navigation. To enable the S/C to successfully navigate within both scenarios, it is equipped with a comprehensive sensor suite, which is presented as well.

The first scenario will generally be introduced and related work will be briefly presented. The goal is to reduce the uncertainty of the state estimate by actively observing celestial bodies and stars. Subsequently, active perception, using value iteration to plan a sequence of rotations, will be proposed for particle filter (PF) based heliocentric navigation in Sect. 3.2.2 [Nakath et al., 2016]. Here, a novel active perception method of directly estimating the impact of a measurement on the uncertainty is introduced for the first time. It operates in 2D and directly quantifies the estimated impact based on the projection of the measurement direction on scaled eigenvectors of the covariance matrix. Subsequently, an improved version of the direct impact estimation of measurements on the uncertainty is proposed. The active perception approach for heliocentric navigation, presented in Nakath et al. [2018], operates in 3D and selects actions in a reactive manner. The measurement of the respective impact is obtained by projecting the covariance onto vectors or planes, placed according to the correctional information of the respective sensors. The active perception approach works extremely fast, as two angles are iterated in parallel and only a desired pose is determined. This pose is passed to a proportional-derivative attitude controller introduced in Nakath et al. [2017], which computes the control torque based on the angular velocity and the geodesic difference on the state manifold in one step. Finally, the approach is evaluated with an extended Kalman filter (EKF), an unscented Kalman filter (UKF), as well as a PF, jointly operating on a manifold state representation.

The work related to active asteroid-centric deep space navigation is presented in the subsequent Sect. 3.2.3. The goal of the active 3D Graph SLAM is to select trajectories, such that the localization performance can be traded-off against the exploration of the asteroid. The approach, presented in Nakath et al. [2019], is evaluated in a realistically modeled local environment around the asteroid Itokawa. The graph is constructed in a tightly-coupled fashion, where an EKF operates in the graph structure. It proposes poses for new nodes and is used to validate relative pose-offset measurements based on a scan-to-scan matching approach. The latter draws on Flash-LiDAR measurements, which additionally serve to establish a point cloud map. In addition, the scan-matching results are fed back into the EKF to bound the – otherwise uncorrected – velocity error. This loop can robustly be used in asteroid-centric navigation scenarios with small error margins.

The point cloud map obtained by the Graph SLAM algorithm is subsequently projected into an evidential grid map to serve as a decision basis for a novel active perception scheme, tailored to rapid trajectory selection in the vicinity of celestial bodies. Within an evaluation sphere placed around the asteroid, approximate trajectories are first generated to identify a subset of endpoints used to compute optimal trajectories for. Both types of trajectories can be evaluated for their respective exploration and localization gain within the evaluation sphere by mimicking the sequential information intake in the course of the trajectories. Finally, localization and exploration performance can be simultaneously improved by introducing an adaptive parameter computed from the covariance marginalized from the graph structure. To the knowledge of the author, this is the first active Graph SLAM approach to be employed in the vicinity of an asteroid.

In Chap. 4 an excursion into sensorimotor object recognition is undertaken. Initially, a very brief introduction to the investigation of human visual perception is given in order to establish a basis of biologically-plausible principles to which a sensorimotor-based autonomous system should adhere. Subsequently, relevant computational models, implementing those principles, are discussed. Finally, a concept of a probabilistic sensorimotor feature is presented, which complies with the beforehand established principles. Based on those preliminaries, a versatile sensorimotor object recognition system operating in 3D is presented in Sect. 4.2.1 [Nakath et al., 2014]. It can be equipped with different frontends, ranging from a robotic arm inspecting real objects with a camera, over objects in a virtual reality, to simple images of objects taken from datasets. The system can learn object classes and afterwards classify test-instances of objects within a probabilistic framework, potentially within arbitrary modalities. However, the employed cluster-based learning scheme suffers from quantization errors. Thus, in Reineking et al. [2015] an improved naive Bayes nearest neighbor-approach for sensorimotor probabilistic classification is proposed.

In general, using sensorimotor information fosters classification performance and enables the computation of expected measurements. Thus, Nakath et al. [2014] use the canonical implementation of the information gain principle, based on expected entropy minimization, as discussed in Sect. 4.2.2. In addition, the feature with the highest value is chosen as the representative for the selected fixation-bin. Hence, the final result is based on bottom-up as well as top-down acquired information. Furthermore, different online and offline information maximization schemes are investigated in Reineking et al. [2015]. They draw on top-down approaches using expected values from previously learned distributions and a bottom-up approach using salient features.

Finally, this thesis concludes with Chap. 5, which comprises a separate discussion of the two main topics.

2 Active Perception for Autonomous Systems

This chapter introduces active perception for autonomous systems from a methodological perspective. Initially, a definition of the class of autonomous systems to be investigated is given in the following Sect. 2.1. Subsequently, general probabilistic state estimation approaches are presented in Sect. 2.2, with a special focus on recursive state estimation in Sect. 2.2.1 and Simultaneous Localization and Mapping (SLAM) in Sect. 2.2.2. All state estimation algorithms operate on an encapsulated manifold state representation which is presented in Sect. 2.2.3.

Subsequently, Sect. 2.3 presents active perception approaches, which are interpreted as planning under uncertainty in Sect. 2.3.1, where an information gain principle, presented in Sect. 2.3.2, governs the action selection process. In the last section of this chapter, additional references to the work presented in the published papers are given, to provide an overview from the methodological standpoint.

2.1. Autonomous Systems

Throughout this thesis, autonomous systems are considered to be robotic systems that are able to

- pursue goals for an extended period of time,
- gain information about the environment,
- move in the environment in accordance with their goals,
- and avoid situations that are harmful to people

without external intervention in a self-sustained manner. To this end, they are typically equipped with hardware and software tailored to solving their respective tasks. The interaction between the autonomous system and the world can be thought of in terms of a closed *perceive-reason-act* loop scheme (see Fig. 2.1 and cf. e.g., Winston [1992]; Russell and Norvig [2003]).

Within the closed-loop, the *perception* module comprises sensors to capture aspects of the surroundings or internal states. In addition, the software-part of that module typically preprocesses the raw sensor data to prepare it for the sensor fusion module.

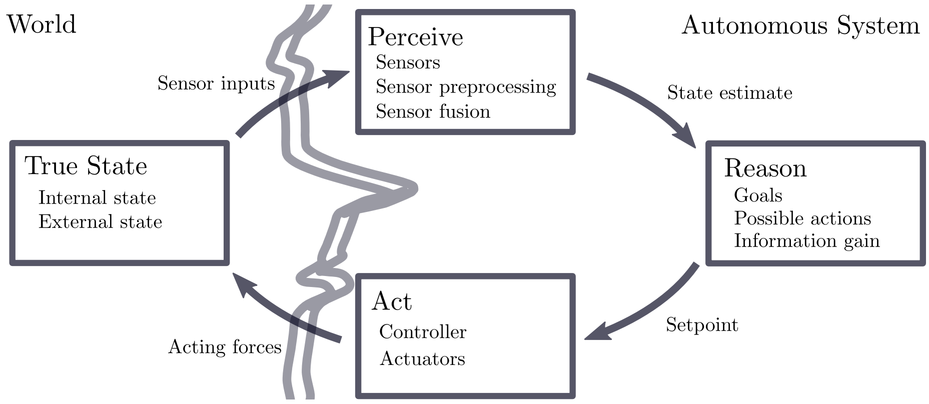


Figure 2.1.: Closed-loop model of an autonomous system interacting with the world.

The task of the latter module is to establish a coherent state estimate, often based on pre-defined models and the current sensor readings.

This estimate is then fed into the *reasoning* module. Depending on the tasks and the available data it may take on a multitude of actual implementations. However, in the course of this thesis, it will always be considering a set of possible next actions to take. They will be evaluated with respect to the goals based on accordingly designed value functions. Throughout this thesis, the goals will usually be defined in terms of uncertainty minimization of a quantity to estimate, i.e., such that information is gained.

The decision of the reasoning module will subsequently be passed to the *action* module in the form of a desired state that the system should take. This so-called setpoint is then processed in the software-part of the *action* module: the controller computes the forces to be actuated based on the setpoint and the current state estimate.

Finally, the gray line in Fig. 2.1 indicates the border between the autonomous system and the world. All information passing this border is subject to noise which may naturally occur in the case of a real-world application or is artificially introduced in a numerical simulation scheme. Thus, the forces actuated by the autonomous system will have a noisy impact on the world which in turn reacts accordingly. This results in a new ground truth (gt) state of the world, comprising the internal state of the autonomous system and all remaining external states. The noisy sensor inputs for the *perception* module – in turn initiating a new cycle of the perceive-reason-act-loop – are then generated based on those states.

2.2. Probabilistic State Estimation

As a basis for reasoning, an autonomous system strives to perceive relevant aspects of the world to subsequently represent them in an appropriate manner. Hence, the result of the perception process of an autonomous system at time t is summarized by the sensor fusion module in the state vector, denoted by x_t. The state vector lives in the state space $\mathcal{X}$, i.e., $x_t \in \mathcal{X}$. It may describe external states, like the position of landmarks, or the occupancy in a grid map, but also internal states may be of interest. For instance, the pose, i.e., the position and attitude of an end effector of a robotic arm, is given by its internal joint positions, i.e., its kinematic state [Murray et al., 1994]. Of major interest is also the estimation of the pose of the autonomous system itself with respect to some external frame of reference. In the 2D-case, the pose is minimally parametrized by the two position coordinates in the plane and an additional heading direction, while in the 3D-case this extends to three position coordinates and three parameters for the angular attitude. This information might also be augmented with bias information or dynamic state variables, like e.g., velocity and acceleration.

Reconsidering the description of the closed-loop system given in Sect. 2.1, several sources of uncertainty can be identified. The exact behavior of the environment is unpredictable and thus a state transition can not be exhaustively described by a motion model. This may be caused by an imprecise model of the environment or an environment with dynamics that are difficult to capture holistically. Sensors are limited in what they can perceive and their readings are subject to noise, resulting in partial and imperfect information about the world. The actuation itself is subject to noise, i.e., the controls are not precisely carried out and the environment might not react as modeled. Another factor is the usage of approximation-techniques and discretizations to ensure the real-time capability at the expense of precision.

As uncertainty cannot be avoided, it is managed alternatively. Hence, it is common to employ some state estimation scheme which captures the associated uncertainties based on the calculus of probability theory [Kolmogorov, 1950]. For a comprehensive introduction into the latter, the interested reader is referred to Ross [2014a,b]. In the field of probabilistic robotics the variables within the state vector x_t are represented by probability density functions over the space of all possible values they may take. This results in a probabilistic belief about the state $bel(x_t)$. For a comprehensive overview of the topic, the reader is referred to Thrun et al. [2005]. In addition, this approach can be thought of in the form of probabilistic graphical models or Bayesian networks. Please see Koller and Friedman [2009] for a comprehensive overview and a wide range of applications. By specifying statistical independencies on joint probability distributions, a factored representation of a problem emerges. If this technique is applied over time, a recursive state estimation problem can be formulated in the form of a dynamic Bayesian network (DBN) (see Fig. 2.2a). Here, the belief about the current state x_t shall be estimated over time, commonly based on the observed controls u_t and measurements z_t. In addition, a first order Markov assumption is usually applied, i.e., all information is assumed to be accumulated in the preceding state x_{t-1}, which implies, that all prior information may safely be discarded.

2.2.1. Recursive State Estimation

The most general approach to this kind of problem is the Bayes filter [Thrun et al., 2005, Chap. 2.4] which recursively estimates the posterior $p(x_t|u_{1:t}, z_{0:t})$[1] over time. It

[1] Here, the shorthand notation $\circ_{0:t} = \circ_0, \cdots, \circ_t$ is used.

consists of (i) a prediction step and (ii) a correction step, where the latter step corrects the state proposed by the motion model with the likelihoods of the sensor readings

$$p(x_t|u_{1:t}, z_{0:t}) \propto \underbrace{p(z_t|x_t)}_{\text{sensor model}} \underbrace{p(x_t|u_{1:t}, z_{0:t-1})}_{\overline{bel}(x_t)}, \tag{2.1}$$

where the proposal distribution $\overline{bel}(x_t)$, in turn, is obtained in the prediction step

$$p(x_t|u_{1:t}, z_{0:t-1}) = \int \underbrace{p(x_t|u_t, x_{t-1})}_{\text{motion model}} \underbrace{p(x_{t-1}|u_{1:t-1}, z_{0:t-1})}_{bel(x_{t-1})} dx_{t-1}\,. \tag{2.2}$$

Here, $p(x_t|u_t, x_{t-1})$ is called the motion model for state transition, and the forward sensor model $p(z_t|x_t)$ computes the likelihood of a measurement given the predicted state. However, in most real situations, it is infeasible to obtain closed form solutions for the two equations given above. Thus, their distribution is often approximated by either assuming a Gaussian shape or by a set of samples, or – put in other words – by parametric or nonparametric methods.

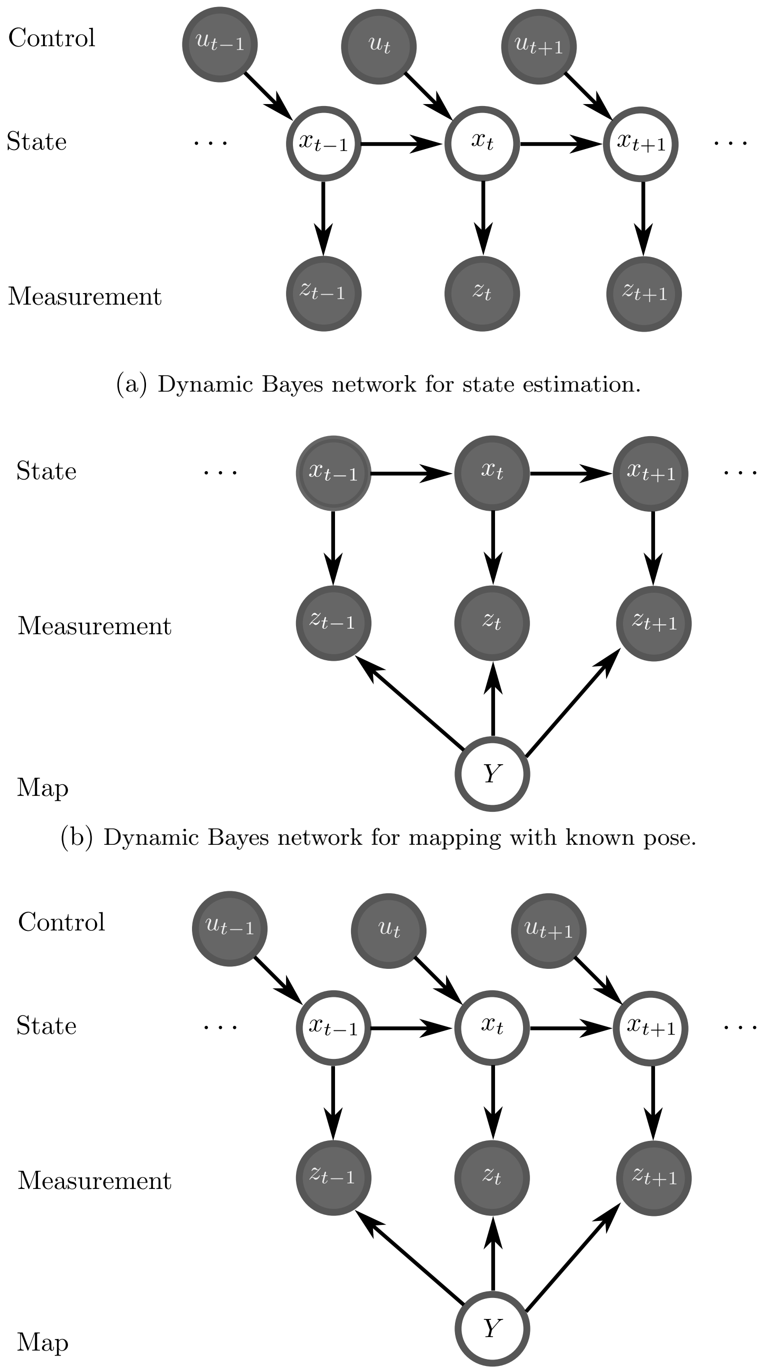
The Kalman filter (KF) is the foundation for the family of Gaussian filters. It assumes the posterior to be normally distributed [Kalman, 1960], thus

$$x_t|u_{1:t}, z_{0:t} \sim \mathcal{N}(\mu_t, \Sigma_t)\,, \tag{2.3}$$

where μ_t and Σ_t denote the mean and the covariance of a normal distribution, respectively. However, this assumption entails that the motion model in the prediction step and the sensor models in the correction step take a certain linear form. When the assumptions hold, an extremely fast and optimal closed-form solution can be obtained [Thrun et al., 2005]. However, they are most certainly violated in real-world applications.

Hence, the EKF [Jazwinski, 1970] replaces the linear motion model and the linear sensor model by respective nonlinear functions g and h with additive Gaussian noise. However, passing a Gaussian through these functions would yield a distribution violating the Gaussian shape. Hence, the nonlinear functions are linearized around the mean of the Gaussian to be mapped. Thus, the mean of a Gaussian can be mapped by a nonlinear function, while the respective covariance is approximated by linearization, maintaining its properties. This leads to an efficient filtering approach, which can be applied to real-world problems.

Of course, the linearization of the covariance introduces a certain error. The UKF seeks to mitigate it by approximating the shape of the distribution in a stochastic regression scheme. To this end, it takes a fixed number of deterministic samples – so-called sigma points – from a Gaussian and passes them through the nonlinear functions g and h [Julier and Uhlmann, 1997]. Subsequently, mean and covariance are recovered from a weighted sum of the samples. As opposed to the EKF, the UKF does not depend on the Jacobians i.e., the matrix containing all first-order partial derivatives. In Van Der Merwe [2004] it is reported that UKFs perform slightly better than EKFs, albeit at the cost of a slightly higher computational demand.



(a) Dynamic Bayes network for state estimation.

(b) Dynamic Bayes network for mapping with known pose.

(c) Dynamic Bayes network for Simultaneous Localization and Mapping.

Figure 2.2.: Models for state estimation and mapping on time-series data. The white nodes are the quantities to be estimated, while the dark nodes are the observed variables. They particularly denote a state vector x, sensor measurements z, controls u, and a map Y. Figures adopted from Thrun et al. [2005].

Finally, the PF is a prominent example of a nonparametric filter, which does not presuppose a fixed functional form of the models or the posterior. It thus represents the latter in terms of M samples, called particles

$$\mathcal{X}_t = x_t^{[1]}, x_t^{[2]}, \cdots, x_t^{[m]}, \cdots, x_t^{[M]}, \tag{2.4}$$

where each $x_t^{[m]}$ is a full state vector [Doucet et al., 2001]. In the prediction step, all particles are individually propagated by virtue of the motion model. In the subsequent correction step, the single particles are weighted in proportion to the sensor likelihood. Finally, resampling is conducted, redrawing samples according to their importance weight. The latter step ensures, that enough samples are drawn from the posterior i.e., the part of the sate space which should be represented. This approach allows for an arbitrarily-shaped and even multi-modal distribution of the posterior. In addition, a sample based approach can model arbitrary nonlinear transformations of random variables. However, the particle filter suffers from high computational demands, as every particle has to be individually propagated. To mitigate this effect, techniques like an improved proposal distribution can be employed to selectively sample only the relevant parts of the state space [Grisetti et al., 2007]. In addition, when resampling is executed in every step, the distribution might become too concentrated. On the contrary, if it is not carried out often enough, the distribution might degenerate, spreading over a big area of the state space. Thus, the application of adaptive resampling techniques is recommended (cf. [Grisetti et al., 2005; Del Moral et al., 2012]).

In Chap. 3 the performance of different approaches to recursive filtering is shown in active deep space navigation scenarios (cf. González Peytaví et al. [2015]; Nakath et al. [2016, 2017, 2018, 2019].

2.2.2. Simultaneous Localization and Mapping

In Fig. 2.2b a dynamic Bayesian network is depicted, which assumes that the states x_t can be observed. Additionally, a random variable Y, representing parts of the environment to be estimated, is introduced. Depending on the type of model for Y, it can be used for active sensorimotor classification, as presented in Chap. 4, or for mapping of the environment, as presented in Chap. 3. If a state x_t and a corresponding measurement z_t are known, a map Y can be recursively updated utilizing an inverse sensor model $p(Y|x_t, z_t)$.

An ad hoc approach would be to estimate a trajectory $x_{0:t}$, i.e., a sequence of poses of a robot and subsequently update the map Y, using some combination of the approaches shown in Fig. 2.2a and Fig. 2.2b. However, this approach yields suboptimal results as the mutual correctional potential of the trajectory and the map is not leveraged. This leads to the statement of the Simultaneous Localization and Mapping (SLAM) problem as depicted in Fig. 2.2c [Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006], which corresponds to the estimation of the joint probability distribution

$$p(x_{0:t}, Y|z_{0:t}, u_{1:t}) \tag{2.5}$$

of the trajectory $x_{0:t}$ and the map Y given observed measurements $z_{0:t}$ and controls $u_{1:t}$.

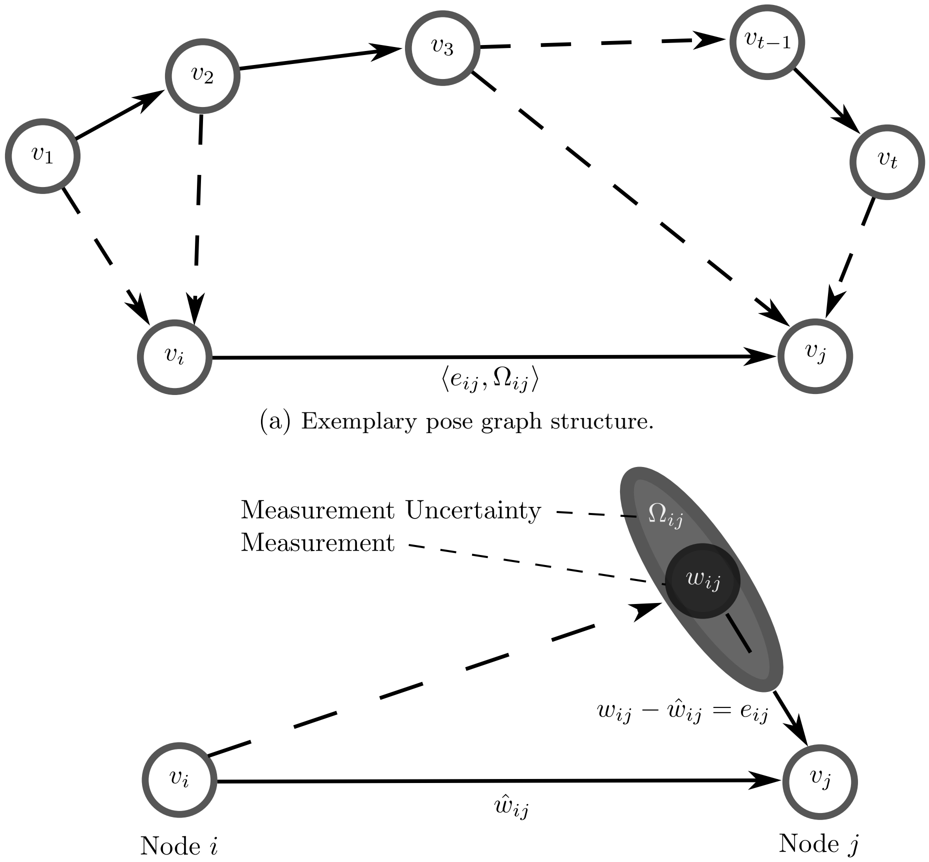
Again, filtering approaches can be employed to tackle this problem in a recursive manner. A popular approach is the Rao-Blackwellized particle filter (RBPF), which operates on separate distributions marginalized from state space [Murphy, 2000; Doucet et al., 2000]. This, in turn, allows for different estimation schemes for the single variables from the joint distribution. The well-established FastSLAM algorithms [Montemerlo et al., 2002, 2003] use this opportunity by representing the trajectory $x_{0:t}$ by samples. Each of those samples is in turn accompanied by a corresponding landmark-based map estimate, represented by Gaussians, which is analytically updated from the respective trajectory. In the same manner an occupancy grid map can be employed as the map representation [Grisetti et al., 2007]. It typically discretizes the world into grid cells and represents the occupancy estimate of each cell in terms of a binary random variable [Moravec and Elfes, 1985]. Recently, Reineking and Clemens [2013]; Clemens et al. [2016] introduced Evidential FastSLAM, which uses an evidential grid map in a RBPF-SLAM scheme. It represents the occupancy belief in terms of belief functions [Smets and Kennes, 1994; Shafer, 1976], which allows for a more distinguished representation of the dimensions of uncertainty [Reineking and Clemens, 2014]. In addition to the belief of an area being occupied or empty, unknown areas or conflicting measurements can explicitly be represented.

An alternative approach to the SLAM-problem is time-series smoothing. Here, a system of equations based on errors introduced by spatial constraints is globally optimized [Lu and Milios, 1997]. As opposed to the filter-based approaches presented above, this family of approaches do not resort to the first order Markov assumption. Thus, they can make use of all preceding measurements. However, first an appropriate structure has to be identified to deal with this amount of information in an efficient manner. To this end, a set of poses and relating measurements is interpreted as a pose graph as depicted in Fig. 2.3a [Grisetti et al., 2010a]. It consists of nodes v_i and v_j, each representing quantities to be estimated. The nodes are related to each other by their expected offset $\hat{w}_{ij}$ and are additionally constrained by measurements denoted w_{ij}. These measurements typically stem from observing the same fractions of the world from two different nodes or poses, thus establishing an indirect differential measurement between them. Considering Fig. 2.3a, the emergent structure naturally forms a sparse graph [Golfarelli et al., 1998]. Based on this insight, an error function can be defined, measuring the tension between the predicted pose offset $\hat{w}_{ij}(v_i, v_j)$ and the actually measured pose offset w_{ij} by

$$e(v_i, v_j, w_{ij}) \overset{\text{def}}{=} e_{ij}(v_i, v_j) = w_{ij} - \hat{w}_{ij}(v_i, v_j)\,. \tag{2.6}$$

In Fig. 2.3b an error function between two nodes v_i and v_j is shown in detail, where the information matrix $\Omega_{ij} = \Sigma_{ij}^{-1}$ represents the measurement uncertainty of w_{ij}. Considering all constrained pairs of vertices in the graph, a set of nonlinear quadratic constraints, each weighted by the corresponding measurement uncertainty, can be defined as

$$F(x) = \sum_{\langle i,j\rangle \in \mathcal{C}} e_{ij}(v_i, v_j)^T \, \Omega_{ij} \, e_{ij}(v_i, v_j)\,, \tag{2.7}$$



(a) Exemplary pose graph structure.

(b) Detaitled view on an error function e_{ij} within a Graph SLAM structure.

Figure 2.3.: Pose graph representation for a time-series data smoothing approach. In Figure (a), the solid lines signify spatial constraints with consecutive indices which can be obtained by, e.g., odometer measurements, while dashed lines denote spatial constraints resulting from observations of the same part of the environment from different poses. In Figure (b) the solid line denotes an expected offset $\hat{w}_{ij}$ between a node v_i and v_j, while the dashed line indicates the actual measurement w_{ij}, with the corresponding uncertainty represented as the information matrix Ω_{ij}. Figures adopted from Grisetti et al. [2010a].

where $x = [v_0, \cdots, v_n]^\top$ is a vector of parameters describing the poses of the nodes, and $\mathcal{C}$ is the set of node-pair indices constrained by a measurement w_{ij}. Subsequently, $F(x)$ can be optimized by altering the poses of the nodes such that the error function is minimized, or put more formally

$$x^* = \arg\min_x F(x)\,. \tag{2.8}$$

In the course of time, different approaches to efficiently solve this kind of problem have been proposed. In Frese et al. [2005] methods for multi-level-relaxation are described, while Dellaert and Kaess [2006] exploit the sparse structure of the system. Another possibility to gain efficiency is the introduction of hierarchies [Bosse et al., 2003], or independent local maps [Estrada et al., 2005].

Finally, Graph SLAM can be thought of being composed of a frontend and a corresponding backend. The former takes care of the construction of the graph, by inserting nodes and constraining them with edges, if applicable. While the latter seeks to optimize the resultant structure in an efficient smoothing approach. The g^{2}o-framework [Kümmerle et al., 2011] seeks to provide such an optimization backend. It solves the system of nonlinear equations with local linearization techniques, while efficiently exploiting the sparse structure. In Chap. 3, a combination of an evidential grid map and Graph SLAM, resting on the g^{2}o-backend, will be presented (cf. [Nakath et al., 2019]).

2.2.3. Manifold State Representation

As introduced above, a probabilistic state estimate $bel(x_t)$ comprises a mean and associated uncertainties. As an autonomous system moves over time in space, the problem of how to represent a sequence of relative movements and the associated uncertainties arises. A possible solution is to define a special compounding operator for uncertain states, which connects consecutive state estimates in a sound manner. For example, the $\oplus$-operator, which has been presented in Smith et al. [1990], directly compounds the means and adapts the associated uncertainties accordingly via local linearization.

However, if the state-space $\mathcal{X}$ is completely or partly defined on a manifold, a proper parametrization of the former has to be considered. Of special interest is the attitude in 3D. The latter is an element of the set of all rotations about the origin of 3D space called the Special Orthogonal Lie-Group $SO(3)$ [Murray et al., 1994; Warner, 2013]. Its three degrees of freedom can – inter alia – be described in terms of Euler angles, axis-angle-representations, unit quaternions, or direction cosine matrices (DCMs). Euler angles provide a parametrization coinciding with the degrees of freedom, but are subject to gimbal lock phenomena and singularities [Ang and Tourassis, 1987]. On the contrary, overparametrized representations of the attitude by quaternions or DCMs overcome these problems, but render the application of standard algorithms impossible, which typically operate directly on the degrees of freedom of the state vector.

In Hertzberg et al. [2013] state encapsulation is proposed, which approaches those problems in an elegant manner, resting on two properties of manifolds. First, on topological n-manifolds a bidirectional mapping between a local neighborhood of a point on the manifold and $\mathbb{R}^n$ always exists, i.e., n-manifolds are locally homeomorphic to $\mathbb{R}^n$. And second, a state comprising different components can be represented by a compound manifold resulting from the Cartesian product of the manifolds representing the single components. By additionally exploiting the fact that Lie-Groups can be thought of as manifolds at the same time, a lower-dimensional local vector space $\mathbb{R}^n$ is defined around every group element of the state space $\mathcal{X}$ [Iserles et al., 2000; Murray et al., 1994; Warner, 2013]. Hence, states can be represented overparametrized on the manifold, while their differences can be expressed in vector space in terms of a minimal parametrization. Thus, the states are represented in a globally consistent manner and their attitude-part does not suffer from the shortcomings of minimal parametrizations anymore, while algorithms directly operating on the degrees of freedom can be employed in the vector space. Furthermore, ordinary Gaussian distributions can be expressed in the local vector space around a state. The latter is an opportunity to express uncertainty, but also to sample noise and apply it to the state on the manifold.

The mapping between the state in the manifold space $\mathcal{X}$ and the vector space $\mathbb{R}^n$ is based on two operators. The operator $\boxplus : \mathcal{X} \times \mathbb{R}^n \to \mathcal{X}$ adds a state-delta expressed in the vector space to a state in the manifold space, resulting in a new state. The corresponding $\boxminus$-operator expresses the difference between two states in terms of a delta-vector in $\mathbb{R}^n$, hence $\boxminus : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^n$. In addition, if a state comprises different components, so-called compound operators can be defined to jointly encapsulate those manifolds [Hertzberg et al., 2013].

On the $\boxplus$-manifold, as defined in Hertzberg et al. [2013], standard filter algorithms like the EKF [Clemens and Schill, 2016], UKF [Hertzberg et al., 2013], or the PF [Nakath et al., 2016] can be employed with minimal adaptations [Nakath et al., 2018]. Another possible application is the usage in the error function of an attitude controller [Chaturvedi et al., 2011]. A proportional derivative (pd) attitude controller computes the control torque from the current offset from the setpoint, i.e., the desired attitude, and the current angular velocity [Bullo and Murray, 1995]. As presented in Nakath et al. [2017], the offset can be computed on the manifold, yielding a result in vector space $\mathbb{R}^3$ using the $\boxminus$-operator. In addition, the Graph SLAM approach, resting on the g^{2}o-framework, also relies on local linearization of the error functions [Grisetti et al., 2010b; Kümmerle et al., 2011] within a least-squares minimization scheme on the $\boxplus$-manifold. Thus, in Nakath et al. [2019] its edges were adopted to fit the $\boxplus$-scheme employed here, using the $\boxminus$-operator to yield a minimally-parametrized result of the error function in $\mathbb{R}^n$ (cf. (2.6)). In addition, the compounded structure of the $\boxplus$-operator results in a representation of the uncertainty which is independent from the attitude as opposed to, e.g., $\oplus$-based representations. Hence, the $\boxplus$-approach is applied throughout all work presented in Chap. 3 (cf. Nakath et al. [2016, 2017, 2018, 2019]).

2.3. Active Perception

Active perception deals with the question of which actions u_t to perform, in order to optimize a sequential information intake process flawed by uncertain information in the light of some goals. Or put in more technical terms, how the parameters of sensors should be adapted over time, such that they adhere to a certain sensing strategy [Bajcsy, 1988]. Throughout this thesis, adapting parameters usually means changing the orientation of sensors by moving the autonomous system they are attached to. However, in other domains, also the adaptation of intrinsic parameters might be of interest (see e.g., Micheloni and Foresti [2009]).

The state estimation approaches presented above commonly work on time-series data stemming from such a imperfect sequential information intake process. The flaws originate from the constraints of the process: only parts of the world can be perceived and even those only in an imperfect fashion. This is due to the facts that the sensors point

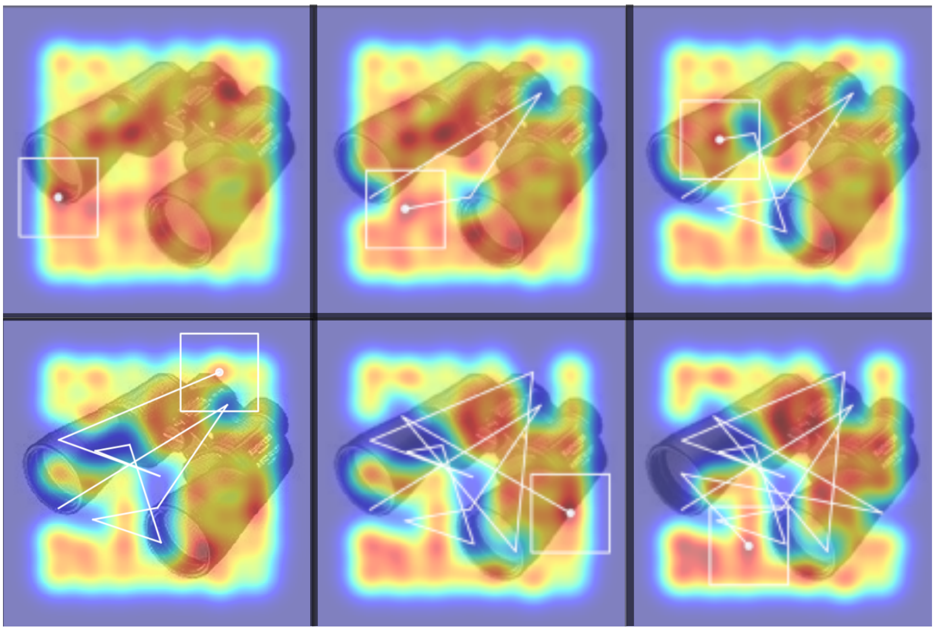


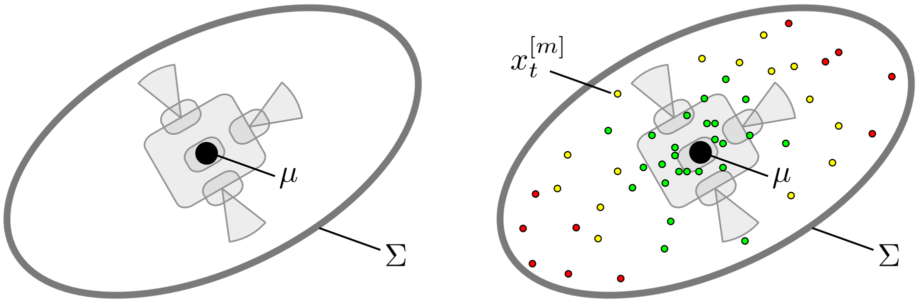
Figure 2.4.: Active perception approach exploring a state-dependent value function on an image over time. The white box indicates the current fixation bin, while the white lines show the fixation path. The heatmap shows a color gradient, where a low informational value is depicted in blue and a high informational value in red. Figure adopted from Reineking [2014]; Reineking et al. [2015].

into different directions, are restricted by a field of view (FOV), operate only in certain modalities, and – on top of that – suffer from noise. In addition, other sources of uncertainty in the system have to be considered, e.g., a non-deterministic state transition, when executing some action u_t. Under these conditions, active perception can be thought of as planning under uncertainty.

2.3.1. Active Perception as Planning

A planning scheme taking robot motion uncertainty into account can be modeled in terms of a Markov decision process (MDP) [Cassandra et al., 1996; Kaelbling et al., 1998; Thrun et al., 2005, Chap. 14.2]. As the outcome of an action is not known in advance, a plan to maximize future payoff in expectation has to be derived for a range of possible states. A plan which maps all possible states x in state space to a subsequent action u is called a control policy π. Each of those policies have an associated value function which measures its specific expected value or the cumulative discounted future payoff [Thrun et al., 2005, Chap. 14.3]. However, it is assumed that the state of the environment is fully observable at all times. If this assumption is dropped, the planning scheme becomes a partially observable Markov decision process (POMDP) [Åström, 1965; Kaelbling et al., 1998; Thrun et al., 2005, Chap. 14.2].

Holistic optimization approaches to those kind of problems are often infeasible. How-



(a) Direct.

(b) Indirect.

Figure 2.5.: Gaussian state estimation.

ever, a dynamic programming paradigm can be employed by dividing the optimization problem in smaller chunks and subsequently solving the individual parts [Bellman, 1957; Howard, 1960]. Using this paradigm, a value iteration algorithm can be devised, recursively maximizing the expected cumulative payoff

$$R_T = \mathrm{E}\left[\sum_{\tau=1}^{T} \gamma^\tau r_{t+\tau}\right], \tag{2.9}$$

where T denotes the planning horizon, such that a plan can be computed ranging from the current time t to $t + T$. The factor γ^τ exponentially discounts future payoffs and ensures certain convergence properties. The payoff function $r(x, u)$ at time t, in turn, is a function of a state x and an action u, typically returning the reward and cost associated with an action, when executed in a given state. The joint representation of cost and reward allows a consistent trade-off between them under uncertainty.

The cost of an action can – inter alia – be defined in terms of time or in a kinematic manner, e.g., by taking into account distances or the number of discrete steps between states. It is also possible to define it in a dynamic manner, i.e., where velocity and acceleration are taken into account. The latter approach can model situations, where a close state might be associated with higher costs than a state which is more remote, due to velocity or acceleration pointing in the direction of the latter. The reward can be defined in terms of reaching a goal or – at least – taking a step into the direction of accomplishing one. Autonomous systems may have diverse goals and – thus – corresponding versatile rewards, depending on the area of application. For example, Fig. 2.4 depicts an evaluation of a reward function for feature selection over time, which is based on expected information gain under the measure of entropy. The costs are simply given as the number of steps carried out, thus not considering the actually traversed distances. Throughout this thesis, the reduction of uncertainty in some form will be the ultimate goal of all actions taken and thus governs their expected reward.

2.3.2. Information Gain

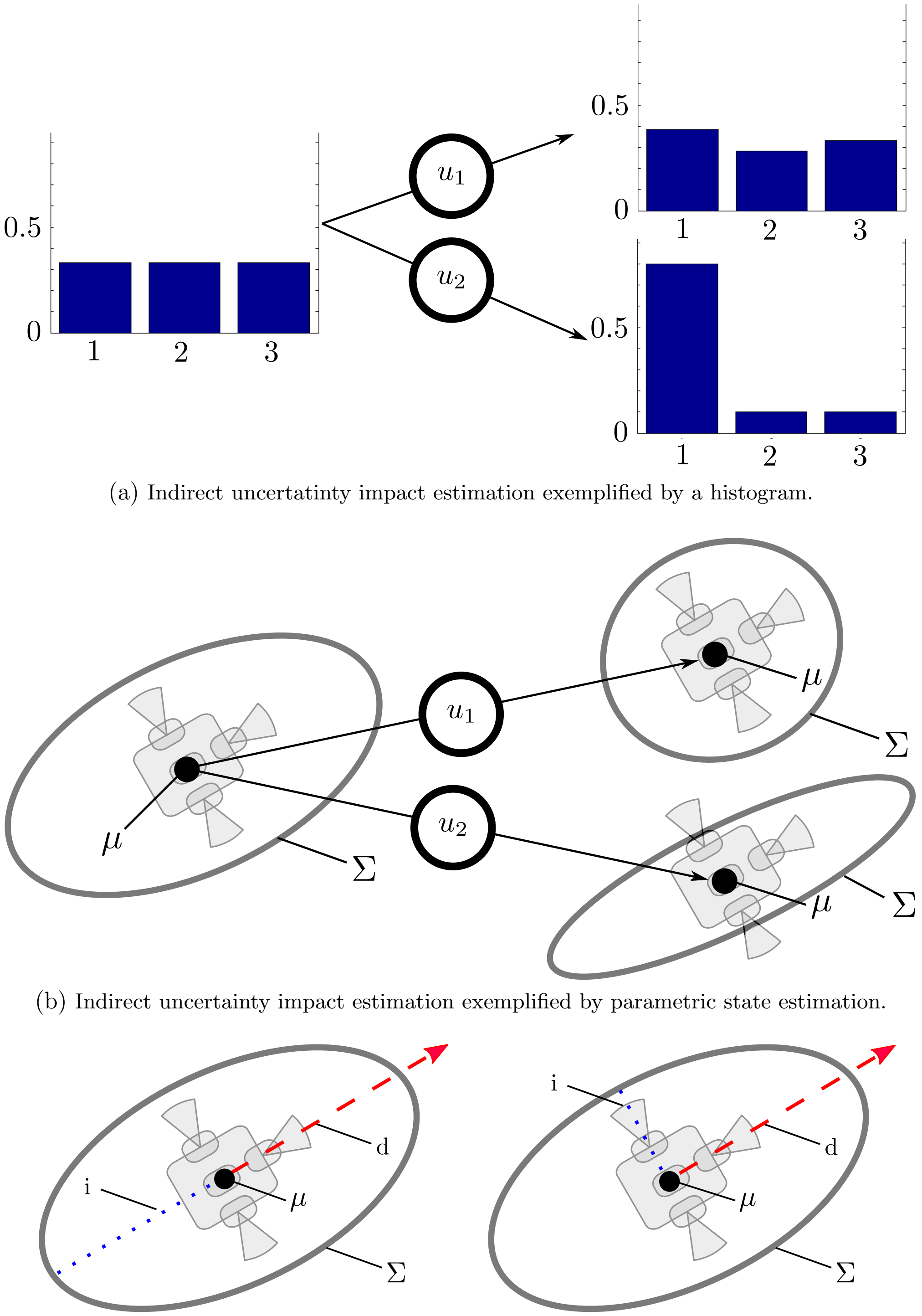
To actively reduce uncertainty, the information gain principle [MacKay, 1992] can be employed by measuring the expected difference induced by an action in the resultant uncertainty. To this end, the uncertainty has to be properly represented, it has to be measured, and it has to be possible to counterfactually, i.e., without actually carrying it out, estimate the impact of an action.

In all work presented in the following, the uncertainty is measured based on the current state estimate, thus serving as a basis for action selection. If the state estimate is provided as a Gaussian, the parameter μ denotes the mean, while the covariance Σ expresses the corresponding uncertainty. This is – of course – the case when Gaussian filters and their derivatives (see Sect. 2.2.1) are employed (see Fig. 2.5a). However, it is also possible to extract a parametric representation of the estimated state in an ex-post approach. This becomes necessary when, e.g., a PF is used, in which case mean and covariance of the nonparametric state estimate are obtained by fitting a Gaussian on the $\boxplus$-manifold (cf. Figs. 2.5a and 2.5b as well as Hertzberg et al. [2013] and Nakath et al. [2016, 2018]). In the case of Graph SLAM, the current parametric pose uncertainty can readily be marginalized from the joint covariance (cf. [Grisetti et al., 2010a; Nakath et al., 2019]).

The parametric uncertainty can then be directly inspected, e.g., by using simple measures like the trace of the covariance, to capture the variance on the principal axes. It is also possible to consider the hyperellipsoid, bounding the uncertainty, by using the determinant as a criterion for action selection. The latter measure in turn results in minimizing the expected variance, which is, in turn, given by the product of the eigenvalues λ of the covariance [Sim and Roy, 2005].

In the scope of the theory of optimal experiments, more involved optimality criteria can be defined, corresponding to a minimization of uncertainty under a certain measure [Fedorov, 1972]. The a-optimality criterion was proposed for trajectory selection in active SLAM, as it maximizes the log-likelihood of the data, assuming a Gaussian error. This, in turn, is equivalent to minimizing the squared error between the model and the data, which is favored by Sim and Roy [2005] over the d-optimality criterion. The latter is proportional to the volume of the hyperellispoid bounding the uncertainty and is thus said to minimize the volume of the model uncertainty [Carrillo et al., 2012]. It can be computed based on the scaled determinant which can be – in the case of a square matrix – obtained by the scaled product of its eigenvalues λ. The latter property allows to drive one dimension of uncertainty to zero, yielding a minimal d-optimality, while the remaining dimensions of uncertainty might take arbitrary values. According to Sim and Roy [2005], this might lead to undesirable solutions in the optimization process. However, the d-optimality criterion is invariant to re-parameterizations as well as linear transformations and may be obtained in logarithmic space to avoid round-off errors in dimensions with small uncertainty [Carrillo et al., 2012]. In addition, in a SLAM scenario, especially for autonomous deep space navigation, near-zero uncertainties in particular dimensions of the position estimate are practicably impossible to achieve.

Hence, in Nakath et al. [2019] the d-optimality criterion is employed to measure the



(a) Indirect uncertatinty impact estimation exemplified by a histogram.

(b) Indirect uncertainty impact estimation exemplified by parametric state estimation.

(c) Direct uncertatinty impact estimation – impact aligned with d.

(d) Direct uncertatinty impact estimation – impact orthogonal to d.

Figure 2.6.: Information gain principles. In Figures (b) and (c) the dashed red line indicates the measurement direction d, while the dotted blue line indicates the correctional impact direction denoted as i.

pose uncertainty encoded in the graph structure to dynamically control an active deep space navigation approach (see Sect. 3.2.3). The latter rests on a twofold trajectory evaluation scheme, tailored to a deep space navigation scenario in the vicinity of asteroids. To this end, approximate trajectories are initially generated to identify a subset of end points to compute optimal trajectories for. For evaluation of the proposed trajectories their expected exploration payoff is compared to their expected localization payoff. The former measure is given by the expected belief about unexplored cells of an evidential grid map [Reineking and Clemens, 2013; Clemens et al., 2016] to be observed from the corresponding trajectory. The latter is measured in terms of the expected belief about occupied cells to be observed and is thus defined to quantify the chance of incorporating additional correctional information by explicit loop closing. The resultant optimal trajectories may have different lengths, depending on the end point distance and the chosen path. This approach can thus be interpreted as having a dynamic planning horizon, based on a discretized subset of choices.

Contrary to that, the heliocentric deep space navigation approaches presented in Sect. 3.2.2 have planning horizons of fixed length. The task here is to pick an attitude of a spacecraft in order to minimize the expected uncertainty. To omit a counterfactual update of the current state with expected measurements (see Fig. 2.6b), the expected influence on the uncertainty is directly measured, drawing on prior knowledge about the nature of the correctional information as well as the measurement direction (cf. Figs. 2.6c and 2.6d). The approach presented in Sect. 3.2.2 employs value iteration for finite-horizon planning. The direct uncertainty impact estimation employed in Nakath et al. [2016] is based on the projection of the measurement direction on the scaled eigenvectors (see Fig. 3.9a). The improved approach also presented in Sect. 3.2.2 uses a greedy one-step planning horizon for active perception [Nakath et al., 2018]. The latter is based on projection of the covariance on the measurement direction to directly estimate the correctional impact of the measurements (see Figs. 3.9b and 3.9c).

However, the distribution does not necessarily have to be Gaussian to serve as a basis for action selection. In addition, it is beneficial – but not mandatory – to omit counterfactual updates like proposed above due to the following reasons. It might be difficult to obtain or process expected measurements, e.g., when they are LiDAR measurements and thus are only meaningful in combination with other LiDAR measurements or a map. A counterfactual update might also be computationally involved, as all possible measurements have to be considered over all possible actions. Sometimes such considerations might even range over several sensors which, in turn, would lead to an even bigger combinatorial explosion. Furthermore, it might not always be immediately obvious how to carry out a counterfactual update, especially when Monte Carlo – i.e., sampling-based – methods are employed. And finally, expected measurements are not easy to process, when mutually dependent distributions have to be estimated, which is the case in SLAM-scenarios.

Despite those considerable issues, the common approach is to update the current distribution with expected measurements, in turn yielding a new distribution for each expected measurement, which can then be compared to the original one [Feder et al., 1999]. Fig. 2.6a exemplifies the indirect implementation of the information gain principle on the basis of a simple histogram. The current distribution is depicted on the left, while two actions, denoted by u_1 and u_2, lead to updated distributions, based on respective expected measurements. A common measure for such a comparison is the Kullback–Leibler divergence [Kullback and Leibler, 1951] to estimate the difference between two distributions. It can thus be seen as a measure of *surprise* induced by a new distribution [Carlone et al., 2014]. This measure would favor u_2, as this would yield the biggest expected difference in the distributions. However, the inverse action u_2^{-1} would yield the same difference, resulting in a uniform distribution. Therefore, it is customary to use the Shannon entropy [Shannon and Weaver, 1949] as a measure for the mutual information [Cover and Thomas, 1991] after a counterfactual update [MacKay, 1992]. It favors *determined* distributions and thus would also vote for u_2 in Fig. 2.6, but it would disregard the inverse step, leading to a uniform distribution. In a seminal paper, Roy et al. [1999] introduced coastal navigation for robots based on an expected entropy minimization approach for active uncertainty reduction in the state estimate.

Due to its properties, the entropy measure is particularly suited for active sensorimotor object-recognition, implemented as a classification task, as presented in Chap. 4. Thus, it is the basis of the information gain strategy presented in Nakath et al. [2014], which is a top-down greedy approach for feature selection. In addition, a bottom-up component is introduced by selecting the feature with the biggest salience score within a fixation bin, indicated by the white box in Fig. 2.4. Finally, an introduction and comparison of several greedy approaches is given in Reineking et al. [2015]. Namely, the approaches consist of several top-down information gain strategies and one saliency-based bottom-up approach.

3 Active Autonomous Deep Space Navigation

In this chapter, different active autonomous deep space navigation approaches will be presented. Initially, an overview of relevant deep space missions will be given in section Sect. 3.1, with a special focus on scientific missions in Sect. 3.1.1 and commercial missions in Sect. 3.1.2. Subsequently, autonomous deep space navigation will be discussed in Sect. 3.2, comprising a description of the sensor suite of the S/C in Sect. 3.2.1 to be employed in either an heliocentric or an asteroid-centric autonomous navigation scheme. The former is discussed in Sect. 3.2.2 which, in turn, is divided in a presentation of the 2D approach and a subsequent discussion of the complete 3D case. The chapter finishes with the presentation of the relative autonomous navigation and mapping approach in Sect. 3.2.3. In the following, the publications covered in this chapter are listed.

Probst et al. [2015] Probst, A., González Peytaví, G., Nakath, D., Schattel, A., Rachuy, C., Lange, P., Clemens, J., Echim, M., Schwarting, V., Srinivas, A., Gadzicki, K., Förstner, R., Eissfeller, B., Schill, K., Büskens, C., and Zachmann, G. (2015). Identifying the Challenges for Cognitive Autonomous Navigation and Guidance for Missions to Small Planetary Bodies. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation.

My share is 10%.
Published at a conference.
Attached in Appx. A.1 (pp. 89ff).

In this conference paper, an overview of the KaNaRiA project and the particular problems to be tackeled within its scope is given. I contributed the sensor fusion part of the paper. It was accepted for presentation at the 66th International Astronautical Congress, Jerusalem, Israel.

González Peytaví et al. [2015] González Peytaví, G., Clemens, J., Nakath, D., Probst, A., Förstner, R., Schill, K., and Eissfeller, B. (2015). Autonomous Orbit Navigation for a Mission to the Asteroid Main Belt. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation.

My share is 20%.
Published at a conference.
Attached in Appx. A.2 (pp. 103ff).

In this conference paper an evaluation of the heliocentric navigation concept is presented, comparing an unscented Kalman filter and a particle filter. I partly designed the research, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 66th International Astronautical Congress, Jerusalem, Israel.

Nakath et al. [2016] Nakath, D., Rachuy, C., Clemens, J., and Schill, K. (2016). Optimal Rotation Sequences for Active Perception. In *Proc. SPIE*, volume 9872, pages 987204–987204–13. International Society for Optics and Photonics, doi: 10.1117/12.2223027.

My share is 45%.
Published at a conference.
Attached in Appx. A.3 (pp. 119ff).

In this conference paper, active perception for autonomous heliocentric navigation is introduced. The evaluation investigates the localization performance of a particle filter in 2D, where projections of the measurement direction onto eigenvectors are used to estimate the expected impact of measurements on the localization uncertainty. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the SPIE Commercial + Scientific Sensing and Imaging, 2016, Baltimore, Maryland, United States.

Nakath et al. [2017] Nakath, D., Clemens, J., and Rachuy, C. (2017). Rigid Body Attitude Control Based on a Manifold Representation of Direction Cosine Matrices. volume 783, page 012040, doi: 10.1088/1742-6596/783/1/012040.

My share is 60%.
Published at a peer-reviewed conference.
Attached in Appx. A.4 (pp. 131ff).

This conference paper introduces a proportional derivative (pd) attitude controller operating on the boxplus-manifold. In an evaluation, its behavior while conducting different maneuvers is investigated. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 13th International Conference on Advanced Control and Diagnosis ACD 2016, Lille, France.

Nakath et al. [2018] Nakath, D., Clemens, J., and Schill, K. (2018). Multi-sensor fusion and active perception for autonomous deep space navigation. In *21st International Conference on Information Fusion (FUSION 2018)*, pages 2596–2605. IEEE, doi: 10.23919/icif.2018.8455788.

My share is 70%.
Published at a peer-reviewed conference.
Attached in Appx. A.5 (pp. 145ff).

This conference paper presents an improved version of the active heliocentric navigation system: it operates in 3D and directly projects the affected parts of the covariance on the measurement direction to estimate its impact. In an evaluation, different implementations of Bayesian filters are compared on different orbit geometries with respect to their reaction on particular active perception strategies. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 21st International Conference on Information Fusion (FUSION 2018), Cambridge, UK.

Nakath et al. [2019] Nakath, D., Clemens, J., and Rachuy, C. (2019). Active Asteroid-SLAM – Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario. *Journal of Intelligent & Robotic Systems*, doi: 10.1007/s10846-019-01103-0.

My share is 75%.
Published in a scientific journal.
Attached in Appx. A.6 (pp. 155ff).

This journal paper introduces the active autonomous asteroid-centric navigation approach which comprises Graph SLAM for the estimation of the trajectory and the map as well as a twofold trajectory generation and evaluation scheme. In an evaluation, the active Graph SLAM approach is investigated in a realistic local environment around the asteroid Itokawa. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It is accepted for publication by the Journal of Intelligent and Robotic Systems.

3.1. Deep Space Missions

Deep space missions may be conducted by manned S/C or unmanned autonomous S/C for scientific or commercial reasons. Typically, the latter kind of missions heavily rely on ground supported navigation and decision-making as a lot of knowledge is already available on Earth and the cost for errors is extremely high. While it is still customary to pursue this kind of approach, autonomous behavior steadily gains importance [Gillette et al., 2018].

This development is driven by several factors. The ever-growing computational capabilities eventually reach on-board computers with a delay, as the hardware has to be space proof. The latter – inter alia – is given by a special radiation shielding to diminish the probability of bit-flipping errors, redundancy, as well as a low weight, and size [Sebestyen et al., 2018]. Also navigation ground support like European Space Agency (ESA)'s European Space Tracking (ESTRACK) network [Budnik et al., 2004] and NASA's Deep Space Network (DSN) [Curkendall and Border, 2013] might not be applicable due to hight cost. On the one hand, this cost is caused by operating ground stations on Earth. On the other, the signal roundtrip time might take up to several hours in deep space missions, caused by the physical limitations of radio signals and the additional time for decision-making on Earth [Bhaskaran, 2012]. An actual example is the two-way control signal roundtrip time in the case of Mascot, one lander of the Hayabusa II mission. Its limited battery capacity renders ground support infeasible and thus makes autonomous decisions compulsory [Ho et al., 2017]. Another example are autonomous planetary subsurface exploration approaches like DLR's Enceladus Explorer (EnEx), which – by design – operates in deep space without ground support [Dachwald et al., 2014; Clemens and Reineking, 2014; Kowalski et al., 2016].

Thus, an increasing level of autonomy becomes more relevant for state estimation and on-board decision-making. This holds true for the relatively new class of nanosatellites [Kayal et al., 2018], but also for conventional S/C and missions. The tasks might range from docking maneuvers with potentially uncooperative objects [D'Souza, 2014] and protection against impacts of celestial bodies on Earth [Landis and Johnson, 2018], over scientific missions, to potential commercial deep space missions. The latter two mission types are described in detail in the following.

3.1.1. Scientific Deep Space Missions

Private space flight is only beginning to develop, thus the most advanced deep space missions are conducted by public institutions for scientific purposes. The NASA-operated spacecraft Dawn started its journey to the asteroid Vesta and the dwarf planet Ceres in 2007 [Russell and Raymond, 2011]. It was the first S/C to visit an asteroid in the main belt in 2.3 astronomical units (au)[1] distance from the Sun (see Figs. 3.1a and 3.1b), to gather information about the early development stages of the solar system.

The Rosetta spacecraft, launched by ESA in 2004, reached its destination, the comet Tschurjumow-Gerassimenko, in 2014 [Glassmeier et al., 2007]. It subsequently deployed its DLR lander Philae to be the first to land on a comet. However, as the anchoring harpoons failed on the first landing approach, the lander bumped two times from the surface. Finally, it landed in a scarcely illuminated area resulting in a weak energy supply which, in turn, rendered parts of the mission impossible to conduct [Biele et al., 2015].

Of special interest in the scope of this thesis are asteroid sample-return missions. They seek to retrieve samples from asteroids and return them to Earth for further scientific investigations. The first scientific asteroid sample-return mission ever to be conducted was Hayabusa I, which reached the asteroid Itokawa in 2005 and successfully returned a sample of its surface in 2010 after having landed on it [Yoshikawa et al., 2006]. In 2018, the consecutive mission Hayabusa II reached the asteroid Ryugu to drop its lander Mascot [Grundmann et al., 2015] after an initial mapping phase [Tsuda et al., 2013]. Both missions were conducted by the JAXA in close cooperation with the DLR. In addition, NASA launched the Origins-Spectral Interpretation-Resource Identification-Security-Regolith Explorer (OSIRIS REx) [Berry et al., 2013]. The mission is designed

[1] One astronomical unit (au) is defined as the mean distance between Earth and Sun, i.e., $149,597,870,700$ m.

to study asteroids as samples of the leftover debris from the solar system formation process. To this end, it strives to return a sample from the near-Earth asteroid Bennu retrieved in the course of an autonomous touch-and-go maneuver with the robotic arm TAGSAM [Lauretta et al., 2017].

3.1.2. Commercial Deep Space Missions

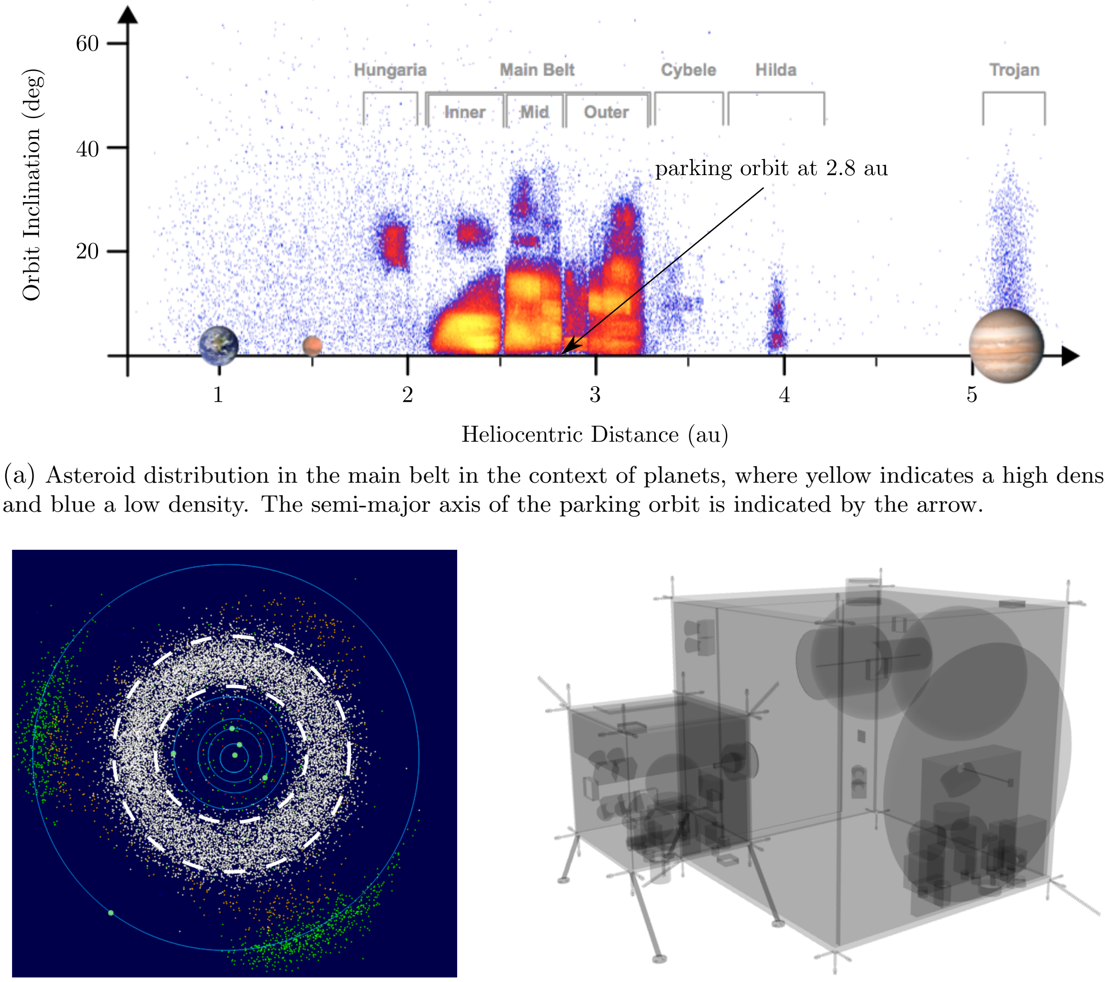
Currently, private space flight is mainly confined to low Earth orbit (LEO) communication and sensing satellites as well as crew and cargo service flights to the International Space Station (ISS). However, there exist a lot of commercial missions in the planning phase, ranging from touristic flights [Seedhouse, 2013], over lunar and Mars missions, as well as private modular space stations [Seedhouse, 2015], to various other potential missions in deep space. For the latter, private companies might provide exploratory S/C, corresponding instrumentation and equipment, deep space communication systems and a corresponding infrastructure [Zapata, 2017]. In addition, in-space stages for leaving and returning to low-Earth orbits as well as landers for delivering and retrieving cargo and crew to surfaces in deep space are of high interest [Zapata, 2017].

Potential commercial asteroid mining missions, which are yet only in the planing phase, are of special importance in the scope of this thesis. Deep space mining missions mainly strive either for in-situ usage of the mined material or for returning the mined resources to Earth. For example, water can be easily extracted and subsequently be used in-situ as a propellant, for life support, or radiation shielding [van Susante and Gertsch, 2018]. Of commercial interest is also to mine platinum-group metals (PGMs) or rare Earth elements (RREs) on asteroids and subsequently return them to Earth [Badescu, 2013]. For the actual work on the ground, various concepts for autonomous mining robots already exist, see e.g., Mueller and Van Susante [2012]. Recently, two companies worth mentioning were founded, actively developing solutions for asteroid mining. Planetary resources[2] was founded in 2012, with the goal of finding suitable asteroids with deep space telescopes and a subsequent closer investigation by autonomous spacecraft [Lewicki et al., 2013]. One year later, Deep Space Industries (DSI)[3] was founded. This company pursues the idea of cheap access to deep space by in-situ usage of mined water as propellant, thus lowering the cost by avoiding frequent encounters with the Earth [Lewis, 2015]. In 2016, DSI also presented their first asteroid prospecting S/C [Foust, 2016]. In addition, companies like Honeybee Robotics[4] actively develop planetary drill systems for those types of missions [Zacny et al., 2008]. Those developments are backed by new legal frameworks for deep space resources mining. In 2015, the United States of America passed the Commercial Space Launch Competitiveness Act, allowing US citizens to engage in the commercial exploration and exploitation of space resources including water and minerals [Tronchetti, 2015]. Luxembourg passed a similar bill in 2017, where companies based in Luxembourg are obliged to obtain a license for deep space resource mining [Foust, 2017]. The corresponding deep space mining initiative

[2] https://www.planetaryresources.com/

[3] http://deepspaceindustries.com/

[4] https://www.honeybeerobotics.com/



(a) Asteroid distribution in the main belt in the context of planets, where yellow indicates a high density and blue a low density. The semi-major axis of the parking orbit is indicated by the arrow.

(b) Heliocentric view of asteroid main belt.

(c) Potential Target Characterization Module (PTCM) spacecraft with attached lander.

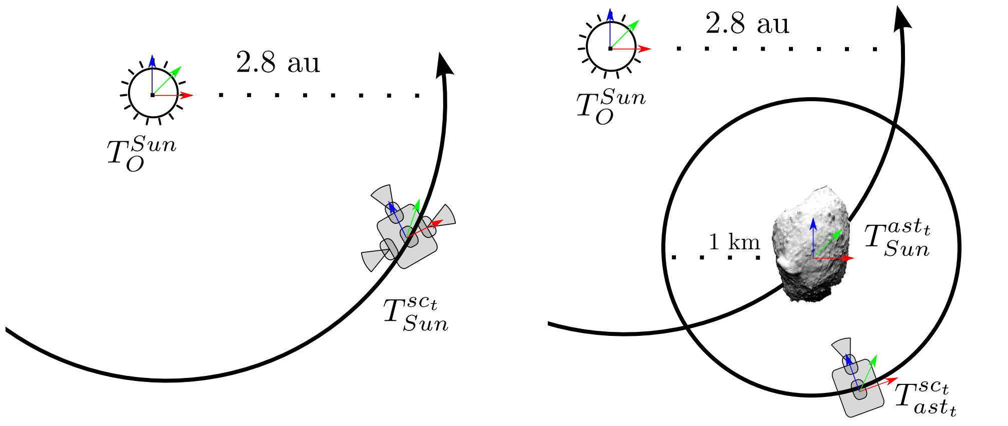
Figure 3.1.: Asteroid distribution in the main belt and PTCM-spacecraft. Figure (a) adopted from DeMeo and Carry [2014], Figures (b) and (c) reprinted from Nakath et al. [2019].

Space Resources[5] strives for mining water and oxygen as propellant and to gather further information about deep space asteroid mining. It – inter alia – comprises the companies introduced above: Deep Space Industries and Planetary Resources.

3.2. Autonomous Deep Space Navigation

The specific scenario investigated throughout this thesis stems from the exploration phase of a commercial asteroid mining mission called Cognition-based, Autonomous Navigation for Deep-Space Resource Mining (KaNaRiA) [Probst et al., 2015, 2016]. The

[5] https://spaceresources.public.lu/en.html



(a) Absolute (heliocentric) navigation.

(b) Relative (asteroid-centric) navigation.

Figure 3.2.: Active autonomous deep space navigation scenarios. To clarify notation: T_B^A denotes a rigid transformation in the special Euclidean group $SE(3)$, in e.g., homogeneous coordinates, by describing the position and orientation of an orthonormal basis of A with respect to B, thus sending a vector v_A defined in A to B, when multiplied from the left, i.e., $v_B = T_B^A v_A$.

mission is designed to consist of a fleet of cooperating S/C operating from the KaNaRiA parking orbit (PO) in a 2.8 au distance from the Sun. It is placed in the Kirkwood gap between the middle and outer asteroid main belt. The Kirkwood gaps correspond to the locations of orbital resonances with Jupiter and constitute relatively safe orbits free of interfering objects (cf. Figs. 3.1a and 3.1b) [DeMeo and Carry, 2014]. At the same time, the gap at 2.8 au is close to the asteroids to be examined, residing in the middle and outer main belt.

The fleet consists of the KaNaRiA Mothership (KMS) establishing a basis on the PO for multiple prospective S/C called Potential Target Characterization Module (PTCM) (see Fig. 3.1c). Each PTCM has an attached lander and is designed for examining a sequence of multiple asteroids in the main belt. Its dimensions are 2.4 m × 2.4 m × 2.4 m and it weighs 3000 kg, of which approximately 650 kg is propellant for the thrusters [Probst et al., 2016; Probst and Förstner, 2017]. For attitude control, it can draw on chemical thrusters and reaction wheels. The thrusters are also used for movement in the vicinity of celestial objects, while for orbit changes an electrical ion propulsion system is used (see e.g., Brophy et al. [2003]). Particularly, the PTCM aims for the characterization of asteroids by sensing available resources and determining the corresponding internal composition using the on-board scientific instrumentation comprising radio tomography, spectrometers, and microscopes [Probst and Förstner, 2017].

Within the KaNaRiA mission context, two scenarios are considered in the following: (i) the cruise phase of the PTCM on the PO, which relies on absolute – or heliocentric – autonomous navigation (see Fig. 3.2a), and (ii) the proximity phase, which relies on relative – or asteroid-centric – autonomous navigation of the PTCM (see Fig. 3.2b) [Probst et al., 2015].

Autonomous deep space navigation is the self-sufficient state estimation of a S/C omitting any ground support [Bhaskaran, 2012]. In the former scenario, the state estimate is given relative to the Sun, while in the latter it is given relative to an asteroid. In either case, an ad-hoc approach might be to define a motion model and integrate over time from a known starting state. Such a dead reckoning approach would – however – lead to an ever-increasing error of the corresponding estimate. Thus, it is customary to correct the predicted position based on observables with known positions (see Sect. 2.2.1). In the cruise phase, the observables will be the Sun, stars, and planets, where precise models of motion exists for the latter (see Sect. 3.2.2). Whereas in the proximity phase, the correction will be with respect to an asteroid model, which has to be established on the fly as a map within a SLAM-scheme (see Sect. 3.2.2).

Both scenarios will be evaluated based on the PTCM, whose sensing capabilities will be described in the following.

3.2.1. Sensor Suite for Autonomous Deep Space Navigation

The PTCM S/C can draw on a pool of sensors, which enables autonomous navigation both in the cruise phase as well as in the proximity operations phase. The particular elements of the PTCM sensor suite will be described in detail in the following.

Inertial Measurement Unit

The inertial measurement unit (IMU) (see Fig. 3.3g) can be employed to obtain the actually applied controls, as only the actuated forces are felt on board of a S/C. In the standard configuration, IMUs contain three orthogonal rate gyroscopes and three orthogonal accelerometers [Woodman, 2007]. The linear acceleration is measured by the latter sensor arrangement, while the former captures angular velocities.

Thus, the effect of thruster controls, directly impacting the acceleration, as well as torque controls, indirectly influencing the angular velocity, can be measured. Those quantities can subsequently be used in a twofold manner. Either for prediction in the motion model of the sensor fusion algorithms or for a correction of the predicted forces derived from directly incorporated controls.

Resonance Scattering Interferometer

The underlying physical principle of the interferometer measurement is the Doppler-effect. The term describes observable wavelength differences in a signal when the source and receiver are moving relatively to each other. The Doppler-effect for electromagnetic waves can be captured by Doppler-spectroscopy (see e.g., [Montmerle et al., 2010]). That approach can be used to measure the speed of celestial objects or whole galaxies approaching or receding from Earth in terms of their respective radial velocities.

This principle can be used for S/C by mounting a resonance scattering interferometer [Scherrer, 2005]. It measures the radial velocity with respect to the Sun by observing the respective frequency shifts of the solar optical spectra as induced by the motion of the

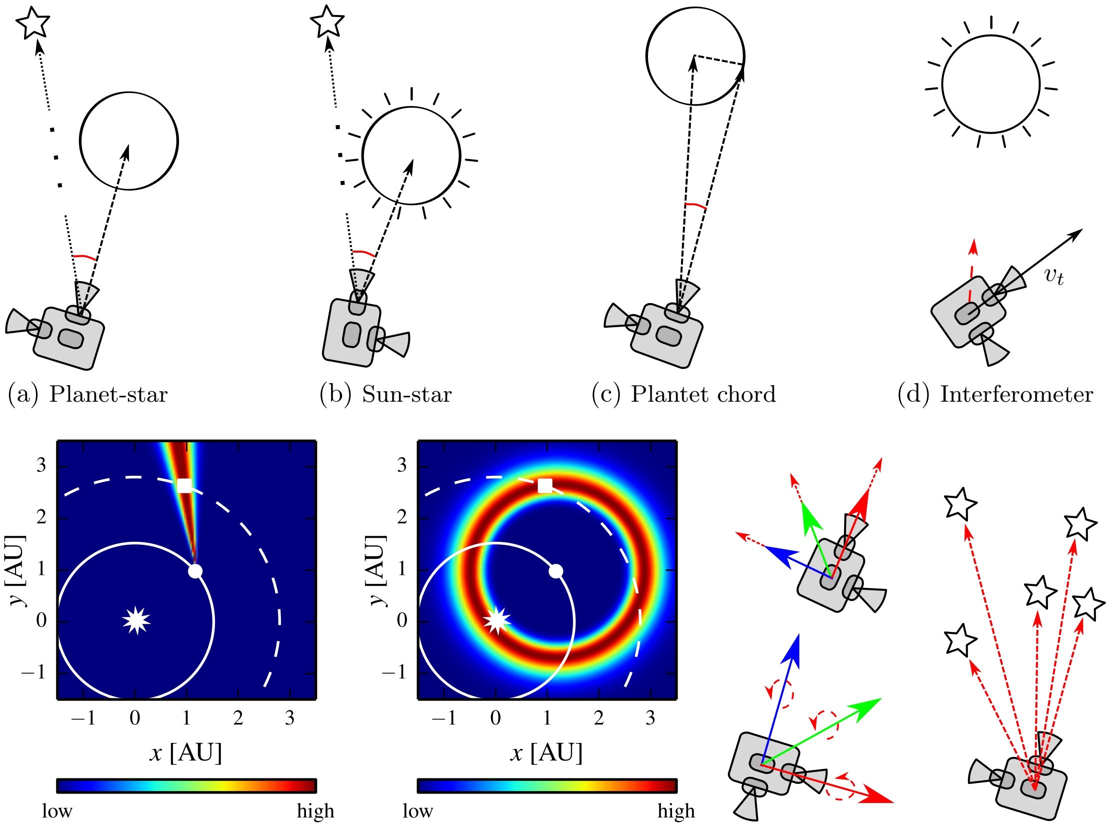


Figure 3.3.: Sensors used in the active autonomous deep space navigation scenarios. The red-colored lines indicate the measured quantities. In Figures (e) and (f) the star-shape denotes the Sun, the dot denotes the observed celestial object on its circular orbit, and the box indicates the S/C. Figures (a) to (c), adopted from Nakath et al. [2018], Figures (e) and (f) reprinted from Nakath et al. [2018] (courtesy of Joachim Clemens), and Figures (d) and (g) adopted from Nakath et al. [2017].

S/C relative to the Sun (see Fig. 3.3d). In Guo [1999], such measurements are proposed to be used for velocity correction purposes in a heliocentric autonomous navigation scheme.

Star Tracker

The star tracker is basically an optical camera which observes stars in its FOV (see Fig. 3.3h). Subsequently, the observations are compared to a previously collected star catalog, e.g., the Hipparcos or the Gaia. The former was compiled from over 300GB of data collected by the High Precision Parallax Collecting Satellite (Hipparcos)-spacecraft, launched in 1989 by ESA [Perryman et al., 1997], and was updated in 2007 [Van Leeuwen, 2007]. In 2013, the successor of Hipparcos, called Gaia, was launched by ESA. It collected

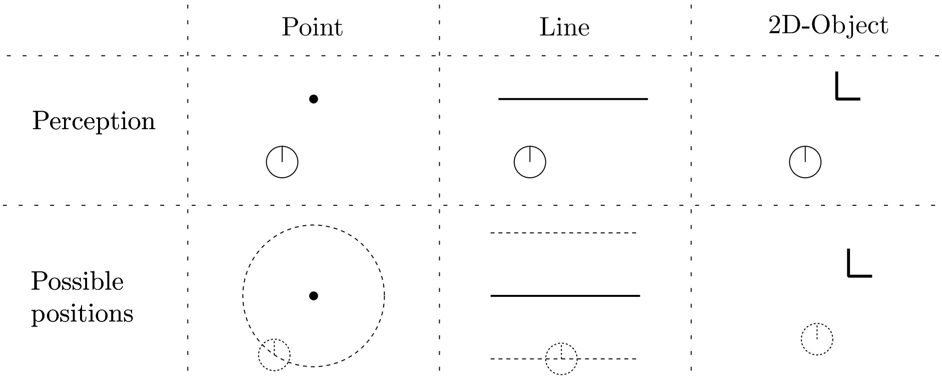


Figure 3.4.: Measurements and their corresponding correctional potential with respect to observed landmarks sketched in 2D. Figure adopted from Meyer and Filliat [2003].

data about 1.7 billion celestial objects, which was subsequently compiled into the Gaia DR2 catalog [Brown et al., 2018].

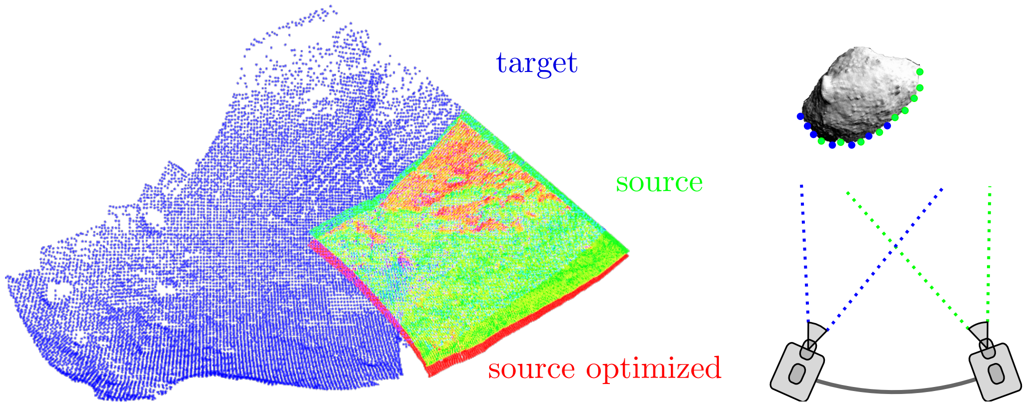
Thus, the star tracker is able to provide an absolute attitude estimation in the heliocentric navigation frame [Liebe, 1995] by using pattern recognition on databases whose size and precision are still increasing.

Sun and Planet Sensors

These types of sensors are typically based on charge-coupled device (CCD)-cameras used in optical autonomous navigation schemes [Paluszek et al., 2010]. Planet sensors can observe planets in their respective FOV and measure either the planetary chord angle (see Fig. 3.3c) or a planet-star angle to corresponding far stars with known position (cf. Fig. 3.3a and the star catalogs introduced above). As the Sun is too bright to measure its chord, only Sun-star angles can be obtained from this celestial object (see Fig. 3.3b).

This kind of information can be used for position correction, albeit in different dimensions. The planetary chord angels can be interpreted as distance measurements, resulting in a circular observation likelihood in 2D (cf. Figs. 3.3f and 3.4). The Sun-star and planet-star measurements provide information, which is orthogonal to the chord-angle measurement with respect to the measurement direction (cf. Figs. 2.6c and 2.6d). This is due to the fact that, instead of the distance, the relative position offset is now subject to correction (cf. Figs. 3.3e and 3.3f, as well as Fig. 3.4). This insight is used in the 2D active autonomous deep space navigation approach, presented in Sect. 3.2.2 (see Nakath et al. [2016]).

If this scheme is thought of in 3D, the planetary-chord angle still provides a one-dimensional distance estimate to a planet, while the star-based measurements provide correctional information on a 2D plane perpendicular to the measurement direction. This extended interpretation is the basis for the 3D active autonomous deep space navigation approach, presented in Sect. 3.2.2 (see Nakath et al. [2018]).



(a) Scan matching of a point cloud patch *source* to a section of Itokawa *target*, resulting in the point cloud *source optimized*.

(b) Scan to scan.

Figure 3.5.: LiDAR-based scan matching on the asteroid Itokawa.

3D Flash-LiDAR

A LiDAR (light detection and ranging) is an active sensing apparatus emitting laser beams, whose reflections in the surroundings are, in turn, recorded. Subsequently, range values can be obtained for the single beams, using a time-of-flight approach. The resulting measurements can be seen as samples from an underlying surface.

A Scanning-LiDAR uses mirrors to gradually deflect a laser beam and obtains the responses with a single sensor, thus it eventually *scans* a scene over time. Opposed to that, the Flash-LiDAR emits one laser beam and uses a sensor array to obtain a full depth image in every dataframe. The latter approach results in a faster frame rate, as the whole scene is captured in one take and additionally in all pixels being correlated in space and time. Furthermore, the absence of mirrors entails the absence of moving parts, which results in lower weight and volume as well as being less prone to errors [Dissly et al., 2012]. Thus, the Flash-LiDAR technology has been considered for numerous deep space applications (see e.g., Amzajerdian et al. [2010]; Christian and Cryan [2013]). Currently, NASA's OSIRIS-REx [Berry et al., 2013] mission is flying a 3D Flash-LiDAR to be employed in the autonomous touch-and-go maneuver for sample retrieval.

LiDAR scans are the only – yet powerful – measurements, as they are used for scan to scan alignment in relative autonomous navigation schemes (see Fig. 3.5b). Since the measurement geometry is known, their alignment provides information about the relation between the respective poses from which the measurements were taken. In addition, they are essential for explicit loop closing, which allows for a globally consistent map within a Graph SLAM scenario. Thus, scan matching, being a technique for aligning the point clouds stemming from two different LiDAR measurements source and target, is of utmost importance for the approach presented in Nakath et al. [2019] (cf. Figs. 3.5a and 3.5b).

In general, the point cloud registration problem is posed as finding an alignment between a target cloud and a source cloud (again, cf. Figs. 3.5a and 3.5b). Each of the clouds contain a set of points to represent the x, y, and z coordinate of an underlying surface sample (which is taken from the asteroid Itokawa in this case).

A simple approach is to run a principal component analysis (PCA) on the point cloud pairs and subsequently align the directions of biggest variance as indicated by the eigenvectors and their corresponding values [Bellekens et al., 2014]. However, as the point distribution within the clouds has to be approximately Gaussian, it is very vulnerable to outliers.

If the correspondences, i.e. the assumed counterpart of a point in the target cloud in the source cloud, are known, a better approach would be to directly minimize the sum of their distances. Formulated as a linear least squares problem, the latter can be solved by singular value decomposition (SVD) [Bellekens et al., 2014]. In practice, however, the measurements suffer from noise and other confounding factors, thus making this approach infeasible due to the perfect data assumption.

To overcome this, Besl and McKay [1992] introduced the Iterative Closest Point (ICP)-algorithm. The key idea is to (i) find a set of correspondences, then (ii) find a transformation based on that set using the SVD-approach, and finally (iii) reject correspondences to create a refined basis for the next transformation estimation. A multitude of variants of this algorithm have been presented since then, mostly modifying the correspondence estimation e.g., *point to point* vs. *point to plane*, and the transformation estimation approach, e.g., *linear* vs. *nonlinear* [Pomerleau et al., 2013; Rusinkiewicz and Levoy, 2001]. The most mentionable extension is the *Generalized ICP* approach by Segal et al. [2009], who formulate the point cloud matching problem within a probabilistic framework and show that all former approaches reduce to mere special cases.

However, the desired scan matching quality in a 1 km (cf. Figs. 3.2b and 3.5b) distance from the center of mass of the asteroid could not be reached with all approaches stated above. Namely, various ICP and GICP-pipelines using implementations from the Mobile Robotics Programming Toolkit (MRPT) [Claraco, 2008] as well as from the well-established Point Cloud Library (PCL) [Rusu and Cousins, 2011a,b] were evaluated. Finally, satisfying results could be obtained using Normal Distribution Transform (NDT), which was originally introduced as a compact representation of surfaces using Gaussians by Biber and Straßer [2003]. For actual scan matching, a mixture model of a normal and a uniform distribution is used to account for outliers in scans [Biber et al., 2004]. This approach was extended to 3D NDT by Magnusson [2009], which is similar to correlative scan matching [Olson, 2008]. This family of approaches seeks to maximize the log-likelihood of a source scan being sampled from a surface representation of the target scan to find an optimal transformation.

To obtain satisfying results in terms of matching quality and speed, the LiDAR-simulation as well as the corresponding 3D NDT scan matching approach are implemented on the GPU. A thorough description of the pipeline is presented in Nakath et al. [2019].

3.2.2. Absolute Active Autonomous Deep Space Navigation

Absolute autonomous deep space navigation denotes the state estimation of a S/C in the heliocentric reference frame without any ground support. In particular, autonomous optical navigation rests on the observations of celestial bodies with CCD-cameras, mounted on board the S/C, which was proposed as the AutoNav system by Bhaskaran et al. [1996]; Riedel et al. [1996].

In Guo [1999] an optical autonomous navigation scheme was introduced, based on observing the radial velocity with respect to the Sun using an interferometer and a corresponding observation of the line of sight (LOS) to the Sun. To obtain the latter, a corresponding instrument, called coupled Sun-star tracker, was developed by Guo [2003]. In Yim et al. [2000], the performance of an EKF was investigated, which draws on observations of the radial velocity, the LOS to Sun, and, in addition, the LOS to Earth. Meanwhile, a novel approach for UKF-based autonomous attitude estimation using gyroscopes was presented by Crassidis and Markley [2003].

In Paluszek et al. [2010] an UKF-based optical navigation system for position and attitude determination combining IMU-measurements and optical observations of celestial bodies, including the Sun, planets, and far stars is presented. And finally, AutoNav was the first optical autonomous navigation system to be actually employed. In NASA's Deep Impact mission, it enabled comet and asteroid fly-by maneuvers, which would be infeasible with sole ground control [Bhaskaran, 2012].

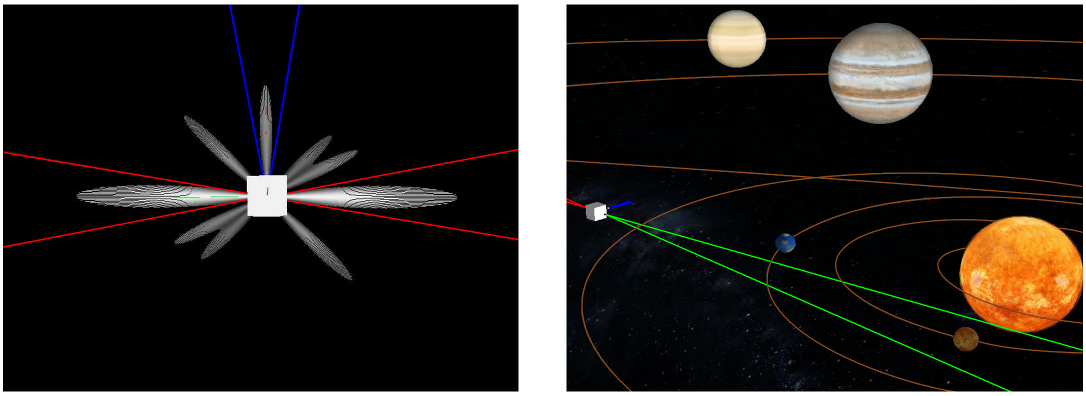
In González Peytaví et al. [2015], the approach by Guo [1999] is combined with the optical observation approaches from Paluszek et al. [2010] and Bhaskaran [2012]. An UKF and a PF are compared for their position estimation performance on various orbit geometries. However, the attitude of the S/C is neglected and it is assumed that the sensors always perceive the observables, i.e., they have a full 360 deg FOV. In the following sections, absolute *active* autonomous deep space navigation approaches will be presented. Here, the attitude of the S/C will be controlled, such that valuable observables appear in the restricted FOV of the particular sensors.

2D Absolute Active Autonomous Deep Space Navigation

In Nakath et al. [2016], an absolute active autonomous navigation approach on an Earth-like 0.8 au orbit around the Sun is proposed (see Fig. 3.6). A PF operating in 2D on the $\boxplus$-manifold estimates position, attitude, and the linear velocity of the S/C with respect to the Sun. A 2D scenario is a reasonable first approximation, as the orbit geometry exhibits the lowest variations in the z-axis.

The S/C can draw on a probabilistic motion model for orbit integration and a star tracker for an absolute attitude estimate. The velocity can be corrected with interferometer measurements and the position with the planet chord, planet-star, and Sun-star measurements as introduced above (see Figs. 3.3b–3.3c). It is assumed that the S/C follows an orbit and only seeks to control the attitude for best possible information intake. The corresponding controls are executed by an abstracted attitude controller, which can directly impact the velocity with a limited 2-norm of 1 deg $/s$.

The active perception approach is modeled in the form of an MDP [Howard, 1960; Cassandra et al., 1996] and solved by value iteration (see Sect. 2.3.1 and Thrun et al. [2005]). Within this scheme, the expected reward for an action is computed by considering the impact of expected observations weighted by their respective likelihoods for each



(a) Reward functions (white cones) and FOV of different sensors (indicated by blue and red lines) of a S/C.

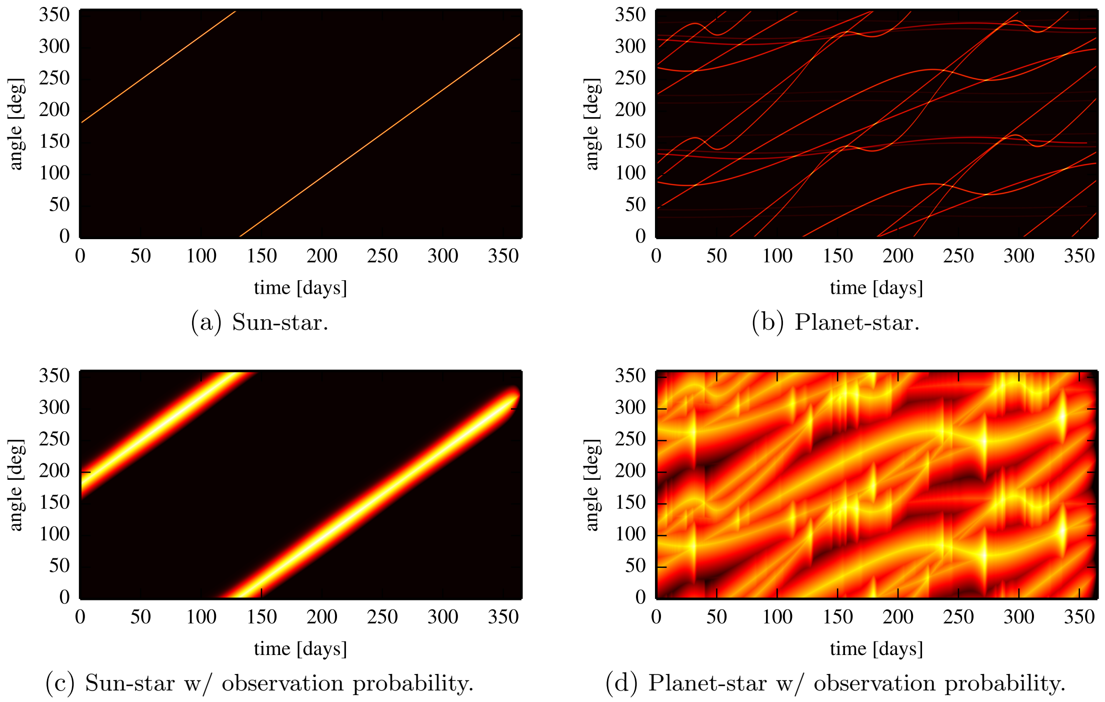
(b) Overview of respective scenario, with Venus in the FOV (green lines).

Figure 3.6.: Reward functions and respective observations in an active autonomous deep space navigation scenario. Figures reprinted from Nakath et al. [2016].

of the sensors. Exemplary reward-functions for the Sun-star sensor and the planet-star sensor are shown in Fig. 3.7. Considering the pairs Figs. 3.7a and 3.7c as well as Figs. 3.7b and 3.7d, the impact of the incorporation of the measurement likelihood can clearly be observed. It is given by a FOV-based evaluation of a cumulative distribution function (CDF) of a normal distribution centered at the position of the observable (see Fig. 3.8a). Additionally, the planet chord measurement is weighted by the respective expected chord angle to prefer nearby planets. Finally, the joint reward of all sensors can be obtained, as depicted in Fig. 3.8b for an exemplary rotation planning space ranging over one year.

Subsequently, the impact of the measurement direction d on the current uncertainty can be estimated by projecting it onto the respective scaled eigenvectors λ of the current covariance matrix Σ_t (see 3.9a). The latter is obtained by fitting a Gaussian to the particle distribution on the $\boxplus$-manifold (see Sect. 2.2.3 and Fig. 2.5b). The expected impact on the uncertainty ellipsoid by the Sun-star and planet-star measurements is computed orthogonally to the respective measurement direction d. This approach accounts for the fact that the correctional information provided by this kind of measurements is effective in the orthogonal direction of the position of the observables relative to the S/C (please cf. Sect. 3.2.1 – Sun and Planet Sensors and cf. Figs. 3.3a–3.3c, Fig. 3.4, as well as Figs. 3.3e and 3.3f). This approach allows to directly estimate the impacts of the measurements on the particular dimensions of uncertainty. Thus, a computationally involved update step over all expected measurements can be omitted. Such a step has to be performed for canonical active perception solutions based on expected entropy (cf. e.g., Roy et al. [1999] and Fig. 2.6, as well as Sect. 2.3.2).

The value of an action, which constitutes the basis for the planning space, is then given by the reward discounted by the respective rotation delta induced by an action.



(a) Sun-star.

(b) Planet-star.

(c) Sun-star w/ observation probability.

(d) Planet-star w/ observation probability.

Figure 3.7.: Reward functions for different sensors in the planning space 360 deg × 1 year. Figures reprinted from Nakath et al. [2016].

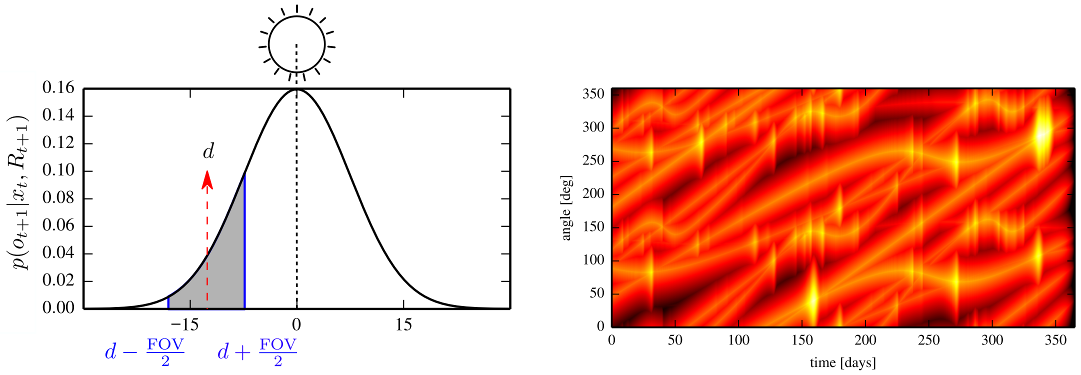
To make planning tractable in that space, it is discretized in its temporal and angular dimensions.

For evaluation, a planning horizon of one week is chosen. Thus, a random agent picks a random observable to follow for one week. A heuristic engineering agent alternates between observing the Sun or the planet with the biggest chord for one week. And finally, an active agent uses the policy derived by value-iteration to pick an observable. The control policy is re-planned every week, to ensure a common decision frequency of all agents.

In Nakath et al. [2016] it is shown, that an active perception approach outperforms random observations of celestial bodies as well as the engineering solution of observing the planet with the biggest chord in alternation with the Sun. The increased performance can be observed with respect to the precision of the position estimate, but also with respect to the cumulative rotations, i.e., the energy spent.

3D Absolute Active Autonomous Deep Space Navigation

In Nakath et al. [2018] an improved active perception approach for absolute navigation of an autonomous S/C in 3D is presented. Different scenarios are simulated within a custom-build simulator based on the Robot Operating System (ROS) [Quigley et al., 2009]. They comprise versatile orbit geometries ranging from an Earth-like orbit (semi-



(a) Cumulative distribution function (CDF) to model the observation likelihood of a celestial object.

(b) Joint reward function in 2D in the planning space 360 deg × 1 year.

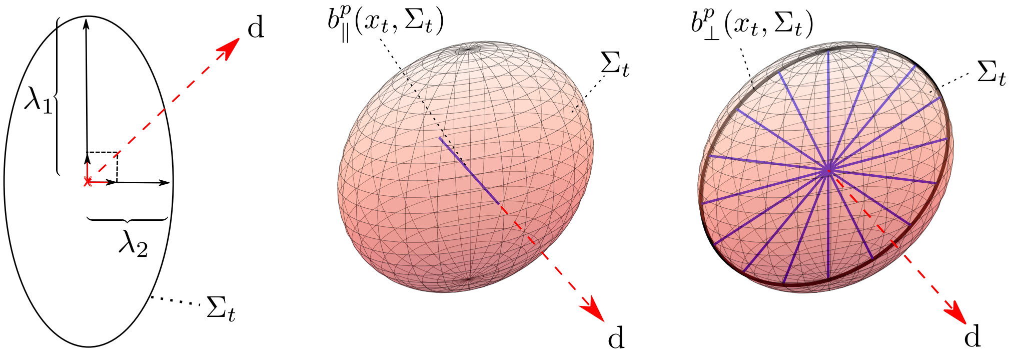
Figure 3.8.: Observation likelihood computation and joint reward function. Figure (a) adopted from Nakath et al. [2018], Figure (b) reprinted from Nakath et al. [2016].

major axis 1.2 au, eccentricity: 0.25, inclination: 50 deg), over an intermediate orbit (semimajor axis 2.0 au, eccentricity: 0.125, inclination: 25 deg), to the KaNaRiA PO (semimajor axis 2.8 au, eccentricity: 0.0, inclination: 0.0 deg). The varying geometries of the orbits entail big variations of the observable's positions relative to the S/C, as well as big differences in the velocity of the S/C itself.

The state is estimated by different implementations of Bayesian filters and comprises the position, the attitude (given as a direction cosine matrix (DCM)), and the velocity. Namely, the filters employed for recursive state estimation are an EKF [Clemens and Schill, 2016], an UKF [Hertzberg et al., 2013], and a PF [González Peytaví et al., 2015; Nakath et al., 2016], jointly adopted to the ⊞-manifold. The latter uses 500 particles and an improved proposal distribution [Grisetti et al., 2007] based on the interferometer and star tracker likelihoods.

For state estimation, the S/C's sensor fusion module can draw on an interferometer for radial velocity measurements (see Fig. 3.3d) and a star tracker for absolute attitude measurements (see Fig. 3.3h). For position correction, a Sun-star sensor, a planet-star sensor, as well as a planet chord sensor are available (see Figs. 3.3b–3.3c, respectively). The latter is re-designed to exhibit a dynamic noise, dependent on the distance to the measured object. This results in a better information quality of nearby celestial objects. Additionally, this implies, that the preference for nearby planets only ensured versatile observations in different directions in the setting presented above (cf. Sect. 3.2.2).

As before, the S/C follows its respective orbit and seeks to assume attitudes which are beneficial for obtaining correctional information. Unlike above, this is a greedy approach, which only considers the attitude of the next step, i.e, the planning horizon is 1. The expected reward for the single sensors is computed as described above (see Sect. 3.2.2) and subsequently weighted with the respective observation likelihood (see Fig. 3.8a). The latter is computed on two orthogonal planes, as the visibility of a celestial object has to be estimated in 3D.



(a) 2D impact estimation of measurement based on projection of its direction d on the single eigenvectors λ_1 and λ_2.

(b) 3D impact estimation of 1D correctional information, coincinding with the measurement direction d by projection onto d.

(c) 3D impact estimation of 2D correctional information on a plane perpendicular to the measurement direction d by iterative projection on vectors perpendicular to d.

Figure 3.9.: Estimation of impact on uncertainty of measurements in different directions and with different correctional potential. Figure (a) adopted from Nakath et al. [2016], Figures (b) and (c) adopted from Nakath et al. [2018].

Again, the directional observations are inspected for their expected impact on the current uncertainty ellipsoid Σ_t to serve as a weighting factor. It can be obtained directly from the EKF and UKF, as they assume a Gaussian state distribution. In the case of the PF, the respective parameters μ_t and Σ_t are fitted on the $\boxplus$-manifold, as introduced above (cf. Sect. 3.2.2 and Figs. 2.5a and 2.5b).

As the active perception scheme is extended to 3D, the impact estimation of measurements has to be reconsidered. The planet chord measurement can still be thought of as one-dimensional correctional information, effective in the measurement direction with respect to the current uncertainty ellipsoid Σ_t. However, the remaining two dimensions are now corrected by Sun-star and planet-star measurements on a plane perpendicular to the measurement direction (again, please cf. Sect. 3.2.1 – Sun and Planet Sensors and cf. Figs. 3.3a–3.3c, Fig. 3.4 as well as Figs. 3.3e and 3.3f).

To additionally avoid to solve for the eigenvectors and corresponding eigenvalues of the covariance Σ_t in every step, the latter is directly projected onto the measurement direction (cf. Figs. 3.9a and 3.9b). To estimate the covariance volume intersecting with the 2D plane perpendicular to the measurement direction, a circular integration is performed. To speed this operation up, angle discretization is employed and only the half sphere is integrated, as the covariance is symmetric (cf. Figs. 3.9b and 3.9c). Although implemented differently, the same idea as above is pursued: As the expected impact of the measurements in the respective dimensions of the current covariance Σ_t is directly measured, no counterfactual updates with the expected measurements have to be conducted. This approach generally reduces computational cost for multiple updates and it can be employed when it is difficult to obtain the instantiations of the expected measurements or the processing of such measurements is computationally involved.

The optimal attitude is then evaluated in terms of a setpoint, i.e., the desired attitude the S/C should assume. To find the optimal setpoint in 3D, two angles are iterated and in every step the corresponding reward is computed for each sensor. To yield a significant speedup, discretized pre-computed poses in an Open Multi-Processing (OpenMP) [Chapman et al., 2008] based parallelization scheme are used.

This setpoint is subsequently passed to the $\boxminus$-controller operating on the $\boxplus$-manifold presented in Nakath et al. [2017]. As it is a proportional-derivative attitude controller, it computes a control torque from the weighted sum of the proportional term, i.e., the offset between the current attitude and the desired setpoint, as well as the derivative term, i.e., the current angular velocity [Bullo and Murray, 1995; Chaturvedi et al., 2011]. As the state is defined on a compound $\boxplus$-manifold (see Sect. 2.2 – Manifold State Representation), the velocity is given in vector space, while the attitude is given in overparametrized form. When the offset between the attitude and the setpoint is computed with the $\boxminus$-operator, the shortest path on the manifold is returned in the vector space [Hertzberg et al., 2013]. Thus, the final control torque can be obtained with simple vector space operations.

As it is a greedy approach, an inhibitive behavior is induced to keep the S/C from frequent target switching, i.e., instable behavior. To this end, a weighting factor is introduced which penalizes large attitude changes for most of the time, while leaving a window for large attitude changes with a fixed frequency. Finally, the so-weighted rotational costs are subtracted from the expected rewards for every attitude to obtain their respective value.

In Nakath et al. [2018] a numerical evaluation is conducted on the orbits stated above. The performance is measured in terms of the positioning error in the heliocentric reference frame. The EKF and UKF behave very similar to each other, thus the improved linearization performance of the UKF seems not to be of a big advantage. The PF has the worst performance throughout all conditions, using 500 particles. In general, the three filters behave in a similar way throughout the various conditions, where the general localization performance decreases with increasing orbit size.

Again, three agents are compared to each other: The random agent picks a setpoint corresponding to alternatively observing the Sun and a random planet. The heuristic engineering agent alternates between the Sun and the planet with the biggest expected chord angle. And finally, the active agent simply picks the actions with the best value in every timestep. The first two agents pick their new observable every 5.2 days. To yield synchronized actions, the attitude penalization factor of the active agent is set to the same value as well.

The evaluation shows, that the active agent approach outperforms the two other agents. In addition, it can be observed that the more uncertain the state estimate is, the more beneficial the active perception approach becomes. The PF benefits the most throughout all conditions, and the effect steadily increases for all filters, the bigger the orbit becomes. This empirical finding complies with the theoretical consideration that the active perception scheme controls the influx of correctional information.

All active perception approaches presented so far are commonly operating in an absolute – i.e., heliocentric – frame of reference with known models of the observables. When relative navigation with respect to a celestial object with unknown shape is required, active SLAM-approaches provide a good choice. Such an approach – tailored to the relative navigation in the vicinity of an asteroid – is presented in the following.

3.2.3. Relative Active Autonomous Deep Space Navigation

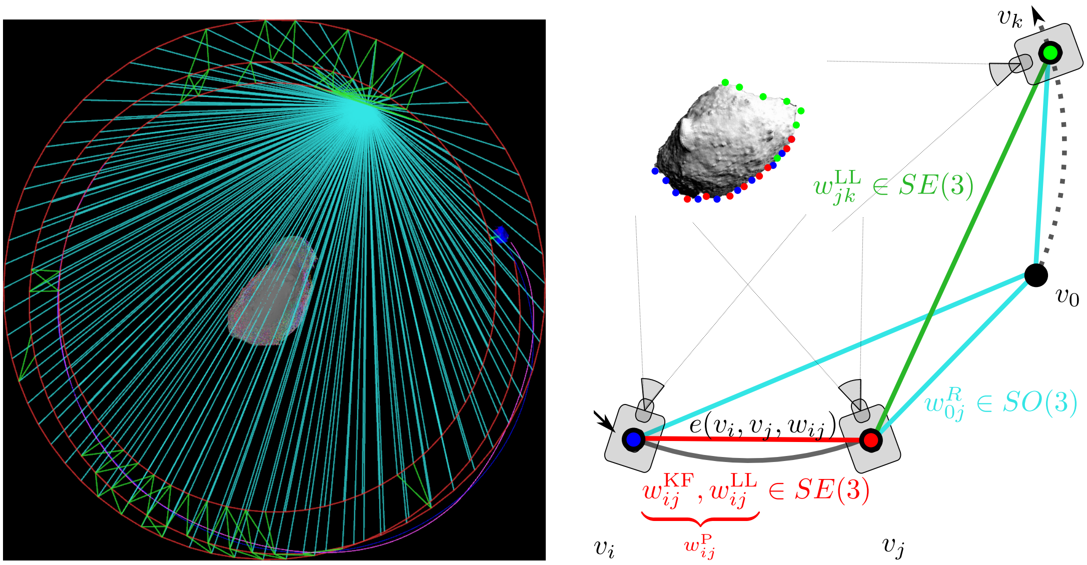
Relative autonomous deep space navigation denotes the state estimation relative to an object, which itself is moving with respect to the heliocentric reference frame (see Fig. 3.2b). Such an autonomous navigation scheme is relevant for docking maneuvers with other S/C or space stations [D'Souza, 2014], but also for navigation in the vicinity of celestial objects.

The latter scenario is examined in the vicinity of a hypothetical asteroid, within a custom-build simulator based on ROS [Quigley et al., 2009]. The asteroid is considered to be part of the main belt, to comply with the KaNaRiA mission concept. As no precise models from asteroids in this distance are available, they have to be established in-situ by the S/C itself. At the same time, the simulation is posed with the problem of presenting a realistic environment to the S/C. To ensure the latter, actual models from the encounter of the asteroid Itokawa in 2005, which has a 1.3 au mean distance from Earth, are used. They comprise spherical harmonics to simulate the gravitation [Klosko and Wagner, 1982; Scheeres et al., 2006] and a 3D shape model, which is used for the LiDAR simulation [Yoshikawa et al., 2006].

Again, a naive approach to the autonomous navigation problem in such a scenario would be to define adequate equations of motion and subsequently integrate them in order to provide a relative position estimate. However, this approach would very fast lead to an accumulating error, as the tolerances within this scenario are small. This is due to the fact that the S/C reacts sensitively to controls, as it is affected by a weak, inhomogeneous, and rotating gravitational field. Additionally, when the sensors are not able to keep the asteroid in their respective FOV anymore, the ability for relative correction is completely lost, resulting in a rapid decrease of positioning quality.

Thus, more involved approaches to mapping and pose estimation, or trajectory estimation, i.e., a sequence of poses, have been recently considered for this scenario. In Pesce et al. [2018] a POMDP-based trajectory selection scheme for mapping of small celestial bodies in deep space is presented. While Dietrich and McMahon [2016] show the feasibility of Flash-LiDAR-based relative navigation around the asteroids Itokawa and Bennu. In a subsequent publication, a robust solution, which, however, presupposes a previously known low-fidelity asteroid shape model, is presented [Dietrich and McMahon, 2018].

Those approaches, however, consider mapping and trajectory estimation as separate processes and thus do not leverage the mutual correctional potential of the map and the trajectory distributions. The beneficial joint estimation of those distributions is called SLAM (see Sect. 2.2). In [Cocaud and Kubota, 2010, 2012] a Monocular RBPF-SLAM using Speeded Up Robust Features (SURF)-based [Bay et al., 2006, 2008] landmarks for relative navigation in a 700 m distance around Itokawa is presented. Andert et al.


v_k
$w_{jk}^{\mathrm{LL}} \in SE(3)$
v_0
$w_{0j}^{R} \in SO(3)$
$e(v_i, v_j, w_{ij})$
$\underbrace{w_{ij}^{\mathrm{KF}}, w_{ij}^{\mathrm{LL}}}_{w_{ij}^{\mathrm{P}}} \in SE(3)$
v_i
v_j

(a) Graph structure as seen in the simulation system.

(b) Scetch of graph structure.

Figure 3.10.: Graph structure of active asteroid SLAM. Figures (a) and (b) have the same color coding, i.e., the red edges denote pose offset measurements in $SE(3)$, the cyan edges denote attitude offset measurements in $SO(3)$, and the green edges denote pose offset measurements in $SE(3)$ resulting from loop closing. Figures reprinted from Nakath et al. [2019].

[2015] also employ RBPF-SLAM, where the landmarks are based on crater models whose parameters are estimated, based on camera and LiDAR data.

However, those are passive approaches, where the S/C does not actively choose beneficial trajectories. The term active SLAM was first coined by Feder et al. [1999] and describes the active selection of actions, which are beneficial either for the quality of the trajectory estimate or the map estimate. Such an approach was already used in a scenario of relative pose estimation of a chaser satellite, where the integrated control, planning, and state estimation of a S/C is considered [Kontitsis et al., 2016]. However, the pose estimation is conducted relatively to only four landmarks and the scenario is confined to the 2D-case. In Nakath et al. [2019], Graph SLAM-based relative navigation combined with an active SLAM approach within a realistic scenario is presented. The S/C can draw on the IMU for the incorporation of the controls, the star tracker for an attitude estimate, and the Flash-LiDAR for relative pose offset estimates as well as mapping. A numerical evaluation is conducted in the relative navigation scenario introduced above. It shows that the norm of the localization error over all axes never exceeds 40 m, thus complying with the small error margins of this scenario. Furthermore, the trajectories can be actively chosen, such that they either favor the exploration of the map or localization performance. Finally, the trajectory selection process can be dynamically controlled, based on a position-uncertainty measurement within the graph. This results in trajectories which are slightly more beneficial for exploration as well as localization

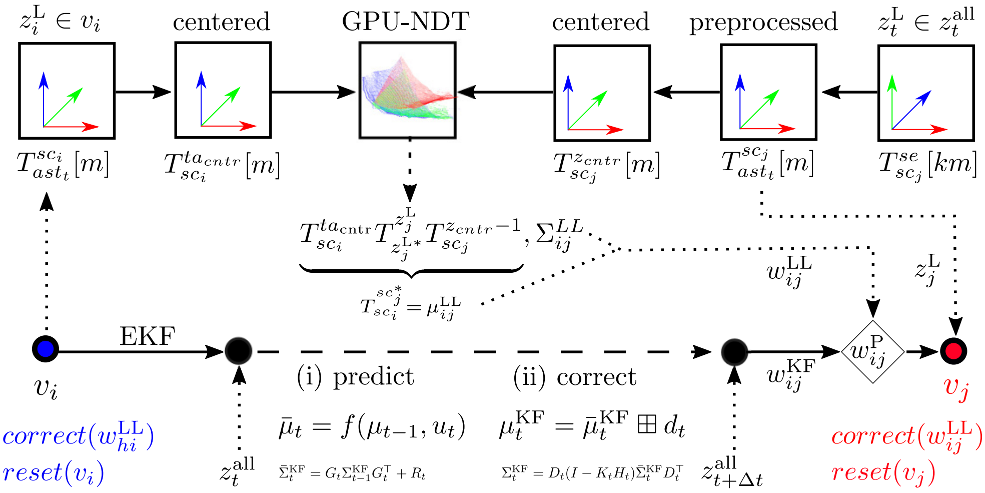

$z_i^{\mathrm{L}} \in v_i$
centered
GPU-NDT
centered
preprocessed
$z_t^{\mathrm{L}} \in z_t^{\mathrm{all}}$
$T_{ast_t}^{sc_i}[m]$
$T_{sc_i}^{ta_{cntr}}[m]$
$T_{sc_j}^{z_{cntr}}[m]$
$T_{ast_t}^{sc_j}[m]$
$T_{sc_j}^{se}[km]$
$\underbrace{T_{sc_i}^{ta_{\mathrm{cntr}}} T_{z_j^{\mathrm{L}*}}^{z_j^{\mathrm{L}}} T_{sc_j}^{z_{cntr}-1}}_{T_{sc_i}^{sc_j^*} = \mu_{ij}^{\mathrm{LL}}}, \Sigma_{ij}^{LL}$
w_{ij}^{LL}
z_j^{L}
EKF
v_i
(i) predict
(ii) correct
w_{ij}^{KF}
w_{ij}^{P}
v_j
$correct(w_{hi}^{\mathrm{LL}})$
$reset(v_i)$
z_t^{all}
$\bar{\mu}_t = f(\mu_{t-1}, u_t)$
$\bar{\Sigma}_t^{\mathrm{KF}} = G_t \Sigma_{t-1}^{\mathrm{KF}} G_t^\top + R_t$
$\mu_t^{\mathrm{KF}} = \bar{\mu}_t^{\mathrm{KF}} \boxplus d_t$
$\Sigma_t^{\mathrm{KF}} = D_t(I - K_t H_t)\bar{\Sigma}_t^{\mathrm{KF}} D_t^\top$
$z_{t+\Delta t}^{\mathrm{all}}$
$correct(w_{ij}^{\mathrm{LL}})$
$reset(v_j)$

Figure 3.11.: Tightly-coupled graph construction based on two relative offset measurements. The upper row shows the processing pipeline of the LiDAR-scans, while the lower row shows the recursive state estimation by the EKF. Figure reprinted from Nakath et al. [2019].

performance. In addition to that, the map is not only used for relative navigation, but also for the discovery of safe landing sites by postprocessing the grid map obtained by SLAM.

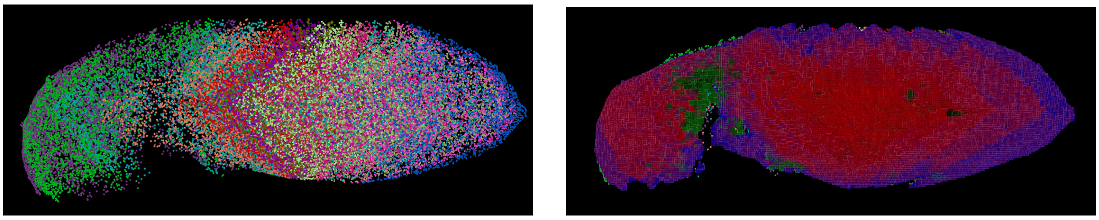
In the following, the Graph SLAM approach will first be presented, followed by a description of the active perception approach.

Graph SLAM for Relative Autonomous Deep Space Navigation

As introduced in Sect. 2.2, Graph SLAM approaches may roughly be divided in (i) a frontend, which builds a graph with vertices connected by edges representing constraints, and (ii) a backend, which solves a nonlinear system of equations derived from the graph structure. In Nakath et al. [2019] the latter is provided by the g^{2}o-framework [Kümmerle et al., 2011]. It is augmented with $\boxplus$-based edges constraining either the attitude of two vertices in $SO(3)$ or the whole pose between two vertices in $SE(3)$ (see Fig. 3.10b).

The vertices and edges are inserted into the graph by the construction frontend, thus incrementally building a graph structure to be optimized. Unlike loosely-coupled approaches employed, e.g., in Hector SLAM [Kohlbrecher et al., 2013], a tightly-coupled graph construction approach is chosen (see Fig. 3.11). This entails that the EKF is operating within the graph structure and, in turn, benefits from correctional measurements obtained from the latter. The graph construction process is thus a heavily intertwined interaction of recursive state estimation and smoothing within the graph structure. It is designed as follows.

Initially, a vertex v_0, aligned with the navigation frame, is inserted into the graph-structure to serve as a reference for the attitude measurements. Every time the S/C traveled a certain distance or significantly changed its attitude with respect to the asteroid, a new vertex is inserted. For all these following vertices, two relative offset



(a) Point cloud map.

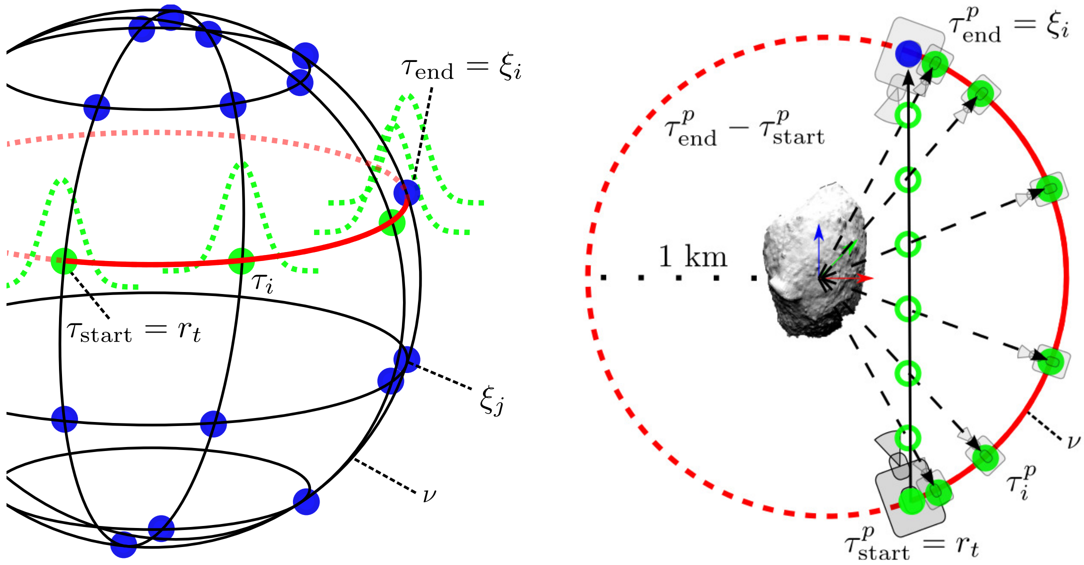
(b) Extracted evidential grid map.

Figure 3.12.: Point Cloud map and corresponding evidential grid map of a section of Itokawa. Figures reprinted from Nakath et al. [2019].

measurements between vertices v_i and v_j are available. First, an EKF is *placed* at the pose of v_i and recursively estimates the offset until v_j is to be inserted, resulting in a relative offset measurement denoted w_{ij}^{KF}. Second, a scan matching result w_{ij}^{LL} can be obtained, based on the LiDAR scans taken at the poses encoded by v_i and v_j (cf. Fig. 3.5). The scan matching result is validated against the EKF estimate for outliers, by ensuring that the pose-offset is below a certain threshold. If the validation is successful, the measurements are merged into a pose-measurement w_{ij}^{P} to avoid a slow optimization process by inserting two pose-edges between consecutive vertices. If the validation fails, only the EKF estimate is written into w_{ij}^{P} and scan matching is re-attempted in the next step. In either case, the measurement w_{ij}^{P} is the constraining measurement for the pose-edge connecting v_i and v_j. In addition, the current star tracker measurement w_{0j}^{R} is inserted into the graph represented by an attitude-edge between the newly inserted vertex v_j and the reference vertex v_0. As the state vector of the EKF comprises the pose and the linear velocity, the latter has to be corrected to avoid the dead-reckoning problem. Thus, upon successful validation of the scan matching result obtained in the graph structure, w_{ij}^{LL} is used in a Kalman-filter update to yield an indirect correction of the velocity estimate. Finally, the EKF is reset to the new pose encoded in v_j by accordingly setting the mean and covariance of its state estimate. Subsequently, explicit loop closing is attempted, using a validate nearest scheme [Olson, 2008]. Finally, an optimization run is carried out in the backend, when a new vertex has been inserted or the graph has not been optimized for a certain time.

The approach presented above yields an optimized trajectory estimate relative to the asteroid, as well as a maximum likelihood scan map of the latter (see Fig. 3.12a). The single scans are subsequently projected into an evidential grid map (see Fig. 3.12b). Thus, the active perception approach can enjoy an extended uncertainty representation based on belief functions (see Sect. 2.2). In addition, the grid map is based on information obtained in a smoothing approach, and is thus corrected, based on all measurements. Hence, it is different to the mapping result in the RBPF-based Evidential FastSLAM presented in Reineking and Clemens [2013]; Clemens et al. [2016], where the first order Markov assumption holds.

In the SLAM community, it is accepted that an EKF-based SLAM approach ultimately converges to a fully correlated map, in the limit of an infinite amount of measurements


$\tau_{\text{end}} = \xi_i$
τ_i
$\tau_{\text{start}} = r_t$
ξ_j
ν
$\tau^p_{\text{end}} = \xi_i$
$\tau^p_{\text{end}} - \tau^p_{\text{start}}$
1 km
ν
τ^p_i
$\tau^p_{\text{start}} = r_t$

(a) Evaluation of trajectories.

(b) Generation of pseudo-trajectories.

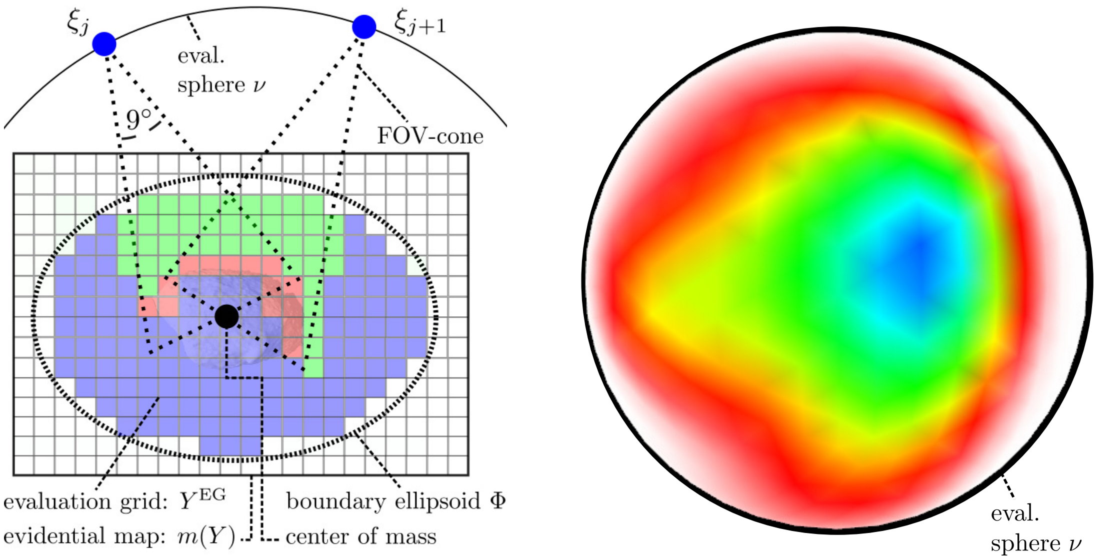
Figure 3.13.: Approaches to approximate trajectory generation and evaluation. Figures reprinted from Nakath et al. [2019].

and time [Dissanayake et al., 2001]. However, simply obtaining a high number of observations is not a feasible approach, as energy and time are extremely scarce resources in deep space missions. Thus, a SLAM approach was developed, which actively trades-off exploration gain and localization gain. In the following, it will be described how the SLAM-based trajectory estimate and evidential grid map are processed for active perception.

Active Perception for Relative Autonomous Deep Space Navigation

The aim of the active perception approach presented in Nakath et al. [2019] is to trade-off between actions yielding localization gain or exploration gain. The approach rests on the assumption, that re-visiting already mapped areas is beneficial for the localization performance, as it provides the Graph SLAM algorithm with opportunities for explicit loop closing. This is due to the fact, that within the Graph SLAM approach consecutive poses are locally corrected by scan-to-scan measurements, but only loop closing allows for a globally consistent map which implicates an improved trajectory estimate.

In general, the active perception scheme works as follows. First, approximate pseudo trajectories (see Fig. 3.13b) are generated from the current position to all evaluation points ξ_i which have not yet been visited. All resulting trajectories are evaluated for their corresponding exploration gain and localization gain (see Fig. 3.13a). From this coarse but rapid evaluation, the best trajectories are picked, and the respective end points ξ_i are passed to the optimal trajectory module. The latter uses the ESA-NLP solver *We Optimize Really Huge Problems* (WORHP) [Büskens and Wassel, 2012] to



(a) Evaluation structure.

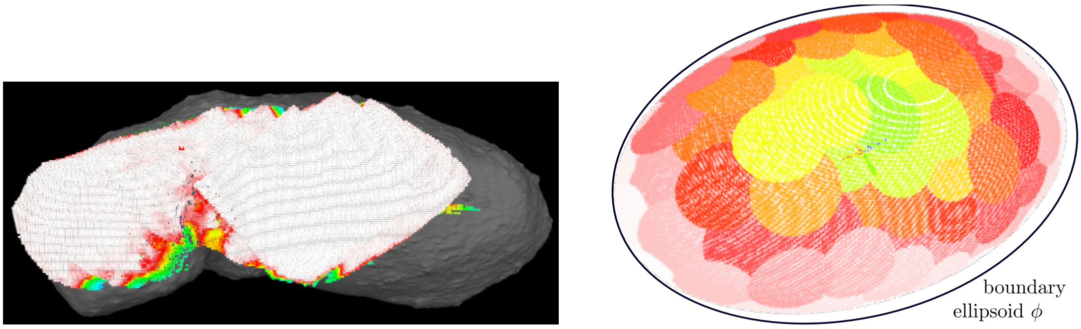
(b) Exploration reward projected onto the evaluation sphere.

Figure 3.14.: Rapid approximation of a reward function based on evaluation nodes ξ_j. The color coding for Figure (a) is blue=unknown, red=occupied, green=free, while for (b) it is a gradient ranging from white corresponding to $\max_i m(Y_i = \Theta_Y)$ to blue corresponding to $\min_i m(Y_i = \Theta_Y)$. Figures reprinted from Nakath et al. [2019].

generate corresponding optimal trajectories. They are subsequently evaluated for their respective information and localization gain values and additionally penalized for frequent directional changes. The best trajectory is selected, based on the current goal of the S/C. Subsequently, it is traversed and its respective endpoint ξ_i is deleted from the set of available target points. When a trajectory is about to be finished, the whole process is restarted. The current goal can be defined by setting a parameter either to 0 for full exploration gain or 1 for full localization gain. In addition, this parameter can be dynamically computed from the position uncertainty marginalized from the graph structure [Grisetti et al., 2010a], and a subsequent measurement of the volume of the hyperellipsoid constituting its boundary (see Sect. 2.3.2) [Carrillo et al., 2012].

Thus, it is not a greedy approach, as it does not act step-wise. However, it can only choose from a discrete set of plans with a dynamic planning horizon. The latter is due to the fact that the length of the trajectories depend on the distance between the starting point τ_{start} and the end of the trajectory τ_{end} and the respective solution provided by WORHP.

In particular, to implement this approach the *actions* to be performed initially have to be generated, as they consist of a trajectory, i.e., a sequence of poses to be assumed over time. Then, those trajectories have to be evaluated for their value with respect to localization and exploration performance in a real-time capable fashion. A common structure and basic assumptions underlie the approach to those two tasks.



(a) Accumulated non-specitivity shown as the evaluation grid map Y^{EG} on Itokawa.

(b) Footstamps of the FOV-cone projected onto the boundary ellipsoid ϕ.

Figure 3.15.: Representations of exploration space. The color coding for Figures (a) and (b) is a gradient ranging from white corresponding to $\max_i m(Y_i = \Theta_Y)$ to blue corresponding to $\min_i m(Y_i = \Theta_Y)$. Figures reprinted from Nakath et al. [2019].

It is assumed that position and orientation of the asteroid are known. In addition, for rapid evaluation, a boundary ellipsoid ϕ is placed around the asteroid, in turn yielding a smaller evaluation grid map Y^{EG} (see Fig. 3.14a). Then, an evaluation sphere denoted as ν with a radius of 1 km is placed around the boundary ellipsoid, which is subsequently tessellated, yielding a set of evaluation nodes ξ_j (see Fig.3.13a). To mimic an information intake process from every ξ_j, a FOV-cone with the size corresponding to the LiDAR's FOV is attached to every node, pointing to the center of mass (COM) of the asteroid. The FOV-cone ends at the COM and the indices for cells inside the FOV are pre-computed and thus can be rapidly considered for evaluation (again, see Fig. 3.14a). The latter approximation rests on the assumption, that the asteroid has no cavities, i.e., that it is not possible to look behind the surface, or put in other words, to look *inside* it from any other pose. Otherwise, it would have been necessary to employ some kind of ray-casting scheme to extract the actual surface from the map.

To be capable of selecting trajectories in real-time, a two-step trajectory generation scheme has been devised. Initially, a rapid coarse estimation of all possible trajectories is conducted, followed by a slower computation of a subset of optimal trajectories. Both types of trajectories share a common basic structure, consisting of a starting pose τ_{start}, intermediate poses τ_{i}, and the final pose τ_{end}.

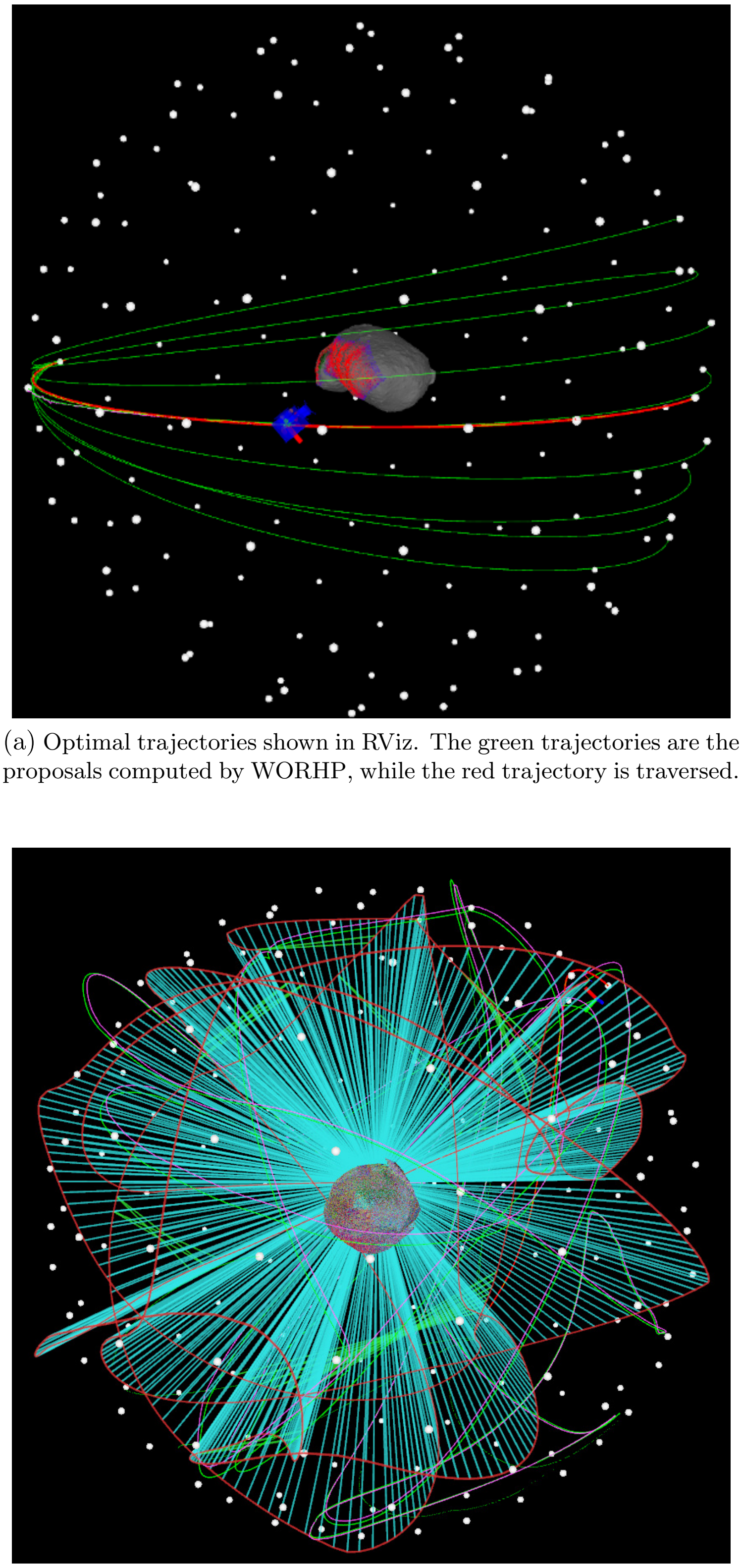
First, so-called pseudo-trajectories are generated from the current position r_t of the S/C to all possible end points, coinciding with the evaluation nodes ξ_i. This is done by spreading equidistant points on a direct connection between the start τ_{start} and the end τ_{end} of a trajectory (see Fig. 3.13b). Subsequently, those points are projected to the evaluation sphere ν to yield an approximation of the geodesic connection between τ_{start} and τ_{end} (cf. Figs. 3.13b and 3.14a).

Those trajectories are evaluated in the evaluation sphere ν for their reward as depicted in Fig. 3.13a. This is done by inserting the trajectory into the evaluation structure. Subsequently, for each trajectory node τ_i the rewards of all evaluation nodes ξ_j

are accumulated with an exponential distance decay. The rewards of the evaluation nodes ξ_i themselves are, in turn, obtained by the mean values of the particular dimensions of uncertainty in the cells lying within their respective FOV-cones (cf. Figs. 3.14a and 3.15b). For the exploration reward, the according measure of uncertainty is the non-specificity, i.e., $m(Y_i = \Theta_Y)$, which measures the belief mass neither assigned to the belief that a cell is occupied or free [Klir, 2005; Reineking and Clemens, 2014]. For the localization reward, simply the occupancy value of a cell is returned, which is measured by the normalized mass, assigned to the belief that a cell is occupied [Reineking and Clemens, 2014]. In Fig. 3.15a the non-specificity dimension of the evaluation grid map Y^{EG} is shown, Fig. 3.15b depicts the corresponding footstamps of the FOV-cones on the boundary ellipsoid ϕ, and finally Fig. 3.14b shows the corresponding non-specificity-values projected onto the evaluation-sphere ν.

In a second step, a set of optimal trajectories, with a subset of the evaluation points ξ_i as targets, is computed. They are obtained by WORHP [Büskens and Wassel, 2012], under consideration of the actual dynamic conditions around Itokawa as proposed by Schattel et al. [2016, 2017]; Schattel [2018]. The evaluation of the optimal trajectories again considers the rewards for exploration gain and localization gain in the fashion described above. In addition, the trajectories are penalized for their curvature as measured by the pairwise ratio of orthogonality of the controls, which are additionally encoded in the optimal trajectories. This avoids abrupt directional changes of the control direction, which turned out to be disadvantageous for the localization performance.

Finally, Fig. 3.16 shows simulator screenshots from an evaluation run, visualized in the tool ROS Robot Visualizer (RViz) [Kam et al., 2015]. In Fig. 3.16a one out of ten optimal trajectories proposed by WORHP is traversed by the S/C, while in Fig. 3.16b the final graph structure after a complete exploration of Itokawa is shown.



(a) Optimal trajectories shown in RViz. The green trajectories are the proposals computed by WORHP, while the red trajectory is traversed.

(b) Final graph structure shown in RViz.

Figure 3.16.: Simulation screenshots. Figures reprinted from Nakath et al. [2019].

4 – Excursion – Active Sensorimotor Object Recognition

This chapter will take an excursion into the topic of active sensorimotor object recognition. The latter can be thought of as a bio-inspired probabilistic object classification approach with online feature selection. Initially, a very brief history of the study of human visual perception will be given in Sect. 4.1.1. Subsequently, computational models of sensorimotor perception, building on the findings achieved in the course of time, will be presented in Sect. 4.1.2. Based on the insights of Sect. 4.1, a sensorimotor feature will be derived in Sect. 4.2 to be used in a probabilistic classification scheme stated in Sect. 4.2.1. In addition, several active perception approaches will be presented in Sect. 4.2.2. They foster classification speed in a process driven by either top-down or bottom-up acquired information. In the following, the publications covered in this chapter are listed.

Kluth et al. [2013] Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2013). Sensorimotor Integration Using an Information Gain Strategy in Application to Object Recognition Tasks. In *Perception*, volume 42, pages 223–223. Pion LTD 207 Brondesbury Park, London NW2 5JN, UK, doi: 10.1177/03010066130420s101.

My share is 40%.
Published at a conference.
Attached in Appx. A.7 (pp. 201ff).

In this poster and a corresponding abstract, the first implementation of the active sensorimotor object recognition system was presented. I partly designed the research, implemented the system, conducted the evaluation, and designed the poster. It was accepted for presentation at the 36th European Conference on Visual Perception ECVP 2013 in Bremen, Germany and the corresponding abstract was subsequently published.

Nakath et al. [2014] Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., and Schill, K. (2014). Active Sensorimotor Object Recognition in Three-Dimensional Space. In *Spatial Cognition IX*, pages 312–324. Springer, doi: 10.1007/978-3-319-11215-2˙22.

My share is 55%.
Published at a peer-reviewed conference.
Attached in Appx. A.8 (pp. 203ff).

In this conference paper, a comprehensive description and evaluation of the sensorimotor object recognition system is given. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the Spatial Cognition IX (International Conference on Spatial Cognition 2014), Bremen, Germany.

Kluth et al. [2014] Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2014). Affordance-Based Object Recognition Using Interactions Obtained from a Utility Maximization Principle. In *Computer Vision-ECCV 2014 Workshops*, pages 406–412. Springer, doi: 10.1007/978-3-319-16181-5˙29.

My share is 40%.
Published at a peer-reviewed conference.
Attached in Appx. A.9 (pp. 217ff).

In this workshop paper, the active classification system is interpreted and evaluated in terms of visual affordances. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the Second Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis at the European Conference of Computer Vision (ECCV), Zurich, Switzerland.

Reineking et al. [2015] Reineking, T., Kluth, T., and Nakath, D. (2015). Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification. In *Computer Analysis of Images and Patterns*, pages 350–361. Springer, doi: 10.1007/978-3-319-23192-1˙29.

My share is 20%.
Published at a peer-reviewed conference.
Attached in Appx. A.10 (pp. 223ff).

This conference paper presents an active classification system using kernel density estimation to obtain the likelihoods. In an evaluation, different top-down as well as a bottom-up approaches were compared. I partly designed the research, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015), Valetta, Malta.

4.1. Perception and Action

As shown in the previous chapter, there is a heavily intertwined relation between perception of the outside world and the actions an autonomous system carries out. Thus, algorithmic concepts can – inter alia – benefit from the studies of the process of human visual perception which exhibits the same interactional properties. Those kind of studies have already been conducted for a long time in the history of science.

4.1.1. Studies of Visual Perception

As early as 300 B.C. Euclid tried to explain problems related to the perception of the size of objects with a beam emitted *from* the eyes [Freedheim and Weiner, 2003]. Roughly 900 years later, Alhazen corrected this flawed notion and postulated that visual perception is actually induced by light falling into the eyes [Freedheim and Weiner, 2003]. In addition, he found that the actual perceptual process does not take place in the eye itself, but in the brain. As a proof, he showed that the actual outcome of the perceptual process depends on the experience of the observer. A further mentionable finding is the distinction between foveal and peripheral vision, introduced by da Vinci [Freedheim and Weiner, 2003]. And finally, Newton came to the conclusion that the perception of color depends on the wavelength of light, after studying the refraction of light by prisms [Freedheim and Weiner, 2003].

Building on those findings, Von Helmholtz [1867] proposed the first modern theory of visual perception. He thought of the eyes as rather primitive optical devices, while their movements only serve to obtain depth information in a stereoscopic manner. Hence, the resulting process of visual perception is a sequence of snapshots on top of which an inductive process takes place, which draws on the respective individual experiences.

Early studies of eye movements for informational purposes were conducted by Yarbus et al. [1967], who made subjects with fixated heads scan presented stimuli. Glass contact lenses with attached pencils were used to record the resulting saccades. The latter are extremely rapid eye movements to sequentially fixate regions of interest with the fovea. While the eye is moving, nothing can be perceived, which partly constitutes phenomena described by the widely known change-blindness paradigm [Blackmore et al., 1995]. Furthermore, saccadic eye movements are subject to an inhibition of return behavior as found by Posner and Cohen [1984]. This means that subjects do not fixate the same spot twice to avoid the intake of already acquired information. The interested reader is referred to Klein [2000] as well as Lupiáñez et al. [2006] for further readings on this topic.

However, sole eye movements cannot describe all perceptual phenomena, as Gibson found when he tried to investigate the optical flow effect a pilot encounters while approaching a landing strip [Eysenck, 2001]. He concluded that, in experiments striving to capture all aspects of the perceptual process, the whole observer has to be able to move. Based on this insight, a whole theory of the perception of moving observers was finally introduced in Gibson [1977]. The theory of affordances is closely related to Gibson's *Ecological Approach to Visual Perception* [Gibson, 1977]. It postulates that the environment offers interaction opportunities to an individual, which can immediately be perceived [Jenkins, 2008]. Today, it is often employed in design processes of new products to foster their usability [Norman, 2013].

This integrated notion of action and perception is further supported by the common coding [Prinz, 1990] and event coding [Hommel et al., 2001] theories. This family of theories assume that the planning of motor actions as well as the perceptual process share the same coding stages within the brain. This implies that the visual perception of an action is – at least partly – similarly coded to actually executing that particular action. This line of thought allows for explaining how complex actions can be learned by imitation and suggests a common representational structure for perception and action.

The findings presented above do not hold exclusively for perception, but may range over multiple modalities. In a seminal paper, Bach-y Rita et al. [1969] showed that vision can partly be replaced by other modalities. A camera was connected with a tactile array, depicting the camera images, fixed at the back of study participants. Surprisingly, the subjects adapted to the mapped information presented in this unfamiliar modality and were able to *see* things, even when blindfolded.

4.1.2. Computational Models of Sensorimotor Visual Perception

The theories and findings presented above are mainly related to the human visual system. Recently, more formalized theories of perception and action were proposed, paving the way for computational models of sensorimotor visual perception.

O'Regan et al. [2001a] postulate that perception results from a tight coupling of sensor and motor information, which makes perception an inherently spatial process. As perception is an extended pattern of activity, it can not entirely take place in the brain. In fact, what one perceives is determined by what one does [Noë, 2004]. According to the Sensorimotor Account of Vision and Visual Consciousness [O'Regan et al., 2001b], the perceptual process is based on so-called sensorimotor contingencies. Agents are said to have perceptual sensations in arbitrary modalities, when they exercise their mastery of the sensorimotor contingencies which hold in that situation. The latter are learned for each particular modality and encode the relation between an action and the corresponding change in the sensory influx. The particular modalities may, e.g., be olfactory, tactile, or visual, and are governed by the interplay of the structure of the outer world and the corresponding sensor. For example, the visual sensorimotor contingencies are governed by the interplay of the 3D structure of the environment and the corresponding visual apparatus. Thus, a general principle underlies sensorimotor perception, while the single modalities can still be distinguished by virtue of the modality-specific sensorimotor relations.

Based on such an integrated sensorimotor representation of the world, it is possible to reconstruct certain aspects with reduced prior knowledge [Philipona et al., 2003, 2004]. Specifically, an autonomous system without knowledge about the topological structure of the world, or without knowledge about the spatial outcomes of its actions or the specific modalities its sensors operate in, can still learn models of the latter. Here, especially manifold approaches are of major interest, as they naturally allow for an integrated representation of the state and respective measurements.

Furthermore, a sensorimotor representation can be used to actively explore an environment, as implemented by the Sensorimotor Explorer (SMX). In Zetzsche et al. [1998] and Schill et al. [2001], a system architecture for active sensorimotor exploration of the surroundings of an autonomous system is introduced. It maintains the belief about hypotheses regarding its whereabouts in the framework of the Dempster-Shafer theory [Shafer, 1976; Smets and Kennes, 1994] within a hierarchical space [Schill et al., 1999]. Furthermore, it employs bio-inspired Intrinsic 2-Dimensional Features (I2D) as an integrated approach to describe and find salient points [Zetzsche et al., 2008] (see Fig. 4.4b). Finally, it can chose actions by a differential inspection of the belief distribution before and after a counterfactual update based on expected measurements. Thus, it can gain information by executing actions which maximize new information in expectation.

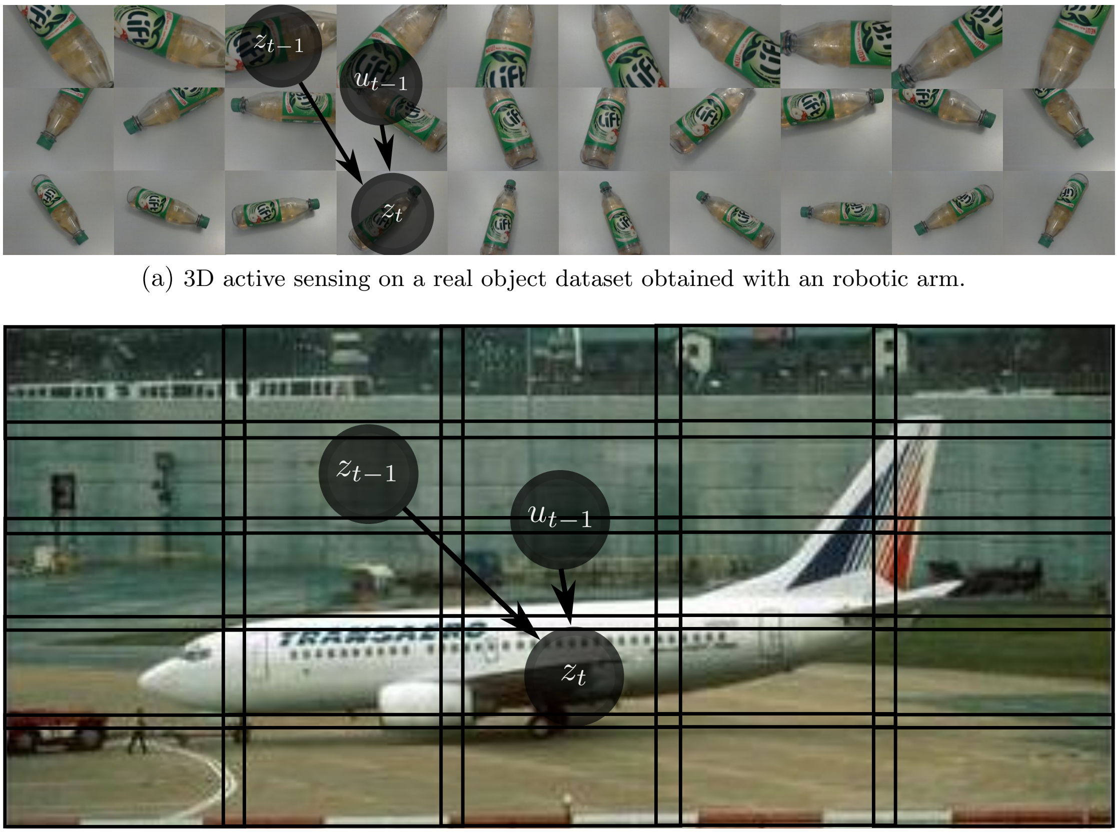
In computational neuroscience, models of eye movements have been devised and successfully tested against human subjects. In Najemnik and Geisler [2005, 2008], a maximum a posteriori (MAP) searcher is compared to an ideal searcher, both in a trial with known target. They found that former search strategy seeks to chose fixations that most likely contain features associated with the target, while the latter seeks to maximize information about the target's location.

Recently, deep neural networks have had a big success in the computer vision community [LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016]. An interesting family of deep learning models are recurrent neural networks, as they operate on time-series data, while maintaining some kind of state. Based on them, the idea of affordance-based object recognition, as introduced in Kluth et al. [2014], was further pursued with deep neural networks by Thermos et al. [2017]. Subsequently, Thermos et al. [2018] extended their approach to a Long Short-Term Memory (LSTM) network, resting on a sensory and an action stream. The general idea of active sensorimotor object recognition, as proposed in Nakath et al. [2014], has been implemented using deep reinforcement learning techniques by Liu et al. [2018a,b]. Furthermore, as the computational capacity of neural networks is limited, the general concept of sequential information processing with guidance by an attentional mechanism can be of interest [Ba et al., 2014]. Following this line of thought, an integrated network has been trained in a reinforcement learning scheme to read house numbers in stepwise manner and simultaneously generate actions by Mnih et al. [2014].

4.2. Modeling Active Sensorimotor Object Recognition

Building on the work by Schill and Zetzsche [Zetzsche et al., 1998; Schill et al., 1999, 2001; Zetzsche et al., 2008], an active sensorimotor object recognition system was devised, adhering to the following principles. Sensorimotor perception should be modeled as a sequential information intake process with commonly coded actions and perceptions. This sequence consists of consecutive snapshots, which implies that nothing is perceived between the fixations. A state should be maintained and comprise all previously inspected positions and a belief about the object class, which is currently presented to the system. An inhibition of return behavior should ensure that no position is fixated twice, to omit the intake of redundant information. And finally, the information intake process should be actively guided, based on previously acquired knowledge as well as the current state of the system.

In Kluth et al. [2013, 2014] and Nakath et al. [2014] such a system, based on a probabilistic sensorimotor feature (SMF), is presented. The latter should be learnable, foster classification performance, and provide a basis from which expected measurements can



(a) 3D active sensing on a real object dataset obtained with an robotic arm.

(b) 2D active sensing on an instance of the Caltech 256 dataset.

Figure 4.1.: Sensorimotor features on 2D and 3D datasets. Figures adopted from Nakath et al. [2014].

be extracted. To this end, a SMF is formally defined as the triple

$$SMF := \{z_{t-1}, u_{t-1}, z_t\}\,, \tag{4.1}$$

where z_{t-1} denotes a preceding sensory input and u_{t-1} an action leading to the current sensory input z_t.

When applied in a classification scheme, the system is presented with an instance from a previously learned set of objects. After actively inspecting the object, a distribution over all object classes is returned, where the maximum density indicates the classification result. The system is specified in a very general form and can thus be equipped with different front ends: A robotic arm can be actively controlled to inspect actual objects with a camera[1]. It is also possible to use a simulated robotic arm to inspect objects within a virtual reality. Furthermore, it is possible to feed images from objects to the system, taken from datasets like Caltech 256 [Griffin et al., 2007]. The images taken by the

[1] One of the first active perception approaches with a robotic arm using a camera as an endeffector was presented in Wilkes and Tsotsos [1993].

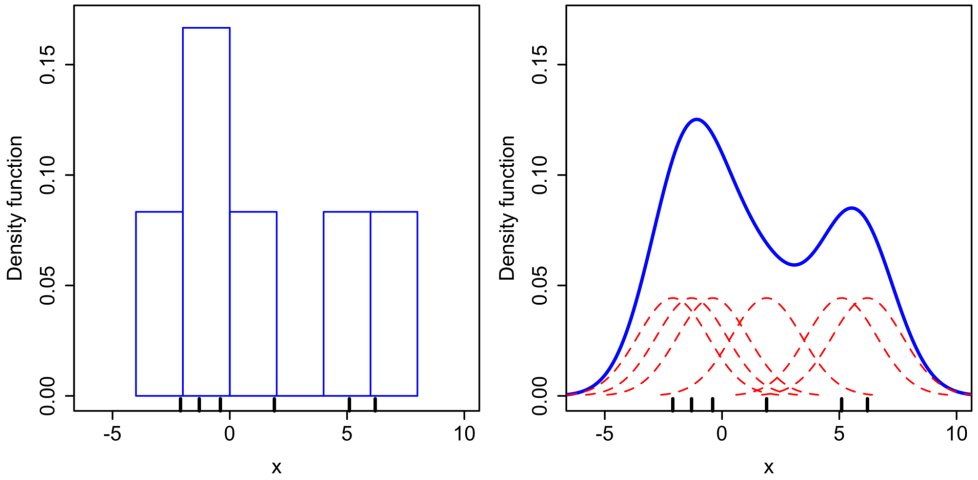


Figure 4.2.: Exemplary estimation of a probability density function (PDF) from data, either by a histogram or by a KDE approach. The resultant density estimation is drawn in blue, the kernels are depicted as red dashed lines, while the black vertical strokes denote the samples.

robotic arm are described with GIST features [Oliva and Torralba, 2006], as the holistic impression is of importance (see Fig. 4.1a). The images of object instances taken from datasets are divided into sub-regions to mimic a sequential information intake process, dependent on the sensor position (see Fig. 4.1b). On the resultant image patches, SURF features [Bay et al., 2006, 2008] are computed. The feature with the highest score is then selected to yield a local description of the respective patch. Afterwards – independent of the used frontend and thus of the employed feature – it is quantized in a previously learned k-means scheme. The resulting SMFs are fed into a classifier, which recursively computes a belief over the object classes. Based on the current uncertainty – measured in terms of entropy – in the belief, the available motor actions, as well as the resultant expected measurements, a new action is picked and carried out by the system. Exemplary SMFs, resulting from this process, are depicted for different frontends in Fig. 4.1.

In Kluth et al. [2013, 2014] as well as Nakath et al. [2014], it is shown that the classification performance benefits from the additional information encoded within the SMFs. In addition, the active perception approach results in a faster peak recognition performance and a corresponding faster minimization of the entropy of the posterior. Those results hold for investigations conducted with the robotic arm as well as for the results obtained on a subset of the Caltech 256 dataset.

4.2.1. Probabilistic Classification

To yield a belief, i.e., a posterior distribution over the object classes, a probabilistic classification scheme is chosen. For a thorough introduction to classification and pattern recognition, the interested reader is referred to Bishop [2006]. Probabilistic classification results can be established by Bayesian approaches, naturally yielding a posterior over the classes. However, it is also possible to train an ensemble of non-probabilistic classifiers and subsequently interpret the versatile results as a distribution over the classes (see e.g., Zhang [2000]).

In Kluth et al. [2013, 2014] and Nakath et al. [2014] a naive Bayes approach is chosen, which recursively estimates the posterior over time. Although being very simple and resting on the naive Bayes assumption this type of classifier often yields satisfying results in the field of computer vision [Boiman et al., 2008]. By applying Bayes' rule and additionally assuming conditional independence of the sensory information, it is then given by

$$p(y|z_{1:t}) \propto p(y) \prod_{i=1}^{t} p(z_i|y)\,, \tag{4.2}$$

where $p(z_i|y)$ is the likelihood and $p(y)$ is the prior for class y. The latter is uniformly distributed, as no prior knowledge should be provided for the classifier. Additionally, as this is a classification scheme, the normalization constant can be omitted. The posterior based on the full information encoded in a SMF can then be estimated recursively by employing the inference rule (see Fig. 4.4a)

$$p(y|z_{1:t}, u_{1:t-1}) \propto p(y)p(z_1|y) \prod_{i=2}^{t} p(z_i|u_{i-1}, z_{i-1}, y)\,, \tag{4.3}$$

where again $p(y)$ is the prior and the likelihoods are factorized according to the system behavior. Initially, only the sensory information is taken into account, as the first fixation is a random position. For all following observations the full sensorimotor information can then be used.

However, the likelihoods can be obtained in different manners, resulting in a varying estimation quality. In Kluth et al. [2013, 2014] and Nakath et al. [2014] a sensorimotor representation (SMR) is simply learned by defining the joint probability distribution

$$\mathit{SMR} := p(\mathit{SMF}, Y) = p(Z_{t-1}, U_{t-1}, Z_t, Y) \tag{4.4}$$

as the relative frequencies of the occurrences of the SMFs in the particular classes y of the training set. In addition, the whole SMR is Laplace-smoothed, to avoid degenerate results. The likelihoods for (4.2) as well as (4.3) can readily be marginalized from a SMR established in such a way. However, the relative frequency information rests on features which were quantized in feature space using k-means (see Sect. 4.2). Quantization can lead to a coarse estimation of the underlying distribution, as depicted in Fig. 4.2. Thus, in Reineking et al. [2015] an active perception scheme on top of a naive Bayes classifier, drawing its likelihoods from a KDE-approach, is proposed (cf. Figs. 4.2 and 4.3a). However, a global – i.e., considering all features acquired in the learning process – KDE-approach is often computationally involved. On top of that, it processes a lot of negligible information due to the exponential decay of the kernels and the general tendency to big distances between data in high-dimensional spaces [Aggarwal et al., 2001]. Thus, it is customary to employ some restrictions such that the local area or even only the nearest neighbor of the query vector z_i is considered (see Fig. 4.3b). The latter approach is called Naive Bayes Nearest Neighbor (NBNN) and despite its simplicity it yields competitive results [Boiman et al., 2008; Tuytelaars et al., 2011]. However, the nearest neighbor search itself is also computationally involved, thus McCann and Lowe [2012] introduced local NBNN. In this scheme, the k-nearest neighbors over all classes are considered, thus establishing a local neighborhood of z_i, denoted as $\Lambda^*_k(z_i)$ in Fig. 4.3c. If a particular class y is not represented in this set, the $k+1$st neighbor $\Lambda_{K+1}(z_i)$ is used as a conservative estimate of the background probability. The latter approach avoids zero-likelihoods and thus degenerate posteriors in a recursive classification scheme. Contrary to that, in Reineking et al. [2015], the whole feature space is searched for nearest neighbors. To mitigate the resultant bottleneck, an approximate search based on the FLANN library [Muja and Lowe, 2009, 2014] is employed.

The feature space, in turn, is established by densely sampling scale-invariant feature transform (SIFT) features [Lowe, 1999] from the training set. Those features are stored class-wise in bins corresponding to the fixation regions as shown in Fig. 4.1b. When an image to be classified is passed to the system, the features from the corresponding fixation bin are passed to the NBNN-classifier. The latter recursively computes the posterior, approximating the likelihood with a Gaussian kernel according to

$$p(y|z_{1:t}) \propto p(y) \prod_{i=1}^{t} p(z_i|y) \propto p(y) \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{t} ||z_i - NN_y(z_i)||^2\right) , \tag{4.5}$$

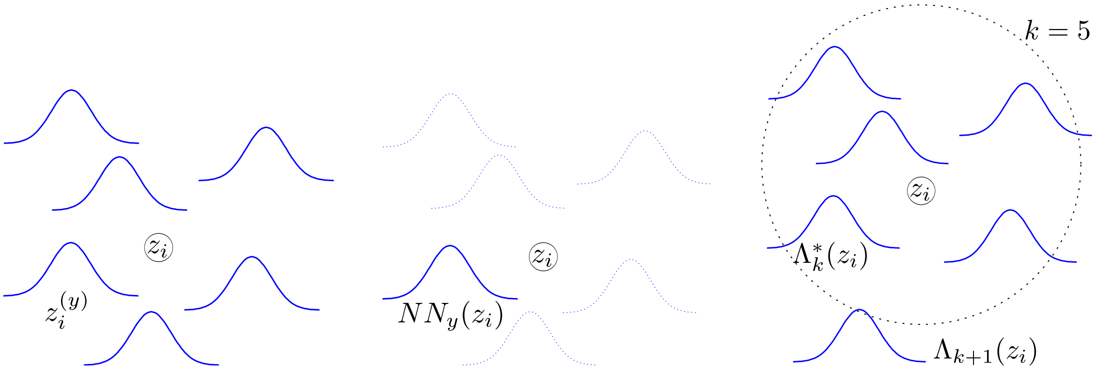
where the prior $p(y)$ is again uniformly distributed, σ is a scaling factor for the Gaussian kernel, and $NN_y(z_i)$ denotes the nearest neighbor of the query vector z_i with respect to class y (see Fig. 4.3b).

4.2.2. Active Perception

Two principles have to be obeyed for a sensorimotor active perception strategy: (i) it should act according to a current internal state of a system and (ii) the next action should be selected with respect to some previously acquired knowledge, i.e., the experience of the observer.

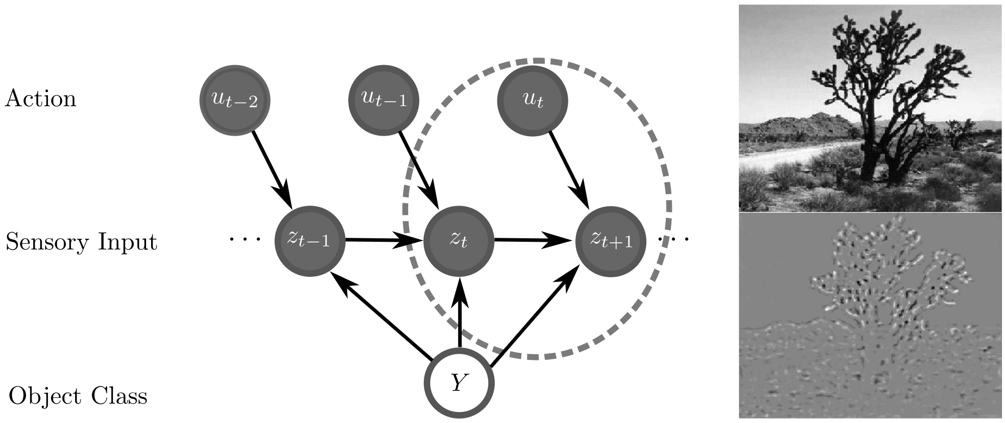
Zetzsche et al. [2008] argued that a belief function-based measure of distribution differences behaves better than the Kullback-Leibler (KL)-divergence in an exploration task, which tends to pick actions favoring hypotheses with low evidence. This is due to the fact that in a probabilistic classification scheme, an active perception approach based on KL-divergence might favor actions leading to uniform distributions (see Fig. 2.6). Hence, the canonical approach to gain information by minimizing the expected entropy is chosen. This favors actions, not only yielding different distributions, which would be sufficient for a KL-divergence measure, but yielding more determined distributions in expectation.

In Kluth et al. [2013, 2014]; Nakath et al. [2014] as well as in Reineking et al. [2015], an optimal action u^*_t is picked which maximizes the expected information gain under the measure of expected entropy. The necessary information to compute the expected



(a) KDE (b) NBNN (c) Local NBNN

Figure 4.3.: Basic principles of the particular KDE-methods for likelihood estimation of z_i. KDE: all class specific labeled values $z_i^{(y)}$ are considered. NBNN: the class specific nearest neighbors $NN_y(z_i)$ are considered. Local NBNN: the k-nearest neighbors over all classes $\Lambda_k^*(z_i)$ are considered. The $k+1$st neighbor $\Lambda_{k+1}^*(z_i)$ serves as a background probability, if the queried class is not contained in the local neighborhood. The blue lines indicate kernels in feature space, they are drawn solid when considered in the approach and dashed otherwise.



(a) Dynamic Bayes net for active sensorimotor object recognition. The dashed lines indicate the top-down information gain area.

(b) I2D-features

Figure 4.4.: Approaches to (a) top-down and (b) bottom-up action selection. Figure (b) adopted from Reineking et al. [2015].

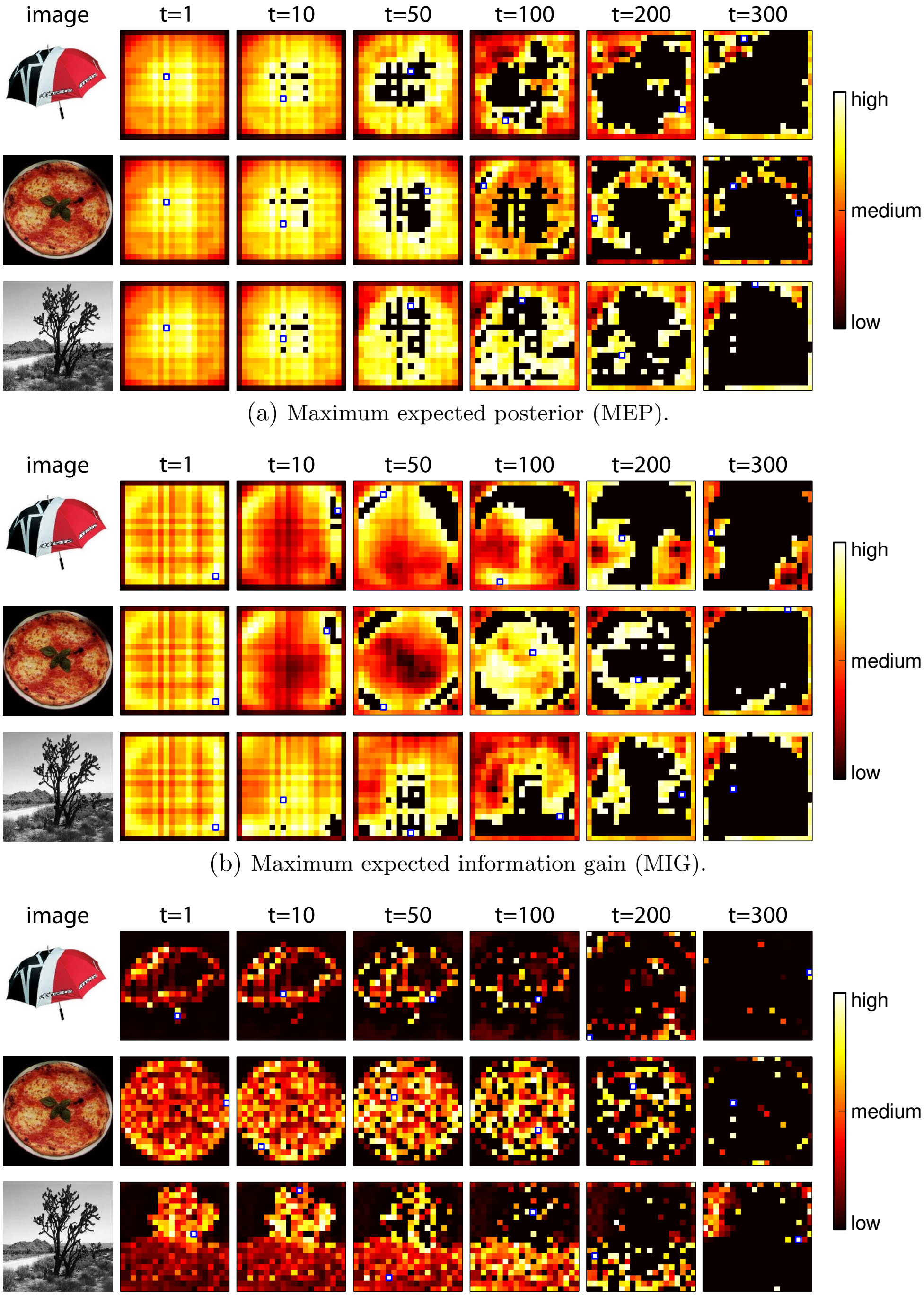
measurements given an action can be marginalized from the SMR, thus drawing on the experience of the system (see the dashed area in Fig. 4.4a). Those are used to update the belief over the object classes, thus taking the current state of the system into account. Finally, an inhibition of return behavior is induced by omitting actions that would lead to already inspected image patches.

Furthermore, in Zetzsche et al. [2008], it is postulated that the information gain process should be governed by a bi-directional top-down and bottom-up process. It can be argued that such a process is implemented by the bottom-up selection of salient SURF features, within an image patch picked by the top-down information gain strategy.

In addition to the canonical formulation using the entropy-based maximum expected information gain (MIG), further guiding principles for action selection have been investigated in Reineking et al. [2015]. They comprise offline methods, neglecting the current state of the system, as well as online methods, which take stepwise decisions depending on the current state. Expect for one bottom-up saliency-based approach, the methods are collectively top-down approaches depending on previously learned distributions.

In particular, the maximum expected likelihood (MEL) strategy ignores the current state and can thus be computed offline. It seeks to pick actions in turn maximizing the normalized expected likelihood of the true class. The latter is known, as those computations are based on the training set which makes this kind of information available. The maximum expected posterior (MEP) strategy stepwise seeks to maximize the expected posterior of the true class, hence taking the previously processed descriptors into account. And finally, the I2D features describe I2D signals, whose magnitude can be used as a bottom up saliency measure (see Fig. 4.4b) [Zetzsche and Barth, 1990].

In Reineking et al. [2015], an evaluation of the performance of the NBNN classifier based on sequential feature selection, as governed by the different strategies, is conducted. As a baseline, two additional strategies are introduced: The linear strategy just scans the image line by line starting at the top of the image, while the random strategy randomly processes the given features. Both baseline strategies are outperformed by the informed strategies. In general, the other strategies behave comparable, with slight advantages for the MEP and MEL strategies, followed by the MIG strategy, and finally the I2D feature. Fig. 4.5 depicts the spatiotemporal behavior of the single search strategies on different object instances of the Caltech 101 dataset [Fei-Fei et al., 2007].



(a) Maximum expected posterior (MEP).

(b) Maximum expected information gain (MIG).

(c) Intrinsic 2-dimensional features (I2D).

Figure 4.5.: Spatiotemporal behavior of the different feature selection strategies. The blue box indicates the cell with the highest value, while cells which have already been selected have value 0, i.e., they are black. Figures reprinted from Reineking et al. [2015].

5 Conclusion

In this thesis, novel active perception approaches are presented which reduce uncertainty in the state estimate of autonomous systems. The latter are considered robotic systems which can pursue certain goals for an extended period of time in a self-sustained fashion. To this end, they are typically equipped with sensors as well as actuators to perceive the outer world and, in turn, manipulate it in accordance with their respective goals. This process may be thought of in terms of a closed-loop scenario exhibiting a perceive-reason-act pattern. The bi-directional interface with the real world then consist of a sequence of imperfect observations, as well as a sequence of actions, carried out in a noisy manner. To formally implement such a loop, the state of an autonomous system has to be maintained and the corresponding uncertainties have to be managed. The active perception approaches presented here work on top of such a probabilistic state estimate to minimize its associated uncertainties. They do so by selecting actions to optimize the sequential information intake process in order to acquire information beneficial for that purpose.

Hence, several works are presented here, striving to answer the question of *where to look next* in an algorithmic fashion. They mainly focus on active perception in an autonomous deep space navigation scenario. In addition, an excursion to active sensorimotor object recognition is undertaken. These two topics are discussed separately in the following.

5.1. Active Autonomous Deep Space Navigation

In this thesis, several approaches for active autonomous deep space navigation are presented. This kind of problems arise within deep space missions, which might be of scientific or of commercial nature, omitting ground support.

A heliocentric navigation scenario and an asteroid-centric navigation scenario are introduced as part of a commercial asteroid mining mission. The latter kind of missions might be doubted for political, environmental, as well as economic reasons. In the political field, the property rights of the resources mined in deep space are still not well-regulated. First attempts for regulation have been undertaken by Luxembourg as well as the United States of America, however they did not remain undisputed. In addition, doubts were raised, if a commercial asteroid mining mission is a business that pays, as launching rockets is extremely expensive. Representatives of Deep Space Industries responded with a concept which proposes to use in-situ mined water as propellant and thus avoids frequent encounters with the earth. This might be an opportunity to draft an actually affordable and sustainable commercial asteroid mining mission.

Autonomous deep space navigation itself is of interest to save cost as well as to enable missions which cannot be conducted with pure ground support. Specifically, absolute navigation is of interest, as optical autonomous navigation enables maneuvers which could not be flown before by S/C with sole ground support. Relative navigation, on the other hand, demands for rapid autonomous decision making in the vicinity of celestial objects, where ground support is impaired by signal roundtrip time. In addition, technical requirements like limited energy supply and subsurface operations might render autonomous behavior mandatory.

Thus, heliocentric active autonomous navigation approaches have been devised, which actively reduce the uncertainty of the state estimate. Instead of the canonical minimization of the expected entropy, a direct uncertainty impact measurement is successfully employed. In addition, a relative active autonomous navigation scheme based on Graph SLAM has been proposed. The joint estimation of the trajectory and the map is conducted using a graph, which is robustly constructed in a tightly-coupled fashion. The trajectory selection scheme works in two steps to limit the number of optimal trajectories to be computed. By that means, real-time capability of the approach is ensured.

All approaches are tested in a numerical simulator. This entails that a gap to the real-world will always persist, even though a lot of effort is made to establish a realistic environment. It is a fundamental property of models not to be able to capture the full complexity of the real world. Thus, when applying the approaches presented here in actual autonomous systems, additional work has to be done to account for the circumstances which have not been covered by the models used within the simulations. Still, all approaches presented work in real-time, which means within the respective time frame posed by a simulation step. To run the algorithms on space-proof hardware, additional adjustments might be necessary, though. However, the accelerated scan-matching approach is already accompanied by advances in GPU-based High Performance Computing (HPC) for S/C (see e.g., [Kraja et al., 2013]).

The approaches presented in this thesis are commonly tailored to the specific scenario they are used in. The thorough modeling of the scenario and the particular solutions based on scenario-related assumptions are valuable as such and might inspire some future developments for active deep space navigation. However, all approaches are built on a general probabilistic structure, thus some of the ideas presented can be used in general by adopting the scenario-specific models. Namely, the proposed direct uncertainty impact measures might readily be used in other probabilistic active perception approaches. Several interesting ideas are also contained in the robust tightly-coupled graph construction scheme. The twofold trajectory evaluation scheme can also be employed in other exploration approaches, when some of the assumptions are adopted accordingly. Finally, all approaches work on the $\boxplus$-manifold, using the state encapsulation approach for a complete autonomous deep space navigation solution.

However, there are ideas are left to be further pursued. In the absolute active perception approaches the attitude offset is used to compute the cost. It seems promising to consider the control torque instead and maybe even compute the expected consumed propellant based on the specific impulse (ISP) of the thrusters.

It is a valid approach to project the point cloud map as obtained with Graph SLAM into an evidential grid map to yield an extended uncertainty representation based on smoothed data. However, the scan matching procedure is based on the point clouds, whose surfaces are in turn approximated with a probabilistic grid structure. It would be interesting to attempt an unification of scan matching and an extended uncertainty representation as given by the evidential grid map. This might be achieved with a kind of correlative scan matching, taking into account the different dimensions of uncertainty provided by the evidential grid map.

In the twofold trajectory evaluation scheme, the values of a trajectory are computed over all single nodes in turn accumulating the values of all evaluation nodes. This computation is subject to exponential decay as a function of distance and thus would greatly benefit from a nearest neighbor or at least a k-nearest neighbor approximation for every node.

The proportional-derivative controller tended to instable behavior in the asteroid-centric navigation scheme. Thus, it might be interesting to use a model predictive control (MPC)-approach based on a ⊟-error-function to yield a tuning free approach (see e.g., Tan et al. [2011]).

In the asteroid-centric navigation scenario, it is assumed that the state of the asteroid is known. At the same time the S/C is drawing on sensing capabilities for absolute as well as for relative navigation. Thus, a further line of work might be the relative estimation of parameters of an asteroid model, while the S/C localizes itself in the heliocentric frame.

And finally, another interesting idea is to use probabilistic regression schemes to generate expected values for unexplored areas of the map. This would not only allow to explore new areas, but also to specifically explore those new areas which are valuable in expectation.

5.2. Active Sensorimotor Object Recognition

The human visual system has been subject to scientific investigations for a long time. Based on those findings, computational models, adhering to basic principles of the human visual system discovered in the course of time, have been derived. Those models might serve two purposes. First, they provide further insight about the human visual system, if successfully tested against human subjects in psychophysical experiments. And second, they benefit in terms of performance by implementing principles which have emerged in the evolutionary process.

To serve the second purpose, an active sensorimotor object recognition system is proposed in this thesis. It rests on a cleanly-designed sensorimotor feature, which constitutes an integrated notion of perception and action. The latter, in turn, fosters and additionally accelerates classification performance by allowing for an active perception scheme. The naive Bayes classification approaches presented in Chap. 4 deliver a probabilistic estimate, however the learning schemes are rather simple. The proposed active perception strategies comprise top-down expected entropy minimization, further top-down probabilistic approaches, and two bottom-up saliency driven approaches.

Provided with appropriate models and sensing capabilities, the system can cope with arbitrary modalities. In addition, it can be used with various frontends, ranging from a robotic arm with an attached camera, over a virtual reality, to a simple interface for images. Thus, the system might also be thought of as a rather general conceptual outline, from which actual high-performance implementations can be derived. This has already been done in the works of Thermos et al. [2017, 2018], as well as in Liu et al. [2018a,b].

As a first step of future work, the applied features might be replaced by advanced Convolutional Neural Network (CNN) features [LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016]. This kind of features can be obtained by omitting the classification stage of a neural network, and interpret the thus-exposed activations as features. Just recently, recursive neural networks with attentional mechanisms have been on the rise in the field of pattern recognition [Ba et al., 2014]. They can serve as a foundation for integrating sequential classification and action generation. In Mnih et al. [2014], such an approach is presented to sequentially read house numbers in images.

It is also very promising to integrate further modalities like tactile sensing into the system. It naturally integrates with robotic arms, exhibiting the respective capabilities (see e.g., Fig. 1.1a), and is well-suited to further improve classification performance. In addition, this course of action would have interesting implications for the active sensing strategy. In the top-down stream, a combined strategy for expected measurements, stemming from the visual and tactile domains, would have to be devised. It could additionally be combined with twofold saliency-information, extracted from the visual as well as from the tactile modality, from the bottom-up-stream.

Finally, most feature selection approaches only consider uncertainty in the distribution over the classes. By additionally considering the pose-uncertainty of the sensor, it would become possible to leverage the mutual correctional potential of those distributions.

Own Publications

González Peytaví, G., Clemens, J., Nakath, D., Probst, A., Förstner, R., Schill, K., and Eissfeller, B. (2015). Autonomous Orbit Navigation for a Mission to the Asteroid Main Belt. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation.

Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2013). Sensorimotor Integration Using an Information Gain Strategy in Application to Object Recognition Tasks. In *Perception*, volume 42, pages 223–223. Pion LTD 207 Brondesbury Park, London NW2 5JN, UK, doi: 10.1177/03010066130420s101.

Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2014). Affordance-Based Object Recognition Using Interactions Obtained from a Utility Maximization Principle. In *Computer Vision-ECCV 2014 Workshops*, pages 406–412. Springer, doi: 10.1007/978-3-319-16181-5˙29.

Nakath, D., Clemens, J., and Rachuy, C. (2017). Rigid Body Attitude Control Based on a Manifold Representation of Direction Cosine Matrices. volume 783, page 012040, doi: 10.1088/1742-6596/783/1/012040.

Nakath, D., Clemens, J., and Rachuy, C. (2019). Active Asteroid-SLAM – Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario. *Journal of Intelligent & Robotic Systems*, doi: 10.1007/s10846-019-01103-0.

Nakath, D., Clemens, J., and Schill, K. (2018). Multi-sensor fusion and active perception for autonomous deep space navigation. In *21st International Conference on Information Fusion (FUSION 2018)*, pages 2596–2605. IEEE, doi: 10.23919/icif.2018.8455788.

Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., and Schill, K. (2014). Active Sensorimotor Object Recognition in Three-Dimensional Space. In *Spatial Cognition IX*, pages 312–324. Springer, doi: 10.1007/978-3-319-11215-2˙22.

Nakath, D., Rachuy, C., Clemens, J., and Schill, K. (2016). Optimal Rotation Sequences for Active Perception. In *Proc. SPIE*, volume 9872, pages 987204–987204–13. International Society for Optics and Photonics, doi: 10.1117/12.2223027.

Probst, A., González Peytaví, G., Nakath, D., Schattel, A., Rachuy, C., Lange, P., Clemens, J., Echim, M., Schwarting, V., Srinivas, A., Gadzicki, K., Förstner, R., Eissfeller, B., Schill, K., Büskens, C., and Zachmann, G. (2015). Identifying the Challenges for Cognitive Autonomous Navigation and Guidance for Missions to Small Planetary Bodies. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation.

Reineking, T., Kluth, T., and Nakath, D. (2015). Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification. In *Computer Analysis of Images and Patterns*, pages 350–361. Springer, doi: 10.1007/978-3-319-23192-1˙29.

Bibliography

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

Amzajerdian, F., Vanek, M., Petway, L., Pierrottet, D., Busch, G., and Bulyshev, A. (2010). Utilization of 3d imaging flash lidar technology for autonomous safe landing on planetary bodies. In *Quantum Sensing and Nanophotonic Devices VII*, volume 7608, page 760828. International Society for Optics and Photonics.

Andert, F., Ammann, N., and Maass, B. (2015). Lidar-aided camera feature tracking and visual slam for spacecraft low-orbit navigation and planetary landing. In *Advances in Aerospace Guidance, Navigation and Control*, pages 605–623. Springer.

Ang, M. H. and Tourassis, V. D. (1987). Singularities of euler and roll-pitch-yaw representations. *IEEE Transactions on Aerospace and Electronic Systems*, (3):317–324.

Åström, K. J. (1965). Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755.*

Bach-y Rita, P., Collins, C. C., Saunders, F. A., White, B., and Scadden, L. (1969). Vision substitution by tactile image projection.

Badescu, V. (2013). *Asteroids: Prospective energy and material resources.* Springer Science & Business Media.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):966–1005.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer.

Bellekens, B., Spruyt, V., Berkvens, R., and Weyn, M. (2014). A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13.

Bellman, R. E. (1957). *Dynamic Programming.* Courier Dover Publications.

Berry, K., Sutter, B., May, A., Williams, K., Barbee, B. W., Beckman, M., and Williams, B. (2013). Osiris-rex touch-and-go (tag) mission design and analysis. In *36th Annual AAS Guidance and Control Conference; 1-6 Feb. 2013; Breckenridge, CO; United States*. American Astronautical Society; Rocky Mountain Section; San Diego, CA, United States.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

Bhaskaran, S. (2012). Autonomous navigation for deep space missions. In *SpaceOps 2012*, page 1267135.

Bhaskaran, S., Riedel, J. E., and Synnott, S. P. (1996). Autonomous optical navigation for interplanetary missions. In *Space Sciencecraft Control and Tracking in the New Millennium*, volume 2810, pages 32–44. International Society for Optics and Photonics.

Biber, P., Fleck, S., and Straßer, W. (2004). A probabilistic framework for robust and accurate matching of point clouds. In *Joint Pattern Recognition Symposium*, pages 480–487. Springer.

Biber, P. and Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. In *IROS*, volume 3, pages 2743–2748.

Biele, J., Ulamec, S., Maibaum, M., Roll, R., Witte, L., Jurado, E., Muñoz, P., Arnold, W., Auster, H.-U., Casas, C., et al. (2015). The landing (s) of philae and inferences about comet surface mechanical properties. *Science*, 349(6247):aaa9816.

Bishop, C. (2006). Pattern recognition and machine learning. *Pattern Recognition and Machine Learning*.

Blackmore, S. J., Brelstaff, G., Nelson, K., and Trościanko, T. (1995). Is the richness of our visual world an illusion? transsaccadic memory for complex scenes. *Perception*, 24(9):1075–1081.

Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalable mapping. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1899–1906. IEEE.

Brophy, J., Garner, C., Nakazono, B., Marcucci, M., Henry, M., and Noon, D. (2003). The ion propulsion system for dawn. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 4542.

Brown, A., Vallenari, A., Prusti, T., de Bruijne, J., Babusiaux, C., Bailer-Jones, C., Collaboration, G., et al. (2018). Gaia data release 2. summary of the contents and survey properties. *arXiv preprint arXiv:1804.09365.*

Budnik, F., Morley, T., and Mackenzie, R. (2004). ESOC's system for interplanetary orbit determination: Implementation and operational experience. In *18th International Symposium on Space Flight Dynamics*, volume 548, pages 387–392.

Bullo, F. and Murray, R. M. (1995). Proportional derivative (pd) control on the euclidean group. In *European Control Conference*, volume 2, pages 1091–1097.

Büskens, C. and Wassel, D. (2012). The esa nlp solver worhp. In *Modeling and optimization in space engineering*, pages 85–110. Springer.

Carlone, L., Du, J., Ng, M. K., Bona, B., and Indri, M. (2014). Active slam and exploration with particle filters using kullback-leibler divergence. *Journal of Intelligent & Robotic Systems*, 75(2):291–311.

Carrillo, H., Reid, I., and Castellanos, J. A. (2012). On the comparison of uncertainty criteria for active slam. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2080–2087. IEEE.

Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. (1996). Acting under uncertainty: discrete Bayesian models for mobile-robot navigation. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 963–972.

Chapman, B., Jost, G., and Van Der Pas, R. (2008). *Using OpenMP: portable shared memory parallel programming*, volume 10. MIT press.

Chaturvedi, N. A., Sanyal, A. K., and McClamroch, H. N. (2011). Rigid-body attitude control. *IEEE Control Systems*, 31(3):30–51.

Christian, J. A. and Cryan, S. (2013). A survey of lidar technology and its use in spacecraft relative navigation. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4641.

Claraco, J. L. B. (2008). Development of scientific applications with the mobile robot programming toolkit.

Clemens, J. and Reineking, T. (2014). Multi-sensor fusion using evidential slam for navigating a probe through deep ice. In *International Conference on Belief Functions*, pages 339–347. Springer.

Clemens, J., Reineking, T., and Kluth, T. (2016). An evidential approach to slam, path planning, and active exploration. *International Journal of Approximate Reasoning*, 73:1–26.

Clemens, J. and Schill, K. (2016). Extended Kalman filter with manifold state representation for navigating a maneuverable melting probe. In *19th International Conference on Information Fusion (FUSION)*, pages 1789–1796. IEEE.

Cocaud, C. and Kubota, T. (2010). Surf-based slam scheme using octree occupancy grid for autonomous landing on asteroids. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, pages 275–282.

Cocaud, C. and Kubota, T. (2012). Autonomous navigation near asteroids based on visual slam. In *Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, California.*

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* John Wiley & Sons, New York, NY, 1 edition.

Crassidis, J. L. and Markley, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542.

Curkendall, D. W. and Border, J. S. (2013). Delta-DOR: The one-nanoradian navigation measurement system of the deep space network—History, architecture, and componentry. *The Interplanetary Network Progress Report*, 42:193.

Dachwald, B., Mikucki, J., Tulaczyk, S., Digel, I., Espe, C., Feldmann, M., Francke, G., Kowalski, J., and Xu, C. (2014). Icemole: a maneuverable probe for clean in situ analysis and sampling of subsurface ice and subglacial aquatic ecosystems. *Annals of Glaciology*, 55(65):14–22.

Del Moral, P., Doucet, A., Jasra, A., et al. (2012). On adaptive resampling strategies for sequential monte carlo methods. *Bernoulli*, 18(1):252–278.

Dellaert, F. and Kaess, M. (2006). Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203.

DeMeo, F. and Carry, B. (2014). Solar system evolution from compositional mapping of the asteroid belt. *Nature*, 505(7485):629.

Dietrich, A. and McMahon, J. W. (2016). Orbit determination using flash lidar around small bodies. *Journal of Guidance, Control, and Dynamics*, 40(3):650–665.

Dietrich, A. B. and McMahon, J. W. (2018). Robust orbit determination with flash lidar around small bodies. *Journal of Guidance, Control, and Dynamics*, pages 1–22.

Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241.

Dissly, R., Weimer, C., Masciarelli, J., Weinberg, J., Miller, K., and Rohrschneider, R. (2012). Flash lidars for planetary missions. In *Workshop on Instrumentation for Planetary Missions*.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media, New York, NY.

Doucet, A., De Freitas, N., Murphy, K., and Russell, S. (2000). Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc.

D'Souza, C. (2014). Autonomous deep-space optical navigation project. Technical report, NASA Johnson Space Center, Houston, TX, United States.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.

Estrada, C., Neira, J., and Tardós, J. D. (2005). Hierarchical slam: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596.

Eysenck, M. W. (2001). *Principles of cognitive psychology*. Psychology Press.

Fairfield, N., Kantor, G., and Wettergreen, D. (2007). Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1-2):03–21.

Feder, H. J. S., Leonard, J. J., and Smith, C. M. (1999). Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Elsevier.

Fei-Fei, L., Fergus, R., and Perona, P. (2007). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70.

Foust, J. (2016). Deep space industries unveils first asteroid prospecting spacecraft. Spacenews.com. `https://spacenews.com/deep-space-industries-unveils-first-asteroid-prospecting-spacecraft/` [accessed on 08-17-2018].

Foust, J. (2017). Luxembourg adopts space resources law. Spacenews.com. `https://spacenews.com/luxembourg-adopts-space-resources-law/` [accessed on 08-17-2018].

Freedheim, D. K. and Weiner, I. B. (2003). *Handbook of Psychology, History of Psychology*, volume 1. John Wiley & Sons.

Frese, U., Larsson, P., and Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207.

Gary, M., Fairfield, N., Stone, W. C., Wettergreen, D., Kantor, G., and Sharp, Jr, J. M. (2008). 3d mapping and characterization of sistema zacatón from depthx (de ep p hreatic th ermal e x plorer). In *Sinkholes and the Engineering and Environmental Impacts of Karst*, pages 202–212.

Gibson, J. (1977). The theory of affordances. *Perceiving, acting, and knowing*, pages 67–82.

Gillette, A., O'Connor, B., Wilson, C., and George, A. (2018). Spacecraft mission agent for autonomous robust task execution. In *2018 IEEE Aerospace Conference*, pages 1–8. IEEE.

Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Kührt, E., and Richter, I. (2007). The rosetta mission: flying towards the origin of the solar system. *Space Science Reviews*, 128(1-4):1–21.

Golfarelli, M., Maio, D., and Rizzi, S. (1998). Elastic correction of dead-reckoning errors in map building. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 905–911. IEEE.

González Peytaví, G., Clemens, J., Nakath, D., Probst, A., Förstner, R., Schill, K., and Eissfeller, B. (2015). Autonomous orbit navigation for a mission to the asteroid main belt. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. Technical report, California Institute of Technology. `http://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001`. [Online; accessed 24-June-2013].

Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010a). A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.

Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., and Hertzberg, C. (2010b). Hierarchical optimization on manifolds for online 2d and 3d mapping. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 273–278. IEEE.

Grisetti, G., Stachniss, C., and Burgard, W. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE.

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46.

Grundmann, J. T., Auster, U., Baturkin, V., Bellion, A., Bibring, J.-P., Biele, J., Boden, R., Bompis, O., Borgs, B., Bousquet, P., Canalias, E., Celotti, L., Cenac-Morthe, C., Cordero, F., Deleuze, M., Evesque, C., Findlay, R., Fredon, S., Glaßmeier, K., Granena, D., Grimm, C., Grott, M., Hamm, V., Hendrikse, J., Hercik, D., Ho, T.-M., Jaumann, R., Krause, C., Kroth, R., Ksenik, E., Lange, C., Lange, M., Mierheim, O., Okada, T., Reill, J., Sasaki, K., Schmitz, N., Sedlmayr, H.-J., Talapina, M., Tangruamsub, S., Termtanasombat, N., Ulamec, S., Wejmo, E., Wrasmann, M., Yoshimitsu, T., Ziach, C., and the MASCOT team (2015). Mobile asteroid surface scout (MASCOT)-design, development and delivery of a small asteroid lander aboard Hayabusa2. In *4th IAA Planetary Defense Conference*.

Guo, Y. (1999). Self-contained autonomous navigation system for deep space missions. *Spaceflight mechanics 1999*, pages 1099–1113.

Guo, Y. (2003). Method and apparatus for autonomous solar navigation. US Patent 6,622,970.

Haag, M. (2015). Kollaboratives arbeiten mit robotern–vision und realistische perspektive. In *Zukunft der Arbeit in Industrie 4.0*, pages 59–64. Springer.

Hertzberg, C., Wagner, R., Frese, U., and Schröder, L. (2013). Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77.

Ho, T.-M., Baturkin, V., Grimm, C., Grundmann, J. T., Hobbie, C., Ksenik, E., Lange, C., Sasaki, K., Schlotterer, M., Talapina, M., et al. (2017). Mascot—the mobile asteroid surface scout onboard the hayabusa2 mission. *Space Science Reviews*, 208(1-4):339–374.

Hommel, B., Müsseler, J., Aschersleben, G., and Prinz, W. (2001). The theory of event coding (tec): A framework for perception and action planning. *Behavioral and brain sciences*, 24(05):849–878.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, MA.

Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P., and Zanna, A. (2000). Lie-group methods. *Acta numerica*, 9:215–365.

Jazwinski, A. H. (1970). *Stochastic processes and filtering theory.* New York, NY: Academic Press.

Jenkins, H. S. (2008). Gibson's "affordances": evolution of a pivotal concept. *Journal of Scientific Psychology*, 12(2008):34–45.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. In *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.

Kam, H. R., Lee, S.-H., Park, T., and Kim, C.-H. (2015). Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60(2):337–345.

Kayal, H., Balagurin, O., Djebko, K., Fellinger, G., Puppe, F., Seipel, D., Serdar, S., Schneider, A., Schwarz, T., and Wojtkowiak, H. (2018). Next level autonomous nanosatellite operations. In *2018 SpaceOps Conference*, page 2690.

Klein, R. M. (2000). Inhibition of return. *Trends in cognitive sciences*, 4(4):138–147.

Klir, G. J. (2005). *Uncertainty and information: foundations of generalized information theory*. John Wiley & Sons.

Klosko, S. and Wagner, C. (1982). Spherical harmonic representation of the gravity field from dynamic satellite data. *Planetary and Space Science*, 30(1):5–28.

Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., and von Stryk, O. (2013). Hector open source modules for autonomous mapping and navigation with rescue robots. In *Robot Soccer World Cup*, pages 624–631. Springer.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kolmogorov, A. (1950). Foundations of the theory of probability.

Kontitsis, M., Tsiotras, P., and Theodorou, E. (2016). An information-theoretic active localization approach during relative circumnavigation in orbit. In *AIAA Guidance, Navigation, and Control Conference*, page 0872.

Kowalski, J., Linder, P., Zierke, S., von Wulfen, B., Clemens, J., Konstantinidis, K., Ameres, G., Hoffmann, R., Mikucki, J., Tulaczyk, S., et al. (2016). Navigation technology for exploration of glacier ice with maneuverable melting probes. *Cold Regions Science and Technology*, 123:53–70.

Kraja, F., Acher, G., and Bode, A. (2013). Designing spacecraft high performance computing architectures. In *Advanced Computing*, pages 137–156. Springer.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE.

Landis, R. and Johnson, L. (2018). Advances in planetary defense in the united states. *Acta Astronautica*.

Lauretta, D., Balram-Knutson, S., Beshore, E., Boynton, W., d'Aubigny, C. D., DellaGiustina, D., Enos, H., Golish, D., Hergenrother, C., Howell, E., et al. (2017). Osiris-rex: sample return from asteroid (101955) bennu. *Space Science Reviews*, 212(1-2):925–984.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Lewicki, C., Diamandis, P., Anderson, E., Voorhees, C., and Mycroft, F. (2013). Planetary resources—the asteroid mining company. *New Space*, 1(2):105–108.

Lewis, J. S. (2015). *Asteroid mining 101: wealth for the new space economy.* Deep Space Industries.

Liebe, C. C. (1995). Star trackers for attitude determination. *IEEE Aerospace and Electronic Systems Magazine*, 10(6):10–16.

Liu, H., Li, F., Xu, X., and Sun, F. (2018a). Active object recognition using hierarchical local-receptive-field-based extreme learning machine. *Memetic Computing*, 10(2):233–241.

Liu, H., Wu, Y., and Sun, F. (2018b). Extreme trust region policy optimization for active object recognition. *IEEE transactions on neural networks and learning systems*, 29(6):2253–2258.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349.

Lupiáñez, J., Klein, R. M., and Bartolomeo, P. (2006). Inhibition of return: Twenty years after. *Cognitive neuropsychology*, 23(7):1003–1014.

MacKay, D. J. (1992). *Bayesian methods for adaptive models.* PhD thesis, California Institute of Technology.

Magnusson, M. (2009). *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection.* PhD thesis, Örebro Universitet.

McCann, S. and Lowe, D. G. (2012). Local naive bayes nearest neighbor for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3650–3656. IEEE.

Meyer, J.-A. and Filliat, D. (2003). Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317.

Micheloni, C. and Foresti, G. L. (2009). Active tuning of intrinsic camera parameters. *IEEE Transactions on Automation Science and Engineering*, 6(4):577–587.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). Recurrent models of visual attention. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2204–2212. Curran Associates, Inc.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al. (2002). Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al. (2003). Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, pages 1151–1156.

Montmerle, T., Ehrenreich, D., Lagrange, A.-M., Eggenberger, A., and Udry, S. (2010). Detection and characterization of extrasolar planets through doppler spectroscopy. *European Astronomical Society Publications Series*, 41:27–75.

Moravec, H. P. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 19–24.

Mueller, R. P. and Van Susante, P. J. (2012). A review of extra-terrestrial mining robot concepts. In *Earth and Space 2012: Engineering, Science, Construction, and Operations in Challenging Environments*, pages 295–314.

Muja, M. and Lowe, D. G. (2009). Flann, fast library for approximate nearest neighbors. In *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, volume 3. INSTICC Press.

Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.

Murphy, K. P. (2000). Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems*, pages 1015–1021.

Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation.* CRC Press.

Najemnik, J. and Geisler, W. S. (2005). Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391.

Najemnik, J. and Geisler, W. S. (2008). Eye movement statistics in humans are consistent with an optimal search strategy. *Journal of Vision*, 8(3).

Nakath, D., Clemens, J., and Rachuy, C. (2017). Rigid Body Attitude Control Based on a Manifold Representation of Direction Cosine Matrices. *Journal of Physics: Conference Series*, 783(1):012040.

Nakath, D., Clemens, J., and Rachuy, C. (2019). Active Asteroid-SLAM – Graph SLAM with Active Exploration and Landing-Site Discovery in a Deep Space Proximity Operations Scenario. *Journal of Intelligent & Robotic Systems*. Under review.

Nakath, D., Clemens, J., and Schill, K. (2018). Multi-Sensor fusion and active perception for autonomous deep space navigation. In *2018 21st International Conference on Information Fusion (FUSION) (FUSION 2018*, Cambridge, United Kingdom (Great Britain).

Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., and Schill, K. (2014). Active Sensorimotor Object Recognition in Three-Dimensional Space. In *Spatial Cognition IX*, pages 312–324. Springer.

Nakath, D., Rachuy, C., Clemens, J., and Schill, K. (2016). Optimal rotation sequences for active perception. In Braun, J. J., editor, *Proc. SPIE: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, volume 9872. SPIE Press.

Noë, A. (2004). *Action in Perception*. MIT Press.

Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Constellation.

Oliva, A. and Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Prorgess in brain research*, 155:23–36.

Olson, E. B. (2008). *Robust and Efficient Robotic Mapping*. PhD thesis, Cambridge, MA, USA. AAI0821013.

O'Regan, J. K. et al. (2001a). What it is like to see: A sensorimotor theory of perceptual experience. *Synthese*, 129(1):79–103.

O'Regan, J. K., Noë, A., et al. (2001b). A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(5):939–972.

Paluszek, M., Littman, M., and Mueller, J. (2010). Optical navigation system. In *AIAA Infotech@ Aerospace 2010*, page 3462.

Perryman, M. A. C., Lindegren, L., Kovalevsky, J., Hoeg, E., Bastian, U., Bernacca, P. L., Crézé, M., Donati, F., Grenon, M., Grewing, M., et al. (1997). The HIPPARCOS catalogue. *Astronomy and Astrophysics*, 323.

Pesce, V., Agha-mohammadi, A., and Lavagna, M. (2018). Autonomous navigation & mapping of small bodies. In *2018 IEEE Aerospace Conference*, pages 1–10.

Philipona, D., O'regan, J., Nadal, J.-P., and Coenen, O. (2004). Perception of the structure of the physical world using unknown multimodal sensors and effectors. In *Advances in neural information processing systems*, pages 945–952.

Philipona, D., O'Regan, J. K., and Nadal, J.-P. (2003). Is there something out there? inferring space from sensorimotor dependencies. *Neural computation*, 15(9):2029–2049.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148.

Posner, M. I. and Cohen, Y. (1984). Components of visual orienting. *Attention and performance X: Control of language processes*, 32:531–556.

Prinz, W. (1990). *A common coding approach to perception and action*. Springer.

Probst, A. and Förstner, R. (2017). Spacecraft design of a multiple asteroid orbiter with re-docking lander. *Advances in Space Research*.

Probst, A., González Peytaví, G., Eissfeller, B., and Foerstner, R. (2016). Mission concept selection for an asteroid mining mission. *Aircraft Engineering and Aerospace Technology: An International Journal*, 88(3):458–470.

Probst, A., González Peytaví, G., Nakath, D., Schattel, A., Rachuy, C., Lange, P., Clemens, J., Echim, M., Schwarting, V., Srinivas, A., Gadzicki, K., Förstner, R., Eissfeller, B., Schill, K., Büskens, C., and Zachmann, G. (2015). KaNaRiA: Identifying the challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Reineking, T. (2014). Belief functions: Theory and algorithms.

Reineking, T. and Clemens, J. (2013). Evidential fastslam for grid mapping. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 789–796. IEEE.

Reineking, T. and Clemens, J. (2014). Dimensions of uncertainty in evidential grid maps. In *International Conference on Spatial Cognition*, pages 283–298. Springer.

Reineking, T., Kluth, T., and Nakath, D. (2015). Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification. In *Computer Analysis of Images and Patterns*, pages 350–361. Springer.

Riedel, J., Bhaskaran, S., Synnott, S., Bollman, W., and Null, G. (1996). An automomous optical navigation and control system for interplanetary exploration missions.

Ross, S. M. (2014a). *A first course in probability*. Pearson.

Ross, S. M. (2014b). *Introduction to probability models*. Academic press.

Roy, N., Burgard, W., Fox, D., and Thrun, S. (1999). Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 35–40. IEEE.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE.

Russell, C. and Raymond, C. (2011). The dawn mission to vesta and ceres. In *The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres*, pages 3–23. Springer.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.

Rusu, R. B. and Cousins, S. (2011a). 3d is here: Point cloud library (pcl). In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE.

Rusu, R. B. and Cousins, S. (2011b). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

Schattel, A. (2018). *Dynamic Modeling and Implementation of Trajectory Optimization, Sensitivity Analysis, and Optimal Control for Autonomous Deep Space Navigation*. PhD thesis, Center for Industrial Mathematics (ZeTeM), University of Bremen.

Schattel, A., Cobus, A., Echim, M., and Büskens, C. (2017). Optimization and sensitivity analysis of trajectories for autonomous small celestial body operations. In Quintela, P., Barral, P., Gómez, D., Pena, F. J., Rodríguez, J., Salgado, P., and Vázquez-Mendéz, M. E., editors, *Progress in Industrial Mathematics at ECMI 2016*, volume 26. ECMI 2016, Springer.

Schattel, A., Echim, M., and Büskens, C. (2016). Low thrust trajectory optimization for autonomous asteroid rendezvous missions. In *6th International Conference on Astrodynamics Tools and Techniques (ICATT), Darmstadt, Germany.*

Scheeres, D., Gaskell, R., Abe, S., Barnouin-Jha, O., Hashimoto, T., Kawaguchi, J., Kubota, T., Saito, J., Yoshikawa, M., Hirata, N., et al. (2006). The actual dynamical environment about itokawa. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6661.

Scherrer, D. (2005). Using solar science to inspire: The education and public outreach projects of the hmi instrument on nasa's solar dynamics observatory (sdo). In *AGU Spring Meeting Abstracts*.

Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., and Zetzsche, C. (1999). Knowledge-based scene analysis with saccadic eye movements. In *Proc.SPIE*, volume 3644, pages 3644–3644–12.

Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., and Zetzsche, C. (2001). Scene analysis with saccadic eye movements: top-down and bottom-up modeling. *Journal of Electronic Imaging*, 10(1):152–160.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Sebestyen, G., Fujikawa, S., Galassi, N., and Chuchra, A. (2018). Spacecraft digital hardware. In *Low Earth Orbit Satellite Design*, pages 87–90. Springer.

Seedhouse, E. (2013). *SpaceX: making commercial spaceflight a reality*. Springer Science & Business Media.

Seedhouse, E. (2015). Bigelow expandable activity module. In *Bigelow Aerospace*, pages 87–98. Springer.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-icp. In *Robotics: science and systems*, volume 2, page 435.

Shafer, G. (1976). *A mathematical theory of evidence*, volume 42. Princeton university press.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, 1 edition.

Sim, R. and Roy, N. (2005). Global a-optimal robot exploration in slam. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 661–666. IEEE.

Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial intelligence*, 66(2):191–234.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer.

Tan, J., Liu, K., and Turk, G. (2011). Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44.

Thermos, S., Papadopoulos, G. T., Daras, P., and Potamianos, G. (2017). Deep affordance-grounded sensorimotor object recognition. *margin*, 17(31):35.

Thermos, S., Papadopulos, G. T., Daras, P., and Potamianos, G. (2018). Attention-enhanced sensorimotor object recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 336–340. IEEE.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.

Tronchetti, F. (2015). The space resource exploration and utilization act: A move forward or a step back? *Space Policy*, 34:6–10.

Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S. (2013). System design of the hayabusa 2—asteroid sample return mission to 1999 ju3. *Acta Astronautica*, 91:356–362.

Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The nbnn kernel. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1824–1831. IEEE.

Van Der Merwe, R. (2004). Sigma-point kalman filters for probabilistic inference in dynamic state-space models.

Van Leeuwen, F. (2007). Validation of the new hipparcos reduction. *Astronomy & Astrophysics*, 474(2):653–664.

van Susante, P. and Gertsch, L. (2018). Minerals from space: Terrestrial and extra-terrestrial perspectives. In *Conference: Earth and Space 2018 At: Cleveland, OH*. IEEE.

Von Helmholtz, H. (1867). *Handbuch der physiologischen Optik*, volume 9. Voss.

Warner, F. W. (2013). *Foundations of differentiable manifolds and Lie groups*, volume 94. Springer Science & Business Media.

Wilkes, D. R. and Tsotsos, J. K. (1993). Behaviors for active object recognition. In *Optical Tools for Manufacturing and Advanced Automation*, pages 225–239. International Society for Optics and Photonics.

Winston, P. H. (1992). Artificial intelligence.

Woodman, O. J. (2007). An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory.

Yarbus, A. L., Haigh, B., and Rigss, L. A. (1967). *Eye movements and vision*, volume 2. Plenum press New York.

Yim, J., Crassidis, J., and Junkins, J. (2000). Autonomous orbit navigation of interplanetary spacecraft. In *Astrodynamics Specialist Conference*, page 3936.

Yoshikawa, M., Fujiwara, A., and Kawaguchi, J. (2006). Hayabusa and its adventure around the tiny asteroid itokawa. *Proceedings of the International Astronomical Union*, 2(14):323–324.

Zacny, K., Paulsen, G., Davis, K., Mumm, E., and Gorevan, S. (2008). Honeybee robotics planetary drill systems. In *Lunar and Planetary Science Conference*, volume 39, page 1355.

Zapata, E. (2017). The opportunity in commercial approaches for future nasa deep space exploration elements.

Zetzsche, C. and Barth, E. (1990). Image surface predicates and the neural encoding of two-dimensional signal variations. In *Human vision and electronic imaging: Models, methods, and applications*, volume 1249, pages 160–178. International Society for Optics and Photonics.

Zetzsche, C., Schill, K., Deubel, H., Krieger, G., Umkehrer, E., and Beinlich, S. (1998). Investigation of a sensorimotor system for saccadic scene analysis: an integrated approach. In *Proc. 5th Int. Conf. Simulation Adaptive Behav*, volume 5, pages 120–126.

Zetzsche, C., Wolter, J., and Schill, K. (2008). Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. *Cognitive processing*, 9(4):283–297.

Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462.

A
Accumulated Publications

Kluth et al. [2013] Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2013). Sensorimotor Integration Using an Information Gain Strategy in Application to Object Recognition Tasks. In *Perception*, volume 42, pages 223–223. Pion LTD 207 Brondesbury Park, London NW2 5JN, UK, doi: 10.1177/03010066130420s101.

My share is 40%.
Published at a conference.
Attached in Appx. A.7 (pp. 201ff), discussed in Chap. 4.

In this poster and a corresponding abstract, the first implementation of the active sensorimotor object recognition system was presented. I partly designed the research, implemented the system, conducted the evaluation, and designed the poster. It was accepted for presentation at the 36th European Conference on Visual Perception ECVP 2013 in Bremen, Germany and the corresponding abstract was subsequently published.

Nakath et al. [2014] Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., and Schill, K. (2014). Active Sensorimotor Object Recognition in Three-Dimensional Space. In *Spatial Cognition IX*, pages 312–324. Springer, doi: 10.1007/978-3-319-11215-2˙22.

My share is 55%.
Published at a peer-reviewed conference.
Attached in Appx. A.8 (pp. 203ff), discussed in Chap. 4.

In this conference paper, a comprehensive description and evaluation of the sensorimotor object recognition system is given. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the Spatial Cognition IX (International Conference on Spatial Cognition 2014), Bremen, Germany.

Kluth et al. [2014] Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K. (2014). Affordance-Based Object Recognition Using Interactions Obtained from a Utility Maximization Principle. In *Computer Vision-ECCV 2014 Workshops*, pages 406–412. Springer, doi: 10.1007/978-3-319-16181-5˙29.

My share is 40%.
Published at a peer-reviewed conference.
Attached in Appx. A.9 (pp. 217ff), discussed in Chap. 4.

In this workshop paper, the active classification system is interpreted and evaluated in terms of visual affordances. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the Second Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis at the European Conference of Computer Vision (ECCV), Zurich, Switzerland.

Reineking et al. [2015] Reineking, T., Kluth, T., and Nakath, D. (2015). Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification. In *Computer Analysis of Images and Patterns*, pages 350–361. Springer, doi: 10.1007/978-3-319-23192-1˙29.

My share is 20%.
Published at a peer-reviewed conference.
Attached in Appx. A.10 (pp. 223ff), discussed in Chap. 4.

This conference paper presents an active classification system using kernel density estimation to obtain the likelihoods. In an evaluation, different top-down as well as a bottom-up approaches were compared. I partly designed the research, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015), Valetta, Malta.

Probst et al. [2015] Probst, A., González Peytaví, G., Nakath, D., Schattel, A., Rachuy, C., Lange, P., Clemens, J., Echim, M., Schwarting, V., Srinivas, A., Gadzicki, K., Förstner, R., Eissfeller, B., Schill, K., Büskens, C., and Zachmann, G. (2015). Identifying the Challenges for Cognitive Autonomous Navigation and Guidance for Missions to Small Planetary Bodies. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation.

My share is 10%.
Published at a conference.
Attached in Appx. A.1 (pp. 89ff), discussed in Chap. 3.

In this conference paper, an overview of the KaNaRiA project and the particular problems to be tackeled within its scope is given. I contributed the sensor fusion part of the paper. It was accepted for presentation at the 66th International Astronautical Congress, Jerusalem, Israel.

González Peytaví et al. [2015] González Peytaví, G., Clemens, J., Nakath, D., Probst, A., Förstner, R., Schill, K., and Eissfeller, B. (2015). Autonomous Orbit Navigation for a Mission to the Asteroid Main Belt. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation.

My share is 20%.
Published at a conference.
Attached in Appx. A.2 (pp. 103ff), discussed in Chap. 3.

In this conference paper an evaluation of the heliocentric navigation concept is presented, comparing an unscented Kalman filter and a particle filter. I partly designed the research, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 66th International Astronautical Congress, Jerusalem, Israel.

Nakath et al. [2016] Nakath, D., Rachuy, C., Clemens, J., and Schill, K. (2016). Optimal Rotation Sequences for Active Perception. In *Proc. SPIE*, volume 9872, pages 987204–987204–13. International Society for Optics and Photonics, doi: 10.1117/12.2223027.

My share is 45%.
Published at a conference.
Attached in Appx. A.3 (pp. 119ff), discussed in Chap. 3.

In this conference paper, active perception for autonomous heliocentric navigation is introduced. The evaluation investigates the localization performance of a particle filter in 2D, where projections of the measurement direction onto eigenvectors are used to estimate the expected impact of measurements on the localization uncertainty. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the SPIE Commercial + Scientific Sensing and Imaging, 2016, Baltimore, Maryland, United States.

Nakath et al. [2017] Nakath, D., Clemens, J., and Rachuy, C. (2017). Rigid Body Attitude Control Based on a Manifold Representation of Direction Cosine Matrices. volume 783, page 012040, doi: 10.1088/1742-6596/783/1/012040.

My share is 60%.
Published at a peer-reviewed conference.
Attached in Appx. A.4 (pp. 131ff), discussed in Chap. 3.

This conference paper introduces a proportional derivative (pd) attitude controller operating on the boxplus-manifold. In an evaluation, its behavior while conducting different maneuvers is investigated. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 13th International Conference on Advanced Control and Diagnosis ACD 2016, Lille, France.

Nakath et al. [2018] Nakath, D., Clemens, J., and Schill, K. (2018). Multi-sensor fusion and active perception for autonomous deep space navigation. In *21st International Conference on Information Fusion (FUSION 2018)*, pages 2596–2605. IEEE, doi: 10.23919/icif.2018.8455788.

My share is 70%.
Published at a peer-reviewed conference.
Attached in Appx. A.5 (pp. 145ff), discussed in Chap. 3.

This conference paper presents an improved version of the active heliocentric navigation system: it operates in 3D and directly projects the affected parts of the covariance on the measurement direction to estimate its impact. In an evaluation, different implementations of Bayesian filters are compared on different orbit geometries with respect to their reaction on particular active perception strategies. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It was accepted for presentation at the 21st International Conference on Information Fusion (FUSION 2018), Cambridge, UK.

Nakath et al. [2019] Nakath, D., Clemens, J., and Rachuy, C. (2019). Active Asteroid-SLAM – Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario. *Journal of Intelligent & Robotic Systems*, doi: 10.1007/s10846-019-01103-0.

My share is 75%.
Published in a scientific journal.
Attached in Appx. A.6 (pp. 155ff), discussed in Chap. 3.

This journal paper introduces the active autonomous asteroid-centric navigation approach which comprises Graph SLAM for the estimation of the trajectory and the map as well as a twofold trajectory generation and evaluation scheme. In an evaluation, the active Graph SLAM approach is investigated in a realistic local environment around the asteroid Itokawa. I partly designed the research, implemented the system, conducted the evaluation, and wrote the paper. It is accepted for publication by the Journal of Intelligent and Robotic Systems.

IAC-15-A3.IP.15

KANARIA: IDENTIFYING THE CHALLENGES FOR COGNITIVE AUTONOMOUS NAVIGATION AND GUIDANCE FOR MISSIONS TO SMALL PLANETARY BODIES

Alena Probst[*], Graciela González Peytaví[†], David Nakath[‡], Anne Schattel[§], Carsten Rachuy[**], Patrick Lange[††], Joachim Clemens[‡], Mitja Echim[§], Verena Schwarting[**], Abhishek Srinivas[†], Konrad Gadzicki[**], Roger Förstner[*], Bernd Eissfeller[†], Kerstin Schill[‡,], Christof Büskens[§], Gabriel Zachmann[††]**

[*] Institute of Space Technology & Applications, Space Technology Dept., Bundeswehr University Munich, Germany, A.Probst@unibw.de

[†] Institute of Space Technology & Applications, Navigation Dept., Bundeswehr University Munich, Germany, graciela.gonzalez@unibw.de

[‡] Cognitive Neuroinformatics, Sensor Fusion Group, University of Bremen, Germany, dnakath@informatik.uni-bremen.de

[§] Working group Optimization and Optimal Control, University of Bremen, Germany, ascha@math.uni-bremen.de

[**] Cognitive Neuroinformatics, Autonomy Group, University of Bremen, Germany, rachuy@informatik.uni-bremen.de

[††] Computer Graphics and Virtual Reality, University of Bremen, Germany, lange@informatik.uni-bremen.de

With the rapid evolution of space technologies and increasing thirst for knowledge about the origin of life and the universe, the need for deep space missions as well as for autonomous solutions for complex, time-critical mission operations becomes urgent. Within this context, the project KaNaRiA aims at technology development tailored to the ambitious task of space resource mining on small planetary bodies using increased autonomy for on-board mission planning, navigation and guidance.

This paper focuses on the specific challenges as well as first solutions and results corresponding to the KaNaRiA mission phases (1) interplanetary cruise, (2) target identification and characterization and (3) proximity operations.

Based on the KaNaRiA asteroid mining mission objectives, initially, a mission reference scenario as well as a reference mission architecture are described in this paper. KaNaRiA has been proposed as a multi-spacecraft mission to the asteroid main belt. Composed of a flock of prospective scout spacecraft, a mother ship carrying the mining payload and several service modules placed on a 2.8 AU parking orbit around the Sun, KaNaRiA intends to characterize main belt asteroid properties, identify targets for mining and perform a soft-landing for in-situ characterization and mining.

Subsequently, the autonomous navigation system design of KaNaRiA for the interplanetary cruise is presented. The navigation challenges, which arise in phases (1) to (3), are discussed. Particular attention is given to the sensor-technology readiness-level, accuracy, applicability range, mass and power budgets. In order to navigate in the vicinity of an asteroid, an information fusion algorithm is required that aggregates multi-sensor data as well as a-priori knowledge and solves the task known as simultaneous localization and mapping (SLAM). In order to deal with uncertain and inconsistent information and to explicitly represent different dimensions of uncertainty, a belief-function-based SLAM approach is used, which is a generalization of the popular FastSLAM algorithm.

The objective of the guidance task is the autonomous planning of optimal transfer trajectories according to mission driving criteria, e.g. transfer time and fuel consumption. Optimal control problems and the calculation of trajectory sensitivities for on-board stability analysis as well as real-time optimal control are explained.

Bringing cognitive autonomy to a spacecraft requires an on-board computational module as a central spacecraft component. This module is responsible for state evaluation, mission planning and decision-making regarding selection of potential targets, trajectory selection and FDIR. A knowledge-base serves as a database for decision making processes.

With the aim to validate and test our methods, we create a virtual environment in which humans can interact with the simulation of the mission. In order to achieve real-time performance, we propose a massively-parallel software system architecture, which enables very efficient and easily adaptable communication between concurrent software modules within KaNaRiA.

I. INTRODUCTION

Following the developments and the news on current space missions such as Rosetta or Dawn, one of the biggest challenges for small body rendezvous and landing missions is the large communication delay that leads to operational problems. Operations need to be planned thoroughly in advance. Nevertheless failures and anomalies often result in the complete loss of the spacecraft or lander. One approach to improve the reliability of complex operations is to enhance the autonomy, decision making and FDIR (fault, detection, isolation and recovery) capabilities of the spacecraft.

This is the approach that the project KaNaRiA takes up. The German acronym KaNaRiA stands for *Kognitionsbasierte, autonome Navigation am Beispiel des Ressourcenabbaus im All*, which translates into *Cognitive Autonomous Navigation for Deep Space Resource Mining.* As an interdisciplinary project, KaNaRiA focuses on autonomous mission planning, navigation and guidance in a-priori unknown environments dealing with the challenges of future space missions to minor planets. KaNaRiA strives to increase on-board spacecraft autonomy in the context of an asteroid mining scenario. The development of these concepts takes place in a virtual simulation environment, which serves as a test bed for a mission study. In this paper we give an overview of the KaNaRiA mission concept and the individual components of the system.

The paper is structured as follows. In section II and III, the engineering solutions applied to the particular mission scenario of KaNaRiA are presented, specifically the mission concept and reference scenario followed by the navigation system design and autonomous navigation concept.

Section IV covers the contribution of information fusion, which combines a-priori knowledge with sensor data to provide an information basis for autonomous decision-making.

In section V it is explained how the mathematical field of optimization and optimal control is used to calculate optimal interplanetary trajectories by solving infinite-dimensional optimal control problems.

In section VI the central component for on-board mission planning and autonomous decision-making is presented.

Section VII describes functionality of the simulation environment and its underlying software architecture.

II. MISSION: ASTEROID MINING

As an application for the proposed autonomous navigation, guidance and simulation solutions, an asteroid mining mission concept is defined.

The aim of asteroid mining opens up a huge space of scenarios and possibilities to implement a successful mission. The mission design changes depending on the desired resource, the purpose of usage or the location of the asteroid target. In order to specify a scenario, the JPL Rapid Mission Architecture [1] method has been applied.

II.I Mission Processes

The mission concept derivation is based on a separation and identification of processes that have to be fulfilled with the goal of mining a space body. First, the targets have to be mapped and characterized according to their natural resources and potential consideration for mining. These activities are done under the scope of Mapping, Characterization and Resource Determination (MCRD). Second, after having appointed a suitable target, the resource is mined by a separate miner (Resource Extraction and Exploitation, REaE). As an asteroid mining mission is by default a long-term mission, the transportation of the resources from the mining site to the refinery or designated user as well as the maintenance of the space elements involved have to be taken into account. Those activities are covered within the Maintenance and Logistics. For a more detailed description and definition of the mission, it is referred to Probst et al. [2]

As each of the processes involved in a successful mining mission imposes different requirements on the spacecraft architecture, separate spacecraft elements have been selected, each of them specialized for one specific process. The selection trade-off for each mission element architecture was done using a numerical method based on relative judgments with respect to suitable trade-criteria. The selection process is described in Probst et al. [3]

II.II Mission Elements

The following mission elements are involved in the mission scenario:

The *Potential Target Characterization Modules* (PTCMs) are in charge of exploring the considered targets in order to analyse their potential resource character.

The *KaNaRiA Miner Spacecraft* (KMS) lands on the designated target and excavates the resource.

The Refuel- and Repair- Elements (RF/RP) take care of the maintenance problems that occur.

An unmanned, autonomous *Operational Centre* (OC) serves as the main communication and delegation hub. It coordinates the mission elements and their tasks, sustains and collects the data and inherits the overall power of decision.

To complete the mining cycle, *resource transporters* are needed that carry the material from the mining site to the refinery or from there to the costumer.

II.III Mission Reference Scenario

As the mission scenario serves as a basis for the navigation, sensor fusion, guidance and autonomy algorithms and their implementation in a simulator, the mission scenario starts with a mission setup at a circular, Sun-bound parking orbit (2PO) with a semi-major axis of 2.8 AU. [2]

On 2PO, the OC, KMS and the maintenance spacecraft as well as the transporter are stationed whereas several PTCMs swarm out for their investigation and search for potential precious resources. Each PTCM consists of an orbiter and a re-docking lander so that it can visit more than one asteroid without coming back to 2PO. This way it is able to characterize each target thoroughly. The data obtained is relayed to the OC, which selects a definite target to which the KMS will head for mining.

In the simulator and further course of this project, the implementation and design will focus on the design of the PTCM [4] as the developed technologies and algorithms can be transferred and applied to the other involved modules as well.

III. NAVIGATIONAL CONCEPT FOR DEEP SPACE MISSIONS

The KaNaRiA reference mission scenario envisages four main operational phases according to the mining processes described in section II.I: MRCD (Mapping, Characterization and Resource Determination), REaE (Resource Extraction and Exploitation), Maintenance and Logistics. Each of these phases imposes stringent performance requirements for the navigation subsystems of the various mission elements and their navigation autonomy capabilities. Within this section, the MRCD mission operations timeline is presented. The navigation requirements for PTCM spacecraft during the MRCD phase are discussed. The navigation system design of the PTCM is described and an autonomous navigation concept [5] for interplanetary cruise is introduced.

PTCM Mission Operations Timeline

The operational concept for PTCM spacecraft is built upon the on-board autonomous capability for mission planning. Based on available system status information and collected knowledge about the target asteroid shape and dynamics, the spacecraft shall be able to select between 3 main concepts of operations while approaching an asteroid: an encounter mission and a lander mission with an additional option on redcoking the lander with the orbiter. The operational timeline for the scenarios is depicted in Fig. 1.

In an encounter mission scenario, the PTCM will perform remote sensing of the asteroid from a safe distance during a pre-planned time span. After finalization of the remote sensing campaign the PTCM will continue its course to a second target asteroid.

A lander mission scenario is selected if the asteroid target shows promising results after the remote sensing. The lander is released from the PTCM and uses its steering capabilities for safe landing on the designated landing site. The surface operations include a deep investigation of the asteroid's composition with a Low Frequency Radar as well as a Laser-Induced Breakdown Spectroscopy (LIBS) of the surface material. The data shall be relayed to the PTCM orbiter. The lander steering capabilities enable the performance of hopping or hovering manoeuvres between sample sites of interest. Additionally, the PTCM lander can ascent from the asteroid surface and re-dock to the PTCM orbiter in order to continue its course to a new asteroid. In case the PTCM delta-v capability is insufficient to perform a flight to a follow-up asteroid, the PTCM stays in the orbit around the asteroid and awaits - if profitable - the RF for refuelling.

The navigation system design of the PTCM spacecraft has been developed in order to ensure the spacecraft's capability to determine its location either absolutely in space or relatively to the target throughout all mission phases.

PTCM Navigation Requirements

The KaNaRiA mission concept proposes the deployment of 5-15 medium-size spacecraft, called PTCM, from a cargo control centre located in Sun-bound orbit about 2.8 AU distance from the Sun and 1.8 AU from Earth. At such distances, two-way ground-spacecraft communication delays exceed thirty minutes. Free-space transmission losses are as high as 290 dB in Ka-band, in which future deep-space communication

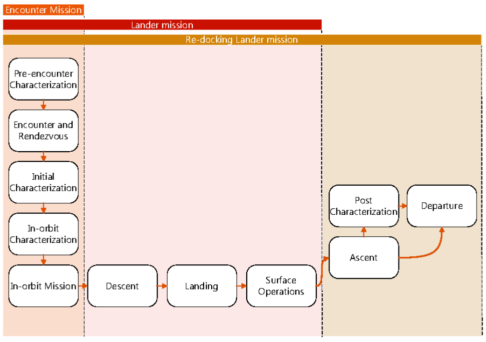


Fig. 1: Operational timeline (from left to right) for PTCM spacecraft,.

infrastructure will operate. The generation of sufficient power to frequently communicate with Earth for tracking purposes, while keeping all subsystems thermally conditioned and performing asteroid characterization operations is not a trivial problem given that the solar flux does not exceed 200 W/m^2. Furthermore, the simultaneous operation of 5-15 missions is a challenge for the already busy tracking and processing schedule of the deep-space ground infrastructure. It is therefore necessary to design the PTCM with a sensible balance between system complexity and self-contained autonomous navigation capabilities.

It has been determined that a PTCM shall be capable of performing on-board orbit determination (OD) at the 100 km precision during cruise in order to support guidance and control during orbit manoeuvring. OD shall be performed fully autonomously without ground support. The stability of the on-board solution shall be guaranteed for a transfer time as long as 4 years. OD updates from ground shall be expected regularly assuming a tracking campaign of maximum 1 week every 5 months.

The PTCM shall be targeted to a rendezvous-plane crossing point between 100 and 500 km from the asteroid surface depending of the volume sphere of influence of the particular object. The 3-σ error ellipsoid at rendezvous condition shall be constraint to 100 m – a requirement that has been fulfilled comfortably by previous asteroid fly-by missions.

During the asteroid in-orbit phase, a thorough characterization of the asteroid surface properties, internal structure as well as landing site selection and mapping will be carried out. During the observation campaigns a position accuracy in the order of meters relative to the asteroid surface shall be achieved.

The landing sequence will consist of a horizontal equalization phase and a subsequent vertical descent. The landing strategy has been designed to ensure soft landing (the survival of the PTCM lander structure), safe landing (safety of landing site avoiding obstacles bigger than 50 cm and slopes higher than 10 degrees) and hazard detection capability up to 10 minutes from touchdown.

Navigation System Design for a KaNaRiA PTCM

The PTCMs have been designed to perform inertial-aided optical navigation throughout all mission phases. In Table 1 a list of the navigation instruments has been provided including their type, mass and primary usage.

Cruise navigation

During cruise the angular observations of planet chords, star-planet and star-Sun angles are combined with the relative Doppler shift of the optical Sun spectra to derive spacecraft position and velocity. The self-contained navigation approach is based on the method proposed by Guo [6] and further investigated by Yim [7].

Instrument	Mass [kg]	Usage
Resonance Scatter Interferometer	42.2	Optical Sun Doppler observations
Coupled Star-Sun tracker	1.98	Stellar attitude and star-planet observations
Fine Sun Sensor	0.65	Coarse Sun attitude
Wide-Angle Camera	2	Asteroid detection and mapping
Narrow-Angle Camera	6	Surface mapping
Lidar Altimeter	3.52	Range finder
3D Lidar	6.5	Asteroid mapping
Space Inertial Reference Unit	7	Inertial position and attitude reconstruction

Table 1: PTCM navigation sensor suite

Spacecraft attitude is reconstructed from the stellar attitude provided by star tracking and from the rate-gyro integration during manoeuvring. Coarse Sun attitude sensors are mounted as back-up solution.

Fig. 2 shows the power flux available from planetary emission and chord lengths of solar system planets in the optical bandwidth as observed by a spacecraft flying a sun-bound circular orbit at 2.8 AU.

Planetary atmospheric and surface albedo has been taken into account. The main selected bodies to be observed for navigation are the Sun, Jupiter and Earth. However other planetary bodies, including the targeted asteroid, are observed when illumination and geometry

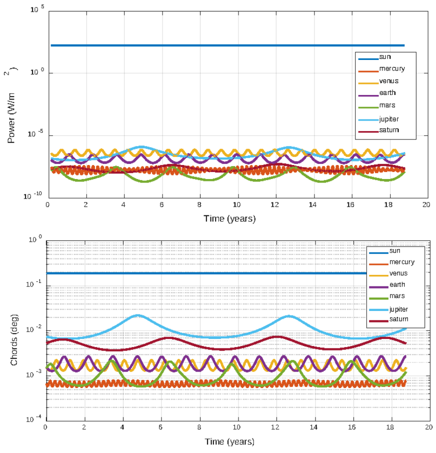


Fig.2: Power flux (top) and angular chord length (bottom) of Sun and solar system planets as observed from a Sun-bound circular orbit with semi-major axis of 2.8 AU.

conditions are favourable.

During cruise, Doppler frequency shift measurements from the Sun optical spectra are used to derive the spacecraft radial velocity. The derived radial velocity measurements are combined with planet chord length angles, planet-star and Sun-star angles. Angular measurements are processed according to standard celestial navigation procedures together with radial velocity measurements in an unscented Kalman filter. A particle filter is simultaneously executed in parallel with timely state updates from the Kalman filter. The particle filter (see section I.V) allows for a robust estimation in mismodelled dynamic environments, as for instance, the vicinity of an asteroid whose gravity field has not been probed. Fig. 3 illustrates the optical cruise navigation system of a PTCM spacecraft.

Optical navigation is aided by means of inertial measurements from the space inertial reference unit during orbit and attitude manoeuvring.

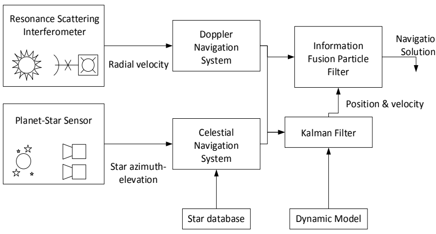


Fig. 3: Integrated celestial and optical Sun Doppler navigation system.

Asteroid relative navigation

In the vicinity of the target asteroid, optical navigation is implemented by means of feature tracking with two optical cameras and a 3D LIDAR. Visual SLAM (simultaneous localization and mapping) is used to reconstruct the asteroid shape and global map, and to locate the spacecraft relative to the generated surface map (see section I.V). A parallel estimation of both, spacecraft state and map, allows for increasing accuracy in the asteroid spin-state knowledge i.e., rotation axis orientation, rotation rate, tumbling modes, etc.

Star trackers are used for stellar attitude reconstruction as long as the asteroid covers between 60 and 80% of the instrument field of view. Rotation-rate measurements are collected from gyros to integrate attitude between stellar-blind phases and during the descent of the PTCM lander.

During descent, the PTCM lander uses a LIDAR altimeter to reconstruct height and vertical speed independently from the main SLAM navigation engine. The LIDAR altimeter solution is fed as input for the collision avoidance decision process handled by the on-board mission planning autonomy.

IV. MULTI-SENSOR FUSION FOR SPACE NAVIGATION

The information fusion subsystem aggregates multi-sensor data and a-priori knowledge to a unified representation, which serves as a basis for cognitive autonomous decision-making (Fig. 4). This bio-inspired model of decision-making relies on perceptions governed by top down as well as bottom up information flows. [8,9]

In particular, the aggregated information is comprised of *i)* top-down a-priori knowledge about the world and the spacecraft as well as *ii)* bottom-up perceived knowledge, which consists of fused data from multiple sensors. In conjunction, this information results in an estimate of the current spacecraft and environment state.

The multi-sensor fusion and state estimation solves the versatile challenges posed by the different mission phases (see section II) within one framework. Throughout all mission phases, a particle filter is used to approximate the desired probability distribution.

In the interplanetary cruise phase, the distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t},\boldsymbol{u}_{1:t})$[‡‡] over the current spacecraft state

$$\boldsymbol{x}_t = [\boldsymbol{r}_t^T, \boldsymbol{q}_t^T, \dot{\boldsymbol{r}}_t^T, \dot{\boldsymbol{q}}_t^T, \ddot{\boldsymbol{r}}_t^T, \ddot{\boldsymbol{q}}_t^T\,]^T$$

given all measurements $\boldsymbol{z}_{0:t}$ and controls $\boldsymbol{u}_{0:t}$ is estimated in a heliocentric reference frame, where $\boldsymbol{r}_t$ is the position, $\boldsymbol{q}_t$ the attitude, $\dot{\boldsymbol{r}}_t$ the velocity, $\dot{\boldsymbol{q}}_t$ the angular velocity, $\ddot{\boldsymbol{r}}_t$ the acceleration, and $\ddot{\boldsymbol{q}}_t$ the angular acceleration of the spacecraft. $\boldsymbol{z}_t$ contains measurements from the interferometer, the coupled Sun-star tracker and the wide-angle camera (see section III).

In the MCRD phase, the camera suite and the mapping LIDAR are able to perceive the asteroid. This enables the multi-sensor fusion module to estimate a map Y of the approached asteroid. This provides a

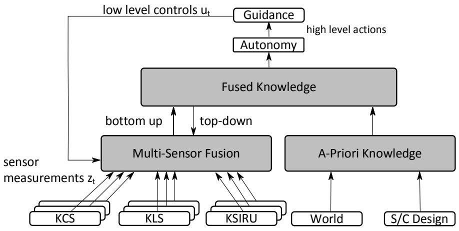


Fig. 4: Knowledge acquisition process for cognitive autonomous decision-making.

[‡‡] For convenience reasons we use $\boldsymbol{a}_{0:t}$ as a short notation for a time series of variables $\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_t$.

physical description of the asteroid and, even more essential, can be used as a reference for relative spacecraft state estimation.

Although the two tasks of localization and mapping can be solved separately, they are not independent of each other. It is a joint estimation problem commonly known as Simultaneous Localization and Mapping (SLAM) [10] (Fig. 5). However, using a conditional independence assumption, the corresponding joint probability distribution can be factorized into one conditional distribution over the trajectory $\boldsymbol{x}_{0:t}$ and one over the map Y:

$$p(\boldsymbol{x}_{0:t}, Y|\boldsymbol{z}_{0:t}, \boldsymbol{u}_{1:t}) = \underbrace{p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t}, \boldsymbol{u}_{1:t})}_{Trajectory}\underbrace{p(Y|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t})}_{Map}.$$

This allows us to use a technique called Rao-Blackwellization. [11] In the first step, the distribution over the trajectory is approximated by the particle filter [12] using controls, measurements and map estimate. In the second step, the current state is assumed to be known and the distribution over the map is computed analytically.

Initially, a landmark-based map is estimated in order to establish robust relative navigation in an asteroid-centric reference frame. The landmarks will be extracted by performing bio-inspired feature detection and description using Intrinsic 2 Dimensional (I2D) features [9,13] on the images obtained by the on-board cameras and with the distance information provided by the LIDAR instruments.

When the landmark map has full coverage and allows for a robust localization, it is extended by a belief-function-based grid-map of the asteroid in the proximity operations phase. It divides the volume into discrete grid cells where each grid cell represents an estimate of a corresponding piece of the physical environment. While the uncertainty regarding the true state is usually represented by a Bayesian probability, we are using belief functions [14,15] here, which allow to assign probability mass not only to the singletons $a \in \Theta$ of a hypothesis space Θ but also to all subsets of the power set $A \subseteq \wp(\Theta)$ including the superset Θ and the empty set $\emptyset$. This approach makes different dimensions of uncertainty explicit. E.g. a full lack of evidence is expressed by assigning all mass to Θ while conflicting evidence is expressed by mass assigned to $\emptyset$. In the Bayesian probability framework, both cases would result in an equal distribution and would be therefore undistinguishable. There are several works on mapping using belief functions [16,17,18] while a belief-function-based SLAM approach as a generalization of the successful grid-map based FastSLAM [19] algorithm was presented by Reineking and Clemens. [20] This approach was already applied in the context of extra-terrestrial exploration. [21,22]

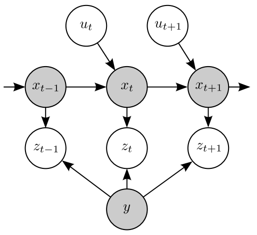


Fig. 5: Bayesian Network depicting the SLAM-problem.

The combination of belief functions and a grid map allow for *i)* a finer representation of the physical environment and *ii)* a better representation of the cognitive uncertainties. [23] This in turn enables the autonomy to pursue advanced exploration strategies to actively investigate possible landing sites, with respect to commodities, hazardous areas and fuel consumption. Based on the uncertainty information in the maps (grid-map as well as landmark based) the autonomy can be provided with desired actions with respect to every navigation instrument. Thus, particular actions can be assessed for their expected information gain.

V. OPTIMAL TRAJECTORY PLANNING

Trajectory planning for deep space missions is a topic of great interest. Mathematical fields like optimization and optimal control can be used to realize autonomous missions while protecting resources and making them safer. A perturbed *optimal control problem* (OCP(p)) has the form

$$\begin{aligned} \min_{x,u} F(x,u,p,t) &:= g(x(t_f),t_f) + \int_0^{t_f} f_0(x(t),u(t),t,p)dt \\ s.t. \qquad \dot{x}(t) &= f(x(t),u(t),t,p) \\ x(0) &= x_0 \\ \Psi(x_0,x(t_f),p) &= 0 \\ C(x(t),u(t),t,p) &\leq 0 \end{aligned}$$

with F being the objective function depending on the state $x(t)$ at time $t \in [0, t_f]$, the vector p describing model perturbations and the control function $u(t)$ by which the system's dynamic f can be influenced via differential equations. The control u has to be chosen in such a way that the constraints C as well as the initial and terminal conditions Ψ are fulfilled while minimizing the objective function F.

In principle, there exist two ways to solve an OCP(p), the so called indirect and direct methods. The indirect methods are being studied since several decades and need advanced skills regarding optimal control theory. Some algorithms are described in Bürlisch [24], Deuflhard [25], Ho and Bryson [26] as well as Miele [27]. The direct approach transcribes the infinite-dimensional OCP(p) into a finite-dimensional *non-linear optimization problem* (NLP(p)) via discretization of states and controls. [28,29] An NLP(p) consists of an objective function F and constraints G:

$$\begin{aligned} \min_{z} \quad & F(z,p) \\ s.t. \quad & G_i(z,p) = 0,\ i = 1,\dots,M_e \\ & G_i(z,p) \leq 0,\ i = M_e+1,\dots,M \end{aligned}$$

The objective function F depends on the optimization vector $z := (x_1^T, \dots, x_{N_t}^T, u_1^T, \dots, u_{N_t}^T)$ with $x_i, u_i, i = 1, \dots, N_t$ representing the former x and u at discrete time points $0 = t_1 < t_2 < \cdots < t_{N_t} := t_f, x_i \approx x(t_i), u_i \approx u(t_i)$ and the perturbation vector p. For a fixed parameter $p = p_0$ an optimal solution is called the *nominal* or *undisturbed solution* indicated by $z(p_0)$.

The OCP(p) formulation's dynamic model describes the movement of the spacecraft due to main gravitational influences of the sun and other planets as well as the thrust commands through ordinary differential equations (ODEs):

$$\dot{x} := \begin{pmatrix} \dot{p}_{sc} \\ \ddot{p}_{sc} \\ \dot{m}_{sc} \end{pmatrix} = \begin{pmatrix} \dot{p}_{sc} \\ \sum_{i \in I} \mu_i \cdot \frac{r_i}{\|r_i\|_2^3} + \frac{T}{m_{sc}} \\ -\frac{\|T\|}{g_0 Isp} \end{pmatrix}$$

Herein p_{sc} is the position vector of the spacecraft, $\mu_i, i \in [Sun, Mars, Jupiter, Saturn]$ is the gravitational constant of the according celestial body and r_i the direction vector between spacecraft and body, $T = [u_x, u_y, u_z]$ is the thrust vector, m_{sc} the spacecraft's recent mass, I_{sp} its specific impulse and g_0 the gravitational constant of Earth.

Within the optimization there exist several methods to solve such ODE systems. One is the so-called full discretization, where all states and controls are calculated for a chosen number of discrete time points. An alternative is to use *multiple shooting methods*. Here the solution space is divided into several sections by so-called *multi-nodes* and for each section a *single shooting method* is applied. [30] It is sufficient to combine the sections by additional constraints in order to gain the correct solution in the end. In the KaNaRiA implementation the position of the multi-nodes is let free for optimization.

These methods will be investigated to achieve a robust and efficient optimization for each of the systemically different navigation phases of a space mission. The resulting non-linear high-dimensional optimization problems are solved using the software package WORHP [31] ('We Optimize Really Huge Problems'). This is especially efficient for solving high-dimensional problems like those resulting from the discretization of optimal control problems as it uses for example the sparsity information of the derivative matrices.

Additionally, an on-board-capable *parametric sensitivity* and *stability analysis* of optimal nominal solutions towards perturbations will be performed in KaNaRiA. Perturbations are for example deviations in the assumed amount of left over fuel, the magnitude of the solar pressure or the asteroid's gravitational influence, which may have a great impact on the practicability of a planned trajectory. Changes in the optimal solution of the undisturbed problem in case of deviating values p from nominal values p_0 can be estimated by calculating the solution vector

$$z(p) \approx z(p_0) + \frac{dz}{dp}(p_0)(p - p_0)$$

while only the nominal solution $z(p_0)$ and its sensitivities $\frac{dz}{dp}(p_0)$ need to be computed.

Whereas offline calculations of optimal trajectories allow for their investigation, a practical online-realization can only be achieved through special *real-time capable methods*. Based on the parametric sensitivity analysis and dependent on the different phases of a space mission and their special claims different trajectory optimization and real-time tracking strategies will be developed for differing time scales. When approaching the asteroid further and especially when entering the landing phase the challenges of efficient real-time capable control interventions increase due to the weak, inhomogeneous gravity field resulting from the relative small mass, irregular form and unknown rotation of the asteroid.

Implementation:

A simple way to achieve an orbit transfer is the Hohmann transfer orbit, but it is only applicable under strong constraints. That is why in KaNaRiA another approach was chosen. For the cruise phase a maximum of three thrust commands may be applied, one at the beginning of a trajectory, one at the end and one at an optimized time point in between. These commands are sufficient regarding the long time frame of the flight without serious perturbation forces. To model impulsive thrusting more accurately an application-adapted model is developed. By using the objective function

$$F = w_{tf} t_f - m_f (1 - w_{tf})$$

with t_f being the total flight time, m_f the spacecraft's final mass and $w_{tf} \in [0,1]$ a weighting factor where any fit between time- and energy-optimization can be chosen. The start mass of the spacecraft is 4000 kg, the fuel mass 1500 kg, the I_{sp} 318 seconds and the thrust is limited to 340 to 440 Newton. The optimization was performed considering the influences of the planets Mars, Saturn and Jupiter. The boundary condition was meeting the position and velocity of the asteroid within a certain range sufficient for the cruise phase. The solutions for full time and full energy optimization can be seen in Fig. 6 and Table 2. With 2157.56 kg of fuel consumption and a total flight time of 796.747 days the flight of the energy-optimal trajectory needs 125.55 kg of fuel less but 30.493 days longer than flying the time-optimal trajectory (Table 2). The energy-optimal trajectory contains two thrust commands whereas the time-optimal trajectory consists of three thrust commands in order to meet the objective. This way in order to meet the energy-optimal objective, the spacecraft might orbit on the original trajectory before thrusting for the first time. The changes in the z-position differs the most since changing the inclination of a trajectory is highly energy consuming. In comparison to the x-/y-positions, the thrusts lead to only a small adjustment in the z-position. For both trajectories the last thrust is applied at the end of the trajectory, whereas only the time-optimal trajectory has a thrust at the beginning of the manoeuvre (Fig. 6).

The solution trajectories show strong differences according to the chosen objective priorities which means being able to save a lot of mission time or fuel consumption according to the mission's needs and allowing for various and considerably different autonomous decisions.

VI. COGNITIVE SPACECRAFT AUTONOMY

The autonomy module is the central component for autonomous reasoning and decision-making regarding normal mission operation as well as emergency situations. It controls all sub-modules of the spacecraft and all processes related to reasoning, plan generation, plan evaluation, plan execution and FDIR during all phases of the mission.

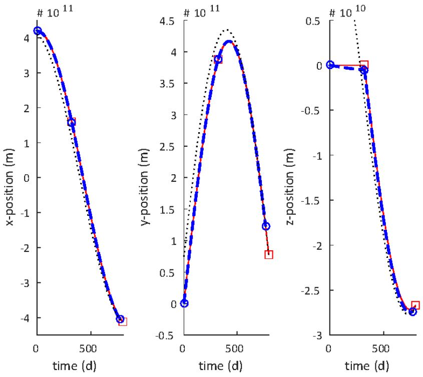


Fig. 6: X-, y- and z-position in meters of time-optimal (dashed blue line) and energy-optimal (solid red line) trajectory over time in days. The dotted black line shows the asteroid's position. Circles (time-optimal) and squares (energy-optimal) show the time points of the thrust commands.

	$w_{tf} = 0$	$w_{tf} = 1$	diff
Opt. criterion	energy	time	
Flight time (d)	796.747	766.254	30.493
Fuel (kg)	2157.56	2283.11	125.55
Line color (Fig. 6)	red	blue	

Table 2: Optimization criterion, flight time in days and fuel consumption in kg for two different mission trajectories.

During normal mission operation, the autonomy module monitors both phase-specific mission objectives and the current state of the spacecraft. Based on these it generates plans to either achieve primary (e.g. locate the asteroid using optical sensors, maintain a stable orbit around the asteroid, perform the docking/landing operation) or secondary objectives (e.g. calculate alternative trajectory to further increase information on possible landing site). As the scenario is of a highly dynamic nature, the system periodically requests re-evaluation of plans to check whether they are still applicable. The appropriate strategy for re-evaluation is based on current system resources and time constraints. The autonomy module has to decide on and ensure commitment to one plan, yet retain the option to reconsider the commitment at a later point – when new information becomes available.

Uncertain knowledge resulting from incomplete or incorrect data poses a central challenge to reasoning and decision making, therefore the system has to consider these kind of uncertainties in the decision making process. Based on the biologically inspired principle of information maximization, the autonomy module seeks to minimize and resolve these uncertainties by employing information gain strategies and active perception to extend and improve the amount and quality of the available knowledge.

As autonomous handling of emergency situations is vital, the module utilizes FDIR algorithms to react to anomalies as they are detected, by reprioritizing primary and secondary mission objectives as well as planning and executing appropriate fault-detection, fault-isolation and fault-recovery plans.

Situation Analysis and Evaluation

To create a basis for decision-making and plan generation, the current state of the spacecraft and all information available to it has to be analysed and evaluated. This includes a-priori knowledge (spacecraft configuration, mission phase specific objectives), internal data (navigation variables, fuel, mass, health status) and external data (sensor measurements, asteroid properties, potential targets). Sensor information from optical cameras, an imaging LIDAR and a LIDAR altimeter are provided by the sensor fusion [8] to the autonomy module and combined to create maps that assign potentially hazardous areas, points of interest and potential landing sites to regions on the asteroid. In addition, boundary conditions for trajectory requests regarding different mission phases and actions are added.

Plan Generation, Assessment and Execution

During plan generation, the system decomposes high-level objectives into a sequence of actions. These are selected from a dynamic set of currently available actions and based on the current beliefs of the spacecraft. At the atomic level, actions can be executed by the spacecraft actuators, which include spacecraft propulsion, reaction wheel control, and communication with other entities, sensor control and deployment of other vehicles (PTCM orbiter and lander). As the environment is dynamic, objectives can become unachievable and thus plans can become obsolete. The autonomy must be able to assess whether a given plan is still feasible and react accordingly.

Attitude and Sensor Control

To fulfil phase specific mission objectives that require distinct sensor and actuator alignments, the autonomy module has to provide an attitude control sequence based on both proposed priority rankings of measurable information and communication requirements.

This attitude control sequence is based on a previously calculated trajectory, where a trajectory is represented as a sequence of positions and time points. This sequence is split into segments at the control points of the trajectory. For each of these segments a spacecraft orientation is calculated for which all available sensors potentially provide the best measurements with respect to the maximisation of gained information.

From these orientations along the trajectory, the required attitude controls can be determined. Taking into account the potential information gain and hazards along this path, a sensor control plan for the trajectory is calculated, which specifies the sensor activation and deactivation at all time points.

Autonomous FDIR

To enable the system to autonomously perform fault detection, isolation and recovery (FDIR), current knowledge about the spacecraft and the world is used to infer about possible erroneous states. Algorithms for anomaly detection are utilized to determine unusual world- or spacecraft state configurations (e.g. conflicting datasets, unusual high uncertainty) that indicate a hard- or software problem. These are analysed regarding fault-identification and fault-recovery. If available, information on error-models of sensors and probabilities for different error scenarios will be incorporated in this analysis. If one or more recovery strategies exist, the necessary actions to be performed and possible constraints on the further action selection and plan generation (e.g. an actuator ceased to function) will be evaluated. In addition findings of this analysis are provided to the sensor fusion to enable this module to adapt the corresponding sensor models accordingly.

VII. MASSIVELY PARALLEL AND PHYSICALLY-BASED SIMULATION

In this section, we highlight two key aspects of KaNaRiA's simulation software. First, we give an overview of our simulation software with a focus on its novel approach to concurrency control management. Second, we will present the challenges for our novel concept of gravity field simulation for irregularly shaped celestial bodies.

Realistic spacecraft simulations have to cover all aspects of a mission scenario in real-world detail. Internal spacecraft components, the space environment with its physical forces and disturbances, the sensor data acquisition chain, and the spacecraft actuator and propulsion systems have to be modelled and simulated.

One key aspect of such simulations is the validation and testing of specific performance aspects (e.g. navigation algorithms), enabling sophisticated analyses for engineers that would otherwise be impossible. These analyses (e.g. spacecraft landing procedure performance) require comprehensive simulation and the monitoring of vast amounts of generated data.

In recent years, simulation has emerged as a key technology for improving and streamlining the conceptualisation and design of vehicles by simulation in "virtual testbeds". [32,33] Virtual testbeds are constituted by a sophisticated physically-based simulation of both the vehicle and its designated environment, as well as real-time, immersive rendering and 3D interaction techniques. These testbeds give engineers the opportunity to interact with the simulated vehicle in order to gain comprehensive understanding of possible design flaws as early as possible during the design process. [32,34]

Consequently, the main challenge of such virtual end-to-end simulations for space missions is real-time simulation with highly responsive interactivity while maintaining realistic physical models. In this context, an enormous amount of software components is working in order to simulate both, spacecraft behaviour and required input data. Additionally, spacecraft engineers would, ideally, have the ability to easily manipulate parameters of the spacecraft(s), change aspects of space environment such as disturbances, add or remove sensors or other spacecraft components, and interactively test the spacecraft(s) under a variety of conditions.

In order to achieve the above stated software requirements, we have proposed and implemented the KaNaRiA virtual simulation (KVS) [35], which proposes an easily adaptable and customizable massively-parallel virtual reality system architecture with a centralized software infrastructure to attain real-time performance of the overall simulation.

Consequently, KVS enables the analysis and testing of autonomous spacecraft operation, spacecraft navigation algorithms, and spacecraft subsystems in an enriched, virtual world. It leverages physics-based spacecraft models in conjunction with high-quality, multimedia visualization and immersive interaction techniques to form an intuitive, accurate engineering tool.

KVS has been designed to take advantage of an open source game engine targeted at the video game industry. Thus, KVS is able to bridge the gap between traditional, high-fidelity analysis tools [36] and graphically realistic, immersive, and interactive simulations.

Some of the highlights of KaNaRiA's virtual simulation include:

- Real-time 3D rendering of complex space environments & spacecraft models
- Real-time simulation of spacecraft subsystems, sensors as well as actuators
- The ability to observe internal spacecraft data intuitively
- Controlled, repeatable testing for advanced simulations
- Intuitive and consistent user interface.

Rendering, internal multi-component spacecraft simulation, and interaction with the overall system happens completely in parallel in KVS. To avoid any latency between those parallel software components, KVS uses our novel concurrency control management (CCM) for wait-free data exchange, with its core being a global hash map, called key-value pool (*KVPool*, Fig. 7). [37,38] The KVPool is a centralized data storage that maintains the complete shared world state of the simulation without being a traditional, heavy-weight database.

Every simulation aspect, such as spacecraft subsystems, sensors, actuators, and any physical models are implemented as *entities*, which can access the KVPool. Other software components can access the data by simply passing the key to the KVPool. The wait-free behaviour of KVS's KVPool results in a dramatic speed-up of several orders of magnitude compared to traditional lock-based approaches (see Fig. 8), while avoiding all their problems like deadlocks or thread starvation. Moreover, it overcomes the well-known many-to-many interface problem of the data-flow-based approach found in many traditional VR system architectures.

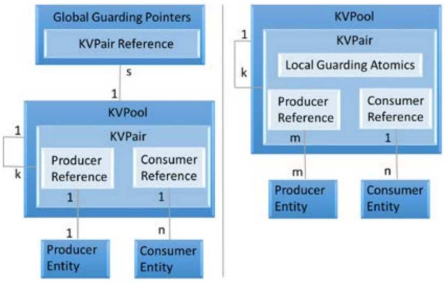


Fig. 7: Entity-relationship diagram of wait-free access to the shared simulation state by global guards (left) [37] and local guards (right). [38]

Furthermore, KVS's software infrastructure facilitates automatic code generation for virtual testbeds via domain specific modelling. [32] In addition, it can also be used for other data-driven simulation domains such as multi-agent-systems. [32]

Testing navigation and autonomous guidance algorithms for landing and orbiting an asteroid, under micro-g or milli-g gravity fields is crucial for developing fail-safe landing procedures. Therefore, KVS has to simulate a realistic gravity field around an asteroid for a given shape model (polygonal mesh) and density distribution at every point in space. We aim for fast and accurate computation of gravitational fields for any given asteroid. Currently spherical or ellipsoid harmonics approaches are the computationally least inexpensive compared with other approaches.

However, spherical harmonics series diverge within the Brillouin sphere [39] (see Fig. 9); hence, the

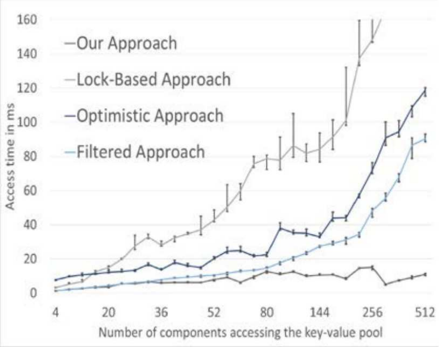


Fig. 8: Timings of a combined read and write operation for massively parallel access to a shared data structure.

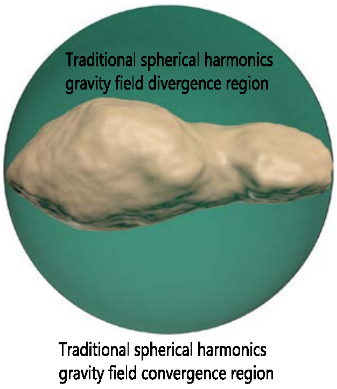

Traditional spherical harmonics gravity field divergence region
Traditional spherical harmonics gravity field convergence region

Fig. 9 : Brillouin sphere of asteroid Toutatis.

gravitational field computed close to the surface of an asteroid is inaccurate. [40] This results in incorrect simulated gravitational forces acting on the spacecraft during landing phase.

Takahashi et al. [41] overcame this issue with interior gravity field approach and alternatively with the interior spherical Bessel gravity field model. [42]

However, the former approach is computationally expensive as different sets of interior spherical harmonic coefficients have to be computed separately for each and every point on the asteroid surface, these different sets of coefficients are only applicable for gravity field computation within their respective interior sphere touching the respective point. [41] On the other hand, the pre-processing in the latter approach is computationally very expensive, which is not suitable for our purposes, since we need to be able to compute the field for any asteroid during runtime of the simulator. In our case, we generate the asteroids' shape models and density distributions procedurally in order to test guidance algorithms for landing on different types of asteroids (with respect to shape and density distributions) as well as sensor fusion algorithms for navigation. Therefore, we are currently working on an approach that computes the gravitational field of an asteroid in real-time while maintaining fast pre-processing. In our approach, we basically compute a sphere packing of a shape model of given asteroid using the modified protosphere algorithm from Weller et al. [42,43] with constraint on the radiuses of spheres based on the known prior asteroid density distribution. This method produces uniform density spheres that can be considered as point masses, then computing gravitational potential/acceleration at any given point is a trivial scalar/vector summation of potentials/accelerations applied by each sphere at that point (see Fig. 10). The sphere packing generation and summation computation for gravitational potential/acceleration are parallelizable. Hence, the pre-processing and gravity field computation are fast, which are suitable for our computational demands.

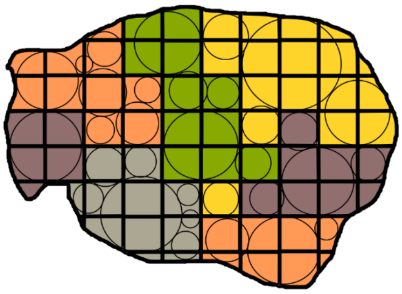


Fig. 10: Sphere packing of an asteroid with density distribution constraint on radius. The colours indicate different densities inside the asteroid.

VIII. CONCLUSION AND OUTLOOK

The KaNaRiA project focuses on the development of new autonomous decision-making, navigation, sensor fusion and guidance methods implemented on a virtual spacecraft. Within the spacecraft design, an autonomy module serves as the central controlling unit managing the data obtained and created by the navigation, fusion and guidance, generating and executing plans as well as controlling the attitude.

As an application for the development of this approach, an asteroid mining mission concept was developed. It considered asteroid mining as a long-term space activity. As the initial mission analysis led to the decision of a parking orbit located in the asteroid main belt, all involved mission elements need advanced autonomy strategies for navigation and guidance as well as mission planning and operations. During the KaNaRiA project, the design and application of the autonomous strategies focuses on the PTCMs consisting of an orbiter and a re-docking lander designed for a multi-rendezvous asteroid mission.

The navigation concept was designed for the PTCM operations timeline. It enables a thorough characterization of the asteroid surface properties as well as mapping including landing site selection with an envisaged position accuracy of several meters. The instrument suite to perform inertial-aided optical navigation under the imposed constraints by the mission concept was presented and the methods to conduct the observations were introduced. For cruise navigation, the navigation system design uses angular observation of planet chords combined with star-Sun relative Doppler shift to obtain the spacecraft position and velocity. The main selected bodies are Jupiter, the Sun and Earth with the option to consider other planets depending on their illumination conditions. The spacecraft radial velocity is calculated using the optical Doppler frequency shift measurement from the Sun and combined with planet chord length angles, planet star and Sun-star angles to determine position. The latter measurements are processed in an unscented Kalman filter whose results are used for timely state updates for the particle filter running in parallel. The Kalman filter ensures a robust estimation for mismodelled dynamic environments (e.g. vicinity of asteroids) such as an unprobed gravity field. For the purpose of relative navigation, two optical cameras and 3D LIDAR are used for feature tracking, optical navigation and independent height and vertical speed reconstruction during descent.

Based on the data obtained by the navigation sensor suite, the multi-sensor fusion subsystem provides all necessary information for cognitive autonomous decision-making. The data is obtained by a particle filter-based SLAM approach with a combination of a landmark-based map with a belief-function-based grid-map. The spacecraft dynamic state and the corresponding maps of the asteroid are estimated with a level of detail corresponding to the respective mission phase. This approach is applicable in every exploration scenario where an autonomous agent has to estimate its own position in an unknown environment and map it at the same time. Furthermore, the uncertainties encoded in the map enable an autonomous system to take cognitive decisions.

The challenge of finding the right interplanetary trajectory is solved using optimal control methods from the mathematical field. In KaNaRiA, the implemented approach allows a maximum of three thrust commands, one at the beginning, one at the end and one at an optimized time point in between. A weighting factor allows a customized fit between time- and energy-optimization. Using the optimal nominal solution as baseline, a parametric sensitivity analysis towards perturbations will be performed. Based on the parametric sensitivity analysis and according to the need for optimality, robustness and calculation time at hand, three real-time capable optimal control methods will be implemented: a method for model-predictive control (MPC), a method for repeated adjustment and an optimal feedback controller. Additionally, the approach of modelling the spacecraft motion will be applied to the task of navigation on the asteroid's surface to investigate an adaptive autonomous consideration of state-space constraints.

Analysing and evaluating the data obtained by navigation, fusion and guidance as well as other information available, the autonomy module assesses the current state of the spacecraft. The module acts as central component for autonomous reasoning and decision-making. The situation assessment is used as input for the decision on the feasibility of applicable mission objectives. Mission objectives are broken down into a sequence of actions, which are used to generate a plan. Due to a dynamic environment, the objectives could become unachievable depending on the spacecraft or environment state. With a changing environment, periodic requests of plan re-evaluations are necessary to either ensure commitment or reconsideration. The autonomy module also takes into account the uncertainty of the obtained knowledge using biologically inspired principles such as information maximisation and active perception. Finally, the execution of the plan is based on the trajectory optimization of the guidance subsystem. Using given time and control points, the actuators can be commanded. Based on the known attitude, a sensor control plan can be generated to specify their de-/activation schedule. Last but not least, erroneous states are inferred from the current knowledge of the spacecraft and world state utilizing anomaly detection algorithms for FDIR. All in all, the methods and algorithms developed in this project can be used to enhance the level of autonomy of future space missions with regards to navigation, plan generation, action selection and FDIR. The system provides the ability to represent uncertainty and incorporate this knowledge into the plan generation step. It can modify existing plans to include utility objectives aiming on reducing uncertainty and therefore enhances the robustness of the system with respect to unexpected situations.

The developed autonomy and navigation methods and algorithms are tested and verified in the KaNaRiA virtual simulator (KVS) using the mission scenario of asteroid mining as application. The KVS uses our novel concurrency control management approach with wait-free data exchange between various software components. A centralized data storage called KVPool is used, which resolves the many-to-many interface problem typically encountered in traditional VR architectures. This wait-free approach outperformed standard approaches in terms of access time as shown in the Fig. 8. The above software infrastructure can also be applied in other data-driven simulation domains. Currently, we are experimenting on a new approach for generating gravity field of asteroid shape models, which is based on sphere packing method [35]. This approach considers variable densities and overcomes the gravity field divergence problem in the Brillouin sphere region (see Fig. 9). However, at the same time our method also focuses on a fast computation of gravity potential and acceleration, and on fast generation of pre-processed data used for computing gravity fields.

The KaNaRiA project had its project kick-off in October 2013 and is designated for a period of four years.

ACKNOWLEDGMENTS

This work has been carried out in the frame of the project KaNaRiA. KaNaRiA is a project financed by the German Ministry of Economy and Energy through the German Aerospace Centre, Space Administration (DLR, Deutsches Zentrum für Luft- und Raumfahrt, FKZ 50NA1318 and 50NA1319). It is a collaboration of the University of Bremen and the Bundeswehr University Munich. The involved institutes as well as their contribution in the paper and the respective point of contacts are listed below:

Section II. Alena Probst, ISTA, Bundeswehr University Munich

Section III. Graciela González Peytaví, ISTA, Bundeswehr University Munich

Section IV. David Nakath, Cognitive Neuroinformatics, University of Bremen

Section V. Anne Schattel, Optimization and Optimal Control, University of Bremen

Section VI. Carsten Rachuy, Cognitive Neuroinformatics, University of Bremen

Section VII. Patrick Lange, Computer Graphics and Virtual Reality, University of Bremen

REFERENCES

[1] R.C. Moeller et al., “Space Mission Trade Space Generation and Assessment using the JPL Rapid Mission Architecture (RMA) Team Approach”, in *IEEE Aerospace Conf.*, 2011.

[2] A. Probst et al., “Reference Mission Scenario Selection for Main Belt Asteroid Mining Missions”, in *Planetary and Terrestrial Mining Science Symp. (PTMSS)*, CIM 2015 Conv., May 10-13, Montréal, Quebec, 2015.

[3] A. Probst et al., “Mission Concept Selection for an Asteroid Mining Mission”, in *Aircraft Engineering and Aerospace Technology*, Vol. 88(3), 2015.

[4] A. Probst et al, “Mission Concept and Preliminary Spacecraft Design for Resource Mining on Main Belt Asteroids,” in *International Astronautical Congress, IAC-15, C1, 7, 4. Proceedings of IAC 2015*, Jerusalem, Israel, Oct. 2015.

[5] G. González Peytaví et al, “Autonomous Orbit Navigation for a Mission to the Asteroid Main Belt,” in *International Astronautical Congress, IAC-15, C1, 7, 4. Proceedings of IAC 2015*, Jerusalem, Israel, Oct. 2015.

[6] Y. Guo, “Self-contained autonomous navigation system for deep space missions”, in *Spaceflight Mechanics*, 1999, pp. 1099-1113.

[7] J. Yim et al., “Autonomous orbit navigation of interplanetary spacecraft”, in *AIAA Astrodynamics Specialist Conf.*, Reston, Virginia, 2000.

[8][IV.4]D. Nakath et al., “Active Sensorimotor Object Recognition in Three-Dimensional Space”, in *Spatial Cognition IX,* Springer International Publishing, 2014, pp. 312-324.

[9] K. Schill et al., “Scene analysis with saccadic eye movements: top-down and bottom-up modelling”, in *Journal of Electronic Imaging*, Vol. 10(1), 2001, pp. 152-160.

[10] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I”, in *IEEE Robotics & Automation Magazine*, 13(2), 2006, 99-110.

[11] A. Doucet et al. “Rao-Blackwellised particle filtering for dynamic Bayesian networks”, in *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence (UAI)*, Morgan Kaufmann Publishers Inc., 2000.

[12] M. Montemerlo and S. Thrun, “FastSLAM 2.0. FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem”, in *Robotics*, 2007, pp. 63-90.

[13] T. Reineking et al., “Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification”, in *16th Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, 2015 (in press)

[14] G. Shafer, “A mathematical theory of evidence”, in *Princeton: Princeton university press*, Vol. 1, 1976.

[15] P. Smets and R. Kennes, “The transferable belief model”, in *Artificial Intelligence*, Vol. 66(2), 1994, pp. 191-234.

[16] D. Pagac et al., “An evidential approach to map-building for autonomous vehicles”, in *IEEE Robotics and Automation*, Transactions on 14.4, 1998, pp. 623-629.

[17] M. Ribo and A. Pinz, “A comparison of three uncertainty calculi for building sonar-based occupancy grids”. in *Robotics and Autonomous Systems*, Vol. 35.3 , 2001, pp. 201-209.

[18] J. Mullane et al., “Evidential versus Bayesian estimation for radar map building”, in *9th IEEE Int. Conf. on Control, Automation, Robotics and Vision (ICARCV'06),* 2006.

[19] D. Hahnel et al., “An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements”, in *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Vol. 1., 2003.

[20] T. Reineking, T. and J. Clemens, “Evidential FastSLAM for grid mapping”, in *IEEE 16th Int. Conf. on Information Fusion (FUSION)*, 2013, pp. 789-796.

[21] J. Clemens and T. Reineking, “Multi-Sensor Fusion Using Evidential SLAM for Navigating a Probe through Deep Ice”, in *Belief Functions: Theory and Applications*, Springer International Publishing, 2014, pp. 339-347.

[22] H. Niedermeier et al., "Navigation system for a research ice probe for antarctic glaciers", in *IEEE / ION Position, Location and Navigation Symp. (PLANS)*, May 2014, pp. 959-975.

[23] T. Reineking and J. Clemens, "Dimensions of Uncertainty in Evidential Grid Maps", in *Spatial Cognition IX*, Springer International Publishing, 2014, pp. 283-298.

[24] R. Bulirsch, "Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen, Technical Report of Carl-Cranz-Gesellschaft e.V., Oberpfaffenhofen, 1971", reprint: *Department of Mathematics, Munich University of Technology*, Germany, 1993.

[25] P. Deuflhard, "A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with applications to multiple shooting", in *Numerische Mathematik*, Vol. 22, 1974, pp. 289-315.

[26] Y. Ho and A.E. Bryson, "Applied Optimal Control Optimization, Estimation and Control", in *Holsted Press Book*, 1975.

[27] A. Miele, "Gradient algorithms for the optimization of dynamic systems", in *C.T. Leondes 17*, 1980, pp. 1-52.

[28] C. Büskens, "Direkte Optimierungsmethoden zur numerischen Berechnung optimaler Steuerprozesse", Diploma Thesis, University of Münster, 1993.

[29] C. Büskens and H. Maurer, "Real-Time Control of an Industrial Robot Using Nonlinear Proramming Methods", in *Proc. of the 4th IFAC Workshop on Algorithms and Architectures*, Vilamoura (Portugal), 1997.

[30] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis", in *Springer-Verlag*, New York, 1980.

[31] C. Büskens and D. Wassel, "The ESA NLP Solver WORHP", in *Modeling and Optimization in Space Engineering*, J. D. Pintér (Hrsg.), Springer Optimization and Its Applications, Springer Verlag, Vol. 73, 2013.

[32] P. Lange et al., "Multi Agent System Optimization in Virtual Vehicle Testbeds", in *8th EAI Int. Conf. on Simulation Tools and Techniques (SIMUtools)*, Athens, Greece, 2015.

[33] M. Cohrs et al. "A Methodology for Interactive Spatial Visualization of Automotive Function Architectures for Development and Maintenance", in *Int. Symp. on Visual Computing (ISVC)*, Crete, Greece, 2013.

[34] A. Gomes de Sa and G. Zachmann, "Virtual Reality as a tool for Verification of Assembly and Maintenance Processes", in *Journal of Computer graphics*, 1999, pp. 389-403.

[35] P. Lange et al., "Virtual Reality for Simulating Autonomous Deep-Space Navigation and Mining", in *24th Int. Conf. on Artificial Reality and Telexistence (ICAT-EGVE),* Bremen, Germany, 2014.

[36] J. Balaram et al.: "DSENDS – A high-fidelity dynamics and spacecraft simulator for entry, descent and surface landing", in *IEEE Aerospace Conf.*, Montana, USA, 2002, pp. 7-3342 – 7-3359.

[37] P. Lange et al., "A Framework for Wait-Free Data Exchange in Massively Threaded VR Systems", in *Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG),* Plzen, Czech Republic, 2014.

[38] P. Lange et al., "Scalable Concurrency Control for Massively Collaborative Virtual Environments", in *ACM Multimedia Systems: Massively Multiuser Virtual Environments (MMVE)*, Portland, USA, 2015.

[39] M. Brillouin, "Equations aux Dériveées partielles du 2e ordre. Domaines à connexion multiple. Fonctions sphériques non antipodes", in *Annales De L 'Institut H. Poincaré*, Vol. 2, 1933, pp. 173–206.

[40] R. A. Werner, "Evaluating Descent and Ascent Trajectories Near Non-Spherical Bodies"*, Tech. Report*, Jet Propulsion Laboratory (JPL), 2010. Available: http://www.techbriefs.com/component/content/article/8726.

[41] Y. Takahashi et al., "Surface Gravity Fields for Asteroids and Comets", in *Journal of Guidance, Control, and Dynamics*, Vol. 36 (2), 2013, pp. 362-374.

[42] R. Weller and G. Zachmann, "ProtoSphere: A GPU-Assisted Prototype-Guided Sphere Packing Algorithm for Arbitrary Objects"*,* in *ACM SIGGRAPH Asia 2010 Sketches*, New York, USA, 2011, pp. 8:1 – 8:2.

[43] R. Weller et al., "Massively Parallel Batch Neural Gas for Bounding Volume Hierarchy Construction", in *Virtual Reality Interactions and Physical Simulations (VRIPhys)*, Bremen, Germany, 2014.

IAC-15-C1.7.4

AUTONOMOUS ORBIT NAVIGATION FOR A MISSION TO THE ASTEROID MAIN BELT

Graciela González Peytaví[*‡], Joachim Clemens[†§], David Nakath[†], Alena Probst[*],
Roger Förstner[*], Kerstin Schill[†], Bernd Eissfeller[*]
[*]Institute of Space Technology & Applications, Bundeswehr University Munich, Germany
[†]Cognitive Neuroinformatics, University of Bremen, Germany
[‡]graciela.gonzalez@unibw.de , [§]clemens@uni-bremen.de

This paper introduces an autonomous navigation concept based on optical observations of the Sun and selected planetary bodies. The concept is self-contained and independent from the mission ground segment. It enables autonomous navigation of spacecraft anywhere in the solar system, including spacecraft flying through the Asteroid Main Belt (AMB). Initially, the novel navigation concept is introduced within its historical and technological context. The sensor technology involved, the navigation observables and their accuracy is discussed. Subsequently, the navigation performance - i.e. position and velocity errors - is analysed for two selected orbit scenarios: an Earth-like orbit around the Sun and an orbit within the AMB at 2.8 AU from the Sun. For each test scenario two estimators are used: an Unscented Kalman Filter (UKF) accounting for third body gravitational perturbations and a Particle Filter (PF). The results are evaluated in terms of the standard deviations of the available measurements, the frequency of the observations and the number of observations simultaneously participating in the sensor fusion process. The optical navigation concept introduced here makes use of a coupled star-Sun tracker to determine direction to the Sun, optical Doppler measurements from a resonance-scattering interferometer to determine the spacecraft radial velocity relative to the Sun and optical cameras to determine the line-of-sight vector towards planetary bodies. The problem of orbit determination is solved at first for an orbital distance of 1 AU from the Sun. This initial test evaluates the benefits of the additional planetary line-of-sight observations as compared to previous concepts based on observations of the Sun only. In a second step, the two implementations based on the UKF and PF are compared with the purpose of evaluating the sensor fusion capability of the overall system. The UKF and the PF solutions are assessed in terms of accuracy of the positioning solution, convergence efficiency and computational requirements. Finally, the performance of the proposed concept is evaluated for the KaNaRiA mission scenario. KaNaRiA is a feasibility study of an asteroid mining mission targeting asteroids in the main belt. Part of the KaNaRiA mission architecture consists of several prospecting spacecraft which will fly autonomously to different target asteroids. In the context of the KaNaRiA mission analysis, a parking orbit at 2.8 AU from the Sun has been selected. The orbit is characterized by slow varying geometry relative to the Sun and good observability opportunities of the near planet Jupiter.

I. INTRODUCTION

The possibility to autonomously navigate a spacecraft throughout the solar system with reduced ground support has been long discussed since the 90s. At the time, NASA's New Millennium Program envisioned a call for multiple, small, dedicated spacecraft operating autonomously over extended periods of time in order to maximise the scientific return of deep-space exploration missions during the 2000s and beyond. Consequently a large amount of first-time concepts for autonomous cruise navigation were proposed.

Folta et al. [1] suggested the usage of standard attitude sensors in combination with one-way Doppler measurements from ground stations to perform on-board navigation. They claimed an achievable positioning accuracy of 100 meter in High Earth Orbit (HEO) and 10 km in Libration Point Orbit (LPO).

Guo [2] introduced the idea of observing the optical Doppler shift due to the motion of the spacecraft relative to the Sun and the Sun direction for orbit determination. Guo [3] even conceived a coupled star-Sun tracker for this purpose and patented the method. Guo et al. [4] studied the performance of the solar navigation approach in the scenario the Solar TErrestrial RElations Observatory (STEREO) mission which led to a claim of 200 km precision at a distance of 1 AU from the Sun. Although the concept was never tested in space, other authors continued to study its viability. Gipsman et al. [5] combined the coupled star-Sun tracker with a set of inertial accelerometers to provide autonomous navigation and guidance for low thrust driven missions. Yim et al. [6] evaluated the performance of an Extended Kalman Filter (EKF) using optical Sun Doppler observables next to Earth line-of-sight (LOS) from a standard attitude sensor. Their simulations led to few km accuracy in positioning and velocity estimates below the mm/s range.

Other proposals made use of optical imaging instruments for celestial navigation. Riedel et al. [7] designed the first fully autonomous deep-space navigation system for NASA/JPL's Deep Space -1 (DS-1). Based on the optical navigation heritage used during fly-bys of the Voyager and Galileo missions, the first on-board autonomous navigation software, AutoNav, was built-in for DS-1. The system used as data type the star-relative astrometric observations of planetary bodies including beacon asteroids (i.e. star-planet angles) and proved its performance during flight, although it was never used as primary navigation system [8]. After the success of DS-1, AutoNav was further developed for rendezvous and landing missions to Mars and small planetary bodies [9]. However it was never used again during cruise.

Similar solutions using Charge-Coupled Device (CCD) cameras and optical flow techniques for image processing were simultaneously developed by other authors [10]. ESA's Technological Research Program on Autonomous Navigation for Interplanetary Missions validated – using real images from the Moon orbiter Smart-1 – an optical navigation concept using beacon asteroids for which precise ephemeris were available [11]. The ObNav software was developed in this framework and its validation lead to the conclusion that 500 km positioning during cruise could be achieved using optical images only [12]. ObNav was foreseen not only for cruise navigation, but also for aerocapture and rendezvous.

The usage of the AutoNav [9] and ObNav heritage packages has been limited to encounter, rendezvous and kinetic impactor operations since then. The need for an autonomous navigation system during cruise lost part of its pull thanks to the rapid and successful development of the ground tracking network and the reduced number of missions demanding simultaneous tracking. Many historical milestones – as the Rosetta rendezvous to comet 67-P [13] – were achieved by means of optical navigation. However, as of today, autonomous cruise navigation has not been a requirement for a handful of simultaneous exploration missions lead by public agencies and institutions.

In the late 2000s the discussion on commercial access to space and commercial space infrastructure has been ignited. Concepts for in-orbit servicing and space resource extraction have been proposed and several companies are developing commercial solutions and services. All these concepts have one key design element in common which aligns with the original ideas of public space exploration programs 30 years back: multiple, small, dedicated spacecraft operating autonomously. The appearance of such commercial initiatives have brought back the importance of developing autonomous spacecraft navigation systems for cruise.

Amongst one of the newly reinvestigated navigation concepts, x-ray pulsar navigation [14] has captured particular attention during the last few years. The NICER/SEXTANT experiment will be placed aboard the International Space Station in 2016 to study and catalogue x-ray timing sources [15]. However, the need for miniaturization of x-ray optics, the development of high precision x-ray detectors and ultra-stable space clocks will delay the integration of a pulsar-based navigation system for spacecraft operations.

This paper takes up the issue of developing a compact, technologically practicable, autonomous cruise navigation system for deep space missions which cannot or can only minimally be supported by the Deep Space Network (DSN). We build upon the solar navigation concept proposed by Guo [2] and combine it with state-of-the-art optical celestial navigation techniques for astrometric observations of planetary bodies. The new navigation concept is described in section II. The dynamic equations of motion and the observation models used in the evaluation of the proposed method are provided in section III. Two different estimators, an Unscented Kalman Filter (UKF) and a Particle Filter (PF) are introduced in section IV and V respectively. These estimators are evaluated for orbit determination in different heliocentric orbit scenarios. A comparison of the performance and system configuration with respect to previous work in this area is presented in section VI. This section additionally comprises an assessment of the system's navigation performance for the KaNaRiA asteroid mining mission scenario. Finally, conclusions are listed in section VII.

II. OPTICAL SUN DOPPLER NAVIGATION

A new type of autonomous navigation for deep space missions is presented here. It combines the solar navigation method introduced by Guo [2] with optical astrometric observations of planetary bodies. The proposed approach uses the Sun as reference body. Three observation types are handled: the direction of the spacecraft relative to the Sun, the optical Doppler shift due to the motion of the spacecraft relative to the Sun and the direction of the spacecraft relative to planetary bodies in our solar system.

II.I Solar directional observations

Digital images from an on-board CCD camera have been long used in space missions for directional data collection. The target body is imaged against a star background which serves as direction reference for the camera boresight orientation. Angular resolutions of 1/10 of the size of a pixel are state-of-the-art.

The Sun's huge brightness in contrast to far stars makes it difficult to form a composite image. Guo [3] conceived a dual-mode imaging system which accounted for a sunlight attenuator and a special mirror to combine light from both, the Sun and background stars. A similar system, denoted Coupled Sun Star Tracker (CSST), was used by Gipsman et al. [5] for a combined autonomous navigation and guidance system.

II.II Optical solar Doppler observations

Passive Doppler shift measurements have been widely applied in astronomy and solar physics research.

Selected absorption lines of the Sun spectrum can be tracked to measure a spacecraft's relative velocity. In the optical band, the large Doppler shift – about 1.6 KHz for a radial velocity of 1 mm/s – allows for instantaneous measurement. It is known that the visible, near ultraviolet and infrared spectral regions, where the flux of Sun's radiation spectra peaks, show the lowest irradiance variations, as compared to the ultraviolet, X-ray and radio portions of the electromagnetic spectrum. The total solar irradiance shows peak-to-peak amplitude variations of order 0.1% due to the 11-year cycle and 0.2% associated with the 27-day Sun rotation cycle. On shorter time scales, five-minute fluctuations of about 0.003% can be expected during calm periods rising as much as to 0.015% during solar flares [16].

Either electro-optical modulators or resonance scattering interferometers can be used to observe the optical solar Doppler shift. The latter known to provide higher accuracy observations up to 1 cm/s [2]. The Michelson Doppler Imager (MDI) aboard the European Solar and Heliospheric Observatory (SOHO) mission is a legacy example of such an interferometer which provided solar Doppler shift measurements based on four filtergram difference ratios with an accuracy of 20 m/s per pixel using technology from the 80s [17].

II.III Astrometric observations of planetary bodies

Optical navigation – traditionally also known as celestial navigation – makes use of digital images of beacon planetary objects, for which ephemeris is available, against a background of catalogued stars. By centroiding the planetary object and determining the astrometric – i.e. angles on the image plane – between the beacon object and the selected stars, the inertial pointing of the camera boresight relative to the LOS of the object can be determined.

In the scope of this work, solar system planets are used as beacon objects. In particular, the largest observable planet is selected for observation at each time. However it is foreseen to introduce asteroid LOS measurements for the KaNaRiA autonomous navigation system during far-approach and rendezvous.

III. NAVIGATION SIMULATOR OVERVIEW

The performance of the optical Sun Doppler navigation concept presented in section II has been evaluated by means of a software-based navigation simulator. Within this section we describe the dynamic equations of motion and observation models adopted for this purpose. A total of three observation types is described: planetary chords, the spacecraft radial velocity derived from solar Doppler measurements and astrometric star-planet and star-Sun angles.

III.I Equations of Motion

The motion of an interplanetary spacecraft in orbit around the Sun is considered for the work presented in this paper. In a first approach, a simplified perturbed two-body problem with the Sun as central body and third-body perturbations is assumed. The three-dimensional acceleration $\ddot{\boldsymbol{r}}$ imposed on the spacecraft centre of mass can be expressed by means of Newtonian dynamics as

$$\ddot{\boldsymbol{r}} = \frac{\mu_s}{\|\boldsymbol{r}\|^3}\boldsymbol{r} + \sum_{p=1}^{9} \frac{\mu_p}{\|\boldsymbol{r}_p - \boldsymbol{r}\|^3} (\boldsymbol{r}_p - \boldsymbol{r}) \qquad [1]$$

where μ_s is the Sun gravitational constant, μ_p is the gravitational constant of the *p-th* planet in our solar system, and $\boldsymbol{r} = [r_x\ r_y\ r_z]^T$ and $\boldsymbol{r}_p = [r_{x,p}\ r_{y,p}\ r_{z,p}]^T$ are the spacecraft position vector and the *p-th* planet position vector respectively in heliocentric coordinate system with the ecliptic as reference plane.

The position of the spacecraft at any epoch can be computed by numerical integration of Eq. 1. A multi-step Runge-Kutta integrator of order 4 and degree 5 is used in the simulation as well as for state prediction in the UKF. For performance reasons, the PF uses a single-step integrator with a fixed step size which is described in detail in the motion model part of section V.

III.II Observation Modelling

The autonomous orbit navigation concept presented in this paper makes use of LOS observations to celestial bodies – i.e., the Sun and solar system planetary bodies – and relative radial motion with respect to the central body which, for interplanetary transfer, is always the Sun. LOS vectors to the Sun and planets are obtained from stellar angular observations – i.e., star-planet and star-Sun angles – and angular chord length observations. The relative radial velocity of the spacecraft can be derived from the optical Doppler frequency shift observed with a resonance scattering interferometer as described in section II.

In the following we present the measurement modelling equations used in the scope of this study.

Planetary chords

The radial angular chord length γ_p of an optically observed planet p can be geometrically reconstructed from its geometrical radius R_p, the heliocentric position of the sensor reference point[*] $\boldsymbol{r}$ at observation time and

[*] For the scope of this study we assume the sensor reference point to be located at the spacecraft centre of mass.

the heliocentric position of the planet centre of mass $\boldsymbol{r}_p$ at the measurement epoch.

$$\tan\gamma_p = \frac{R_p}{\|\boldsymbol{r}_p - \boldsymbol{r}\|} \qquad [2]$$

The values of planetary radius introduced in Eq. 2 for simulation purposes are the equivalent noisy magnitude of mean and standard deviation listed in Table 1.

For each observation epoch, only one planet in the solar system is observed. In order to select the best observable object, planetary chords are computed for all nine planets according to Eq. 2. The chords are then sorted according to magnitude and only the largest observable object is selected for observation.

Spacecraft radial velocity

The relative velocity of a spacecraft orbiting the Sun during interplanetary transfer can be derived from the frequency shift of the solar optical spectra as observed by an on-board resonance scattering interferometer.

For the scope of this work, the spacecraft instantaneous radial velocity v_r can be modelled as the projection of the spacecraft total velocity $\dot{\boldsymbol{r}}$ on the LOS towards the Sun, which is coincident with the spacecraft heliocentric position vector $\boldsymbol{r}$.

$$v_r = \frac{\boldsymbol{r}\cdot\dot{\boldsymbol{r}}}{\|\boldsymbol{r}\|} \qquad [3]$$

Star-planet and star-Sun angles

As a source of star declination and right-ascension values, the Hipparcos star catalogue is used. We consider stars of magnitude 7 or brighter. For each observation epoch, LOS unit vectors $\hat{\boldsymbol{u}}_s$ are reconstructed for each potentially observed star s in the catalogue:

$$\hat{\boldsymbol{u}}_s = \begin{bmatrix} \cos\Omega_s \cos\gamma_s \\ \sin\Omega_s \cos\gamma_s \\ \sin\gamma_s \end{bmatrix}$$

where Ω_s and γ_s are the astronomical right-ascension and declination obtained from the star catalogue.

For star-planet angular measurements, the projection $\hat{\boldsymbol{u}}_s^{\perp} = \hat{\boldsymbol{u}}_s \cdot \hat{\boldsymbol{u}}_p$ of the star s unit vector $\hat{\boldsymbol{u}}_s$ onto the planet LOS direction – i.e., defined by $\hat{\boldsymbol{u}}_p$ as in Eq. 4 – is then computed. For star-Sun measurements, the projection $\hat{\boldsymbol{u}}_s^{\perp}$ is computed on to the LOS towards the Sun defined by the spacecraft heliocentric position $\hat{\boldsymbol{u}}_{Sun}$.

$$\hat{\boldsymbol{u}}_p = \frac{\boldsymbol{r}_p - \boldsymbol{r}}{\|\boldsymbol{r}_p - \boldsymbol{r}\|}$$
$$[4]$$
$$\hat{\boldsymbol{u}}_{Sun} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}$$

A total of three stars in the vicinity of the largest observable planet are selected together with three stars in the vicinity of the Sun for each observation epoch. The criterion for selection is that $\cos\gamma_p < \hat{\boldsymbol{u}}_s^{\perp} < \cos\alpha/2$, where γ_p is the body radial chord length – calculated as in Eq. 2 – and α is the instrument field of view. A conic field of view of 20 degrees has been assumed for the coupled star-Sun tracker.

In addition, three polar stars in the vicinity of the planet are selected. Polar stars can be selected by performing the same vector projection $\hat{\boldsymbol{u}}_s^{\perp}$ of the star unit vector $\hat{\boldsymbol{u}}_s$ with a unit polar vector $\hat{\boldsymbol{u}} = [0\ \ 0\ \ 1]^T$.

In total up to 9 stars can be observed per epoch: 3 + 3 polar stars in the vicinity of the largest observable planet and 3 far stars around the Sun.

The star-planet and star-Sun angles β_s are finally obtained from

$$\cos\beta_s = \hat{\boldsymbol{u}}_s^{\perp} = \hat{\boldsymbol{u}}_s \cdot \begin{Bmatrix} \hat{\boldsymbol{u}}_p \\ \hat{\boldsymbol{u}}_{Sun} \end{Bmatrix}$$

IV. THE UNSCENTED KALMAN FILTER (UKF)

The Unscented Kalman Filter (UKF) is a recursive Minimum Mean Square Error (MMSE) estimator belonging to a family of so-called Sigma-Point Kalman Filters (SPKF) introduced in the late 90s [18]. In opposition to the Extended Kalman Filter (EKF), the UKF does not introduce large errors due to the truncation of higher-terms of the Taylor series expansion appearing from the linearization of the underlying nonlinear system and observations equations. Instead, the UKF attempts to represent the true state distribution – still characterized by a Gaussian random variable – using a minimal set of deterministically chosen sample points: the so-called sigma-points [19].

IV.I The Scaled Unscented Transformation (SUT)

The Scaled Unscented Transformation (SUT) is a method for calculating the statistics of a variable undergoing a nonlinear transformation. Instead of attempting to approximate the nonlinear function $\boldsymbol{y} = g(\boldsymbol{x})$, the SUT directly approximates the statistics of the L-dimensional random variable $\boldsymbol{y}$ based on $2L+1$ weighted samples denoted sigma-points $S_i = \{\omega_i, \boldsymbol{\chi}_i\}$. Assuming that the prior variable $\boldsymbol{x}$ has mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{P}_x$ the sigma-point set is defined as in Eq. 5. The *i-th* column (or row) of the matrix $\sqrt{(L+\lambda)\boldsymbol{P}_x}$ is denoted as $\left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_i$.

$$\begin{aligned} \boldsymbol{\chi}_0 &= \bar{\boldsymbol{x}} \\ \boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} + \left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_i \qquad i = 1,\dots,L \qquad [5] \\ \boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} - \left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_i \quad i = L+1,\dots,2L \end{aligned}$$

Sigma-points are assigned weights $\omega_i^{(m)}$ and $\omega_i^{(c)}$ (see Eq. 6). The superscripts m and c stand for *mean* and *covariance* respectively. The set of weights satisfies $\sum_{i=0}^{2L} \omega_i = 1$. The scaling parameter λ is obtained as $\lambda = \alpha^2(L+\kappa) - L$.

$$\begin{aligned} \omega_0^{(m)} &= \tfrac{\lambda}{L+\lambda} && i = 0 \\ \omega_0^{(c)} &= \tfrac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) && i = 0 \qquad [6] \\ \omega_i^{(m)} &= \omega_i^{(c)} = \tfrac{1}{2(L+\lambda)} && i = 1, \dots, 2L \end{aligned}$$

The filter tuning parameters κ, α and β can be adjusted to minimize the errors in the fourth and higher order terms of the approximation of the prior distribution [20]. The parameter α tunes the size of the sigma-point distribution and should remain small in order to avoid sampling nonlinearities of the prior distribution. Both β and κ are positively defined. The parameter β can be used to incorporate knowledge of the higher order moments – i.e., skew or kurtosis – of the prior, while κ is tuned to guarantee positive definiteness of the covariance matrix. For the purpose of this paper, the parameters have been set to $\alpha = 0.8 \cdot 10^{-3}$, $\beta = 2$ and $\kappa = 0$.

Once the tuning parameters are selected and the sigma-points S_i are calculated, each sigma-point can be propagated through the nonlinear transformation $g(\cdot)$.

$$\boldsymbol{\Upsilon}_i = g(\boldsymbol{\chi}_i) \qquad i = 0, \dots, 2L$$

The mean $\bar{\boldsymbol{y}}$, covariance $\boldsymbol{P}_y$ and cross-covariance $\boldsymbol{P}_{xy}$ are computed as follows:

$$\bar{\boldsymbol{y}} \approx \sum_{i=0}^{2L} \omega_i^{(m)} \boldsymbol{\Upsilon}_i$$

$$\boldsymbol{P}_y \approx \sum_{i=0}^{2L} \omega_i^{(c)} (\boldsymbol{\Upsilon}_i - \bar{\boldsymbol{y}})(\boldsymbol{\Upsilon}_i - \bar{\boldsymbol{y}})^T$$

$$\boldsymbol{P}_{xy} \approx \sum_{i=0}^{2L} \omega_i^{(c)} (\boldsymbol{\chi}_i - \bar{\boldsymbol{x}})(\boldsymbol{\Upsilon}_i - \bar{\boldsymbol{y}})^T$$

IV.II UKF Initialization

The UKF algorithm updates the mean $\bar{\boldsymbol{x}}_k$ and covariance $\boldsymbol{P}_{x,k}$ of the posterior distribution at time step k considering a Gaussian approximation. For a more compact expression, we may define the state random variable as the concatenation of the original state $\boldsymbol{x}_k$ together with the process and observation noise random variables, $\boldsymbol{v}_k$ and $\boldsymbol{n}_k$ respectively. The concatenated new state is the augmented state $\boldsymbol{x}_k^a = [\boldsymbol{x}_k \ \boldsymbol{v}_k \ \boldsymbol{n}_k]^T$. Similarly, the augmented covariance matrix $\boldsymbol{P}^a$ is built by concatenation of the state covariance $\boldsymbol{P}_x$, the observation noise covariance $\boldsymbol{R}_v$ and the process noise covariance $\boldsymbol{R}_n$.

$$\boldsymbol{P}^a = \begin{bmatrix} \boldsymbol{P}_x & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_v & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_n \end{bmatrix}$$

The filter is initialized such that the state covariance $\boldsymbol{P}_{x,0} = diag\{\sigma_{\boldsymbol{r}}\, \sigma_{\boldsymbol{r}}\, \sigma_{\boldsymbol{r}}\, \sigma_{\dot{\boldsymbol{r}}}\, \sigma_{\dot{\boldsymbol{r}}}\, \sigma_{\dot{\boldsymbol{r}}}\}$ with $\sigma_{\boldsymbol{r}} = 1\, km$ and $\sigma_{\dot{\boldsymbol{r}}} = 1\, cm/s$. The initial state $\hat{\boldsymbol{x}}_0^a = E[\hat{\boldsymbol{x}}_0 \quad 0 \quad 0]^T$ and covariance $\boldsymbol{P}_0^a$ can be defined with $\boldsymbol{R}_v = \boldsymbol{R}_n = \boldsymbol{0}$.

The measurement noise covariance $\boldsymbol{R}_v$ contains the *p-th* planet chord length noise STandard Deviation (STD) σ_{γ_p}, the radial velocity noise STD σ_{v_r}, and the angular measurement noise STD σ_s of the s star-based observables: $\boldsymbol{R}_v = diag\ \{\sigma_{\gamma_p}^2\, \sigma_{v_r}^2\, \sigma_{s=1}^2 \dots \sigma_{s=9}^2\}$. Where $\sigma_{v_r} = 1\, cm/s$ and $\sigma_s = 1.745.10^{-6}\, rad$ according to [6]. The angular chord length measurement of the largest visible planet is computed as $\sigma_{\gamma_p} = \sqrt{n_1^2 + n_2^2}$ with:

$$n_1 = \tan^{-1}\left(\frac{\epsilon \cdot l_{pixel}}{f_L}\right)$$

$$n_2 = \frac{R_p \cdot \sigma_{R_{pt}}}{\cos\left(\sin^{-1}\left(\frac{R_p}{\|\boldsymbol{r} - \boldsymbol{r}_p\|}\right)\right)}$$

A pixel size l_{pixel} of 10 μm was assumed for the coupled star-Sun tracker with a focal length f_L of 43.3 mm and centroiding capacity ϵ of 1/10 of a pixel.

The values for planetary radius R_p and corresponding radius measurement STD σ_{R_p} are listed in Table 1. Only those planets considered for observation have been included.

The spacecraft heliocentric position vector r and the planet position $\boldsymbol{r}_p$ are computed for each measurement epoch. Planetary ephemeris is obtained from the Astronomical Almanac [21].

p	Planet	R_p [km]	σ_{R_p} [km]
2	Venus	6051.9	1
3	Earth	6378	1
4	Mars	3397	1
5	Jupiter	71492	100
6	Saturn	60268	600
7	Uranus	25559	1250
8	Neptune	24767	2250

Table 1: Planetary radius and its measurement STD as used in the generation of the measurement noise covariance matrix.

IV.III UKF Implementation

For each UKF iteration k, three steps are performed: calculation of sigma-points, time update and measurement update.

Calculation of sigma-points

Sigma-points can be obtained by applying the SUT as provided in Eq. 5 to the augmented state $\boldsymbol{x}^a$ as:

$$\boldsymbol{\chi}^a_{k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}^a_{k-1} & \hat{\boldsymbol{x}}^a_{k-1} + \gamma\sqrt{\boldsymbol{P}^a_{k-1}} & \hat{\boldsymbol{x}}^a_{k-1} - \gamma\sqrt{\boldsymbol{P}^a_{k-1}} \end{bmatrix}^T$$

The scaling factor $\gamma = \sqrt{L + \lambda}$ is a composite parameter not to be confused with the planet-chord angular measurements γ_p. The superimposed hat in $\hat{\boldsymbol{x}}^a_k$ denotes estimated variable.

Time update

In the following we drop the superscript a to denote augmented variables for clarity. All variables are augmented unless a superscript indicating the corresponding random variable is included.

Each sigma-point $\boldsymbol{\chi}^x_k$ can be updated using the motion model $f(\cdot)$.

$$\boldsymbol{\chi}^x_{k|k-1} = f(\,\boldsymbol{\chi}^x_{k-1}, \boldsymbol{\chi}^v_{k-1}, \boldsymbol{u}_{k-1}\,)$$

As a weighted combination of sigma points, the updated state $\hat{\boldsymbol{x}}_k$ can be reconstructed. We introduce an apostrophe to mark the updated character of the variable.

$$\hat{\boldsymbol{x}}'_k = \sum_{i=0}^{2L} \omega_i^{(m)} \boldsymbol{\chi}^x_{i,k|k-1}$$

With the updated sigma-points and state, the covariance of the augmented state $\boldsymbol{P}'_{x,k}$ can be updated too. The ω_i terms are the weights as computed in Eq. 6.

$$\boldsymbol{P}'_{x,k} = \sum_{i=0}^{2L} \omega_i^{(c)} \left(\boldsymbol{\chi}^x_{i,k|k-1} - \hat{\boldsymbol{x}}'_k\right)\left(\boldsymbol{\chi}^x_{i,k|k-1} - \hat{\boldsymbol{x}}'_k\right)^T$$

Measurement update

The sigma-points $\boldsymbol{\Upsilon}_k$ representing the distribution of the measurements are derived from the observation equations h using as input the updated sigma-points. The updated observations $\hat{\boldsymbol{y}}'_k$ are reconstructed as weighted sum of sigma-points $\boldsymbol{\Upsilon}_k$.

$$\boldsymbol{\Upsilon}_{k|k-1} = h(\boldsymbol{\chi}^x_{k|k-1}, \boldsymbol{\chi}^n_{k-1})$$

$$\hat{\boldsymbol{y}}'_k = \sum_{i=0}^{2L} \omega_i^{(m)}\, \boldsymbol{\Upsilon}_{i,k|k-1}$$

The covariance $\boldsymbol{P'}_{y,k}$ and cross-covariance $\boldsymbol{P'}_{xy,k}$ are calculated as weighted-sum of the actual deviations.

$$\boldsymbol{P'}_{y,k} = \sum_{i=0}^{2L} \omega_i^{(c)} \left(\boldsymbol{\Upsilon}_{i,k|k-1} - \hat{\boldsymbol{y}}'_k\right)\left(\boldsymbol{\Upsilon}_{i,k|k-1} - \hat{\boldsymbol{y}}'_k\right)^T$$

$$\boldsymbol{P'}_{xy,k} = \sum_{i=0}^{2L} \omega_i^{(c)} \left(\boldsymbol{\chi}^x_{i,k|k-1} - \hat{\boldsymbol{x}}'_k\right)\left(\boldsymbol{\Upsilon}_{i,k|k-1} - \hat{\boldsymbol{y}}'_k\right)^T$$

Finally, the state $\hat{\boldsymbol{x}}_k$ and the covariance of the posterior $\boldsymbol{P}_{x,k}$ are corrected by means of the Kalman gain $\boldsymbol{K}_k = \boldsymbol{P}_{xy,k}\boldsymbol{P}^{-1}_{y,k}$.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}'_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}'_k)$$

$$\boldsymbol{P}_{x,k} = \boldsymbol{P}'_{x,k} - \boldsymbol{K}_k\boldsymbol{P'}_{y,k}\boldsymbol{K}^T_k$$

V. THE PARTICLE FILTER (PF)

The PF [22] [23] approximates the target distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k}, u_{1:k})$[†] over the current state $\boldsymbol{x}_k$ given all measurements $\boldsymbol{z}_{0:k}$ (here, planet chord length, star-planet angles and star-Sun angles) and state transition measurements $u_{1:k}$ (here, Doppler measurements) using a finite set of random particles $\mathcal{S}_k$. The particle set consists of M state hypotheses $\boldsymbol{x}^{[m]}_k$, with $1 \leq m \leq M$, and is updated recursively over time: In the first step, a new set of particles $\tilde{\mathcal{S}}_k$ is drawn from the proposal distribution

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, u_{1:k}) = p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1}, u_{1:k-1})\; p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_k) \qquad [7]$$

where $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1}, u_{1:k-1})$ is given by the previous particle set $\mathcal{S}_{k-1}$ and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_k)$ is the *motion model*, which depends on the state transition measurement u_k. In a second step, an importance weight $w^{[m]}_k$ is calculated for each particle that is defined as the ratio between the target and proposal distribution and can be computed from the measurement likelihood according to

$$w^{[m]}_k = \frac{target\ distribution}{proposal\ distribution} = \frac{p\left(\boldsymbol{x}^{[m]}_k \middle| \boldsymbol{z}_{0:k}, u_{1:k}\right)}{p\left(\boldsymbol{x}^{[m]}_k \middle| \boldsymbol{z}_{0:k-1}, u_{1:k}\right)} \propto p\left(\boldsymbol{z}_k \middle| \boldsymbol{x}^{[m]}_k\right) \qquad [8]$$

[†] To simplify the notations, we use $a_{0:k}$ as short hand for $a_0, a_1, \dots, a_k$.

where $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ denotes the measurement likelihood provided by a *measurement model*. In a third and final step, the particles are resampled with a probability proportional to their importance weight. The resulting particle set $\mathcal{S}_k$ represents the desired approximation of the target distribution.

The advantage of the PF over KFs is that it makes almost no demands on the target distribution, the motion model, and the measurement models, e.g. regarding linearity or type of distribution. Thus, it can approximate almost any distribution, including non-Gaussian, non-parametric, and multi-modal ones. Furthermore, the motion- and measurement models can be of any form as well and are capable of describing non-linear relations. The only requirement is that one needs to be able to sample from the proposal distribution (see Eq. 7).

On the downside, in general, the PF requires more computational power than a KF, because the prediction and correction has to be performed for each particle. The complexity is $O(M)$, which means that it grows linearly with the number of particles. However, the number of particles required to adequately approximate the target distribution can be reduced with a good proposal distribution (i.e., a good motion model) as well as with a technique called *adaptive resampling*. Both are described in the following along with the measurement models required for the PF. A quantitative evaluation regarding the number of particles is provided in the performance analysis section V.

V.I Motion model

The motion model $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_k)$ is required to calculate the proposal distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k-1}, u_{1:k})$ from the prior distribution $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1}, u_{1:k-1})$ (see Eq. 7). It is used to sample a new state hypothesis $\boldsymbol{x}_k$ given the previous state $\boldsymbol{x}_{k-1}$ and the state transition measurement u_k. As for the UKF, the state is defined as

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{r}_k \\ \dot{\boldsymbol{r}}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \end{bmatrix}$$

where $\boldsymbol{r}_k$ denotes the spacecraft positon vector and $\dot{\boldsymbol{r}}_k$ its first derivation, i.e. the velocity $\boldsymbol{v}_k$. The state transition measurement u_k is provided by the optical Sun Doppler measurements which can be directly translated as the Sun-relative radial velocity $v_{k;r}$.

By assuming conditional independence between orbit dynamics and the Doppler measurement, the proposal distribution can be rewritten as

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_k) = \int p(\tilde{\boldsymbol{x}}_k|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_k|\tilde{\boldsymbol{x}}_k, u_k) d\tilde{\boldsymbol{x}}_k$$

where $p(\tilde{\boldsymbol{x}}_k|\boldsymbol{x}_{k-1})$ is given by the orbit dynamics and $p(\boldsymbol{x}_k|\tilde{\boldsymbol{x}}_k, u_k)$ by the Doppler measurement. This allows to sample from the orbit dynamics first, in order to predict $\boldsymbol{x}_{k-1}$ from $k-1$ to k (denoted by $\tilde{\boldsymbol{x}}_k$), followed by sampling with the Doppler measurement to correct the velocity accordingly.

To sample from $p(\tilde{\boldsymbol{x}}_k|\boldsymbol{x}_{k-1})$, the relation between $\boldsymbol{x}_{k-1}$ and $\tilde{\boldsymbol{x}}_k$ as well as the central acceleration caused by the Sun are taken into account. This results in

$$\tilde{\boldsymbol{x}}_k = \begin{bmatrix} I_{3\times3} & I_{3\times3}\Delta k \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} I_{3\times3}\dfrac{\Delta k^2}{2} \\ I_{3\times3}\Delta k \end{bmatrix} \frac{-\mu_{Sun}}{\|\boldsymbol{r}_{k-1}\|^3} \boldsymbol{r}_{k-1} + \boldsymbol{\varepsilon}_{x_k}$$

where $I_{3\times3}$ is the identity matrix of dimension 3×3, $0_{3\times3}$ the zero matrix of dimension 3×3, Δk the time difference between $k-1$ and k, and μ_{Sun} the standard gravitational parameter of the Sun. Planetary influences (e.g. from Saturn or Jupiter) can simply be included in the second summand of the equation. $\boldsymbol{\varepsilon}_{x_k}$ is additive Gaussian noise with zero mean and covariance Σ_{x_k} that is proportional to Δk. To reduce the approximation error, the prediction is calculated in L steps with $\Delta k' = \Delta k/L$, i.e. $\boldsymbol{x}'_{k-1+\Delta k' l} \sim p(\boldsymbol{x}'_{k-1+\Delta k' l}|\boldsymbol{x}'_{k-1+\Delta k'(l-1)})$, $1 \le l \le L$, $\boldsymbol{x}'_{k-1} = \boldsymbol{x}_{k-1}$, and $\tilde{\boldsymbol{x}}_k = \boldsymbol{x}'_k$. It is also possible to calculate the prediction based on the UKF's motion model (see Equations of Motion in section III). However, this would increase the computational complexity because of the used Runge-Kutta method.

After computing the prediction, it is sampled from $p(\boldsymbol{x}_k|\tilde{\boldsymbol{x}}_k, u_k)$ in order to include the Doppler measurement by correcting the predicted radial velocity with respect to the measured (noisy) radial velocity $v_{k;r}$. This is done according to

$$\boldsymbol{x}_k = \begin{bmatrix} \tilde{\boldsymbol{r}}_k \\ \tilde{\boldsymbol{v}}_k + \left(\tilde{v}_{k;r} - \tilde{\boldsymbol{v}}_k \cdot \dfrac{\tilde{\boldsymbol{r}}_k}{\|\tilde{\boldsymbol{r}}_k\|}\right) \dfrac{\tilde{\boldsymbol{r}}_k}{\|\tilde{\boldsymbol{r}}_k\|} \end{bmatrix}$$

where $\tilde{\boldsymbol{r}}_k$ and $\tilde{\boldsymbol{v}}_k$ are the position and velocity vectors of $\tilde{\boldsymbol{x}}_k$ and $\tilde{v}_{k;r}$ is sampled from the measurement distribution with

$$\tilde{v}_{k;r} \sim \mathcal{N}\left(v_{k;r}, \sigma_r^2\right)$$

where σ_r^2 is the variance of the measurement.

Incorporating the Doppler measurement during the prediction step ensures that the majority of the particles are located in the relevant range of the target distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k}, u_{1:k})$, resulting in a reduction of the number of required particles. Considering this measurement type in the correction step instead would result in a lot of particles with small likelihoods, because of the small variance of the measurement. Hence, more particles would be required to adequately cover the relevant range of the target distribution.

V.II Measurement model

The measurement model provides the measurement likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ and is used to calculate the importance weight (see Eq. 8). The measurement vector $\boldsymbol{z}_k$ contains the planets' and Sun's angular chord measurements $\boldsymbol{z}_{k;c}$ and the planet and Sun to star measurements $\boldsymbol{z}_{k;s}$. Both are independent of each other. Hence, the measurement likelihood can be factorized into

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) = p(\boldsymbol{z}_{k;c}|\boldsymbol{x}_k)p(\boldsymbol{z}_{k;s}|\boldsymbol{x}_k)$$

The measurements of each celestial body (incl. the Sun) as well as the respective measurements to the stars are also independent of each other. Hence, the likelihood can be further factorized according to

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) = \prod_{p\in P} p\left(z_{k;c_p}\middle|\boldsymbol{x}_k\right) \prod_{n\in N_p} p\left(z_{k;s_{p,n}}\middle|\boldsymbol{x}_k\right)$$

where P denotes the set of observed celestial bodies, N_p denotes the set of stars observed for celestial body p, $z_{k;c_p} = \gamma_p$ denotes the cord measurement of celestial body p, and $z_{k;s_{p,n}} = \beta_s$ the angular measurement between celestial body p and star n.

The celestial body chord measurement likelihood is given by

$$p\left(z_{k;c_p}\middle|\boldsymbol{x}_k\right) = \phi\left(z_{k;c_p}; \tan^{-1}\frac{c_p}{\|\boldsymbol{r}_{k;p} - \boldsymbol{r}_k\|}, \sigma_{c_p}^2\right)$$

where ϕ is the probability density function of the normal distribution, $\boldsymbol{r}_{k;p}$ denotes the current position vector of planet p, $c_p = R_p$ is its planetary chord length, and $\sigma_{c_p}^2$ is the variance of the measurement.

The celestial body to star measurement likelihood is given by

$$p\left(z_{k;s_{p,n}}\middle|\boldsymbol{x}_k\right) = \phi\left(z_{k;s_{p,n}}; \cos^{-1}\frac{\left(\boldsymbol{r}_{k;p} - \boldsymbol{r}_k\right)\cdot \boldsymbol{r}_{k;p,n}}{\|\boldsymbol{r}_{k;p} - \boldsymbol{r}_k\|}, \sigma_{s_{p,n}}^2\right)$$

where $\boldsymbol{r}_{k;p,n}$ denotes the unit position vector of the n-th star observed for planet p, and $\sigma_{s_{p,n}}^2$ is the variance of the measurement.

V.III Adaptive Resampling

The frequency of resampling is crucial for the performance of the PF. If it is performed too frequently, e.g., after each correction step, the particle set loses its diversity. When – on the other hand – it is performed too infrequently, the particle set degenerates and an insufficient number of particles is left in the relevant range of the target distribution.

To overcome this issue, we are using adaptive resampling [22], where resampling takes place when the particle set does not represent the target distribution well enough. This state is quantified by the effective number of particles M_{eff}, defined as

$$M_{eff} = \frac{\left(\sum_{m=1}^{M} w_k^{[m]}\right)^2}{\sum_{m=1}^{M}\left(w_k^{[m]}\right)^2}$$

If M_{eff} drops below a certain threshold (e.g., $M/2$), resampling is performed.

VI. NAVIGATION PERFORMANCE

In order to evaluate the navigation performance of the optical Sun Doppler navigation concept proposed here, two main test scenarios were selected: an Earth-like orbit (scenario A) and the parking orbit of the KaNaRiA mission (scenario B).

In scenario A we have opted for an orbit geometry similar to that of planet Earth around the Sun. Both estimators – the UKF and the PF – are tested in this setup using first Sun-based measurements only – i.e., radial velocity and star-Sun angles – and later using planet chord and planet-star measurements additionally. Scenario A allows to compare our results with those produced by Guo [4] and Yim [6] and to quantize the performance of our UKF and PF relative to Extended Kalman Filter (EKF) implementation by the latter.

Scenario B is built upon the reference mission scenario of the project KaNaRiA [24, 25]. The KaNaRiA mission concept foresees the operation of autonomous space infrastructure for the mining of raw materials on asteroids. Within this broad goal, the KaNaRiA project focuses on the design and simulation of one of the involved mission elements: the Potential Target Characterization Modules (PTCMs). PTCMs are prospecting spacecraft in charge of the characterization of the potential target asteroids prior to the mining mission. They are deployed from a carrier module on a parking orbit within the Asteroid Main Belt from where they swarm to their specified target asteroids.

The size of the KaNaRiA parking orbit employed here for the navigation performance analysis was designed upon the number of accessible asteroids, provided the Δv capability of PTCMs. The JPL Small-Body Database was used for this study. With a planned maximum Δv of 1500 m/s, the largest number of asteroids can be reached when starting from a parking orbit with a semi-major axis of 2.8 AU [24], see Fig. 1.

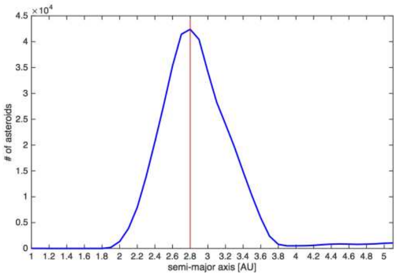


Fig. 1: Number of accessible asteroids using Hohmann transfer with varying semi-major axis for the starting heliocentric orbit. A delta-v limit of 1500 m/s was set for the analysis [24].

The design of the parking orbit, apart from its size, was subject of study in order to optimize the navigation performance as well as the accessibility of the asteroids in terms of transfer time and energy demands.

Unless otherwise noted, the results presented in this section for the PF have been computed using 50 particles.

VI.I Scenario A: Earth-like orbit

Two tests were carried out in the scenario of an Earth-like orbit geometry: navigation using Sun-based measurements only and navigation using both Sun and planetary measurements.

In the first case, radial velocity measurements from the optical interferometer and star-Sun angular measurements from the coupled star-Sun tracker were synthesized. A STD of 1 cm/s for the interferometer measurements and 1.7453 μrad for the star tracker were assumed. These values were selected in accordance with [6] in order to produce comparable results.

In the second test case, the radial chord length of the largest observable planet and star-planet angular measurements were considered additionally. The same noise STD for star-planet measurements as for star-Sun observables was used.

Simulations were performed over a time span of one Earth year starting on December, 31st of year 2029 at midnight 23:59:59.

The Keplerian elements defining the test orbit are listed in Table 2.

Orbit element	Magnitude	Unit
Semi-major axis	1.5426 x 10^8	km
Numerical eccentricity	0.033	-
Orbital inclination	1.92	deg
Right ascension node	0	deg
Argument of perigee	297.9	Deg

Table 2: Orbital elements for Scenario A.

The estimated error time series of the spacecraft heliocentric position and velocity vectors using Sun-based measurements only are depicted in Fig. 2 for the UKF. The 3σ STD of the coordinate errors are plotted along. The error time series were calculated as direct differences between the Runge-Kutta orbit-propagator output of the simulation and the estimates of the filters. The same *true* orbit and observations were used in the calculation for both, the UKF and the PF. In the following, when discussing estimated errors, we always refer to the before mentioned difference between estimated and computed. The lack of out-of-plane observables causes the z-components to be poorly observed, leading to larger variances of the filter estimates as compared to other vector components.

When planet LOS observations are additionally considered – as presented in Fig. 3 – the filter convergence time and estimated variance of the out-of-plane components are significantly reduced.

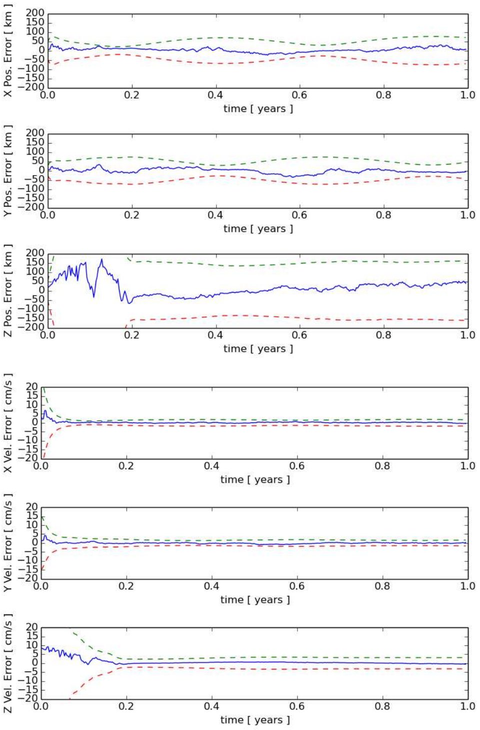


Fig. 2: Position (top) and velocity (bottom) error with 3σ STD of the Unscented Kalman Filter estimates using Sun-based measurements only.

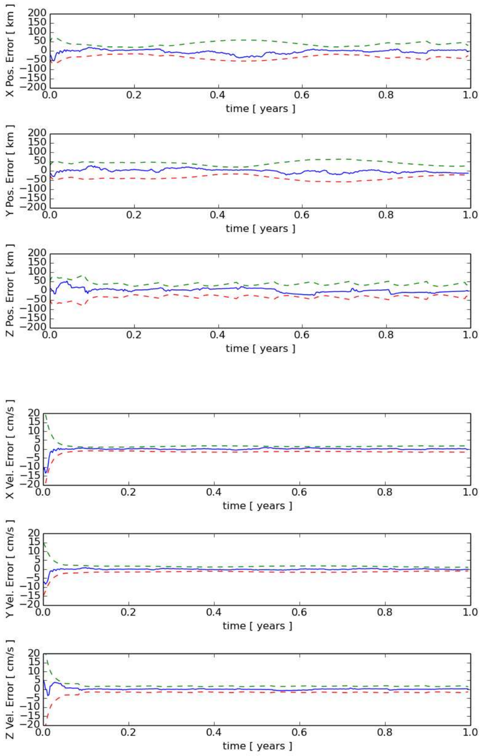


Fig. 3: Position (top) and velocity (bottom) error with 3σ STD of the Unscented Kalman Filter estimates using both Sun-based and planet-based measurements.

Daily observations were collected in every case. Each day the largest observable planet was selected for observation when no solar conjunctions occurred. Over 100 observations of Earth and Jupiter were used, together with approximately 50 observations of Venus and 70 of Saturn.

The same tests were carried out using the PF as estimator. The 3σ STD of the estimated errors at the end of the one-year simulations are summarized in Table 3. The 3σ STD of similar tests using an EKF [6] are also listed for comparison.

From the provided 3σ STD it could be understood that both the UKF and PF perform significantly worse than the EKF. However LOS measurements were calculated in [6] directly from the true spacecraft position and planetary ephemeris. We believe our observation modelling – considering the Hipparcos catalogue and true astrometric measurements – to be consistently more representative of the expected true measurement performance.

	Comp.		EKF [6]	UKF	PF
Sun-based measurements	x	km	8.57	70.59	112.34
	y	km	30.13	43.77	172.48
	z	km	21.20	159.75	39.29
	$\dot{x}$	mm/s	5.09	17.52	48.77
	$\dot{y}$	mm/s	3.14	16.38	211.25
	$\dot{z}$	mm/s	3.93	11.79	9.22
Sun and planetary measurements	x	km	5.39	27.51	43.63
	y	km	14.77	22.71	38.29
	z	km	2.50	26.25	33.84
	$\dot{x}$	mm/s	2.86	14.58	87.82
	$\dot{y}$	mm/s	1.45	9.90	378.06
	$\dot{z}$	mm/s	0.73	14.67	13.21

Table 3: Standard deviations (3σ) of position and velocity components at end of simulation for an Earth-like orbit geometry.

A positioning accuracy in the order of 200 km can be expected when using Sun-based observations only, which coincides with the results by Guo [4]. This performance is improved by a factor of four when planetary LOS are included in addition.

Velocity estimates accurate to 20 mm/s can be expected for the UKF. The PF implementation however uses a lighter single-step integrator which can rapidly introduce higher errors in the along-track velocity component.

VI. II Scenario B: KaNaRiA parking orbit

In the scenario of a heliocentric orbit within the main asteroid belt, several tests were carried out covering three main optimization aspects for improved navigation performance: orbit geometry, density of sampling particles for the PF and measurement update rate. In the following, we present and discuss results derived from these three studies.

Simulations were performed over a time span of 5 Earth years starting on December, 31st of year 2029 at midnight 23:59:59 UTC.

Orbit Geometry

In order to quantify the effect of the observation geometry on the selected parking orbit and to support the optimal selection of KaNaRiA mission trajectories, several test cases were studied. For each of these test cases, simulations were performed applying parametric variations of one orbital element at a time.

In particular, the following orbit design options were evaluated: orbit size (1), orbit shape (2), orbit orientation within its plane (3), orientation of the orbital plane (4) and orbit phasing (5). For the sake of compactness, orbit geometry test cases are identified by *BXY* where *B* stands for the scenario, *X* stands for the orbit design parameter involved – as listed above, from (1) to (5) – and *Y* stands

X	Y	Varied parameter	Magnitude	Units
2	1	Numerical eccentricity e	0	-
	2		0.005	
	3		0.05	
	4		0.5	
3	1	Argument of perigee ω	0	deg
	2		45	
	3		90	
	4		180	
	5		270	
4	1	Orbit inclination i	0	deg
	2		5	
	3		15	
	4		25	
	5		45	
5	1	Mean anomaly M	0	deg
	2		45	
	3		90	
	4		180	
	5		270	

Table 4: Orbit geometry test cases for Scenario B.

Orbit element	Magnitude	Unit
Semi-major axis	2.8	AU
Numerical eccentricity	0.033	-
Orbital inclination	5	deg
Right ascension node	0	deg
Argument of perigee	180	deg

Table 5: Keplerian elements for orbit geometry test cases within Scenario B, KaNaRiA parking orbit.

for the index of the actual value given to the design parameter as listed in Table 4. Besides the intended parametric variations, the fixed orbit configuration is listed in Table 5.

Sun-based and planetary observation updates were generated every 3 hours. In Fig. 4, statistics of the performance of the UKF and PF are presented for the orbit geometry test cases listed in Table 4. In particular, the mean value of the l^2-norm of the estimated 3D position and velocity error time series are depicted as a bar chart. The STD of the series are provided as overlaid error bars.

The PF position estimates are consistently accurate to 50 km throughout all test cases, with the exception of case *B24* ($e = 0.5$) where the large orbit eccentricity causes the single-step propagator to show its downside. UKF position estimates show mean error of about 200 km with significant dependencies on the orbit inertial orientation and the spacecraft phasing (see cases *B3Y* and *B5Y*). Solar optical Doppler shift measurements aid the determination of spacecraft the velocity. Estimates in the 1.5 cm/s range are obtained with both the UKF and the PF.

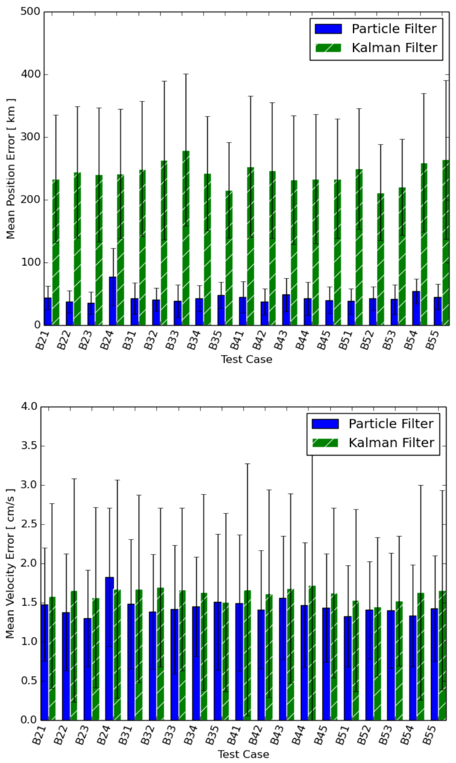


Fig. 4: Mean values of the l^2-norm of the 3D position (top) and velocity (bottom) estimation error with varying orbit geometry for the UKF and PF orbit geometry test cases. STD of the time series are depicted as error bars.

Density of sampling particles

All orbit geometry tests described in the previous section (see Table 4) were conducted with the PF with a varying number of sampling particles. A minimum of 1 and a maximum of 1000 particles were considered. The computational demand and consequent processing delay when using a high number of particles would make the PF solution unsuitable for real-time on-board navigation (see section IV). Therefore it is desirable that the major improvements in navigation accuracy are achieved with a small number of sampling particles.

Fig. 5 shows the mean position and velocity estimation errors overall test cases listed in Table 4 as function of the number of particles in the PF. The mean error of the UKF overall test cases is also depicted as constant threshold for comparison purposes.

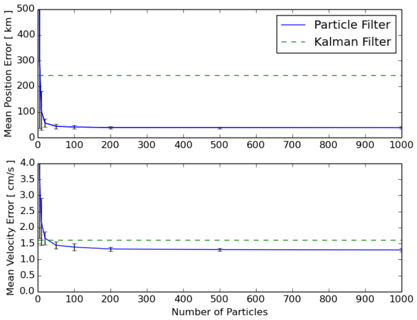


Fig. 5: Mean error of the l^2-norm of the estimated 3D position (top) and velocity (bottom) over 19 runs with varying orbit geometry as function of the number of particles for the PF. STD over the test cases are shown as error bars.

Using as little as 25 particles the PF matches the performance of the UKF. With few more particles, the UKF is clearly outperformed by the PF. This leads us to the conclusion that the PF can be introduced for peak performance only using as little as 50 particles, thus being suitable for the limited on-board computational resources. The stability of the PF is however sensitive to sudden measurement noise variations due to the low trust granted to the motion model based on a single-step integrator. To overcome this problem, a combination of both, the UKF and the PF, would deserve further study in the authors' opinion.

Measurement update rate

Based on the outcome of the orbit geometry *BXY* test cases presented above, a parking orbit for the KaNaRiA mission was selected which is optimal in the navigation sense. The selected orbit configuration is listed in Table 6. The spacecraft mean anomaly is defined for December, 31st of year 2029 at midnight 23:59:59 UTC.

Orbit element	Magnitude	Unit
Semi-major axis	2.8	AU
Numerical eccentricity	0.05	-
Orbital inclination	15	deg
Right ascension node	0	deg
Argument of perigee	270	deg
Mean anomaly	270	deg

Table 6: Optimal configuration of the KaNaRiA parking orbit for navigation.

Simulations with varying measurement update rates were performed in order to quantify both the sensitivity of the system to the measurement update rate and the lower boundaries of the measurement update rate for precise navigation in the 100 km precision range, which is the requirement for KaNaRiA PTCMs during cruise.

Five-year long tests were carried out with the UKF and the PF using Sun-based and planetary measurements. Due to the proximity of their orbit and their large chord length, mainly Jupiter and Saturn were selected for observations. However observations of the Earth, Mars and Venus were also used when the observation geometry allowed for it. Measurement update rates varying from as short as 15 min to once per week were also studied.

In Fig. 6, the mean estimation error for both filters is presented for different measurement update intervals. The STD of the time series are depicted as overlaid error bars. The UKF relies strongly on the dynamic equations of motion for navigation. Given the slow varying system geometry (about one degree every 3 Earth days relative to the Sun) the UKF performs at best with daily measurement updates. A mean 3D position estimation error of approx. 85 km is achieved.

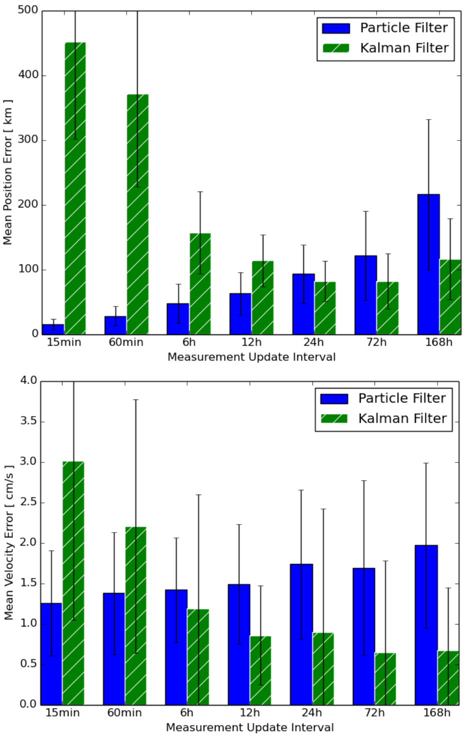


Fig. 6: Mean value of position (top) and velocity (bottom) estimated errors for different measurement update intervals. The STD of the error time-series are depicted as error bars.

Higher measurement rates increase the filter noise and do not contribute to the enhancement of performance in the case of the UKF. On the other hand, the PF uses a simplified single-step integrator with fixed step sized which causes the filter to be highly reliant on observations. The positioning performance of the PF increases with increasing measurement update rate leading to a few km position mean error at 15 minute intervals. In terms of velocity estimates, both filters show similar performance with best solutions yielding 1 cm/s mean error. The UKF achieves its peak performance at low update rates, while the PF sees its best at high-rate updates.

Figs. 7 and 8 show the estimated position and velocity errors for the PF and the UKF in their best performing measurement configuration scenario. The PF series has been obtained with 15 min measurement update interval, while the UKF uses daily updates.

The PF clearly outperforms the UKF for this measurement rate in both position and velocity estimate errors. However, the cost of using a high measurement rate might turn unacceptable during nominal mission operations as the spacecraft subsystems require the available computational power for other maintenance tasks. The access of the navigation task to these resources can be very limited.

In Table 7, the 3σ STD of the position and velocity estimated errors at end of simulation are listed for 15 minute and daily measurement updates. At a high measurement rate 100 km and 10 cm/s STD are obtained for the UKF, whereas the PF achieves 30 km in position and 3 cm/s in velocity.

Using daily update rates both filters show a comparable performance of 150 km STD in position and 3.5 cm/s in velocity. Both filters fulfil the standard navigation requirements for in-orbit cruise and orbit correction manoeuvring.

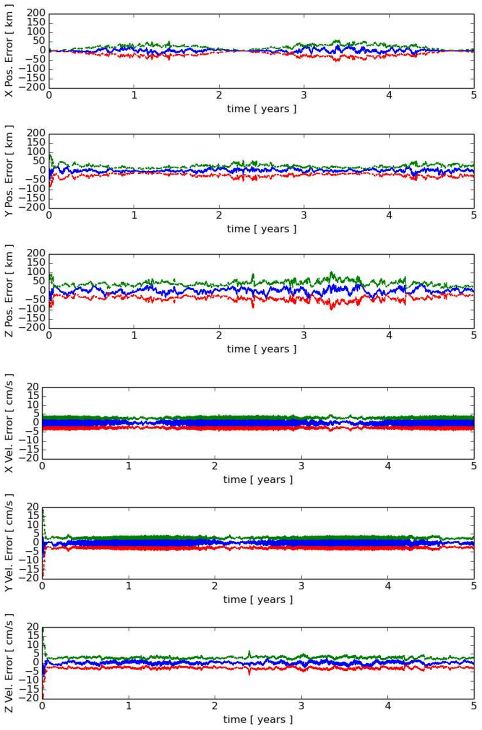


Fig. 7: Position (top) and velocity (bottom) error with 3σ STD of the PF estimates using 15 minute measurement updates on a KaNaRiA parking orbit geometry.

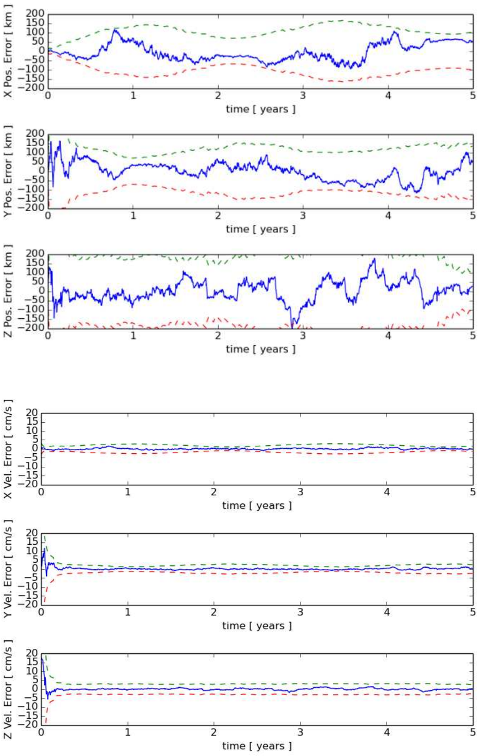


Fig. 8: Position (top) and velocity (bottom) error with 3σ STD of the UKF estimates using daily measurement updates on a KaNaRiA parking orbit geometry.

	Component		UKF	Particle Filter
15 min	x	km	108.12	9.10
	y	km	48.30	29.69
	z	km	44.97	23.37
	$\dot{x}$	mm/s	92.40	31.90
	$\dot{y}$	mm/s	86.25	31.02
	$\dot{z}$	mm/s	82.83	29.07
24 hour	x	km	102.75	58.98
	y	km	134.79	135.80
	z	km	119.40	79.23
	$\dot{x}$	mm/s	14.73	33.73
	$\dot{y}$	mm/s	24.60	33.79
	$\dot{z}$	mm/s	26.82	31.77

Table 7: Standard deviations (3σ) of estimated errors at end of simulation for a KaNaRiA parking orbit geometry. Statistics for measurement updates rates of 15 minutes and 24 hours are provided.

VII. CONCLUSION

A method for autonomous spacecraft navigation suitable for interplanetary cruise has been presented in this work. The approach combines the solar navigation approach conceived by Guo [2] with state of the art optical navigation. Three observation types are used: the optical Doppler shift due to the relative motion of the spacecraft relative to the Sun, the LOS towards the Sun and the LOS towards planetary bodies with known ephemerides. The optical Doppler shift allows to derive the spacecraft radial velocity relative to the Sun at all times during transfer. The LOS towards the Sun can be accurately obtained by means of astrometric measurements from a coupled star-Sun tracker. The LOS towards planetary bodies can be similarly measured using a digital CCD camera.

The performance of the navigation system has been assessed with two different estimators: a UKF highly-reliant on its dynamic motion model and a PF with a computationally light single-step integrator. The estimators have been evaluated in the scenario of an Earth-like orbit mission and for the KaNaRiA asteroid mining mission considering a distance of 2.8 AU from the Sun. In the Earth-like orbit scenario, results have been compared to previous work by other authors yielding comparable outcomes.

Software-based simulations confirm that a 3σ STD of estimated errors of about 50 km in position and 2 cm/s can be obtained at 1 AU using daily measurement updates. The UKF presents a more robust behaviour as compared to the PF for almost circular orbits and semi-major axis of circa 1 AU. For larger orbit sizes and high-rate measurements – i.e. 15 minute updates – the PF clearly outperforms the UKF. Deviations of approx. 30 km in position and 3 cm/s in velocity (3σ) can be achieved with the PF at 2.8 AU using high-rate measurements.

At lower rates – e.g. daily up to weekly measurement update intervals – the UKF and the PF show comparable performance yielding 150 km and 3.5 cm/s position and velocity 3σ STD at 2.8 AU.

The PF was tuned to perform comparably or better than the UKF using a limited number of 50 particles to approximate the target distribution. This results confirms the suitability of the PF solution to be operated with restricted on-board computational resources.

The combined optical navigation concept presented here fulfils the accuracy requirements demanded for mission cruise and in-orbit manoeuvring of most interplanetary missions today. During approach the LOS towards the target asteroid could be used as additional observable.

In the scope of KaNaRiA plans are set to investigate Simultaneous Localization and Mapping (SLAM) algorithms for asteroid proximity navigation. These algorithms, together with the cruise navigation concept presented here, will complete the design of the end-to-end autonomous navigation system for KaNaRiA asteroid prospecting spacecraft.

REFERENCES

[1] D. C. Folta, C. J. Gramling, A. C. Long, D. S. Leung and S. V. Belur, “Autonomous navigation using celestial objects,” in *Proceedings of the AAS/AIAA Astrodynamics Symp. 1999*, San Diego, California, 2000.

[2] Y. Guo, “Self-contained autonomous navigation system for deep space missions,” in *Proceedings of the AAS/AIAA Spaceflight Mechanics Conference 1999*, 1999.

[3] Y. Guo, “Method and apparatus for autonomous solar navigation”. United States Patent US006622970B2, 23 September 2003.

[4] Y. Guo and T. E. Strikwerda, “Performance Simulation of Autonomous Solar Navigation,” in *Proceedings of the 1999 Flight Mechanics Symposium*, 1999.

[5] A. Gipsman, M. Guelman and A. Kogan, “Autonomous navigation and guidance system for low thrust driven deep space missions,” *Acta Astronautica,* vol. 44, no. 7-12, pp. 353-364, 1999.

[6] J. R. Yim, J. L. Crassidis and J. L. Junkins, “Autonomous orbit navigation of interplanetary spacecraft,” in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference 2000*, Denver, Colorado, 2000.

[7] J. Riedel, S. Bhaskaran, S. Synnott, W. Bollman and G. Null, “An autonomous optical navigation and control system for

interplanetary exploration missions," in *Proceedings of the 2nd IAA International on Low-cost Planetary Missions IAA-L-506*, 1996.

[8] S. Bhaskaran, J. Riedel and S. Synnott, "Autonomous optical navigation for interplanetary missions," in *Proceedings SPIE 2810. Space Sciencecraft Control and Tracking in the New Millennium*, Denver, CO, 1996.

[9] J. Riedel, S. Bhaskaran, D. Eldred, R. Gaskell, C. Grasso, B. Kennedy, D. Kubitscheck, N. Mastrodemos, S. Synnott and A. W. R. Vaughan, "AutoNav Mark3: Engineering the next generation of autonomous onboard navigation and guidance," *Proceedings of the AAS/AIAA GNC Conference,* p. 18, 22-25 August 2006.

[10] M. Vasile, M. Romano and F. Triniti, "An Optical Based Strategy for Deep Space Autonomous Navigation," in *Proceedings of the 4th ESA Intenational Conference on Spacecraft Guidance, Navigation and Control Systems 1999*, ESTEC, Noordwijk, The Netherlands, 2000.

[11] B. Polle, B. Frapard, J. Gil-Frenández, E. Millic, M. Graziano, J. Rebordao and P. Motrena, "Autonomous navigation for interplanetary missions. Performance achievements based on real and flight images.," *Proceedings of the 6th International ESA Conference on Guidance, Navigation and Control Systems 2005,* January 2006.

[12] B. Polle, B. Frapard and T. Voirin, "Vision Based Navigation for Planetary Exploration Opportunity for AURORA," in *Proceedings of the 54th International Astronautical Congress of the International Astronautical Federation*, Bremen, Germany, 2003.

[13] P. Munoz, F. Budnik, B. Godard, T. Morley, V. Companys, U. Herfort and C. Casas, "Preparations and Strategy for Navigation During Rosetta Comet Phase," in *Proceedings of the International Symposium on Space Flight Dynamics*, Pasadena, California, 2012.

[14] M. Bernhardt, W. Becker, T. Prinz, F. M. Breithuth and U. Walter, "Autonomous spacecraft navigation based on pulsar timing information," in *Proceedings of the 2nd International Conference on Space Technology*, 2011.

[15] K. Gendreau, "NICER Mission Overview, Objectives and Status," Presentation at the 593rd WE-Heraus Seminar on Autonomous Spacecraft Navigation , Bonn, Germany, 2015.

[16] C. Fröhlich and J. Lean, "Solar Radiative Ouput and its Variability: Evidence and Mechanisms," *The Astronomy and Astrophysics Review,* vol. 12, pp. 273-320, 2004.

[17] P. H. Scherrer, R. S. Bogart, R. I. Bush, J. T. Hoeksema, A. G. Kosovichev, J. Schou, W. Rosenberg, L. Springer, T. D. Tarbell, A. Title, C. J. Wolfson and I. Zayer, "The Solar Oscillations Investigation - Michelson Doppler Imager," *Solar Physics,* vol. 162, no. 1-2, pp. 129-188, 1995.

[18] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE - The International Society for Optical Engineering,* 1999.

[19] R. van der Merwe, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models,* Dissertation from Oregon Health & Science University, 2004.

[20] S. J. Julier, "The Scaled Unscented Transformation," in *Proceedings of the American Control Conference*, 2002.

[21] "Explanatory Supplement to the Astronomical Almanac," 2013, p. Table 5.8.2 on page 316.

[22] A. Doucet, N. de Freitas and N. Gordon, "Sequential Monte Carlo Methods in Practice," *Springer Science & Business Media,* 2001.

[23] S. Thrun, W. Burgards and D. Fox, Probabilistic Robotics, MIT press, 2005.

[24] A. Probst, G. González Peytaví, B. Eissfeller and R. Förstner, "Reference Mission Scenario Selection for Main Belt Asteroid Mining Missions," in *Planetary and Terrestrial Mining Science Symposium (PTMSS), 2015 May 10-13*, Montréal, Quebec, 2015.

[25] A. Probst, G. González Peytaví, B. Eissfeller and R. Förstner, "Mission Concept Selection for an Asteroid Mining Mission," *Journal on Aircraft Engineering and Aerospace Technology,* vol. 88, no. 3, 2015.

ACKNOWLEDGEMENT

This work was conducted within the KaNaRiA project. KaNaRiA is supported by the German Federal Ministry of Economics and Technology (BMWi) on the basis of a decision by the German Bundestag, coordinated by the German Aerospace Centre (DLR) and executed by the University of Bremen (FKZ 50NA1318) and the Bundeswehr University in Munich (FKZ 50NA1319).

Rigid Body Attitude Control Based on a Manifold Representation of Direction Cosine Matrices

David Nakath, Joachim Clemens, Carsten Rachuy

Cognitive Neuroinformatics, University of Bremen,
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

E-mail: {dnakath,jclemens,rachuy}@informatik.uni-bremen.de

Abstract. Autonomous systems typically actively observe certain aspects of their surroundings, which makes them dependent on a suitable controller. However, building an attitude controller for three degrees of freedom is a challenging task, mainly due to singularities in the different parametrizations of the three dimensional rotation group $SO(3)$. Thus, we propose an attitude controller based on a manifold representation of direction cosine matrices: In state space, the attitude is globally and uniquely represented as a direction cosine matrix $R \in SO(3)$. However, differences in the state space, i.e., the attitude errors, are exposed to the controller in the vector space $\mathbb{R}^3$. This is achieved by an operator, which integrates the matrix logarithm mapping from $SO(3)$ to $\mathfrak{so}(3)$ and the map from $\mathfrak{so}(3)$ to $\mathbb{R}^3$. Based on this representation, we derive a proportional and derivative feedback controller, whose output has an upper bound to prevent actuator saturation. Additionally, the feedback is preprocessed by a particle filter to account for measurement and state transition noise. We evaluate our approach in a simulator in three different spacecraft maneuver scenarios: (i) stabilizing, (ii) rest-to-rest, and (iii) nadir-pointing. The controller exhibits stable behavior from initial attitudes near and far from the setpoint. Furthermore, it is able to stabilize a spacecraft and can be used for nadir-pointing maneuvers.

1. Introduction

Autonomous systems typically need to keep track of their position and attitude, i.e., their pose, as well as properties of their surroundings for successful operations. The gathered information is mutually beneficial, as – on the one hand – precise pose information improves the ability to describe the surroundings and – on the other hand – a precise description of the latter improves the ability to determine an exact pose. To this end, it is often useful for autonomous systems to actively observe certain aspects of their surroundings, which makes them dependent on a suitable controller.

This especially holds for space missions, where active observation of planetary objects can be used for absolute navigation [11, 18] as well as for nadir-pointing maneuvers for relative navigation in the vicinity of a celestial object, e.g., within a LIDAR-based SLAM approach [6, 20]. However, the scenario may also apply to airplanes, underwater vehicles [9], ground vehicles, and robotic systems, which have to inspect certain aspects of their surroundings by aligning some sensor to a point of interest.

The only global and unique representation of the set of all attitudes a rigid body might assume in three dimensions is the *direction cosine matrix* (DCM) [5]. Parameterizations like

Euler angles have the disadvantage of geometric and kinematic singularities as well as different conventions. The (modified) Rodriguez parameters, which are derived from unit quaternions, suffer from geometric singularities. Even the globally defined unit quaternions can lead to undesired rotations, which is known as controller unwinding [22, 2, 16]. The latter problem stems from the non-unique map between the unit quaternion space $\hat{\mathbb{H}}$ and the rotation space $SO(3)$ [2, 5].

As the computational power and available memory of autonomous systems constantly increases, it has become feasible to directly employ DCMs for pose representations. Thus, the drawbacks of the parametrized attitude representations can be omitted by running control algorithms directly on DCMs [10].

In this paper, we present a filtered proportional derivative saturation preventing rigid body controller. It represents the attitude in $SO(3)$ using a DCM, while exposing the differences between two attitudes in the vector space $\mathbb{R}^3$ employing the $\boxplus$-method [13]. The derivative term is bounded by a nonlinearity, to ensure a bounded control torque in each particular axis. The feedbacks are preprocessed by a particle filter to account for state and measurement noise. The tuning of the controller in the respective maneuver scenarios can be considered as another modest contribution.

This paper is structured as follows. First, in Sect. 2, the problems to be addressed and the corresponding mathematical and geometrical preliminaries will be presented. In Sect. 3, the spacecraft (s/c)-state and a particle filter-based (PF) estimation scheme will briefly be introduced. Based on these foundations, the controller will be derived in Sect. 4. Subsequently, it will be evaluated and discussed in Sect. 5. Finally, this paper concludes in Sect. 6.

2. Preliminaries

In the following, we will briefly cover the mathematical foundations of the derivation of the controller.

2.1. Attitude

Consider a rigid body representing a s/c.[1] An attitude is the rotation of the principal axes of some body-fixed frame F_b relative to some global inertial frame F_a, which is uniquely described by a DCM denoted by R_{ba}. The set of all attitudes is composed of all orthonormal rotation matrices with determinant equal to one, i.e., $\{R \in \mathbb{R}^{3\times3} | R^T R = 1, \det R = +1\}$. This set forms the special orthogonal group $SO(3)$ of rigid body rotations in 3D [17].

2.2. $\boxplus$-Method

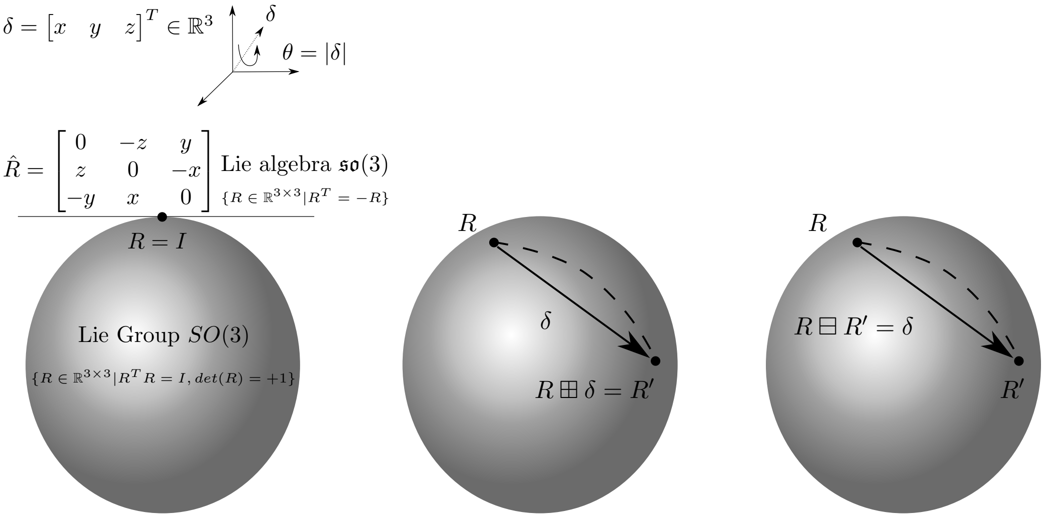
We use a $\boxplus$-encapsulation of $SO(3)$ introduced by Hertzberg et al. [13], which is capable of integrating a higher dimensional state representation $\mathcal{S}$ with a local lower dimensional state representation $\mathbb{R}^n$ (cf. Fig. 1). This is achieved by the mappings

$$\begin{aligned} \boxplus &: \mathcal{S} \times \mathbb{R}^n \to S, \\ \boxminus &: \mathcal{S} \times \mathcal{S} \to \mathbb{R}^n. \end{aligned} \tag{1}$$

The DCM-specific operators encapsulate the manifold by combining the matrix exponential ($\boxplus$-operator) or logarithm ($\boxminus$-operator) respectively with the isomorphism between the Lie-algebra $\mathfrak{so}(3)$ and $\mathbb{R}^3$, in particular

$$\begin{aligned} x \boxplus \delta &= x \exp \delta, \\ y \boxminus x &= \log(x^{-1} y), \end{aligned} \tag{2}$$

[1] We will only take attitude control into account in this paper, as it can be treated independently of translation in our scenario.


$\delta = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$
δ
$\theta = |\delta|$
$\hat{R} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$ Lie algebra $\mathfrak{so}(3)$
$\{R \in \mathbb{R}^{3\times3} | R^T = -R\}$
$R = I$
Lie Group $SO(3)$
$\{R \in \mathbb{R}^{3\times3} | R^T R = I, det(R) = +1\}$
R
δ
$R \boxplus \delta = R'$
R
$R \boxminus R' = \delta$
R'

(a) Joint structure of $SO(3)$, $\mathfrak{so}(3)$, and $\mathbb{R}^3$.

(b) $\boxplus$-operator: $\delta \in \mathbb{R}^3$ added to element $R \in SO(3)$, resulting in $R' \in SO(3)$.

(c) $\boxminus$-operator: Distance computation between R and R' on the $SO(3)$-manifold with result $\delta \in \mathbb{R}^3$.

Figure 1: (a) Joint structure of $SO(3)$, $\mathfrak{so}(3)$, and $\mathbb{R}^3$ with corresponding encapsulating operators in (b) and (c). Dashed lines indicate the geodesics on the manifold, while solid arrows indicate the corresponding vectors in $\mathbb{R}^3$. Following [13], the $\boxplus$-principles are shown using the unit sphere S^2 to simplify visualization.

with $x, y \in SO(3)$ and $\delta \in \mathbb{R}^3$. In the case of 3D rotations, the matrix exponential can be computed in closed-form by the Rodriguez-formula [13, 17, 14]. Combined with the hat map from $\mathbb{R}^3$ to $\mathfrak{so}(3)$ this results in

$$\exp \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta + cx^2 & -sz + cxy & sy + cxz \\ sz + cxy & \cos\theta + cy^2 & -sx + cyz \\ -sy + cxz & sx + cyz & \cos\theta + cz^2 \end{bmatrix}, \tag{3}$$

where $\theta = \sqrt{x^2 + y^2 + z^2}$, $s = \frac{\sin\theta}{\theta}$, and $c = \frac{1-\cos\theta}{\theta^2}$. The inverse mapping is a combination of the matrix logarithm and the hat map from $\mathfrak{so}(3)$ to $\mathbb{R}^3$, which is computed according to

$$\log x = \frac{\theta}{2\sin\theta} \begin{bmatrix} x_{32} - x_{23} \\ x_{13} - x_{31} \\ x_{21} - x_{12} \end{bmatrix}, \tag{4}$$

where $\theta = \operatorname{acos}\frac{\operatorname{tr} x - 1}{2}$ and $|\theta| < \pi$. Thus, the global state is represented uniquely in $SO(3)$, while state differences δ, i.e., state transitions or errors, are uniquely represented in $\mathbb{R}^3$ for angles $< \pi$.

2.3. Dynamic Setpoint Calculation

To have an autonomous system looking at some observable, we have to rotate the sensor normal into the desired view direction. The rotation of a unit vector A to some desired unit vector B

(both defined in F_a) can be described as a 2D rotation on a plane with the normal vector $A \times B$. The corresponding rotation matrix is given by

$$G = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} A \cdot B & -||A \times B|| & 0 \\ ||A \times B|| & A \cdot B & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where ϕ is the rotation angle. However, to apply the rotation, we have to temporarily change the basis with

$$F = \begin{bmatrix} A & \frac{B-(A \cdot B)A}{||B-(A \cdot B)A||} & B \times A \end{bmatrix}^{-1}. \tag{6}$$

The final rotation matrix R_e is obtained by a transformation from the inertial frame F_a to the rotation frame, followed by a rotation about G, and a transformation back to the inertial frame. Thus, we get

$$R_e = F^{-1} G F. \tag{7}$$

If we now assume, for the sake of simplicity, that we want to orient a sensor whose sensor normal points along the z_{ba} axis of R_{ba}, we have that $A = z_{ba}$ and $B = z_{da}$. Thus, we can calculate a rotation matrix R_e following the procedure described above. The resulting R_e can also be applied to another rotation matrix, as $SO(3)$ is closed under matrix multiplication. Hence, we can obtain the desired attitude R_{da}, i.e., where z_{ba} is rotated to z_{da}, by rotating the current attitude R_{ba} according to

$$R_{da} = R_e R_{ba}. \tag{8}$$

In order to obtain a suitable representation for the controller, the rotation from R_{ba} onto R_{da} can be retrieved in vector space using Eq. (2). This yields

$$\delta = R_{ba} \boxminus R_{da}, \tag{9}$$

where the $\boxminus$-operator ensures that δ represents the geodesic, i.e., the shortest rotation on $SO(3)$ [13]. The approach described above will be used for time-dependent setpoint (SP) calculation $R_{da}(t)$ within the nadir-pointing maneuver (cf. Sect. 5).

2.4. Attitude Kinematics and Dynamics

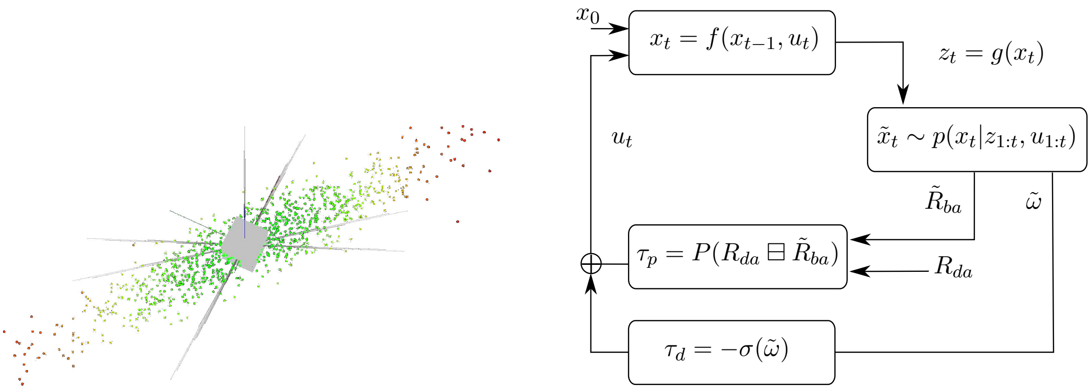
Applying this geometric framework to attitude kinematics, we let $\omega \in \mathbb{R}^3$ be the angular velocity relative to F_a given in F_b, then the time rate of change of the rigid body attitude is

$$\dot{R}_{ba} = R_{ba} \hat{\omega}. \tag{10}$$

Here, $\hat{\omega}$ is a skew symmetric matrix, as it is an element of the Lie algebra $\mathfrak{so}(3)$ (cf. Fig. 1a), and $\dot{R}_{ba}$ maps the body coordinates of a point to the spatial velocity of that point. The attitude dynamics are the rate of change of angular velocity expressed in the body frame given by

$$I\dot{\omega} = I\omega \times \omega + u, \tag{11}$$

where I is the inertia tensor and u is the sum of external moments applied to the body given in F_b. Finally, throughout this paper we assume that the s/c is fully actuated regarding attitude, i.e., angular moments can be applied on all three axes.


x_0
$x_t = f(x_{t-1}, u_t)$
$z_t = g(x_t)$
u_t
$\tilde{x}_t \sim p(x_t|z_{1:t}, u_{1:t})$
$\tilde{R}_{ba}$
$\tilde{\omega}$
$\tau_p = P(R_{da} \boxminus \tilde{R}_{ba})$
R_{da}
$\tau_d = -\sigma(\tilde{\omega})$

(a) Particle filter

(b) Controller structure

Figure 2: (a) shows the s/c visualization with the corresponding particle distribution, where green and red represents a high and low weight respectively. (b) depicts the control diagram of the filtered pd controller.

3. Particle Filtered Measurements

The controller proposed in the next section relies on measurements in the closed loop. To account for noise in state transition as well as sensor measurements we use a particle filter (PF) [8, 21] for state estimation (cf. Fig. 2a). This type of filter is chosen, as it integrates well with the SLAM approach [6] to be used for relative navigation in the vicinity of asteroids. The basic concepts and integration of attitude sensors (star tracker and 3-axis gyroscope), which are relevant in the context of this paper, are described in the following. Further details on our quaternion-based approach including all sensor models can be found in [11, 18]. The state space is defined as $\mathcal{S} = SE(3) \times \mathbb{R}^6$, where $SE(3) = \mathbb{R}^3 \times SO(3)$ represents the space of position and attitude. The remaining $\mathbb{R}^6$ is the space of the corresponding first derivatives, i.e., the linear and angular velocities. The PF estimates a probability distribution of the current state $x_t \in \mathcal{S}$ of the following form

$$\tilde{x}_t \sim p(x_t|z_{1:t}, u_{1:t}), \tag{12}$$

where $\tilde{x}_t \in \mathcal{S}$ is the estimated state, $z_{1:t}$ are all preceding measurements and $u_{1:t}$ are all preceding controls. This probability density function is approximated using a set of particles, where each particle represents a state hypothesis. The particle set is updated recursively over time. In each step, the particles are predicted from the previous to the current time step by sampling from the proposal distribution. This is done by integrating the dynamic model of the s/c, which depends on the previous state estimate and the most recent control u_t, using a fourth order Runge-Kutta integrator and applying additive normally-distributed noise with zero mean. Subsequently, the distribution is corrected using importance resampling. The importance measure is obtained by weighting the particles according to the measurement likelihood given the sampled state. Finally, new particles are sampled with a probability proportional to the weights.

A star tracker is used for determining the attitude R_{ba} of the s/c. It operates by measuring angles to stars (see Fig. 3b) and comparing these to known star positions indexed in a star catalog, e.g., the *HIPPARCOS* [19]. For simplicity, we assume that the resulting attitude measurement z_s corresponds to the true attitude R_{ba} and additive normally-distributed noise with zero mean. Hence, the measurement model is given by

$$z_s = R_{ba} \boxplus \varepsilon_s, \tag{13}$$

with $\varepsilon_s \sim \mathcal{N}(0, \Sigma_s)$, where Σ_s is covariance of the noise.

Furthermore, we use the 3-axis gyroscope integrated in an inertial measurement unit (IMU) to obtain the angular velocity ω (see Fig. 3c) of the s/c. The corresponding measurement z_g is also assumed to be affected by normally-distributed noise with zero mean, which results in

$$z_g = \omega + \varepsilon_g, \tag{14}$$

with $\varepsilon_g \sim \mathcal{N}(0, \Sigma_g)$ and Σ_g being the noise covariance matrix.

In order to retrieve a single estimate from the set of particles as an input for our controller, the posterior distribution is approximated using a normal distribution. This basically corresponds to a weighted average for the mean and calculating the covariance with respect to the estimated mean using the $\boxminus$-operator (see [18] for further details). This avoids computationally expensive least squares estimations [12]. The controller proposed in the following section will rely on the angular velocity $\tilde{\omega}$ and the attitude $\tilde{R}_{ba}$ estimated by the PF as part of $\tilde{x}_t$.

4. Filtered Proportional Derivative Controller

The proportional derivative controller (cf. Fig. 2b) computes the current control u_t as a sum of the proportional τ_p and the derivative τ_d control torque according to

$$u_t = \tau_p + \tau_d. \tag{15}$$

In [4], a feedback controller is presented, which uses the matrix logarithm mapped into vector space to expose the geodesic, i.e., the shortest path on $SO(3)$, as the error term. This approach has been adopted by [1, 15], who additionally show the stability of proportional derivative controllers using such an error term. Analogously, the proportional control term can be computed according to a weighted error function based on the $\boxminus$-operator, resulting in

$$\tau_p = P\,\delta = P(R_{da} \boxminus \tilde{R}_{ba}), \tag{16}$$

where P is the proportional gain matrix. The error function calculates the difference $\delta \in \mathbb{R}^3$ between the desired $R_{da} \in SO(3)$ and the current estimated attitude $\tilde{R}_{ba} \in SO(3)$.

To implement an actuator saturation prevention mechanism [3, 7] based on the proportional term, we observe that δ is guaranteed to never exceed $|\pi|$ in its components. This is asserted, as the error function in Eq. (16) returns the vector representation of the geodesic, whose magnitude can never be greater than π. We initially set $Q = diag(\bar{u}_1/\pi, \bar{u}_2/\pi, \bar{u}_3/\pi)$, where $\bar{u}_i$ indicates the maximum allowed torque per axis, to ensure a bounded maximum value on every axis. For tuning purposes, this value can be scaled with the factor $K = diag([0,1],[0,1],[0,1])$ (cf. Sect. 5). Thus, we let $P = KQ$, where $K_{ii} = 0$ puts all weight on the derivative term, and $K_{ii} = 1$ puts all weight on the proportional term of the respective axis i.

The derivative term is the negative output of the nonlinearity σ, which acts upon the estimated angular velocity $\tilde{\omega}$

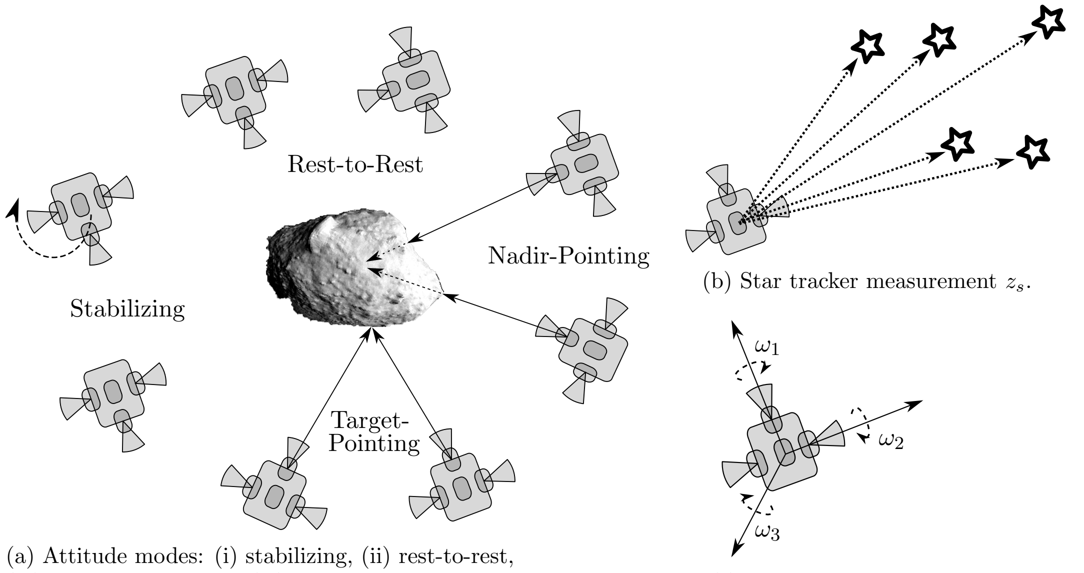
$$\tau_d = -\sigma(\tilde{\omega}). \tag{17}$$

The nonlinearity applies upper and lower bounds to the estimated angular velocity $\tilde{\omega}$ [10]

$$\sigma(\tilde{\omega}) = [\sigma_1(\tilde{\omega}_1), \sigma_2(\tilde{\omega}_2), \sigma_3(\tilde{\omega}_3)]^T \tag{18}$$

$$\sigma_i(\tilde{\omega}_i) = \begin{cases} \bar{\omega}_i & \tilde{\omega}_i > \bar{\omega}_i \\ \tilde{\omega}_i & -\bar{\omega}_i \leq \tilde{\omega}_i \leq \bar{\omega}_i, \qquad i = 1,2,3 \\ -\bar{\omega}_i & \tilde{\omega}_i < -\bar{\omega}_i, \end{cases} \tag{19}$$

where the bounding term $\bar{\omega}_i$, $i = 1, 2, 3$ is computed as the difference between the maximum allowed torque and the maximal value the scaled error term might assume, i.e., $\bar{\omega}_i = \bar{u}_i - P_{ii}\pi$. The indirect dependence on the choice of K ensures that the tuned control u_t never exceeds the maximum allowed torque $\bar{u}$.



(a) Attitude modes: (i) stabilizing, (ii) rest-to-rest, (iii) nadir-pointing, and (iv) target-pointing.

(b) Star tracker measurement z_s.

(c) Gyroscope measurement z_g.

Figure 3: (a) s/c-attitude modes exemplified around the Itokawa asteroid, (b) gyroscope measurement, and (c) star tracker measurement geometry.

5. Evaluation

We conducted the evaluation in a custom written software, which simulates the solar system bodies as well as rigid body kinematics and dynamics using a fourth order Runge-Kutta integrator. The attitude propagation is based on Eqs. (10) and (11), where the inertia tensor is $I = diag(I_1, I_2, I_3)$, with $I_1 = 3305.49\,kgm^2$, $I_2 = 4653.63\,kgm^2$, and $I_3 = 4814.78\,kgm^2$. For evaluation, we chose an orbit around an asteroid with the gravitational parameter $2.342520648\,m^3s^{-2}$, which is orbiting with a semi-major axis of 3 AU around the Sun. The orbital elements defining the orbit of the s/c around the asteroid are: semi-major axis 0.00000000235 AU $\approx 0.34\,km$, inclination 0.0°, eccentricity 0.1, ascending node 0.0°, periapsis 114.6°, and longitude 57.3°. The simulation operated with a timedelta $dt = 60\,s$ per step. The initial attitude in the evaluation is $R_{ba}(0) = I$ in the **T1/T2** scenarios and varied in the **T3** scenario. Of the possible maneuvers shown in Fig. 3a, we evaluate the following ones under different conditions:

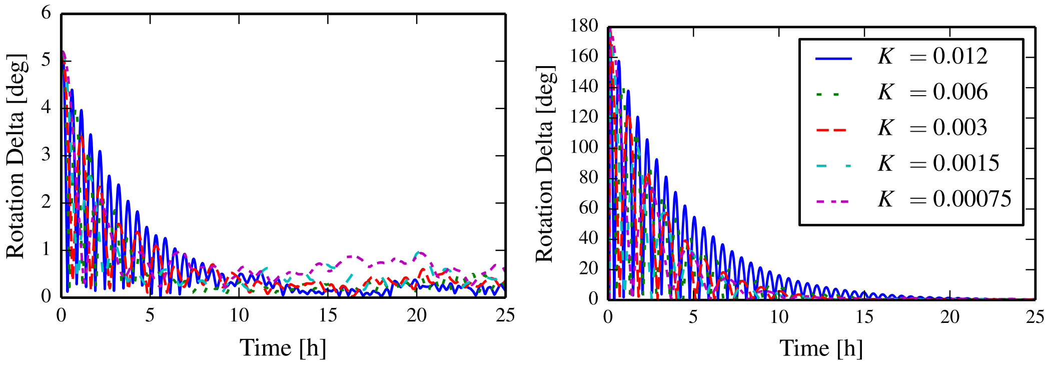
T1 *Rest-to-rest*: From a resting state with $\omega(0) = 0$, we try to reach a constant setpoint R_{da}.

- **T1A** The setpoint (SP) is close to the initial state $R_{da} = I \boxplus [3°, 3°, 3°]^T$, and
- **T1B** the setpoint is exactly 180° rotated with respect to the initial state $R_{da} = diag(-1, -1, 1)$.

T2 *Stabilization*: From an initial angular velocity $\omega(0) = [\pi/2, \pi/2, \pi/2]^T$, we try to reach a constant setpoint R_{da}.

- **T2A** Again, the setpoint is close to the initial state $R_{da} = I \boxplus [3°, 3°, 3°]^T$, and
- **T2B** the setpoint is exactly 180° rotated with respect to the initial state $R_{da} = diag(-1, -1, 1)$.

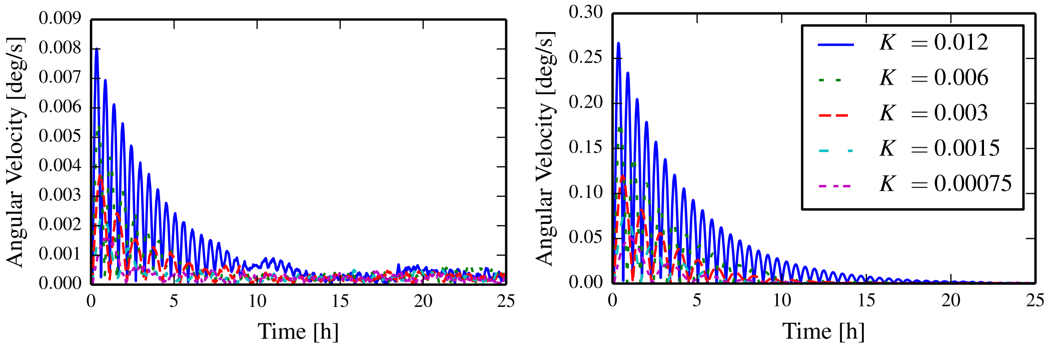
T3 *Nadir-pointing*: From a resting state $\omega(0) = 0$, we try to follow a dynamic setpoint $R_{da}(t)$, i.e., we want to orient our sensors towards the center of mass of the celestial object to be observed. As the setpoint is determined by the position of the s/c and the asteroid, we now change the initial attitude of the s/c to achieve the desired initial position deltas.



(a) T1A: SP close to the initial attitude.

(b) T1B: SP 180° rotated wrt. to the initial attitude.

Figure 4: T1A/B: 2-norm of rotation deltas for rest-to-rest maneuvers. (Note the different scalings of the Y-axes.)



(a) T1A: SP close to the initial attitude.

(b) T1B: SP 180° rotated wrt. to the initial attitude.

Figure 5: T1A/B: 2-norm of angular velocities for rest-to-rest maneuvers. (Note the different scalings of the Y-axes.)

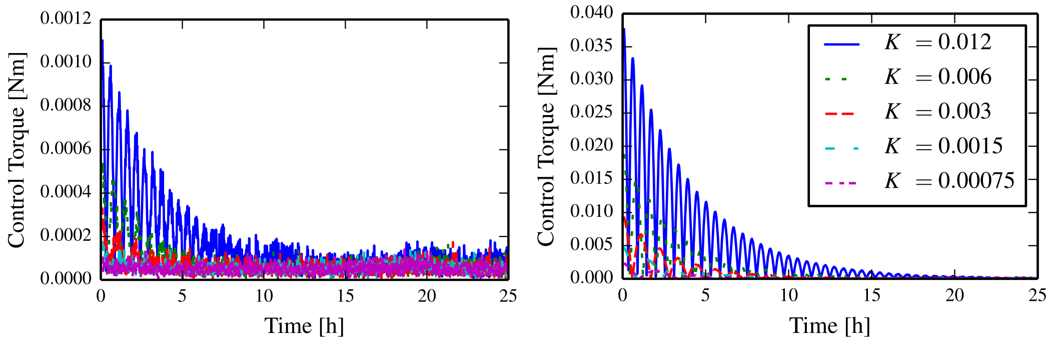
T3A Here, the initial state is modified to differ from the first setpoint setting $R_{ba}(0) = R_{da}(0) \boxplus [3°, 3°, 3°]^T$, and

T3B the initial state is exactly 180° rotated with respect to the initial setpoint $R_{ba}(0) = R_{da}(0)\, diag(-1, -1, 1)$.

We evaluated these scenarios with different gain scaling factors $K_{ii} = 0.00075, 0.0015, 0.003, 0.006, 0.012$, applied uniformly over all axes i. For simplicity, the factor will just be denoted K in the plots in Figs. 4–10. The rotation deltas in Figs. 4–10 are computed in every epoch t as the 2-norm of the current attitude error in vector space

$$\Delta R(t) = \|R_{ba}(t) \boxminus R_{da}(t)\|. \tag{20}$$

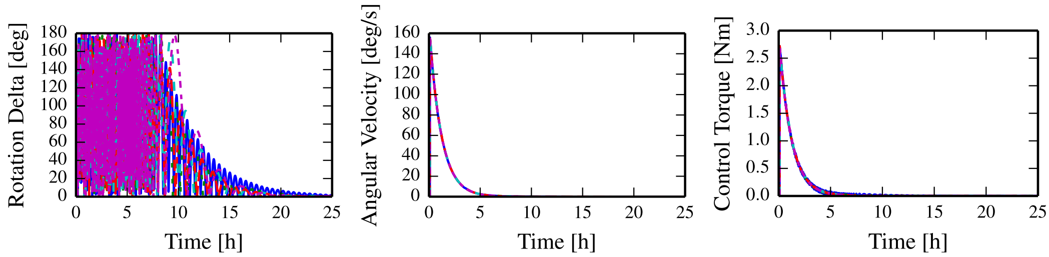
Thus, the error states the angle, which has to be carried out about a unique rotation axis to rotate $R_{ba}(t)$ onto $R_{da}(t)$. Similarly, the angular velocities and the control torques are shown in the plots as the 2-norm over all axes.



(a) T1A: SP close to the initial attitude.

(b) T1B: SP 180° rotated wrt. to the initial attitude.

Figure 6: T1A/B: 2-norm of spent control torques for rest-to-rest maneuvers. (Note the different scalings of the Y-axes.)

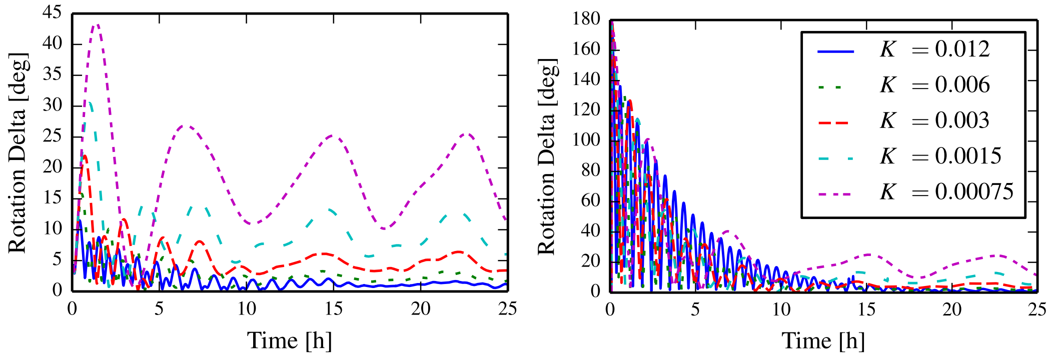


(a) 2-norm of rotation deltas.

(b) 2-norm of angular vel.

(c) 2-norm of spent torques.

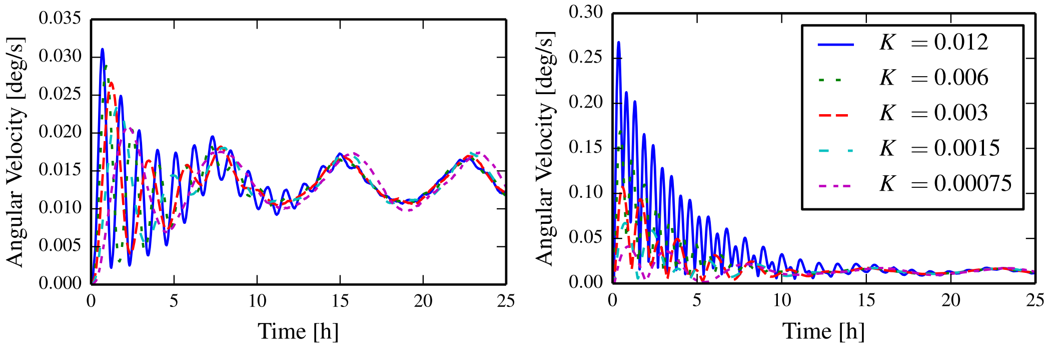
Figure 7: T2A: Stabilizing with a setpoint close to the initial attitude. (Note that the error is bounded by 180°.)



(a) T3A: SP close to the initial attitude.

(b) T3B: SP 180° rotated wrt. to the initial attitude.

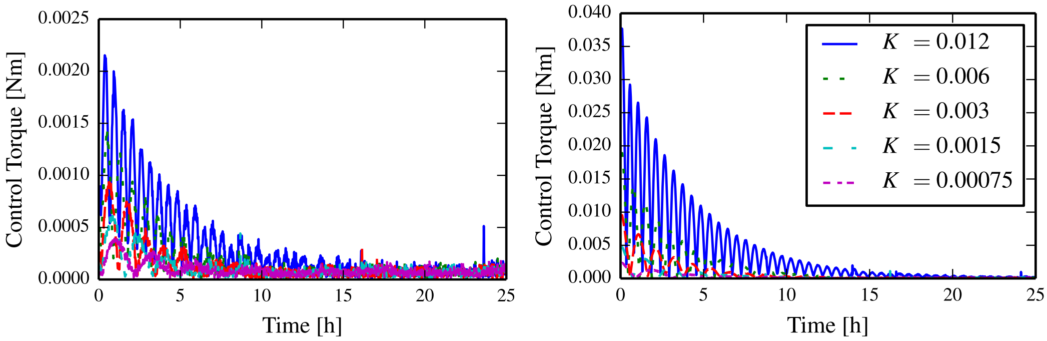
Figure 8: T3A/B: 2-norm of rotation deltas for nadir-pointing maneuvers. (Note the different scalings of the Y-axes.)



(a) T3A: SP close to the initial attitude.

(b) T3B: SP 180° rotated wrt. to the initial attitude.

Figure 9: T3A/B: 2-norm of angular velocities for nadir-pointing maneuvers. (Note the different scalings of the Y-axes.)



(a) T3A: SP close to the initial attitude.

(b) T3B: SP 180° rotated wrt. to the initial attitude.

Figure 10: T3A/B: 2-norm of spent control torques for nadir-pointing maneuvers. (Note the different scalings of the Y-axes.)

In the first scenario T1A, the controller is able to reach the setpoint within a range $< 1°$ in roughly 6h (cf. Fig. 4). The residual error is due to the state transition and measurement noise, which cannot be entirely smoothed by the filter. As expected, a small K is not able to keep track of the setpoint, while a higher K shows better results at the cost of more spent torque (cf. Fig. 6). In the second scenario T1B, the controller shows its robustness by stabilizing at the setpoint in about 15h (cf. Fig. 5). In T1A and T1B, the gain factor has a similar proportional impact on the control torque although the error starts from a much higher level (cf. Fig. 6).

The scenarios T2A and T2B show that the controller successfully stabilizes the s/c from tumbling after 20h (cf. Fig. 7). Starting from a high error the gain factor K is not that important anymore. The controller neatly decelerates over all axes and decreases the spent torque regardless of the gain factor (cf. Figs. 7b and 7c). We do not show plots of T2B here, as the initial pose does not have a high impact compared to the high initial angular velocity. Thus, the results are qualitatively equal to those of T2A. The gain factor $K = 0.006$ seems to be preferable over $K = 0.012$ for the T1 / T2 scenarios, as it yields a comparable performance while applying significantly less control torque.

In the nadir-pointing scenarios T3A and T3B, the gain factor becomes important again (cf. Figs. 8 and 9). It is obvious that a stiff controller is needed for such a demanding maneuver, as the weaker K are not able to reach the setpoint, neither in condition A (cf. Fig. 8a) nor in condition B (cf. Fig. 8b). In this scenario, we furthermore note a constant residual angular velocity, which is essential to follow the dynamic setpoint (cf. Figs. 9a and 9b). The controllers with a higher gain are able to apply higher control torques (cf. Fig. 10), and thus are able to follow the moving setpoint. Hence, for scenario T3 a gain factor of $K = 0.012$ seems to be preferable over $K = 0.006$ as it (i) approaches the dynamic setpoint faster and (ii) is able to maintain a slightly lower residual error.

6. Conclusion

This paper presented a proportional derivative controller working on a manifold encapsulation of direction cosine matrices. The controller uses particle filtered angular velocity and attitude estimates to compute control torques. The proportional term is calculated from the weighted error between two attitudes on the manifold $SO(3)$ which is mapped to the vector space $\mathbb{R}^3$. The derivative term is bounded by a nonlinearity, which depends on the proportional gain to prevent actuator saturation. In a numerical evaluation, we examined its behavior in different maneuvers: (i) stabilizing, (ii) rest-to-rest, and (iii) nadir-pointing. The controller exhibits stable behavior near the setpoint and from instability-prone initial attitudes. Furthermore, it is able to stabilize a s/c rotating about all three axes and can be used for nadir-pointing maneuvers, which involve a dynamic setpoint.

For future work, a combination with an information driven selection [18] of the setpoint with the control approach presented here is of high interest. Furthermore, the controller can be combined with a SLAM-approach [6] to examine possible landing sites by actively reducing uncertainty. This would result in a highly dynamic setpoint, which is not only determined by decisions as well as the motion of the autonomous system, but also by the motion of the observable itself. However, achieving this would allow autonomous systems to actively select dynamic observables and directly come up with a corresponding control.

Acknowledgments

This work was supported by the German Aerospace Center (DLR) with financial means of the German Federal Ministry for Economic Affairs and Energy (BMWi), project “KaNaRiA” (grant No. 50 NA 1318) and project “EnEx-CAUSE” (grant No. 50 NA 1505).

References

[1] S Bharadwaj, M Osipchuk, K D Mease, and F C Park. Geometry and inverse optimality in global attitude stabilization. *Journal of Guidance, Control, and Dynamics*, 21(6):930–939, 1998.

[2] S P Bhat and D S Bernstein. A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems & Control Letters*, 39(1):63–70, 2000.

[3] J D Bošković, S Li, and R K Mehra. Robust adaptive variable structure control of spacecraft under control input saturation. *Journal of Guidance, Control, and Dynamics*, 24(1):14–22, 2001.

[4] F Bullo and R M Murray. Proportional derivative (pd) control on the euclidean group. In *European Control Conference*, volume 2, pages 1091–1097, 1995.

[5] N A Chaturvedi, A K Sanyal, and H N McClamroch. Rigid-body attitude control. *IEEE Control Systems*, 31(3):30–51, 2011.

[6] J Clemens, T Reineking, and T Kluth. An evidential approach to SLAM, path planning, and active exploration. *International Journal of Approximate Reasoning*, 73:1–26, 2016.

[7] A H J De Ruiter. Adaptive spacecraft attitude control with actuator saturation. *Journal of Guidance, Control, and Dynamics*, 33(5):1692–1696, 2010.

[8] A Doucet, N de Freitas, and N Gordon. *Sequential Monte Carlo methods in practice.* Springer Science & Business Media, 2001.

[9] N Fairfield, G Kantor, and D Wettergreen. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1):3–22, 2007.

[10] J R Forbes. Direction-cosine-matrix-based attitude control subject to actuator saturation. *IET Control Theory & Applications*, 9(11):1653–1661, 2015.

[11] G González Peytaví, J Clemens, D Nakath, A Probst, R Förstner, K Schill, and B Eissfeller. Autonomous orbit navigation for a mission to the asteroid main belt. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation, 2015.

[12] R Hartley, J Trumpf, Y Dai, and H Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013.

[13] C Hertzberg, R Wagner, U Frese, and L Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[14] A Iserles, H Z Munthe-Kaas, S P Nørsett, and A Zanna. Lie-group methods. *Acta Numerica 2000*, 9:215–365, 2000.

[15] T Lee. Exponential stability of an attitude tracking control system on SO(3) for large-angle rotational maneuvers. *Systems & Control Letters*, 61(1):231–237, 2012.

[16] C G Mayhew, R G Sanfelice, and A R Teel. On quaternion-based attitude control and the unwinding phenomenon. In *Proceedings of the 2011 American Control Conference*, pages 299–304. IEEE, 2011.

[17] R M Murray, Z Li, and S S Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[18] D Nakath, C Rachuy, J Clemens, and K Schill. Optimal rotation sequences for active perception. In *Proc. SPIE*, volume 9872, pages 987204-1–987204-13, 2016.

[19] M A C Perryman, L Lindegren, J Kovalevsky, E Hoeg, U Bastian, P L Bernacca, M Crézé, F Donati, M Grenon, M Grewing, et al. The HIPPARCOS catalogue. *Astronomy and Astrophysics*, 323, 1997.

[20] A Probst, G González Peytaví, D Nakath, A Schattel, C Rachuy, P Lange, J Clemens, M Echim, V Schwarting, A Srinivas, K Gadzicki, R Förstner, B Eissfeller, K Schill, C Büskens, and G Zachmann. KaNaRiA: Identifying the challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies. In *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel.* International Astronautical Federation, 2015.

[21] S Thrun, W Burgard, and D Fox. *Probabilistic Robotics.* MIT Press, Cambridge, MA, 2005.

[22] B Wie and P M Barba. Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):360–365, 1985.

Optimal Rotation Sequences for Active Perception

David Nakath[a], Carsten Rachuy[a], Joachim Clemens[a], and Kerstin Schill[a]

[a]Cognitive Neuroinformatics, University of Bremen, Germany

ABSTRACT

One major objective of autonomous systems navigating in dynamic environments is gathering information needed for self localization, decision making, and path planning. To account for this, such systems are usually equipped with multiple types of sensors. As these sensors often have a limited field of view and a fixed orientation, the task of active perception breaks down to the problem of calculating alignment sequences which maximize the information gain regarding expected measurements. Action sequences that rotate the system according to the calculated optimal patterns then have to be generated. In this paper we present an approach for calculating these sequences for an autonomous system equipped with multiple sensors. We use a particle filter for multi-sensor fusion and state estimation. The planning task is modeled as a Markov decision process (MDP), where the system decides in each step, what actions to perform next. The optimal control policy, which provides the best action depending on the current estimated state, maximizes the expected cumulative reward. The latter is computed from the expected information gain of all sensors over time using value iteration. The algorithm is applied to a manifold representation of the joint space of rotation and time. We show the performance of the approach in a spacecraft navigation scenario where the information gain is changing over time, caused by the dynamic environment and the continuous movement of the spacecraft.

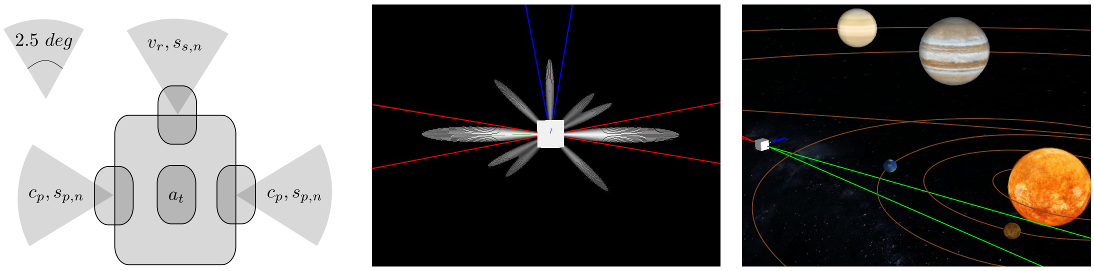
Keywords: Markov decision process, manifold representation, particle filter, value iteration, multi-sensor fusion, optimal decision making, active perception, information gain

1. INTRODUCTION

Applications of autonomous navigation concepts have already been successfully implemented by missions like Hayabusa I[1] and Rosetta.[2] Long term autonomy will become even more important in consecutive missions like Haybusa II.[3] This also holds for autonomous navigation of landing systems and ground exploration systems like Mascot[4] and Enceladus Explorer (EnEx).[5,6] In addition, prospective mission scenarios like KaNaRiA[7] explicitly build on the prior condition of autonomous navigation. In all the missions and scenarios mentioned, autonomous decision making is an essential prerequisite due to the fact that ground-based decisions is expensive and take too much transmission time in certain situations. This type of navigation in deep space is challenging as autonomous systems have to cope with complex dynamic environments without the support of the Global Navigation Satellite Systems (GNSS) or the Deep Space Network (DSN). Instead, they have to rely on different kinds of sensors with a limited field of view, delivering information about certain aspects of the environment. Under this conditions, the orientation of an autonomous system can be optimized over time by trading off expected dynamic information influx and movement costs, thus improving localization performance while minimizing effort. Building on prior work on active perception,[8–10] active exploration[11] and autonomous navigation for space missions,[5,6,12] we present an approach for active perception of an autonomously navigating spacecraft based on optimal rotation sequences.

We start with outlining the autonomous deep space navigation concept in Sect. 2, where we describe the observables and the computation of the respective sensor measurements. In Sect. 3, the rotation planning algorithm is presented, which comprises a reward and value function tailored to the different observables as well as a planning step. Subsequently, the multi-sensor fusion algorithm used for state estimation based on a particle filter is described in Sect. 4. In Sect. 5, we present the result of our approach and show its benefits compared to other strategies for choosing rotation sequences. Finally, the paper concludes with a summary and outlook in Sect. 6.

E-mail: {dnakath,rachuy,jclemens,kschill}@informatik.uni-bremen.de



(a) Sensor configuration of the spacecraft: interferometer (v_r), Sun sensor ($s_{s,n}$), planet sensors ($c_p, s_{p,n}$), and star tracker (a_t).

(b) The spacecraft in the simulation environment, with cones indicating the FOV of the respective sensor and the reward functions of the viewing directions.

(c) Overview of the simulation environment, showing the solar system.[*]

Figure 1: Sensor configuration of the spacecraft and the corresponding visualization in the simulation environment.

2. AUTONOMOUS DEEP SPACE NAVIGATION

To meet the challenges of autonomous navigation posed by space missions like the ones described above, we build on the instrumentation set presented in Ref. 12. It combines the solar navigation method introduced in Ref. 13 with astrometric measurements of planetary bodies. In addition, we use a star tracker to determine the attitude of the spacecraft. This set of instruments is tailored to the observables Sun, planets, and stars with respect to the spacecraft's current state. All sensors are assumed to have a field of view (FOV) of 2.5 degrees, while the star tracker has a FOV of 1 degree. However, since the star tracker can always observe enough stars despite of its small FOV, it provides a measurement at every time step. The sensors are arranged as depicted in Fig. 1a. We further assume that the spacecraft can change its attitude in each direction. The corresponding attitude control system, which is usually realized with small thrusters and reactions wheels, is abstracted by allowing omnidirectional attitude changes with a limited $l2$-norm of the angular velocity of $1°/s$.

The state space of the spacecraft is defined as $\mathcal{X} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 = SE(3) \times \mathbb{R}^3$, where $SO(3)$ denotes the 3-dimensional rotation group, which is the set of all rotations in 3-dimensional Euclidean space $\mathbb{R}^3$, while $SE(3)$ denotes the 3-dimensional special Euclidean group, i.e., the joint space of position and attitude in $\mathbb{R}^3$. The third component is the velocity space of the spacecraft. The state $x_t \in \mathcal{X}$ at time t is given by

$$x_t = \begin{pmatrix} r_t \\ q_t \\ \dot{r}_t \end{pmatrix} = \begin{pmatrix} r_t \\ q_t \\ v_t \end{pmatrix}, \tag{1}$$

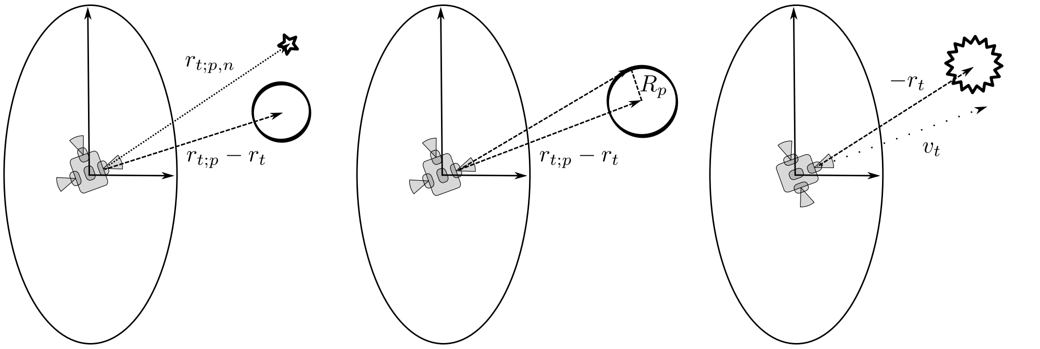
where r_t denotes the position vector, q_t the rotation quaternion which states the attitude of the spacecraft, and $\dot{r}_t = v_t$ the velocity vector.

The radial velocity of the spacecraft can be observed by measuring the frequency shift of the solar optical spectra with a resonance scattering interferometer (see Fig. 2c). The observation $z_{t;v_r}$ is modeled by projecting the linear velocity v_t onto the position vector r_t and a subsequent normalization

$$z_{t;v_r} = \frac{r_t \cdot v_t}{||r_t||} + \varepsilon_v. \tag{2}$$

The sensor-specific noise ε_v is assumed to be additive and normally distributed with $\varepsilon_v \sim \mathcal{N}(0, \sigma_v^2)$ and standard deviation σ_v. The position of the sensor in the spacecraft body-fixed frame can be ignored for this and all other

[*] The planet textures used here are provided under CC BY 3.0 license by http://solarsystemscope.com/nexus/textures (accessed at 2016-02-10).



(a) Sun-star $z_{t;s_{s,n}}$/ planet-star angle $z_{t;s_{p,n}}$ measurement.

(b) Planet chord angle measurement $z_{t;c_p}$.

(c) Interferometer measurement $z_{t;v_r}$.

Figure 2: Exemplary uncertainty ellipses relating sensor type and corresponding measurement geometries.

sensors, because it is negligibly small compared to the position of the spacecraft in the heliocentric frame (few centimeters vs. millions of kilometers). However, the attitude of the sensors is considered when determining the visibility of a celestial object, i.e., whether it is in the FOV.

The planet and Sun sensors use cameras to measure the angle between the center of a particular celestial body and known stars in the FOV (see Fig. 2a). The planet-star and Sun-star angle measurements between the n-th observed star and the planet p or the Sun s are denoted as $z_{t;s_{p,n}}$ and $z_{t;s_{s,n}}$ respectively. Both measurements are generated by computing the unit vector $r_{t;s_{p,n}}$ (or $r_{t;s_{s,n}}$ respectively) to the star[†] and subsequently obtaining the measurement from the position r_t of the spacecraft with

$$z_{t;s_{p,n}} = \cos^{-1} \frac{(r_{t;p} - r_t) \cdot r_{t;p,n}}{||r_{t;p} - r_t||} + \varepsilon_{s_{p,n}}, \tag{3}$$

$$z_{t;s_{s,n}} = \cos^{-1} \frac{-r_t \cdot r_{t;s,n}}{||r_t||} + \varepsilon_{s_{s,n}}, \tag{4}$$

where $r_{t;p}$ denotes the position of planet p, while the position of the Sun is zero, since it is located in the center of the heliocentric navigation frame. The noise terms $\varepsilon_{s_{p,n}}$ and $\varepsilon_{s_{s,n}}$ are additive and normally distributed.

In contrast to the Sun sensor, the planet sensors also provide an angular measurement of the size of the planetary chord (see Fig. 2b). We compute the corresponding measurement $z_{t;c_p}$ of planet p from its radius R_p, its position $r_{t;p}$, and the position r_t of the spacecraft by

$$z_{t;c_p} = \tan^{-1} \frac{R_p}{||r_{t;p} - r_t||} + \varepsilon_{c_p}, \tag{5}$$

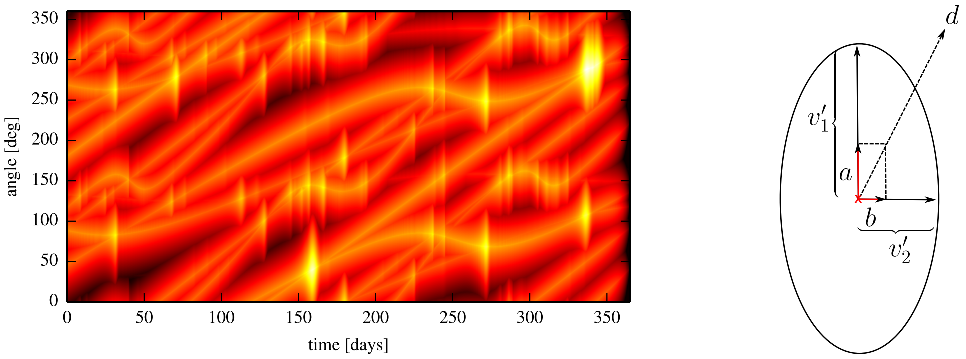
where ε_{c_p} again corresponds to additive and normally distributed noise.

The star tracker provides the attitude quaternion q_t with additional noise as a measurement to the spacecraft

$$a_t = q_t \boxplus \varepsilon_a, \tag{6}$$

where ε_a is sampled from a multi-variate normal distribution $\mathcal{N}(0, \Sigma_a)$ with zero mean and covariance Σ_a. The $\boxplus$-operator[14] used here is in general defined as $\boxplus : \mathcal{S} \times \mathbb{R}^n \to \mathcal{S}$. It encapsulates a globally consistent representation of an arbitrary manifold state space $\mathcal{S}$ using a local vector space $\mathbb{R}^n$, where n corresponds to the number of degrees of freedom in $\mathcal{S}$. This allows to use standard (sensor-fusion) algorithms on $\mathcal{S}$ without

[†] Since the distance to the star is in the magnitude of light years, its actual position can be replaced by the corresponding unit vector.



(a) Value function $\hat{V}(\hat{x}_t)$ for the reward shown in Fig. 4f.

(b) Uncertainty ellipse with scaled eigenvectors v_1' and v_2', a measurement direction d and resulting value vectors a, b.

Figure 3: Value function for all sensors and uncertainty ellipse.

the need for representation-specific modifications. In this paper, the 3-dimensional rotation group $SO(3)$ is part of the state space $\mathcal{X}$, which is represented by a rotation quaternion (see Eq. (1)). This representation is over-determined, since quaternions in general have four parameters, while there are three degrees of freedom in $SO(3)$. However, a representation of the minimal dimension, like Euler angles, would have multiple disadvantages including singularities and discontinuities. The $\boxplus$-operator overcomes the limitations of both representations. For example in Eq. (6), it is used to add local changes expressed in $\mathbb{R}^3$, i.e., multi-variate normally distributed noise with zero mean, to a globally consistent representation, i.e., a rotation quaternion. More details on this topic can be found in Ref. 14.

3. PLANNING OPTIMAL ROTATION SEQUENCES

The task of finding optimal rotation sequences for localization in a dynamic environment based on noisy measurements is modeled as a Markov decision process (MDP).[15,16] The goal is to find the best action $u \in \mathcal{U}$ for a given state x_t, where $\mathcal{U}$ is set of all actions. Since we only aim on optimizing the rotation of the spacecraft, we only take actions into account that alter the spacecraft orientation and therefore have no impact on the trajectory. The best action sequence is the one that maximizes expected cumulated discounted reward

$$E\left(\sum_{t=1}^{\infty} \gamma^{t-1} v(x_{t-1}, u_t)\right), \tag{7}$$

where the *reward function* $v : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ quantifies the reward of being in a state x_{t-1} and executing action u_t. The discount rate $0 \leq \gamma < 1$ weights immediate rewards higher and thus guarantees convergence for all time horizons.[17] We have to consider the expectation of the cumulated reward because of the stochastic nature of the Bayesian motion model used here (see Sect. 4.1). For the actual computation of the expected cumulative reward, a value function $V : \mathcal{X} \to \mathbb{R}$ is used, which is computed recursively using value iteration[18]

$$V(x) = \max_{u \in \mathcal{U}} \left(v(x, u) + \gamma \int_{\mathcal{X}} V(x')\, p(x'|x, u) dx' \right). \tag{8}$$

In order to make planning tractable, we discretize the state space in attitude and time. For each time point, the position is given by assuming that the spacecraft continues flying on its orbit, which can be predicted for a sufficiently small time span using the motion model (see Sect. 4.1). The discretized state space is denoted as $\hat{\mathcal{X}}$

and Eq. (8) can be rewritten as

$$\hat{V}(\hat{x}) = \max_{u \in \mathcal{U}} \left(v(\hat{x}, u) + \gamma \sum_{\hat{x}' \in \hat{\mathcal{X}}} \hat{V}(\hat{x}')\, p(\hat{x}'|\hat{x}, u) \right). \tag{9}$$

Note that $p(\hat{x}'|\hat{x}, u)$ is zero for all $\hat{x}'$ that are not in the subsequent time slice of $\hat{x}$, since the spacecraft cannot move backward on the orbit. As a consequence, $\hat{V}(\hat{x})$ can be computed efficiently in one iteration beginning at the planning horizon backward to the current time slice. Fig. 3a[‡] shows the value function in the discrete planning space, while the corresponding reward function is described in the following.

The reward of a specific state $\hat{x}_t$ and action u is defined as the weighted sum of the reward of the different sensors and the costs of the action

$$v(\hat{x}_t, u) = \|u\| f_u + \sum_{y \in Y} v_y(\hat{x}_t) f_y, \tag{10}$$

where $v_y : \hat{\mathcal{X}} \to \mathbb{R}$ is the reward function for sensor y, f_y is a sensor-specific weighting factor, Y is the set of all sensors, and f_u is the (negative) cost factor for the action, i.e., the angle difference.

The interferometer information reward function $v_v(\hat{x}_t)$ is defined by the product of the probability $p(o_{t;v}|\hat{x}_t)$ of making an observation with this sensor and the expected influence on the uncertainty ellipse e_v (see Fig. 4a)

$$v_v(\hat{x}_t) = p(o_{t;v}|\hat{x}_t) e_v. \tag{11}$$

The reward function $v_c(\hat{x}_t)$ of the planet chord angle is given by the product of the probability $p(o_{t;p}|\hat{x}_t)$ of observing planet p, the expected influence e_p, and the expected chord angle $\bar{c}_p$ (see Fig. 4b)

$$v_c(\hat{x}_t) = \sum_{p \in P} p(o_{t;p}|\hat{x}_t) e_p \bar{c}_p, \tag{12}$$

where the scaling with the expected chord angle accounts for the fact that a larger measured angle is of higher value and P denotes the set of all planets The reward function $v_p(\hat{x}_t)$ of the planet-star measurement is computed according to

$$v_p(\hat{x}_t) = \sum_{p \in P} p(o_{t;p}|\hat{x}_t) e_p^{\perp} \left(1 - \min\left(\frac{\|r_{t;p} - r_t\|}{12\,\mathrm{AU}}, 0.9 \right) f_{dst} \right). \tag{13}$$

The weighting with the distance from spacecraft to planet $\|r_{t;p} - r_t\|$ is used, because nearer planets are of higher value for the spacecraft, since their relative change in position is larger. The distance is scaled to a reasonable range, while the minimum operation ensures that the value does not exceed 0.9. The factor $f_{dst} \in [0, 1]$ controls the amount of the distance-dependent discount (see Fig. 4c and Fig. 4d). In contrast to interferometer and chord angle, the planet-star measurements discriminate the position in perpendicular direction to the line of sight to the planet. Hence, the expected influence with respect to the uncertainty ellipse is computed accordingly (indicated by $\perp$). The calculation of the reward function $v_s(\hat{x}_t)$ for the Sun sensor (Sun-star angle) is equivalent to the planet-star measurement, except for the distance discount, i.e.,

$$v_s(\hat{x}_t) = p(o_{t;s}|\hat{x}_t) e_s^{\perp}, \tag{14}$$

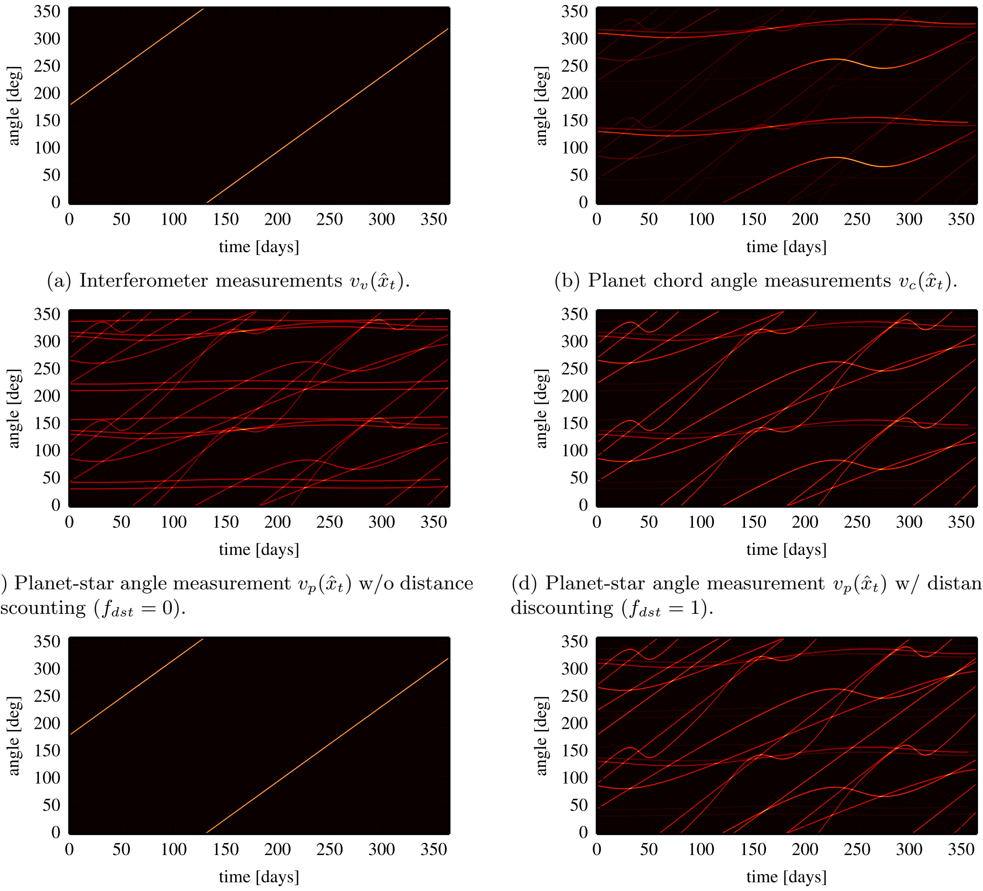
where $p(o_{t;s}|\hat{x}_t)$ is the probability of observing the Sun. The result is shown in Fig. 4e.

In Eqs. (11) – (14), the probability $p(o_{t,y}|\hat{x}_{t,i})$ of making a specific observation in the discretized state space is computed for state $\hat{x}_t$ and the respective sensor y with

$$p(o_{t,y}|\hat{x}_t) = \Phi(\alpha_{best}; \alpha + \Delta\alpha/2, \sigma_\alpha^2) - \Phi(\alpha_{best}; \alpha - \Delta\alpha/2, \sigma_\alpha^2), \tag{15}$$

where Φ is the cumulative distribution function (CDF) of the normal distribution, α is the orientation angle of $\hat{x}_t$, $\Delta\alpha$ is the angular discretization, α_{best} the best orientation for making the observation for y, and σ_α chosen based on the FOV of y.

[‡] For illustration purposes, we demonstrate our algorithm for the 2D case. However, it can simply be applied to 3D.



(a) Interferometer measurements $v_v(\hat{x}_t)$.

(b) Planet chord angle measurements $v_c(\hat{x}_t)$.

(c) Planet-star angle measurement $v_p(\hat{x}_t)$ w/o distance discounting ($f_{dst} = 0$).

(d) Planet-star angle measurement $v_p(\hat{x}_t)$ w/ distance discounting ($f_{dst} = 1$).

(e) Sun-star angle measurement $v_s(\hat{x}_t)$.

(f) All measurements $v(\hat{x}_t, u_t)$ with the correct scaling.

Figure 4: Reward functions for different sensors in the planning space of 360 degrees and 1 year. Note the 180 degrees symmetry of the reward for the planetary sensors, since one of them is mounted at each side of the spacecraft (see Fig. 1a).

The expected influence e_y of sensor y is obtained as follows. Initially, we compute the scaled eigenvectors v_1', v_2' of the uncertainty ellipse from the eigenvectors v_1, v_2 and the respective eigenvalues λ_1, λ_2, which we derive from the estimated covariance matrix of the particle filter (see Sect. 4) and subsequently normalize according to

$$v_1' = \frac{\lambda_1}{\lambda_{max}} v_1, \tag{16}$$

$$v_2' = \frac{\lambda_2}{\lambda_{max}} v_2, \tag{17}$$

with $\lambda_{max} = max(\lambda_1, \lambda_2)$. We calculate the expected influence e_y of the measurement direction vector d by projecting it on the scaled eigenvectors. The $l2$-norm of the resulting vectors a and b then defines the value of the respective sensor information with respect to the shape of the current estimation uncertainty (see Fig. 3b)

$$e_y = \sqrt{(d \cdot v_1')^2 + (d \cdot v_2')^2} = \sqrt{a^2 + b^2}. \tag{18}$$

The expected influence of the Sun and the planet sensor is calculated with respect to the position block of the covariance matrix, while for the interferometer, the velocity block is used. This done due to the fact that the corresponding sensor measurements only affect the respective quantities.

4. MULTI-SENSOR FUSION

To estimate the current state of the spacecraft based on multiple, uncertain measurements, a Bayesian filter can be used. It computes the belief over the state variable x_t, which is an abbreviation for the posterior or target distribution, i.e.,

$$bel(x_t) = p(x_t | z_{1:t}, a_{1:t}). \tag{19}$$

This states the probability distribution of the state x_t conditioned by all preceding sensor measurements $z_{1:t}$ and state transition measurements $a_{1:t}$. As the direct application of the Bayes filter is computationally not feasible in real-world settings, we use a non-parametric implementation of the Bayes filter, the particle filter (PF). It represents $bel(x_t)$ by a finite set of random state samples drawn from the posterior. These samples are referred to as particles and are concrete instantiations of the state x_t. The set of particles is denoted as

$$\mathcal{M}_t := x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]}, \tag{20}$$

where M is the total number of particles. The single particles can be seen as hypotheses of x_t, whose likelihood to be included in the particle set $\mathcal{M}_t$ should be proportional to the respective posterior $bel(x_t)$

$$x_t^{[m]} \sim p(x_t | z_{1:t}, a_{1:t}), \tag{21}$$

with $1 \leq m \leq M$. Based on these essential ideas the basic PF algorithm can be derived (see Algorithm 1). It consists of three main steps:

1. In line 3, a state hypothesis $x_t^{[m]}$ is generated based on the preceding hypothesis $x_{t-1}^{[m]}$ and the state transition measurements a_t. To obtain $x_t^{[m]}$, one has to sample from the proposal distribution. A detailed description of this process can be found in Sect. 4.1.

2. In line 4, the *importance factor* $w_t^{[m]}$ is calculated for every particle $x_t^{[m]}$. This factor is given by the ratio between the target distribution and proposal distribution

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x_t^{[m]} | z_{1:t}, a_{1:t})}{p(x_t^{[m]} | z_{1:t-1}, a_{1:t})} \propto p(z_t | x_t^{[m]}), \tag{22}$$

which is proportional to the measurement likelihood $p(z_t | x_t^{[m]})$. For its computation, specifically tailored sensor models are needed. The description of the model for each particular sensor is given in Sect. 4.2.

3. The for-loop starting from line 7 comprises the resampling step. The algorithm draws M particles from the temporary set $\widetilde{\mathcal{M}}_t$. The probability of drawing the m-th particle is given by its respective weight $w_t^{[m]}$. Thus, by means of resampling, the temporary set $\widetilde{\mathcal{M}}_t$ of cardinality M is transformed into the set $\mathcal{M}_t$ of the same cardinality. The difference is, that the particles in the former set $\widetilde{\mathcal{M}}_t$ are distributed according to $\widetilde{bel}(x_t)$, while the latter set $\mathcal{M}_t$ is distributed according to

$$bel(x_t) \propto p(z_t|x_t^{[m]})\ \widetilde{bel}(x_t). \tag{23}$$

Put in other words, the resampling process chooses the particles which fit best to the current posterior after all measurements have been incorporated in the form of importance weights.

Algorithm 1: Particle Filter.[18, 19]

```
   Data: $\mathcal{M}_{t-1}, a_t, z_t$
1  $\widetilde{\mathcal{M}}_t = \mathcal{M}_t = \emptyset$                                   // initialize
2  for $m = 1$ to $M$ do
3      sample $x_t^{[m]} \sim p(x_t|x_{t-1}^{[m]}, a_{1:t})$                  // sample from motion model
4      $w_t^{[m]} = p(z_t|x_t^{[m]})$                                    // update particle weights
5      add $\left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$ to $\widetilde{\mathcal{M}}_t$                          // temporal particle set
6  end
7  for $m = 1$ to $M$ do
8      draw $x_t^{[m]}$ from $\widetilde{\mathcal{M}}_t$ with probability $\propto w_t^{[m]}$                   // resampling
9      add $x_t^{[m]}$ to $\mathcal{M}_t$                                    // final particle set
10 end
11 return $\mathcal{M}_t$
```

The resampling approach described above in bullet point 3 heavily depends on its execution frequency. If resampling is performed too frequently, the particle set looses its diversity. If – on the contrary – the resampling process is performed too infrequently, the particle set degenerates, i.e., an insufficient number of particles is left in the relevant range of the target distribution. In order to tackle this issue, we employ a technique called adaptive resampling, which performs resampling whenever the particle set does not represent the target distribution well enough anymore. This is the case when the number of effective particles

$$N_{\text{eff}} = \frac{(\sum_{m=1}^{M} w^{[m]})^2}{\sum_{m=1}^{M} (w^{[m]})^2} \tag{24}$$

drops below a certain threshold (e.g., $M/2$).

For retrieving the estimated state, one can assume that the particle set represents a multivariate normal distribution over the state space and estimate its parameters, i.e., mean μ_t and covariance Σ_t, using

$$\mu_t = \sum_{m=1}^{M} w_t^{[m]} x_t^{[m]}, \tag{25}$$

$$\Sigma_t = \sum_{m=1}^{M} w_t^{[m]} (x_t^{[m]} \boxminus \mu_t)(x_t^{[m]} \boxminus \mu_t)^T. \tag{26}$$

The $\boxminus$-operator is the counterpart to the $\boxplus$-operator and in general defined as $\boxminus : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^n$. It calculates the difference between two globally represented states in a manifold state space $\mathcal{S}$ and maps it to a local vector space $\mathbb{R}^n$, where n denotes the number of degrees of freedoms in $\mathcal{S}$ (see Ref. 14 for further details).

4.1 Motion Model

To compute the proposal distribution $p(x_t|z_{0:t-1}, a_{1:t})$ from the prior distribution $p(x_{t-1}|z_{0:t-1}, a_{1:t-1})$, we have to define a motion model of the form $p(x_t|x_{t-1}, a_t)$. If we assume conditional independence between the star tracker measurement a_t and the orbit dynamics we can rewrite the motion model

$$p(x_t|x_{t-1}, a_t) = \int p(\tilde{x}_t|x_{t-1})p(x_t|\tilde{x}_t, a_t)d\tilde{x}_t. \tag{27}$$

This factorization allows us to first compute $p(\tilde{x}_t|x_{t-1})$ according to the orbit dynamics. After adding the transition noise Σ_{x_t}, we can compute $p(x_t|\tilde{x}_t, a_t)$ according to the star tracker measurement. The integration of x_{t-1} from $t-1$ to t is approximated by a 4th order Runge-Kutta method (RK4), where the resulting state is denoted by $\hat{x}_t$. The RK4 has to consider the gravitational influences of Sun and larger planets of the solar system. The corresponding acceleration is computed according to

$$\ddot{r} = -\left(\frac{\mu_s}{||r_{t-1}||^3}r + \sum_{p\in P'} \frac{\mu_p}{||r_p - r_{t-1}||^3}(r_p - r_{t-1})\right), \tag{28}$$

where μ_s denotes the gravitational constant of the Sun, μ_p the gravitational constant of planet p, and r_p the position of planet p. Furthermore, $P' = \{Earth, Mars, Jupiter, Saturn\}$ denotes the set of considered planets. The transition noise is added with the $\boxplus$-operator

$$\tilde{x}_t = \hat{x}_t \boxplus \varepsilon_{x_t}, \tag{29}$$

with $\varepsilon_{x_t} \sim \mathcal{N}(0, \Sigma_{x_t})$, where the covariance $\Sigma_{x_t} \propto \Delta t$ is proportional to the time difference. Subsequently, we sample from $p(x_t|\tilde{x}_t, a_t)$ to incorporate the star tracker measurement a_t using

$$x_t = \begin{pmatrix} \tilde{r}_t \\ a_t \boxplus \varepsilon_a \\ \tilde{v}_t \end{pmatrix}, \tag{30}$$

where $\tilde{r}_t$ and $\tilde{v}_t$ are the position and velocity vectors of $\tilde{x}_t$. The sampled measurement noise is normally distributed according to $\varepsilon_a \sim \mathcal{N}(0, \Sigma_a)$, where Σ_a denotes the covariance of the star tracker measurement.

4.2 Sensor Models

The sensor model is used to compute the importance weight according to line 5 of algorithm 1, it describes the likelihood of a measurement z_t given a state x_t. By assuming independence between the single measurements (see Eqs. (2) – (3)), we can factor the sensor model into

$$p(z_t|x_t) = p(z_{t;v_r}|x_t) \prod_{p\in P} p(z_{t;c_p}|x_t)\ p(z_{t;S_{s,n}}|x_t) \prod_{n\in P_N} p(z_{t;S_{p,n}}|x_t). \tag{31}$$

The likelihood of the planet-star $p(z_{t;s_{p,n}}|x_t)$ and of the Sun-star angle measurement $p(z_{t;s_{s,n}}|x_t)$ are computed in the same manner according to

$$p(z_{t;s_{p,n}}|x_t) = \phi\left(z_{t;s_{p;n}};\ \cos^{-1}\frac{(r_{t;p} - r_t)\cdot r_{t;p,n}}{||r_{t;p} - r_t||},\ \sigma^2_{s_{p,n}}\right), \tag{32}$$

$$p(z_{t;s_{s,n}}|x_t) = \phi\left(z_{t;s_{s;n}};\ \cos^{-1}\frac{-r_t\cdot r_{t;s,n}}{||r_t||},\ \sigma^2_{s_{s,n}}\right), \tag{33}$$

where $\sigma^2_{s_{p,n}}$ and $\sigma^2_{s_{s,n}}$ are the variances of the measurements. The planet chord measurement likelihood is computed according to

$$p(z_{t;c_p}|x_t) = \phi\left(z_{t;c_p};\ \tan^{-1}\frac{R_p}{||r_{t;p} - r_t||}, \sigma^2_{C_p}\right), \tag{34}$$

where and $\sigma_{c_p}^2$ denotes the variance of the measurement. The interferometer measurement likelihood is computed according to

$$p(z_{t;v_r}|x_t) = \phi\left(z_{t;v_r};\ \frac{r_t \cdot v_t}{||r_t||},\ \sigma_v^2\right), \tag{35}$$

where σ_v^2 denotes the variance of the measurement. In the above Eqs. (32) – (35), ϕ denotes the probability density function (PDF) of the normal distribution.[§]

5. RESULTS

For the evaluation we generate sensor measurements according to the descriptions in Sect. 2 and propagate the ground truth position with a RK4 using the differential equation stated in Eq. (28). We test on an orbit with a semimajor axis of 0.8 AU, leaving the other orbital element parameters set to zero, as we restrict our evaluation to the 2D case. We implement three different strategies to compute rotation sequences, which are used by the autonomous agent. The latter uses a controller to maintain the selected attitude. The generated sensor measurements are used by the particle filter described in Sect. 4 for state estimation. The resulting state and uncertainty estimates are in turn used by the agent as a basis for further planning and control.

The agent following the *random* strategy randomly picks a celestial body every week and follows it for the rest of the week. That agent yields the highest cumulative RMS error of all strategies tested here (cf. Fig. 5), the most unstable non-cumulative RMS position error (cf. Fig. 6), while using the most rotations (cf. Fig. 7).

The results can be slightly improved by pursuing a heuristic strategy like the *heuristic* agent does: It also changes the observed celestial object every week, by alternating its focus between the Sun and the planet with the largest chord angle at planning time. This is the same strategy that was used in Ref. 12. Like the *random* strategy it sticks to its decision for a whole week and thus can slightly improve the cumulative RMS position error and the non-cumulative RMS position error, while saving some rotations.

The third agent tries to maximize the expected cumulated reward (see Sect. 3 and Eq. (7)). It uses a planning horizon of one week as a) during this time window the geometry of the environment might change therefore planning into the future is reasonable, b) the planning horizon is small enough to prevent the agent from following a plan based on the current covariance distribution for too long, thus perceiving measurements which are not beneficial anymore, and c) it is on par with the reconsidering frequency of the other agents, thus making the approaches comparable. At the beginning of each planning phase, the agent uses value iteration to calculate the reward function $\hat{V}(\hat{x}_t)$ for the current (discretized) state $\hat{x}_t$ according to Eq. (9). Then, it chooses the best action u'_{t+1} that maximizes the expected reward using

$$u'_{t+1} = \operatorname*{arg\,max}_{u \in \mathcal{U}} \hat{V}\left(\hat{x}_t \boxplus (0\ u\ 0)^T\right), \tag{36}$$

where the $\boxplus$-operator is used to apply the action u to the state $\hat{x}_t$, by changing the attitude and leaving the position and velocity unchanged (see Eq. (1)). After that, we have to find the celestial object p' that makes executing u'_{t+1} valuable, i.e.,

$$p' = \operatorname*{arg\,max}_{p \in P \cup \{Sun\}} \sum_{y \in Y} v_y^p(\hat{x}'_{t+1}) f_y, \tag{37}$$

with $\hat{x}'_{t+1} = \hat{x}_t \boxplus (0\ u'_{t+1}\ 0)^T$ and where $v_y^p(\hat{x}'_{t+1})$ denotes the reward function for sensor y (see Eqs. (11) – (14)) in which only the celestial object p is considered. Finally, the agent follows the selected celestial object p' until the end of the current planning horizon is reached, i.e., for one week.

In general, we can observe from Fig. 5 and Fig. 6 that the particle filter needs about 200 days to stabilize. Within the stabilization phase, our approach is already limiting the rotations (cf. Fig. 7) but is not yet able to improve the positioning performance. However, after stabilization the agent using our approach can find a good trade off between rotations and the information values computed by our models. As a consequence it can compute the best action, which efficiently improves the localization performance and, at the same time, minimizes the costs.

[§] For technical reasons, the logarithmic values are used in the actual implementation.

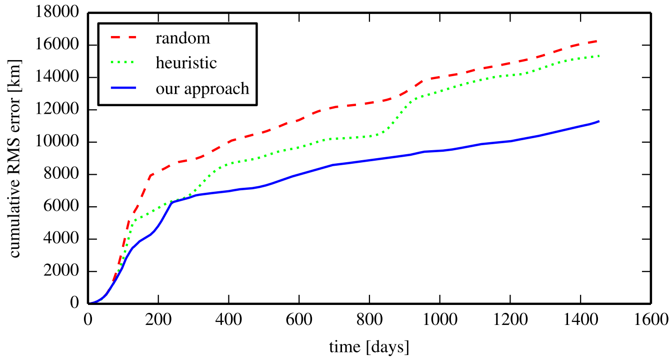


Figure 5: Cumulative root mean squared (RMS) position error over time and averaged over 10 runs.

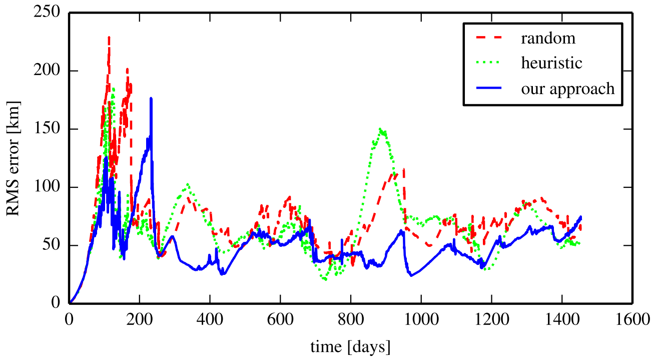


Figure 6: Root mean squared (RMS) position error over time and averaged over 10 runs.

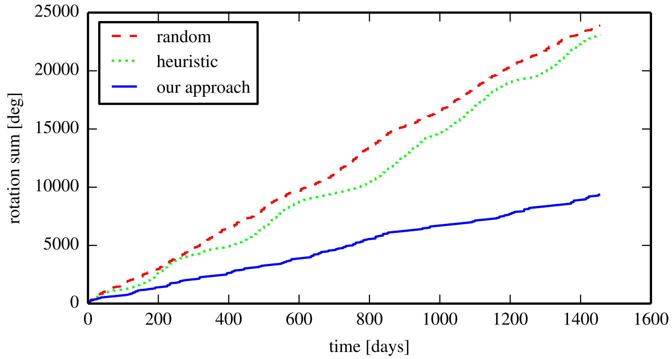


Figure 7: Rotation sum over time and averaged over 10 runs.

6. CONCLUSION

In this paper, we presented an approach for computing optimal rotation sequences based on active perception that takes the dynamic information gain of different sensors and the impact on the estimated uncertainty into account. We applied this approach to autonomous navigation of a spacecraft, for which we provided the required sensors and corresponding models. Furthermore, a particle filter was presented, which is used for multi-sensor fusion and state estimation. For evaluation, we developed a simulation of the spacecraft and its environment, i.e., the solar system with its celestial objects. Our approach is compared to a random strategy, which randomly picked a celestial body to observe, as well as a heuristic strategy, which alternately observes the Sun and the planet with the largest chord angle at planning time. In contrast, our approach chooses the best action based on the expected cumulated reward. This is calculated using value iteration based on the expected impact of the sensor observations on the probability distribution of the filter over time. In comparison to the other strategies, our approach substantially reduces the positioning error, while at the same time minimizing the costs in terms of the number of rotations.

While the results presented here are for planning in 2-dimensional space, the algorithms can be easily extended to 3D. An extensive evaluation in 3D as well as for different orbits and sensor layouts is one of next steps and we are planning to extend our approach by introducing further, not navigation-related constraints like thermal demands and communication. In this context, the automatic optimization of sensor and spacecraft configurations might also be of interest as well. Furthermore, we are planning to apply our approach to active exploration and simultaneous localization and mapping (SLAM) in a space scenario. Here, the rotation has to be optimized together with the position and the estimated map provides additional information regarding the usefulness of a state for different tasks (see Ref. 11 for further details). In general, our approach is not limited to spacecraft navigation. It can also be applied for pose planning of a mobile robot as well as of aerial or underwater vehicles. Another challenge, in a terrestrial or extra-terrestrial context, is information driven action selection for active sensors that can change their orientation, focus, or field of view. The resulting extension of the state and action space requires efficient planning and control algorithms that can cope with the increased dimensionality, for which the approach presented here can serve as a first starting point.

ACKNOWLEDGMENTS

This work was supported by the German Aerospace Center (DLR) with financial means of the German Federal Ministry for Economic Affairs and Energy (BMWi), project “KaNaRiA” (grant No. 50 NA 1318) and project “EnEx-CAUSE” (grant No. 50 NA 1505).

REFERENCES

[1] Kubota, T., Hashimoto, T., Kawaguchi, J., Uo, M., and Shirakawa, K. I., “Guidance and navigation of hayabusa spacecraft for asteroid exploration and sample return mission,” in [*SICE-ICASE, 2006. International Joint Conference*], 2793–2796, IEEE (2006).

[2] Morley, T., Budnik, F., Godard, B., Muñoz, P., and Janarthanan, V., “Rosetta navigation from reactivation until arrival at comet 67P/Churyumov-Gerasimenko,” in [*Proceedings of the 25 th International Symposium on Space Flight Dynamics (ISSFD), Munich, Germany*], (2015).

[3] Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S., “System design of the Hayabusa 2-asteroid sample return mission to 1999 {JU3},” *Acta Astronautica* **91**, 356 – 362 (2013).

[4] Grundmann, J. T., Auster, U., Baturkin, V., Bellion, A., Bibring, J.-P., Biele, J., Boden, R., Bompis, O., Borgs, B., Bousquet, P., Canalias, E., Celotti, L., Cenac-Morthe, C., Cordero, F., Deleuze, M., Evesque, C., Findlay, R., Fredon, S., Glameier, K., Granena, D., Grimm, C., Grott, M., Hamm, V., Hendrikse, J., Hercik, D., Ho, T.-M., Jaumann, R., Krause, C., Kroth, R., Ksenik, E., Lange, C., Lange, M., Mierheim, O., Okada, T., Reill, J., Sasaki, K., Schmitz, N., Sedlmayr, H.-J., Talapina, M., Tangruamsub, S., Termtanasombat, N., Ulamec, S., Wejmo, E., Wrasmann, M., Yoshimitsu, T., Ziach, C., and the MASCOT team, “Mobile asteroid surface scout (MASCOT)-design, development and delivery of a small asteroid lander aboard Hayabusa2,” in [*4th IAA Planetary Defense Conference*], (2015).

[5] Clemens, J. and Reineking, T., "Multi-sensor fusion using evidential SLAM for navigating a probe through deep ice," in [*Belief Functions: Theory and Applications, Third International Conference, BELIEF 2014, Proceedings*], Cuzzolin, F., ed., *Lecture Notes in Computer Science* **8764**, 339–347, Springer International Publishing (2014).

[6] Kowalski, J., Linder, P., Zierke, S., von Wulfen, B., Clemens, J., Konstantinidis, K., Ameres, G., Hoffmann, R., Mikucki, J., Tulaczyk, S., Funke, O., Blandfort, D., Espe, C., Feldmanm, M., Francke, G., Hiecker, S., Plescher, E., Schöngarth, S., Schüller, K., Dachwald, B., Digel, I., Artmann, G., Eliseev, D., Heinen, D., Scholz, F., Wiebusch, C., Macht, S., Bestmann, U., Reineking, T., Zetzsche, C., Schill, K., Förstner, R., Niedermeier, H., and Szumski, A., "Navigation technology for exploration of glacier ice with maneuverable melting probes," *Cold Regions Science and Technology* **123**, 53–70 (Mar. 2016).

[7] Probst, A., González Peytaví, G., Nakath, D., Schattel, A., Rachuy, C., Lange, P., Clemens, J., Echim, M., Schwarting, V., Srinivas, A., Gadzicki, K., Förstner, R., Eissfeller, B., Schill, K., Büskens, C., and Zachmann, G., "KaNaRiA: Identifying the challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies," in [*Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*], International Astronautical Federation (2015).

[8] Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., and Schill, K., "Active sensorimotor object recognition in three-dimensional space," in [*Spatial Cognition IX, International Conference, Spatial Cognition 2014, Proceedings*], Freksa, C., Nebel, B., Hegarty, M., and Barkowsky, T., eds., *Lecture Notes in Computer Science* **8684**, 312–324, Springer International Publishing (2014).

[9] Kluth, T., Nakath, D., Reineking, T., Zetzsche, C., and Schill, K., "Affordance-based object recognition using interactions obtained from a utility maximization principle," in [*Computer Vision-ECCV 2014 Workshops*], *Lecture Notes in Computer Science* **8926**, 406–412, Springer International Publishing (2014).

[10] Reineking, T., Kluth, T., and Nakath, D., "Adaptive information selection in images: Efficient naive bayes nearest neighbor classification," in [*Computer Analysis of Images and Patterns*], 350–361, Springer (2015).

[11] Clemens, J., Reineking, T., and Kluth, T., "An evidential approach to SLAM, path planning, and active exploration," *International Journal of Approximate Reasoning* **73**, 1–26 (2016).

[12] González Peytaví, G., Clemens, J., Nakath, D., Probst, A., Förstner, R., Schill, K., and Eissfeller, B., "Autonomous orbit navigation for a mission to the asteroid main belt," in [*Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*], International Astronautical Federation (2015).

[13] Guo, Y., "Self-contained autonomous navigation system for deep space missions," *AAS/AIAA Space Flight Mechanics Meeting* , 1099–1113 (1999).

[14] Hertzberg, C., Wagner, R., Frese, U., and Schröder, L., "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion* **14**(1), 57–77 (2013).

[15] Howard, R. A., [*Dynamic Programming and Markov Processes*], MIT Press, Cambridge, MA.

[16] Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A., "Acting under uncertainty: discrete Bayesian models for mobile-robot navigation," in [*Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*], **2**, 963–972 (Nov 1996).

[17] Russell, S. and Norvig, P., [*Artificial intelligence: a modern approach*], Prentice Hall (2003).

[18] Thrun, S., Burgard, W., and Fox, D., [*Probabilistic robotics*], MIT Press, Cambridge, MA (2005).

[19] Doucet, A., de Freitas, N., and Gordon, N., [*Sequential Monte Carlo methods in practice*], Springer Science & Business Media (2001).

Multi-Sensor Fusion and Active Perception for Autonomous Deep Space Navigation

David Nakath, Joachim Clemens and Kerstin Schill
Cognitive Neuroinformatics, University of Bremen, Germany
Email: {dnakath,jclemens,kschill}@informatik.uni-bremen.de

***Abstract*—Keeping track of the current state is a crucial task for mobile autonomous systems, which is referred to as state estimation. To solve that task, information from all available sensors needs to be fused, which includes relative measurements as well as observations of the surroundings. In a dynamic 3D environment, the pose of an agent has to be chosen such that the most relevant information can be observed. We propose an approach for multi-sensor fusion and active perception within an autonomous deep space navigation scenario. The probabilistic modeling of observables and sensors for that particular domain is described. For state estimation, we present an Extended Kalman Filter, an Unscented Kalman Filter, and a Particle Filter, which all operate on a manifold state space. Additionally, an approach for active perception is proposed, which selects the desired attitude of the spacecraft based on the knowledge about the dynamics of celestial objects, the kind of information they provide as well as the current uncertainty of the filters. We evaluated the localization performance of the algorithms within a simulation environment. The filters are compared to each other and we show that our active perception strategy outperforms two other information intake approaches.**

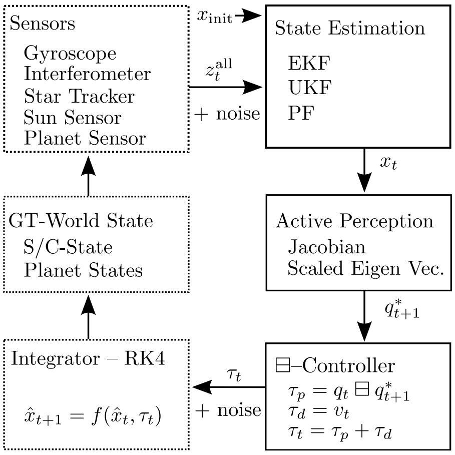


Fig. 1: Overview of the closed-loop system.

I. Introduction

Autonomous systems typically need to continuously estimate their current state to fulfill their designated task. A simple *dead reckoning* approach integrates the equations of motion from a given starting point, but consequently suffers from a cumulating error. To overcome this limitation, the state can be corrected using observations of multiple known objects in the world. This is referred to as multi-sensor fusion, since the information of multiple sensors needs to be combined in order to obtain an estimation of the state. This task is particularly challenging, when the system operates in a dynamic 3D environment, where the objects to be observed change their properties over time. Here, the system has to continuously adapt its pose, or at least its attitude, in order to obtain the most valuable observations, which is commonly referred to as active perception [1]. For bio-inspired scene analysis [2] and in robotic scenarios [3], an expected-entropy-minimization approach has been successfully employed to tackle the active-perception problem in 2D state space. In [4], this approach is extended to an–albeit–discretized and noise-free 3D state space.

Another applicable scenario is the autonomous navigation [5] of a spacecraft during a commercial long-term deep-space mission [6], [7]. Such a system heavily depends on self-sustaining localization and decision making [8] as the round-trip time for control signals is too big and ground observations, e.g., using NASA's deep space network (DSN) [9] or ESA's European Space Tracking (ESTRACK) network [10], are not feasible due to high costs. Thus, the spacecraft has to rely on its own sensors and localize itself by observing celestial objects, like the Sun, planets, and stars, whose dynamics are known. A sensor setup and two sensor fusion algorithms for this kind of navigation approach were presented in [11]. However, the attitude of the spacecraft was neglected and the filters were simply provided with the required observations. An approach for active perception in the given scenario was presented in [12]. It models the planning task as a Markov decision process (MDP) [13], [14], which predicts the movement of the spacecraft and the celestial objects for a certain amount of time. However, the performance was only demonstrated in a 2D environment and only one filter algorithm was used.

In this paper, we overcome the limitations of the previous approaches. We present different algorithms for estimating the full state of the spacecraft in the 3D state space, which we model as position, attitude, and velocity in the heliocentric reference frame. Furthermore, we propose an algorithm for active perception, which calculates the best possible 3D attitude of the spacecraft under consideration of the movements of the celestial objects, the predicted movement of the spacecraft itself, and the impact of the expected observations on the uncertainty of the state estimate.

An overview of our system is shown in Fig. 1. Initially, a state x_{init} is passed into the *state estimation* module, which in turn can make use of either an Extended Kalman Filter (EKF) [15], an Unscented Kalman Filter (UKF) [16], or a Particle Filter (PF) [11], [12] operating on a manifold state representation. The resultant state estimate x_t is subsequently fed into the *active perception* module, which estimates the impact of possible future measurements on the current uncertainty of the estimate, and thus, derives a desired attitude q^*_{t+1} for the next step. Based on this, the *⊟-controller* [17] computes a control torque τ_t, which will be applied noisy in the next state transition step. The transition will be conducted by numerically integrating all states in the world, thus computing a new ground truth world state $\hat{x}_{t+1}$. Certain aspects of the world can then be observed, depending on the current attitude of the agent, as well as the field of view of the particular sensor. These noisy observations are continuously fed back to the state estimation module, thus establishing a closed loop system.

II. State, Motion, and Sensor Models

In the following, we discuss the state representation for the 3D manifold space as well as the motion and sensor models.

A. State Representation

If the state space $\mathcal{X}$ is or contains a manifold space, the representation of the elements has to be chosen with care, as minimal representations suffer from singularities. However, over-parametrized representations of the degrees of freedom in $\mathcal{X}$ can avoid these problems. Therefore, we employ the ⊞-method [16] to enjoy a globally consistent state representation in $\mathcal{X}$ and additionally gain the ability to express small state changes in vector space $\mathbb{R}^n$, where n denotes the number of degrees of freedom in $\mathcal{X}$. In general, the operator $\boxplus : \mathcal{X} \times \mathbb{R}^n \to \mathcal{X}$ adds small changes expressed in $\mathbb{R}^n$ to a state in $\mathcal{X}$, while the operator $\boxminus : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^n$ calculates the difference between two states in $\mathcal{X}$ as a vector in $\mathbb{R}^n$.

The actual state space $\mathcal{S} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 = SE(3) \times \mathbb{R}^3$ is the joint space of the special Euclidean group $SE(3)$ for the pose and a 3D Euclidean space $\mathbb{R}^3$ for the velocity, where $SE(3)$ is composed of a 3D Euclidean space $\mathbb{R}^3$ for the position and the 3D rotation group $SO(3)$ for the attitude. A state element $x_t \in \mathcal{S}$ at time t is given by

$$x_t = \left[r_t^\top q_t^\top v_t^\top\right]^\top , \tag{1}$$

with the position represented by $r_t \in \mathbb{R}^3$, an attitude given as a Direction Cosine Matrix (DCM) $q_t \in \hat{\mathbb{R}}^{3\times3}$, and a velocity represented as $v_t \in \mathbb{R}^3$. Consequently, the corresponding vector space is $\mathbb{R}^9$.

For the state space $\mathcal{S}$, we use the compound ⊞/⊟-operators according to [16]. Our operators $\boxplus_{\mathcal{S}} : \mathcal{S} \times \mathbb{R}^9 \to \mathcal{S}$ and $\boxminus_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^9$ are defined as

$$x \boxplus_{\mathcal{S}} d = \begin{bmatrix} x_r + d_{1:3} \\ x_q \boxplus_{\hat{\mathbb{R}}^{3\times3}} d_{4:6} \\ x_v + d_{7:9} \end{bmatrix}, \quad y \boxminus_{\mathcal{S}} x = \begin{bmatrix} y_r - x_r \\ y_q \boxminus_{\hat{\mathbb{R}}^{3\times3}} x_q \\ y_v - x_v \end{bmatrix} \tag{2}$$

where $x, y \in \mathcal{S}$ are states on the manifold and $d \in \mathbb{R}^9$ represents a small change expressed in the vector space. The position part of x, y is denoted by $x_r, y_r \in \mathbb{R}^3$, and analogously, $x_q, y_q \in \hat{\mathbb{R}}^{3\times3}$ and $x_v, y_v \in \mathbb{R}^3$ are the attitude and velocity parts. Furthermore, $d_{1:3}, d_{4:6}, d_{7:9} \in \mathbb{R}^3$ denote the 1.–3., 4.–6., and 7.–9. elements of d, which corresponds to the change in position, attitude, and velocity, respectively. The operators $\boxplus_{\hat{\mathbb{R}}^{3\times3}} : \hat{\mathbb{R}}^{3\times3} \times \mathbb{R}^3 \to \hat{\mathbb{R}}^{3\times3}$ and $\boxminus_{\hat{\mathbb{R}}^{3\times3}} : \hat{\mathbb{R}}^{3\times3} \times \hat{\mathbb{R}}^{3\times3} \to \mathbb{R}^3$ are the ⊞/⊟-operators for DCMs, the definition of which can be found in [16], [17].

Finally, for probabilistic state estimation, it is important to adequately represent distributions. According to [16], the normal distribution $\mathcal{N}_{\mathcal{X}}$ for a manifold space $\mathcal{X}$ can be defined as

$$\mathcal{N}_{\mathcal{X}}(\mu, \Sigma) = \mu \boxplus \mathcal{N}(0, \Sigma) , \tag{3}$$

with $\mu \in \mathcal{X}$, $\Sigma \in \mathbb{R}^{n\times n}$ and where $\mathcal{N}$ is an ordinary normal distribution on vector space $\mathbb{R}^n$. Hence, the mean is represented globally consistent by a manifold, while the uncertainty is expressed in the vector space around the mean. For a better readability, we omit the space indication of the respective operators in the following given the domain is clearly defined by the arguments.

B. Motion Model

The motion model $p(x_t|u_t, x_{t-1})$ defines the state transition from $t-1$ to t. It takes the form

$$x_t = g(u_t, x_{t-1}) \boxplus \epsilon_t , \qquad \epsilon_t \sim \mathcal{N}(0, R_t), \tag{4}$$

where $u_t \in \mathbb{R}^3$ is a gyroscope measurement, which gives the angular velocity, x_{t-1} is the previous state, and $\epsilon_t \in \mathbb{R}^9$ is additive normally-distributed noise with zero mean and covariance $R_t \in \mathbb{R}^{9\times9}$. The state transition function $g : \mathbb{R}^3 \times \mathcal{S} \to \mathcal{S}$ is defined as

$$g(u_t, x_{t-1}) = x_{t-1} \boxplus \begin{bmatrix} v_{t-1} \\ u_t \\ -a_t \end{bmatrix} \Delta t. \tag{5}$$

Furthermore, $a_t \in \mathbb{R}^3$ is the acceleration caused by celestial bodies according to the orbit dynamics. It is given by

$$a_t = \frac{\mu_{\text{Sun}}}{||r_{t-1}||^3} r + \sum_{p \in \mathcal{P}} \frac{\mu_p}{||p_{t-1} - r_{t-1}||^3} (p_{t-1} - r_{t-1}) , \tag{6}$$

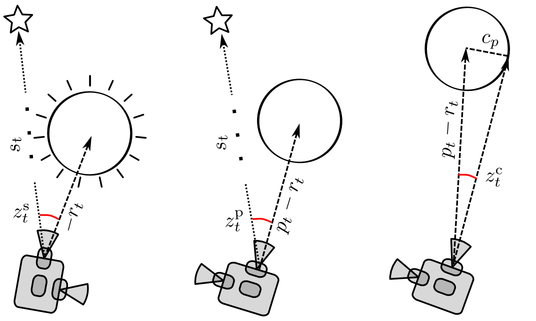
where $\mu_{\text{Sun}} \in \mathbb{R}$ is the gravitational constant of the Sun, $\mu_p \in \mathbb{R}$ and $p_{t-1} \in \mathbb{R}^3$ are the gravitational constant and the position of the planet $p \in \mathcal{P}$, and $\mathcal{P}$ denotes the set of all planets in the Solar system.

The covariance matrix R_t (see (4)) is constructed using

$$R_t = A_t + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & B_t \Delta t & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} , \tag{7}$$

$$A_t = \begin{bmatrix} A_t^r & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & A_t^q & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_t^v \end{bmatrix} , \tag{8}$$

where $A_t \in \mathbb{R}^{9\times9}$ denotes the state transition noise and $A_t^r, A_t^q, A_t^v \in \mathbb{R}^{3\times3}$ are the transition noise of the position, the



(a) Sun-star angle (b) planet-star angle (c) planet chord angle

Fig. 2: Measurement geometries of the different sensors.

attitude, and the velocity, respectively. The matrix $B_t \in \mathbb{R}^{3\times 3}$ is the covariance of the gyroscope's noise.

C. Star Tracker Model

The measurement $z_t^{\mathrm{q}} \in SO(3)$ of the star tracker provides the absolute attitude of the spacecraft represented by a DCM in $\hat{\mathbb{R}}^{3\times 3}$, which is computed by observing stars and comparing them to a star catalog [18]. The measurement likelihood $p(z_t^{\mathrm{q}}|x_t)$ is given by

$$z_t^{\mathrm{q}} = q_t \boxplus \delta_t^{\mathrm{q}}\,, \qquad \delta_t^{\mathrm{q}} \sim \mathcal{N}(0, Q_t^{\mathrm{q}}), \tag{9}$$

where $\delta_t^{\mathrm{q}} \in \mathbb{R}^3$ denotes normally-distributed noise with zero mean and covariance $Q_t^{\mathrm{q}} \in \mathbb{R}^{3\times 3}$. Evaluating the corresponding probability density function (PDF) is not straight forward, since $p(z_t^{\mathrm{q}}|x_t)$ is defined on a manifold space. However, it can be implemented using the $\boxminus$-operator. From (3) and (9) follows

$$\phi_{\hat{\mathbb{R}}^{3\times 3}}(z_t^{\mathrm{q}}; q_t, Q_t^{\mathrm{q}}) = \phi(z_t^{\mathrm{q}} \boxminus q_t; 0, Q_t^{\mathrm{q}})\,, \tag{10}$$

where $\phi_{\hat{\mathbb{R}}^{3\times 3}} : \hat{\mathbb{R}}^{3\times 3} \times \hat{\mathbb{R}}^{3\times 3} \times \mathbb{R}^{3\times 3} \to \mathbb{R}$ is the PDF on $SO(3)$ and $\phi : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{3\times 3} \to \mathbb{R}$ is an ordinary PDF.

D. Interferometer Model

An interferometer measurement $z_t^{\mathrm{v}} \in \mathbb{R}$ gives the radial velocity of the spacecraft relatively to the Sun, which is determined based on the Doppler shift. Its likelihood $p(z_t^{\mathrm{v}}|x_t)$ is defined as

$$z_t^{\mathrm{v}} = h^{\mathrm{v}}(x_t) + \delta_t^{\mathrm{v}}\,, \qquad \delta_t^{\mathrm{v}} \sim \mathcal{N}(0, Q_t^{\mathrm{v}}), \tag{11}$$

where the sensor noise $\delta_t^{\mathrm{v}} \in \mathbb{R}$ is normally-distributed with zero mean and covariance $Q_t^{\mathrm{v}} \in \mathbb{R}$, which is simply the squared standard deviation of the measurement. The measurement function $h^{\mathrm{v}} : \mathcal{S} \to \mathbb{R}$ projects the velocity on the position vector, since the Sun defines the origin of our navigation frame. This results in

$$h^{\mathrm{v}}(x_t) = \frac{v_t \cdot r_t}{\|r_t\|}\,. \tag{12}$$

E. Sun Sensor Model

The Sun sensor is basically an optical camera that observes the Sun in front of a background of stars (see Fig. 2a). Each measurement $z_t^{\mathrm{s}} \in \mathbb{R}$ gives the angle between the Sun and one star. The likelihood $p(z_t^{\mathrm{s}}|x_t)$ takes the form

$$z_t^{\mathrm{s}} = h^{\mathrm{s}}(x_t) + \delta_t^{\mathrm{s}}\,, \qquad \delta_t^{\mathrm{s}} \sim \mathcal{N}(0, Q_t^{\mathrm{s}}), \tag{13}$$

where the sensor noise $\delta_t^{\mathrm{s}} \in \mathbb{R}$ is normally-distributed with zero mean and covariance $Q_t^{\mathrm{s}} \in \mathbb{R}$. The latter is the squared standard deviation of the measurement itself. The measurement function $h^{\mathrm{s}} : \mathcal{S} \to \mathbb{R}$ is defined as

$$h^{\mathrm{s}}(x_t) = \cos^{-1} \frac{-r_t \cdot s_t}{\|r_t\|}\,, \tag{14}$$

where $s_t \in \mathbb{R}^3$ is the unit vector to the specific star, whose true position is known. Usually, the Sun sensor provides measurements to multiple stars around the Sun. The corresponding joint probability can be obtained by multiplying the single likelihoods, since the measurements are conditionally independent of each other given the state x_t.

F. Planet Sensor Model

The planet sensor works similar to the Sun sensor, but observes planets instead of the Sun (see Fig. 2b). A single measurement $z_t^{\mathrm{p}} \in \mathbb{R}$ gives the angle between one planet and one star. Its likelihood $p(z_t^{\mathrm{p}}|x_t)$ is shown in Fig. 3b and given by

$$z_t^{\mathrm{p}} = h^{\mathrm{p}}(x_t) + \delta_t^{\mathrm{p}}\,, \qquad \delta_t^{\mathrm{p}} \sim \mathcal{N}(0, Q_t^{\mathrm{p}}), \tag{15}$$

where the sensor noise $\delta_t^{\mathrm{p}} \in \mathbb{R}$ is normally-distributed with zero mean and covariance $Q_t^{\mathrm{p}} \in \mathbb{R}$. The corresponding measurement function $h^{\mathrm{p}} : \mathcal{S} \to \mathbb{R}$ is defined as

$$h^{\mathrm{p}}(x_t) = \cos^{-1} \frac{(p_t - r_t) \cdot s_t}{\|p_t - r_t\|}\,, \tag{16}$$

where $p_t \in \mathbb{R}^3$ is the planet's position and $s_t \in \mathbb{R}^3$ is the unit vector to the star.

In addition, the planet sensor measures the angle $z_t^{\mathrm{c}} \in \mathbb{R}$ of the planet chord (see Fig. 2c). The corresponding likelihood $p(z_t^{\mathrm{c}}|x_t)$ is shown in Fig. 3a and takes the form

$$z_t^{\mathrm{c}} = h^{\mathrm{c}}(x_t) + \delta_t^{\mathrm{c}}\,, \qquad \delta_t^{\mathrm{c}} \sim \mathcal{N}(0, Q_t^{\mathrm{c}}), \tag{17}$$

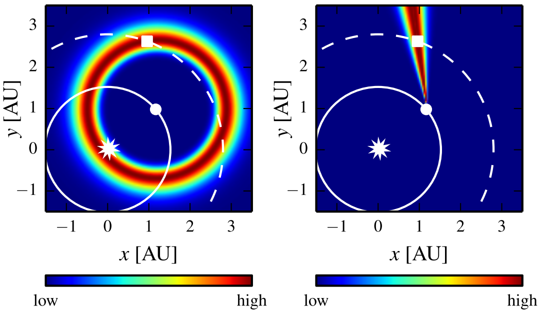
where $\delta_t^{\mathrm{c}} \in \mathbb{R}$ is also normally-distributed noise with zero mean and covariance $Q_t^{\mathrm{c}} \in \mathbb{R}$. The measurement function $h^{\mathrm{c}} : \mathcal{S} \to \mathbb{R}$ is given by

$$h^{\mathrm{c}}(x_t) = \tan^{-1} \frac{c_p}{\|p_t - r_t\|}\,, \tag{18}$$

with $p_t \in \mathbb{R}^3$ being the planet's position and c_p is the planet chord (i.e., radius of the planet). The covariance Q_t^{c} in (17) is computed dynamically for each planet depending on the distance to that planet as well as its size using

$$Q_t^{\mathrm{c}} = \sqrt{(n_1^2 + n_2^2)}\,, \tag{19}$$

$$n_1 = \tan^{-1}\left(\frac{\epsilon \cdot l_{\text{pixel}}}{f_L}\right), \qquad n_2 = \frac{c_p - \sigma_{R_p}}{\cos\left(\sin^{-1}\left(\frac{c_p}{\|p_t - r_t\|}\right)\right)}\,.$$



(a) planet chord angle

(b) planet-star angle

Fig. 3: Measurement likelihoods of a a planet chord angle and a planet to star angle measurement plotted in 2D state space. The position of the Sun is indicated by a star and the orbit of the spacecraft is shown as dashed circle. The spacecraft's position is indicated as a square, while the position of the planet, Mars in this case, is shown as a dot and its orbit as a solid circle (for simplicity, the orbit is assumed to be circular).

Here, l_{pixel}, f_L, and ϵ are the pixel size, focal length, and centroiding capacity of the camera. Following [11], we assume a pixel size of 10 µm, a focal length of 43.3 mm, and a centroiding capacity of 1/10 of a pixel. The planet-specific value σ_{R_p} can also be found in [11].

The planet sensor provides measurements for all planets in the FOV as well as for multiple stars per planet. As for the Sun sensor, the joint probability can be obtained by multiplication, since the measurements are conditionally independent as well.

III. Multi-Sensor Fusion

The most popular approach for calculating the state from a set of measurements is Bayesian filtering [19]. It estimates the state posterior recursively over time using

$$p(x_t|u_{1:t}, z^{\text{all}}_{0:t}) \propto p(x_t|u_{1:t}, z^{\text{all}}_{0:t-1})p(z^{\text{all}}_t|x_t)\,, \tag{20}$$

with $z^{\text{all}}_t = \{z^{\text{q}}_t, z^{\text{v}}_t, z^{\text{s}}_t, z^{\text{p}}_t, z^{\text{c}}_t\}$ and analogously for $z_0 : t^{\text{all}}$. The measurements z^{all}_t are conditionally independent given the state x_t. Thus, their joint likelihood can be factorized according to

$$p(z^{\text{all}}_t|x_t) = \prod_{i\in\mathcal{I}} p(z^i_t|x_t)\,, \tag{21}$$

with $\mathcal{I} = \{\text{q}, \text{v}, \text{s}, \text{p}, \text{c}\}$ being the set of all sensors. The prior is given by

$$\begin{aligned} &p(x_t|u_{1:t}, z^{\text{all}}_{0:t}) = \\ &\qquad \int_{\mathcal{S}} p(x_{t-1}|u_{1:t-1}, z^{\text{all}}_{0:t-1})p(x_t|u_t, x_{t-1})\,dx_{t-1}\,, \end{aligned} \tag{22}$$

where $p(x_{t-1}|u_{1:t-1}, z^{\text{all}}_{0:t-1})$ is the state estimate of the previous time step and $p(x_t|u_t, x_{t-1})$ is the motion model.

There are different practical implementations of (20) and (22), an overview of which can be found in [19]. We consider three of them here, which are the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), and the Particle Filter (PF).

A. Extended Kalman Filter

The EKF [20], [21] estimates the posterior $p(x_t|u_{1:t}, z^{\text{all}}_{0:t})$ in terms of a normal distribution

$$x_t|u_{1:t}, z^{\text{all}}_{0:t} \sim \mathcal{N}(\mu_t, \Sigma_t)\,, \tag{23}$$

with mean $\mu_t \in \mathcal{S}$ and covariance $\Sigma_t \in \mathbb{R}^{9\times9}$. Since our state is an element of a manifold space, the standard algorithm has to be modified. We are using the algorithm proposed in [15], which uses the ⊞-method to implement the prediction and correction step. Since there are some differences to the standard implementation of the EKF, we recall the essential equations here. The prediction step is performed using

$$\bar{\mu}_t = g(u_t, \mu_{t-1})\,, \tag{24}$$

$$\bar{\Sigma}_t = G_t\Sigma_{t-1}G^\top_t + R_t\,, \tag{25}$$

where $\bar{\mu}_t \in \mathcal{S}$, $\bar{\Sigma}_t \in \mathbb{R}^{9\times9}$ are the predicted mean and covariance, g is the state transition function given in (5), and $G_t = \frac{\partial g}{\partial \mu_{t-1}} \in \mathbb{R}^{9\times9}$ denotes the Jacobian of g resulting from the linearization using a first-order Taylor series expansion developed around μ_{t-1}. The Jacobian is given by

$$\frac{\partial g}{\partial \mu_{t-1}} = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & I_{3\times3}\Delta t \\ 0_{3\times3} & \exp -\Delta q_t & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}, \tag{26}$$

where $\Delta q = u_t\Delta_t \in \mathbb{R}^3$ denotes the change in attitude (see (5)). The influence of r_{t-1} on v_t due to (6) is very small because of the large distances, and thus, it can safely be neglected here. The correction step takes the general form

$$K_t = \bar{\Sigma}_t H^\top_t (H_t\bar{\Sigma}_t H^\top_t + Q_t)^{-1}\,, \tag{27}$$

$$d_t = K_t(z_t \boxminus h(\bar{\mu}_t))\,, \tag{28}$$

$$\mu_t = \bar{\mu}_t \boxplus d_t\,, \tag{29}$$

$$\Sigma_t = D_t(I_{9\times9} - K_t H_t)\bar{\Sigma}_t D^\top_t\,, \tag{30}$$

where $z_t \in \mathcal{Z}$ is a particular measurement with measurement space $\mathcal{Z}$, $Q_t \in \mathbb{R}^{n\times n}$ is the covariance of the measurement noise with n being number of degrees of freedom in $\mathcal{Z}$, $K_t \in \mathbb{R}^{9\times n}$ is the Kalman gain, and $d_t \in \mathbb{R}^9$ denotes the state change, i.e., the innovation multiplied by the Kalman gain. Furthermore, $h : \mathcal{S} \to \mathcal{Z}$ denotes the measurement function and $H_t = \frac{\partial h}{\partial \bar{\mu}_t} \in \mathbb{R}^{n\times9}$ denotes its Jacobian resulting from the linearization using first-order Taylor series expansion developed around $\bar{\mu}_t$. Note that, if $\mathcal{Z}$ is a vector space, i.e., $\mathcal{Z} = \mathbb{R}^n$, the ⊟-operator in (28) reduces to an ordinary vector difference. The matrix $D_t = \frac{\partial \bar{\mu}_t \boxplus d_t}{\partial \bar{\mu}_t} \in \mathbb{R}^{9\times9}$ is the Jacobian of (29) derived with respect to μ_t, which is given by

$$\frac{\partial \bar{\mu}_t \boxplus d_t}{\partial \bar{\mu}_t} = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \exp -d_{t;4:6} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}, \tag{31}$$

with $d_{t;4:6} \in \mathbb{R}^3$ being the 4.–6. element of d_t, which corresponds to the change in attitude. That Jacobian is needed because the state uncertainty is defined in the local vector space around the mean (see (3)) and in general we have $\mu \boxplus \mathcal{N}(d, \Sigma) \neq (\mu \boxplus d) \boxplus \mathcal{N}(0, \Sigma)$. Thus, D_t is used in (30) to transform Σ_t with respect to the new reference. In order to apply the correction step to specific measurements, we need to compute the Jacobian H_t for the corresponding measurement functions h. Those are given in the following for all models introduced in Sect. II.

The sensor model of the star tracker is linear with respect to x_t (see (9)). Hence, the Jacobian is simply given by $H_t^{\mathrm{q}} = I_{3\times 3}$.

The measurement function of the interferometer is given in (12). Its Jacobian $H_t^{\mathrm{v}} = \frac{\partial h^{\mathrm{v}}}{\partial x_t} \in \mathbb{R}^{1\times 9}$ is calculated according to

$$\frac{\partial h^{\mathrm{v}}}{\partial x_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{v}}}{\partial r_t} & 0_{1\times 3} & \frac{\partial h^{\mathrm{v}}}{\partial v_t} \end{bmatrix}, \tag{32}$$

$$\frac{\partial h^{\mathrm{v}}}{\partial r_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{v}}}{\partial r_{t;x}} & \frac{\partial h^{\mathrm{v}}}{\partial r_{t;y}} & \frac{\partial h^{\mathrm{v}}}{\partial r_{t;z}} \end{bmatrix}, \tag{33}$$

$$\frac{\partial h^{\mathrm{v}}}{\partial v_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{v}}}{\partial v_{t;x}} & \frac{\partial h^{\mathrm{v}}}{\partial v_{t;y}} & \frac{\partial h^{\mathrm{v}}}{\partial v_{t;z}} \end{bmatrix}, \tag{34}$$

where $\frac{\partial h^{\mathrm{v}}}{\partial r_t}, \frac{\partial h^{\mathrm{v}}}{\partial v_t} \in \mathbb{R}^{1\times 3}$ are the partial derivatives with respect to position and velocity. Furthermore, $r_{t;\circ}, v_{t;\circ} \in \mathbb{R}$, $\circ \in \{x, y, z\}$ refer to the x-, y-, or z-component of the respective vectors and $\frac{\partial h^{\mathrm{v}}}{\partial r_{t;\circ}}, \frac{\partial h^{\mathrm{v}}}{\partial v_{t;\circ}} \in \mathbb{R}$ are the partial derivatives with respect to those components. They are given by

$$\frac{\partial h^{\mathrm{v}}}{\partial r_{t;\circ}} = \frac{v_{t;\circ}}{\|r_t\|} - \frac{r_{t;\circ}(v_t \cdot r_t)}{\|r_t\|^3}, \tag{35}$$

$$\frac{\partial h^{\mathrm{v}}}{\partial v_{t;\circ}} = \frac{r_{t;\circ}}{\|r_t\|}, \tag{36}$$

for all $\circ \in \{x, y, z\}$.

For the Sun sensor's measurement function (see (14)), the Jacobian $H_t^{\mathrm{s}} = \frac{\partial h^{\mathrm{s}}}{\partial x_t} \in \mathbb{R}^{1\times 9}$ results in

$$\frac{\partial h^{\mathrm{s}}}{\partial x_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{s}}}{\partial r_t} & 0_{1\times 3} & 0_{1\times 3} \end{bmatrix}. \tag{37}$$

The partial derivatives with respect to x-, y-, or z-component of r_t are computed using

$$\frac{\partial h^{\mathrm{s}}}{\partial r_{t;\circ}} = \frac{\frac{-r_{t;\circ} r_t \cdot s_t}{\|r_t\|^3} + \frac{s_{t;\circ}}{\|r_t\|}}{\sqrt{1 - \left(\frac{-r_t \cdot s_t}{\|r_t\|}\right)^2}}, \tag{38}$$

for all $\circ \in \{x, y, z\}$ and where $s_{t;\circ}$ denotes the x-, y-, or z-component of the star's position s_t.

Deriving the measurement function of the planet to star measurements (see (16)) gives us the Jacobian $H_t^{\mathrm{p}} = \frac{\partial h^{\mathrm{p}}}{\partial x_t} \in \mathbb{R}^{1\times 9}$ as

$$\frac{\partial h^{\mathrm{p}}}{\partial x_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{p}}}{\partial r_t} & 0_{1\times 3} & 0_{1\times 3} \end{bmatrix}. \tag{39}$$

The partial derivatives with respect to x-, y-, or z-component of r_t are

$$\frac{\partial h^{\mathrm{p}}}{\partial r_{t;\circ}} = \frac{\frac{(r_{t;\circ} - p_{t;\circ})(p_t - r_t) \cdot s_t}{\|p_t - r_t\|^3} + \frac{s_{t;\circ}}{\|p_t - r_t\|}}{\sqrt{1 - \left(\frac{(p_t - r_t) \cdot s_t}{\|p_t - r_t\|}\right)^2}}, \tag{40}$$

for all $\circ \in \{x, y, z\}$ and with $p_{t;\circ}$ referring to the x-, y-, or z-component of the planet's position p_t.

Finally, the measurement function of the planet chord measurements is given in (18). Its Jacobian $H_t^{\mathrm{c}} = \frac{\partial h^{\mathrm{c}}}{\partial x_t} \in \mathbb{R}^{1\times 9}$ is calculated using

$$\frac{\partial h^{\mathrm{c}}}{\partial x_t} = \begin{bmatrix} \frac{\partial h^{\mathrm{c}}}{\partial r_t} & 0_{1\times 3} & 0_{1\times 3} \end{bmatrix}. \tag{41}$$

The partial derivatives with respect to the x-, y-, or z-component of r_t are given by

$$\frac{\partial h^{\mathrm{c}}}{\partial r_{t;\circ}} = \frac{c\,(p_{t;\circ} - r_{t;\circ})}{\|p_t - r_t\|^3 \left(\frac{c^2}{\|p_t - r_t\|^2} + 1\right)}, \tag{42}$$

for all $\circ \in \{x, y, z\}$.

B. Unscented Kalman Filter

The UKF [22] also estimates the state in terms of a normal distribution (see (23)). However, in contrast to the EKF, it uses a set of sigma points instead of a Taylor series expansion in order to linearize the motion and sensor models. Thus, there is no need to provide the analytic derivations for the respective models, which comes at the costs of a slightly higher computational complexity. Due to the manifold state space, we again have to use a modified version of the standard algorithm. Our implementation of the filter uses the general algorithm presented in [16], while the actual models used are the ones given in Sect. II.

C. Particle filter

The PF [23] is implemented as described in [11], [12]. It represents the posterior $p(x_{0:t} | u_{1:t}, z_{0:t}^{\mathrm{all}})$, which is also referred to as target distribution in this context, using a set of particles, where each particle $x_t^{[m]} \in \mathcal{S}$, $1 \leq m \leq M$ is a state hypothesis and M denotes their total number. The state estimate is calculated recursively over time by (i) sampling from a proposal distribution, (ii) calculating the individual importance weights $w_t^{[m]} \in \mathbb{R}$ for each particle, which accounts for the difference between the target and the proposal distribution, and (iii) resampling the particles with a probability proportional to $w_t^{[m]}$.

The standard implementation of the particle filter [19] uses the motion model $p(x_t | u_t, x_{t-1})$ as the proposal distribution. However, in particular when precise sensors are used, a high number of particles may end up in less relevant areas of the target distribution. As a consequence, many particles are needed to adequately cover the distribution. We overcome that disadvantage by following the idea of [24] and use an improved proposal distribution. In particular, we consider the interferometer measurement z_t^{v} and star tracker measurements z_t^{q} in the proposal distribution, which then becomes

$$\begin{aligned} &\tilde{p}(x_t^{[m]} | u_t, x_{t-1}^{[m]}, z_t^{\mathrm{v}}, z_t^{\mathrm{q}}) \\ &\qquad \propto p(x_t^{[m]} | u_t, x_{t-1}^{[m]}) p(z_t^{\mathrm{v}} | x_t^{[m]}) p(z_t^{\mathrm{q}} | x_t^{[m]}). \end{aligned} \tag{43}$$

To draw samples from this distribution, we exploit that the measurements are significantly more accurate than the motion model as well as that the position and velocity are independent of the attitude in the relevant models (see (5), (7)–(9), (11), and (12)). Accordingly, we can first sample r_t and v_t from $p(x_t|u_t, x_{t-1})$. After that, we set q_t and the radial velocity by sampling them according to the respective measurements and their uncertainty (see below). To further reduce the required number of particles, we employ the adaptive resampling technique also proposed in [24].

For expressing the uncertainty of the estimate, no special treatment is needed regarding the manifold state space, since the uncertainty is implicitly given by the distribution of the particles. However, for active perception (see Sect. IV), a Gaussian uncertainty estimation Σ_t has to be provided, which can be obtained as described in [12]. In the prediction step, we have to draw samples in a manifold space. More specifically, we have to sample from $p(z_t^{\mathrm{q}}|x_t) = p(q_t|z_t^{\mathrm{q}})$ (see (9)). This is implemented by using $q_t^{[m]} = z_t^{\mathrm{q}} \boxplus \delta_t$, where $\delta_t \sim \mathcal{N}(0, Q_t^{\mathrm{q}})$ is drawn from an ordinary normal distribution. In the same way, we can draw samples from $p(x_t|u_t, x_{t-1})$, but as explained above, we only sample $r_t^{[m]}$ and $v_t^{[m]}$ from that distribution. Both are elements of vector spaces that can be drawn directly. For calculating the importance weights, we need to evaluate the measurement likelihoods. Here, no modification is needed, since z_t^{q} is the only measurement defined in a manifold space, but it is already considered in the prediction step.

IV. Active Perception

The main task of the spacecraft during the cruise flight is to keep track of its current pose. This requires to continuously change the attitude in order to make observations that help the filters (see Sect. III) to maintain or improve the current localization performance. An approach [11] used by space engineers is to alternate between the Sun and the planet with the largest visible chord. We aim to improve that strategy by taking the expected information gain as well as the costs for changing the attitude into account [12]. In particular, we calculate the best attitude $q_{t+1}^* \in SO(3)$ for the next time step using

$$q_{t+1}^* = \operatorname*{arg\,max}_{\tilde{q}_{t+1} \in SO(3)} \sum_{i \in \mathcal{I}'} IG^i(\tilde{q}_{t+1}, x_t, \Sigma_t) - c(\tilde{q}_{t+1}, x_t, q_t^*) \,, \tag{44}$$

where $\mathcal{I}' = \{\mathrm{c}, \mathrm{p}, \mathrm{s}\}$ denotes the set of all sensors considered here, which are the planet sensor (planet chord and planet to star measurements) and the Sun sensor (Sun to star measurements). Furthermore, $IG^i : SO(3) \times \mathcal{S} \times \mathbb{R}^{9\times 9} \to \mathbb{R}$ is the expected information gain for sensor i when being in state x_t and changing the attitude to $\tilde{q}_{t+1}$, while the estimated uncertainty is given in terms of the covariance matrix Σ_t. Finally, $c : SO(3) \times \mathcal{S} \times SO(3) \to \mathbb{R}$ is a cost function that determines the costs for being in state x_t and changing the attitude to $\tilde{q}_{t+1}$, while the best attitude determined in the previous step is q_t^*. It is also possible to introduce weighting factors for IG^i and for c to trade off between information gain and costs. In order to make the computation of (44) feasible, we discretize $SO(3)$ and evaluate the objective function for the discretized attitudes.

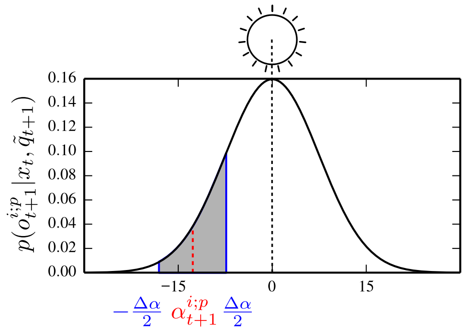


Fig. 4: Probability of observing the celestial object p with sensor i at time $t+1$. It is calculated by integrating the PDF of a normal distribution centered at the celestial object from $\alpha_{t+1}^{i;p} - \frac{\Delta\alpha}{2}$ to $\alpha_{t+1}^{i;p} + \frac{\Delta\alpha}{2}$.

A. Information Gain

The expected information gain for each measurement consists of two parts: the probability of observing the respective celestial object and the expected benefit of the measurement with respect to the estimated uncertainty. The former is calculated using (see Fig. 4)

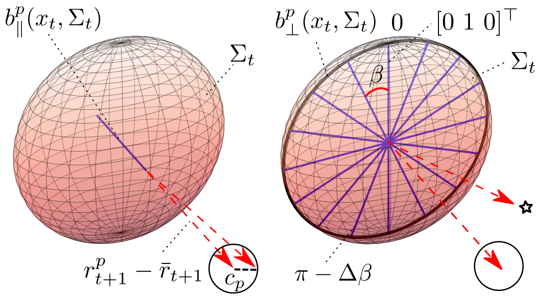
$$p(o_{t+1}^{i;p}|x_t, \tilde{q}_{t+1}) = \Phi(\alpha_{t+1}^{i;p} + \tfrac{\Delta\alpha}{2}; 0, \sigma_p^2) - \Phi(\alpha_{t+1}^{i;p} - \tfrac{\Delta\alpha}{2}; 0, \sigma_p^2), \tag{45}$$

where $o_{t+1}^{i;p} \in \{0, 1\}$ is a binary random variable that models whether the sensor i can observe the celestial object p at $t+1$. Furthermore, Φ is the cumulative distribution function (CDF) of the normal distribution, $\alpha_{t+1}^{i;p}$ is the expected angle between the sensor's view vector and the vector from the spacecraft to the celestial object, $\Delta\alpha$ is the angular discretization used to evaluate (44), and σ_p is chosen according to the field of view of the sensor. The expected angle is given by

$$\alpha_{t+1}^{i;p} = \cos^{-1} \frac{\tilde{q}_{t+1} a^i [1\ 0\ 0]^\top \cdot (r_{t+1}^p - \bar{r}_{t+1})}{\|\tilde{q}_{t+1} a^i [1\ 0\ 0]^\top\| \, \|r_{t+1}^p - \bar{r}_{t+1}\|} \,, \tag{46}$$

where $a^i \in SO(3)$ is the assembly orientation of the sensor and the vector $[1\ 0\ 0]^\top$ accounts for the fact that the viewing direction of each sensor is aligned with the x-direction of its body-fixed frame. Furthermore, $\bar{r}_{t+1}$ is the position part of the expected state for $t+1$. It is calculated using $\bar{x}_{t+1} = g(0_{3\times 1}, x_t)$ (see (5)), i.e., no gyroscope measurements are taken into account, since they are not available yet and they are not needed to predict the position anyway.

The expected benefit of the measurement is calculated based on the covariance matrix provided by the filters. We have to take into account whether the measurement helps to correct the estimate in radial or tangential direction to a celestial object (cf. Figs. 2 and 3). When the measurement allows corrections in radial direction, i.e., along the vector towards the celestial


$b^p_\parallel(x_t, \Sigma_t)$
Σ_t
$r^p_{t+1} - \bar{r}_{t+1}$
c_p
$b^p_\perp(x_t, \Sigma_t)$
0
$[0\ 1\ 0]^\top$
Σ_t
β
$\pi - \Delta\beta$

(a) radial benefit

(b) tangential benefit

Fig. 5: Calculation of the expected benefit of an observation based on the measurement direction and the measurement type (cf. Figs. 2 and 3).

object, the expected benefit is calculated using (see Fig. 5a)

$$b^p_\parallel(x_t, \Sigma_t) = f(r^p_{t+1} - \bar{r}_{t+1}, \Sigma_t)\,, \tag{47}$$

$$f(d, \Sigma) = F \Sigma F^\top\,, \qquad F = \begin{bmatrix} \frac{d^\top}{\|d\|} & 0_{1\times 6} \end{bmatrix}\,, \tag{48}$$

where the function $f : \mathbb{R}^3 \times \mathbb{R}^{9\times 9} \to \mathbb{R}$ projects the position part of the covariance matrix onto the vector $d \in \mathbb{R}^3$, which in this case corresponds to the vector from the spacecraft to the celestial object. Thus, a celestial object is more valuable if the position uncertainty in the radial direction is large.

When the measurement allows corrections in tangential direction, i.e., on the plane that is orthogonal to the vector towards the celestial object, we have (see Fig. 5b)

$$b^p_\perp(x_t, \Sigma_t) = \int_0^\pi f(M^p R^\beta [0\ 1\ 0]^\top, \Sigma_t)\, d\beta\,. \tag{49}$$

The matrix $M^p \in \hat{\mathbb{R}}^{3\times 3}$ defines a rotation from a coordinate frame that has its x-axis aligned with $r^p_{t+1} - \bar{r}_{t+1}$ to the navigation frame and the matrix $R^\beta \in \hat{\mathbb{R}}^{3\times 3}$ rotates about β around the x-axis. In other words, the vector $[0\ 1\ 0]^\top$ is placed orthogonal on the vector from the spacecraft to the celestial object and is rotated around that vector. The covariance is projected on the resulting vector, and thus, the final result corresponds to the area of the slice plane between the covariance ellipsoid and the tangential plane. Since the covariance is symmetric, we only integrate from 0 to π. Furthermore, the actual implementation discretizes the angle β in order to calculate the integral as a sum from 0 to $\pi - \Delta\beta$ in $\Delta\beta$ steps.

A planet chord measurement allows to correct the position in radial direction (see Fig. 3a). Accordingly, its expected information gain is given by

$$IG^{\mathrm{c}}(\tilde{q}_{t+1}, x_t, \Sigma_t) = \sum_{p\in\mathcal{P}} \eta^{\mathrm{c}}_p\, p(o^{\mathrm{c};p}_{t+1} | x_t, \tilde{q}_{t+1})\, b^p_\parallel(x_t, \Sigma_t)\,. \tag{50}$$

The scaling factor $\eta^{\mathrm{c}}_p = h^{\mathrm{c}}(x_t)$ weights the value for each planet with the expected chord angle, which prefers planets with a larger visible chord. The latter accounts for the distance-dependent sensor uncertainty (see (19)). A planet-star measurement corrects the position in radial direction (see Fig. 3b). Thus, its expected information gain is computed using

$$IG^{\mathrm{p}}(\tilde{q}_{t+1}, x_t, \Sigma_t) = \sum_{p\in\mathcal{P}} \eta^{\mathrm{p}}_p\, p(o^{\mathrm{p};p}_{t+1} | x_t, \tilde{q}_{t+1})\, b^p_\perp(x_t, \Sigma_t)\,. \tag{51}$$

Here, the weighting factor is defined as $\eta^{\mathrm{p}}_p = 1.0 - \min(\|r^p_{t+1} - \bar{r}_{t+1}\|/12\,\mathrm{AU}, 0.9)$ (see [12]), which prefers nearby planets. The information gain of a Sun-star measurement is calculated using

$$IG^{\mathrm{s}}(\tilde{q}_{t+1}, x_t, \Sigma_t) = p(o^{\mathrm{s;Sun}}_{t+1} | x_t, \tilde{q}_{t+1})\, b^{\mathrm{Sun}}_\perp(x_t, \Sigma_t)\,. \tag{52}$$

B. Rotation Costs

The costs for changing the attitude from q_t to $\tilde{q}_{t+1}$ comprise

$$c(\tilde{q}_{t+1}, x_t, q^*_t) = \|q_t \boxminus \tilde{q}_{t+1}\| + 1_{t<t'+\lambda} \|q^*_t \boxminus \tilde{q}_{t+1}\|\,. \tag{53}$$

The first summand penalizes new attitudes that have a large distance to the current attitude, and thus require more energy. The second summand results in additional costs if the new attitude has a large distance to last best attitude q^*_t. The latter is taken into account when the indicator function $1_{t<t'+\lambda}$ is 1. This is the case as long as the current time t is smaller than $t' + \lambda$, where t' is the most recent time point when that condition was not fulfilled and λ is a fixed time period.

Put in other words, the spacecraft will prefer attitudes with a small distance to the last best attitude for a period of λ, while each λ time steps, this penalty is not applied and the spacecraft can select an arbitrary new attitude (under consideration of the expected information gain and the actual costs for rotation to that attitude). By that means, we force the spacecraft to observe a celestial object for a certain period and give it time to actually reach the target attitude. At the same time, the spacecraft still has the flexibility to track that celestial object when it moves relatively to the spacecraft.

V. Evaluation

The spacecraft [7] has one sensor of each of the types introduced in Sect. II. The sensors considered for active perception are the Sun sensor and the planet sensor (see (44)). The former points along the x-axis, while the latter points along negative z-axis of the spacecraft's body-fixed frame, thus exhibiting orthogonal viewing directions (see Fig. 2). The evaluation is conducted in a simulation environment. The step size is set to 2.5 h, where in each simulation step a new ground truth position is calculated and the respective sensor measurements are generated. The dynamic of the spacecraft is simulated using (5), where normally-distributed noise with zero mean and a standard deviation of $10^{-8}\,\mathrm{m/s^2}$ is added to the accelerations given in (6) to account for non-modeled influences. The sensor measurements are calculated according to the equations given in Sect. II. We conduct the evaluation on an Earth-like *Orbit 1*, an intermediate *Orbit 2*, and finally on *Orbit 3* as proposed in [7]. The varied orbital element parameters (semimajor axis, eccentricity, and inclination) are

TABLE I: Orbital element parameters for the different orbits.

orbital element	Orbit 1	Orbit 2	Orbit 3
semimajor axis [AU]	1.2	2.0	2.8
eccentricity [—]	0.25	0.125	0
inclination [deg]	50	25	0

given in Tab. I, while the remaining parameters (ascending node, periapsis, and longitude) are set to 0. We evaluate two complete revolutions for each orbit and the number of particles in the PF is set to 500. To mitigate the effects of the random factors of the system, all results are averaged over 50 runs and the standard deviation over the runs is indicated. This also demonstrates the stability of the proposed methods.

The cumulative position error (CPE) for time step t is computed as

$$CPE_t = \frac{1}{N}\sum_{n=1}^{N} CPE_{t-1}^{(n)} + \|\hat{r}_t^{(n)} - r_t^{(n)}\|\,, \tag{54}$$

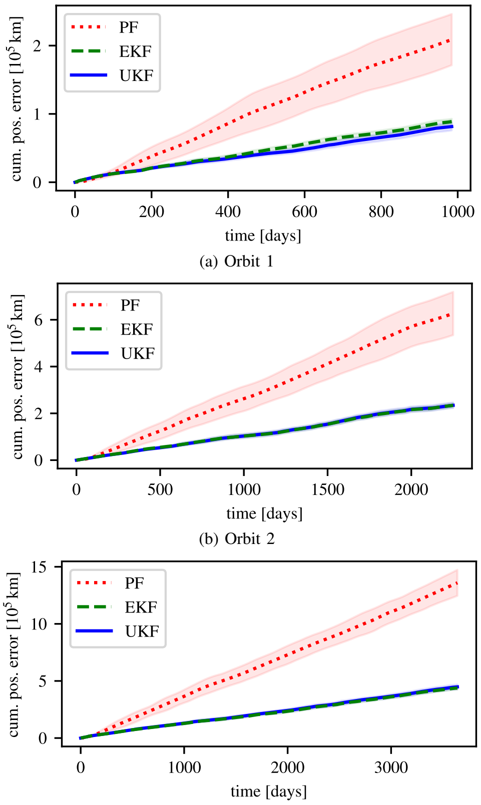
where $N = 50$ is the number of runs, $r_t^{(n)}$ is the position estimate for time step t of the n-th run, and $\hat{r}_t^{(n)}$ is the ground truth position for time step t of the n-th run. The recursion is initialized with $CPE_0^{(n)} = 0$. Note that the CPE grows over time by definition. The root mean square (RMS) position error is computed using

$$RMS = \frac{1}{N}\sum_{n=1}^{N} \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\hat{r}_t^{(n)} - r_t^{(n)}\|^2}\,, \tag{55}$$

where T is the number of time steps. The standard deviation is calculated with respect to the averaging over N.

A. Filter Evaluation

As baseline, we evaluated the filters for state estimation presented in Sect. III without the need to choose any observables, i.e., the FOV of the Sun sensor and the planet sensor is set to 360°. The cumulative position error for all filters on all orbits is shown in Fig. 6. All filters show a stable behavior for the single orbits. In particular, the EKF and UKF are on a similar performance level. The advantage of the UKF, that it potentially copes better with nonlinearities, cannot be exploited, with a weak exception on Orbit 1 (see Fig. 6a). Surprisingly, the PF performs worse, even though it is theoretically even better suited to deal with nonlinearities as well as non-Gaussian distributions. In addition, while showing a stable behavior overall, the particular runs of the PF vary more than the ones of the Kalman filters, which is indicated by a larger standard deviation.



(a) Orbit 1

(b) Orbit 2

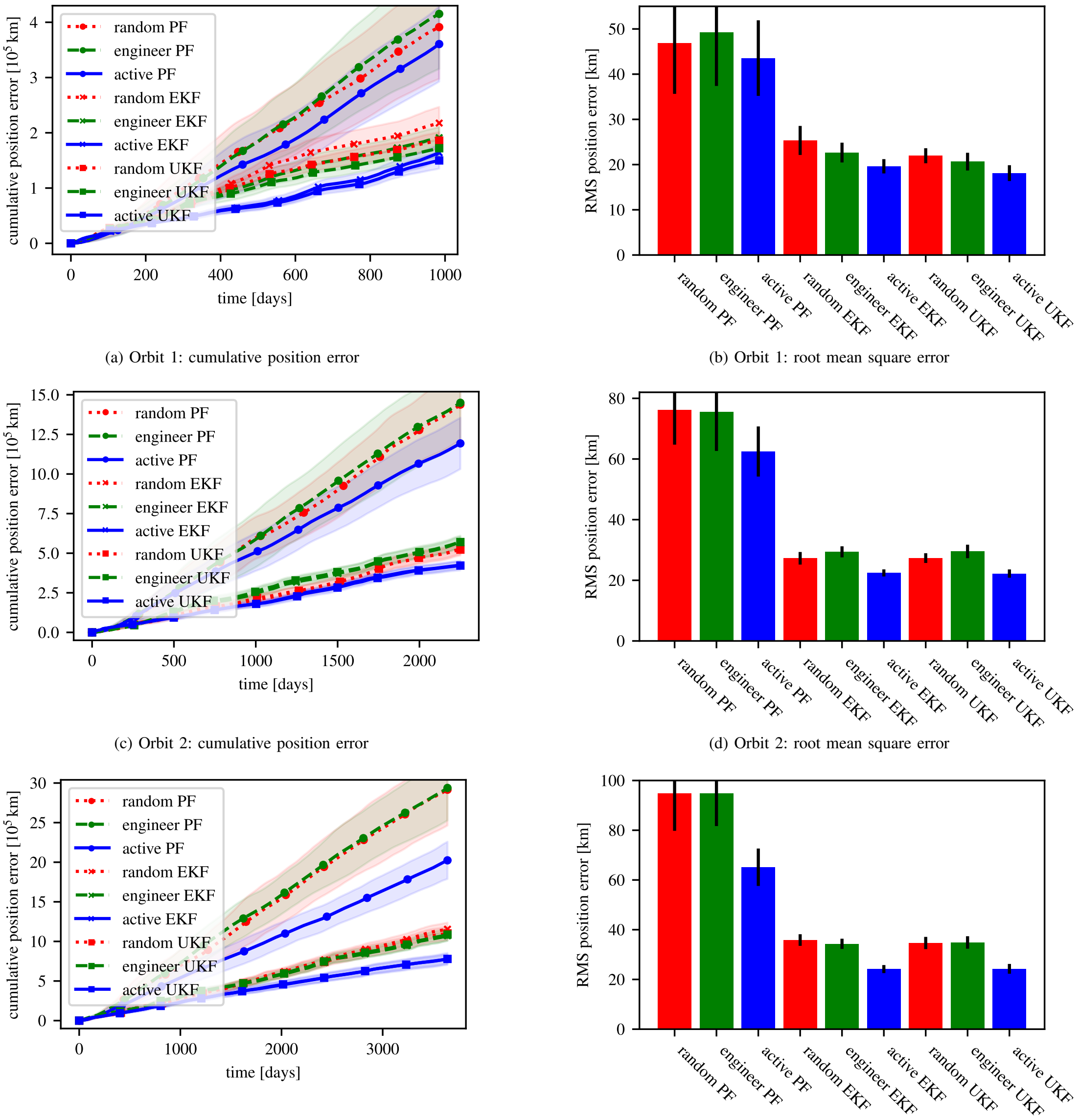
(c) Orbit 3

Fig. 6: Cumulative position error (CPE) of Extended KF., Unscented KF., and Particle F. evaluated with a 360° FOV. Please note the varying scaling of the x and y-axis.

B. Agent Evaluation

In order to evaluate the performance of the active perception approach proposed in Sect. IV, we implement two other agents. The first agent periodically switches between the Sun and a randomly selected planet. Therefore, we refer to this agent as *random agent*. The second agent also switches between the Sun and a planet, but it selects the planet based on the largest visible chord angle as described in [11]. This agent is referred to as *engineering agent*. Finally, we let our agent be the *active agent*. The time to observe a celestial object is set to 5.2 days. The same value is chosen for the fixation parameter λ in (53) for our agent. Furthermore, the angle discretization for evaluating (44) is set to $1.5°$. For this evaluation, we set the FOV of the Sun sensor and the planet sensor to 10°.

The cumulative position error as well as the RMS position error for all orbits is shown in Fig. 7. Again, the EKF and UKF have a similar performance level, while the performance of the PF is worse. The random and the engineering agents exhibit a similar performance, while the active agent outperforms them throughout all orbits and filter conditions: it has a lower cumulative position error (see Figs. 7a, 7c, and 7e) and a lower RMS error (see Figs. 7b, 7d, and 7f). In addition, the standard deviation is lowered, which indicates a more stable localization performance. As the active agent positively affects measurements which are used in the correction step, it performs even better the more uncertainty has to be dealt with. This holds true in two perspectives: (i) looking at every single



(a) Orbit 1: cumulative position error

(b) Orbit 1: root mean square error

(c) Orbit 2: cumulative position error

(d) Orbit 2: root mean square error

(e) Orbit 3: cumulative position error

(f) Orbit 3: root mean square error

Fig. 7: Cumulative position error (CPE) plotted over time and RMS position error of Extended KF., Unscented KF., and Particle F. and all agents. All results are averaged over 50 runs of two revolutions on the respective orbit. Please note the different scaling of the x and y-axis.

orbit, the PF always benefits the most, as it has the biggest error on every particular orbit. But also (ii) throughout all orbits the benefit for all filters increases with a rising general error level. The latter is simply caused by larger distances and evaluation time spans on larger orbits.

VI. Conclusion

In this paper, we presented an Extended Kalman Filter, an Unscented Kalman Filter, as well as a Particle Filter for estimating the state of a spacecraft in a dynamic 3D environment. All filters operate on a globally consistent manifold state representation, while small changes are expressed in a local vector space. In addition, we proposed an approach for active perception, which computes the desired attitude for the next time step based on the expected information gain of different sensors with respect to the current uncertainty of the filters.

All algorithms are evaluated in a simulation system using three different orbits. The filters are compared to each other in a full field of view setting. Furthermore, we compare our strategy for active perception to a random approach and an engineering approach. Their interaction with the different filters in a restricted field of view setting is investigated. We showed that the UKF and EKF have a similar performance level, while the PF cannot compete with them. Furthermore, we found that our active perception approach outperforms both other strategies. The advantage of the active perception approach increases with a higher error level of the state estimation module, as the correctional measurements are improved.

In the future, this approach might also be used to optimize sensor positions for a certain mission, given the knowledge about the sensor noise, the observables as well as the time-window and trajectory to be taken. Furthermore, the methodology can be applied to other scenarios, where the movement of the surroundings can be predicted reasonably well. Examples are automated driving or industrial robotics.

Acknowledgment

We thank Carsten Rachuy for his dedicated work on the simulation software. This work was supported by the German Aerospace Center (DLR) with financial means of the German Federal Ministry for Economic Affairs and Energy (BMWi), project "KaNaRiA" (grant No. 50 NA 1318) and project "EnEx-CAUSE" (grant No. 50 NA 1505).

References

[1] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[2] K. Schill, E. Umkehrer, S. Beinlich, G. Krieger, and C. Zetzsche, "Scene analysis with saccadic eye movements: top-down and bottom-up modeling," *Journal of electronic imaging*, vol. 10, no. 1, pp. 152–161, 2001.

[3] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1999, pp. 35–40 vol.1.

[4] D. Nakath, T. Kluth, T. Reineking, C. Zetzsche, and K. Schill, "Active Sensorimotor Object Recognition in Three-Dimensional Space," in *Spatial Cognition IX*. Springer, 2014, pp. 312–324.

[5] J. Yim, J. Crassidis, and J. Junkins, "Autonomous orbit navigation of interplanetary spacecraft," in *Astrodynamics Specialist Conference*, 2000, p. 3936.

[6] A. Probst, G. González Peytaví, D. Nakath, A. Schattel, C. Rachuy, P. Lange, J. Clemens, M. Echim, V. Schwarting, A. Srinivas, K. Gadzicki, R. Förstner, B. Eissfeller, K. Schill, C. Büskens, and G. Zachmann, "KaNaRiA: Identifying the challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies," in *66th International Astronautical Congress (IAC)*. International Astronautical Federation, 2015, paper IAC-15-A3.IP.15.

[7] A. Probst, G. González Peytaví, B. Eissfeller, and R. Förstner, "Mission concept and preliminary spacecraft design for resource mining on main belt asteroids," in *66th International Astronautical Congress (IAC)*. International Astronautical Federation, 2015, paper IAC-15-D4.1.2.x28035.

[8] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone, "Spacecraft autonomy challenges for next-generation space missions," in *Advances in Control System Technology for Aerospace Applications*. Springer, 2016, pp. 1–48.

[9] D. W. Curkendall and J. S. Border, "Delta-DOR: The one-nanoradian navigation measurement system of the deep space network—History, architecture, and componentry," *The Interplanetary Network Progress Report*, vol. 42, p. 193, 2013.

[10] F. Budnik, T. Morley, and R. Mackenzie, "ESOC's system for interplanetary orbit determination: Implementation and operational experience," in *18th International Symposium on Space Flight Dynamics*, vol. 548, 2004, pp. 387–392.

[11] G. González Peytaví, J. Clemens, D. Nakath, A. Probst, R. Förstner, K. Schill, and B. Eissfeller, "Autonomous orbit navigation for a mission to the asteroid main belt," in *66th International Astronautical Congress (IAC)*. International Astronautical Federation, 2015, paper IAC-15-C1.7.4.

[12] D. Nakath, C. Rachuy, J. Clemens, and K. Schill, "Optimal rotation sequences for active perception," in *Proc. SPIE: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, J. J. Braun, Ed., vol. 9872. SPIE Press, 2016, paper 987204.

[13] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. Wiley, 1994.

[14] R. A. Howard, *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

[15] J. Clemens and K. Schill, "Extended Kalman filter with manifold state representation for navigating a maneuverable melting probe," in *19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1789–1796.

[16] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[17] D. Nakath, J. Clemens, and C. Rachuy, "Rigid body attitude control based on a manifold representation of direction cosine matrices," in *13th European Workshop on Advanced Control and Diagnosis (ACD)*, ser. Journal of Physics: Conference Series, vol. 783, 2016, paper 012040.

[18] M. A. C. Perryman, L. Lindegren, J. Kovalevsky, E. Hoeg, U. Bastian, P. L. Bernacca, M. Crézé, F. Donati, M. Grenon, M. Grewing *et al.*, "The HIPPARCOS catalogue," *Astronomy and Astrophysics*, vol. 323, 1997.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[21] A. H. Jazwinski, *Stochastic processes and filtering theory*. New York, NY: Academic Press, 1970.

[22] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182–193.

[23] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer Science & Business Media, 2001.

[24] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

Active Asteroid-SLAM

Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario

David Nakath · Joachim Clemens · Carsten Rachuy

Received: date / Accepted: date

Abstract In this paper, we propose an active real-time capable 3D graph based simultaneous localization and mapping (Graph SLAM) approach, which actively estimates the state of an autonomous spacecraft relative to a simultaneously established map estimate. The graph is constructed in a tightly-coupled fashion, where an Extended Kalman Filter estimates the relative offset between two of its vertices. An additional relative measurement is derived by matching point clouds obtained by a light detection and ranging (LiDAR) system. In order to yield a significant speed-up, scan matching is implemented on the GPU. To reduce the uncertainty of either the state or the map estimate, we present an approach to actively control the system resting on an extended representation of uncertainty in the map. Furthermore, it adapts its behavior depending on the current uncertainty distribution in order to find a dynamic trade-off between exploitation (improve localization performance) and exploration (improve knowledge about the environment). Finally, we present a post-processing approach to discover landing sites in the map estimate without prior knowledge. The evaluation is conducted in a numerical simulation, where the spacecraft explores the real 3D model of Itokawa in its actual dynamic environment. Within that simulation, we use a shader-based GPU implementation for simulating LiDAR measurements. We evaluate the performance of the active SLAM approach and demonstrate that the use of the adaptive approach improves navigation and exploration performance at the same time.

This work was supported by the German Aerospace Center (DLR) with financial means of the German Federal Ministry for Economic Affairs and Energy (BMWi), project "KaNaRiA" (grant No. 50 NA 1318) and project "EnEx-CAUSE" (grant No. 50 NA 1505). In addition, we gratefully acknowledge the support of NVIDIA Corporation by donating a Titan X Pascal GPU used for this research.

David Nakath
E-mail: dnakath@informatik.uni-bremen.de

Joachim Clemens
E-mail: jclemens@informatik.uni-bremen.de

Carsten Rachuy
E-mail: rachuy@informatik.uni-bremen.de

Cognitive Neuroinformatics, University of Bremen
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

Keywords Autonomous Relative Navigation · Active Graph SLAM · Active Perception · Deep Space · Active State Estimation · Autonomous Systems

1 Introduction

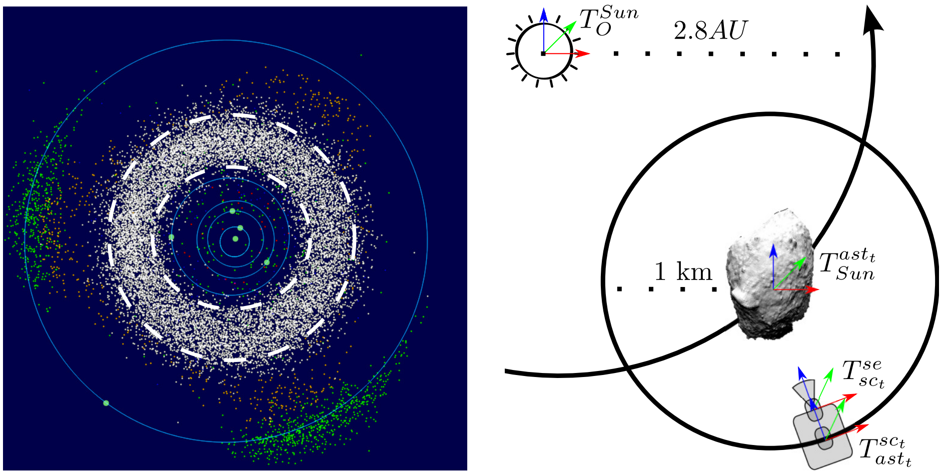
There is an ever-growing interest in relative autonomous deep space navigation around planetary bodies, such as asteroids, or docking maneuvers with (un-)cooperative objects [27]. In addition, future missions strive for mapping of celestial bodies for scientific, commercial, and navigational purposes, which commonly require a metric, globally consistent representation, preferably with associated uncertainties. Specifically, this interest is driven by scientific sample-return asteroid missions, potential commercial missions, and the rising demand for planetary protection [50].

On the one hand, asteroid sample-return missions might be conducted for scientific reasons, like the Hayabusa I [96] and Hayabusa II [92] missions to the asteroids Itokawa and Ryugu, respectively. In addition, NASA's OSIRIS REx mission attempts to retrieve a sample from the near-earth asteroid Bennu in order to gather information about the formation process of the solar system [6]. On the other hand, potential commercial missions to asteroids aim for mining of water [89]. It can be extracted and subsequently be used in-situ as a propellant, for life support, or as radiation shielding. Also of commercial interest are missions mining platinum-group metals (PGMs) or rare earth elements (REEs) on asteroids and returning them to Earth [5]. The specific scenario evaluated in this paper stems from the exploration phase of a commercial asteroid mining mission [67, 68, 69] in the main belt in a mean distance of 2.8 astronomical units (AU) from the Sun (see Fig. 1).

More specifically, we investigate active autonomous relative navigation and mapping in the vicinity (1 km) of an asteroid using a Potential Target Characterization Module (PTCM) spacecraft (S/C) with an attached lander (see Fig. 2) [31, 67]. Due to the lack of a precise asteroid model from that area, we choose Itokowa, the destination of Hayabusa I, as a replacement. This allows us to draw on realistic models obtained during the encounter in 2005 [82]. They comprise a gravity model, given as spherical harmonics, and a 3D-model, which has a resolution of < 1 m.

Ground support for deep space navigation can usually be provided for these types of missions. However, for commercial missions, ESA's European Space Tracking (ESTRACK) [10] network and NASA's deep space network (DSN) [21] are not applicable due to high cost. In addition, technical issues like signal roundtrip time or complex maneuvers might also demand a higher level of autonomy. For example, Mascot, one of the landers of the Hayabusa 2 mission, has a limited battery capacity, thus rendering autonomous decision making mandatory [37].

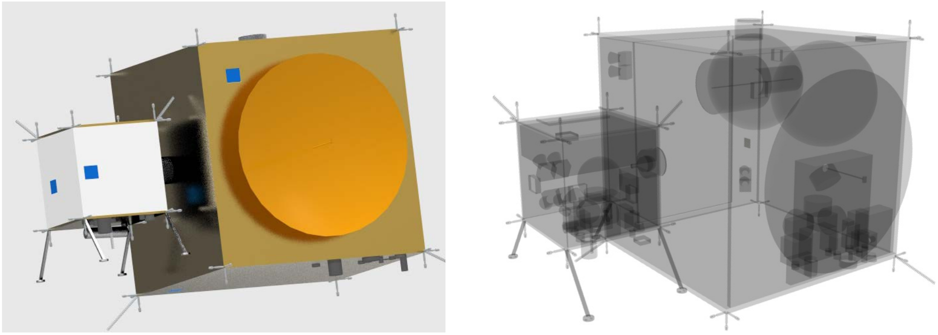
When no ground support is available, the movement of a S/C can be estimated by integrating the equations of motion from a given starting point. However, such a dead reckoning approach results in an ever-accumulating estimation error. The latter can be mitigated by optical sensors, e.g., LiDARs or cameras, to enable optical flow technologies like visual odometry to determine an offset between two poses (i.e., position and attitude). This information can for example be used in the correction step of a Bayesian filter or for a mapping algorithm, which simply registers the measurements into a map. However, this is not sufficient for a globally consistent map estimate, which can be established using a Simultaneous Localization and Mapping (SLAM) approach. The resultant map might not only be used for correctional purposes in the navigation modules but also as a decision-making foundation for the autonomy module of the S/C. Specifically, it can be used in an active perception



(a) Asteroid main belt (white dots).

(b) Scenario and associated reference frames.

Fig. 1: Scenario sketch, relative navigation and mapping (b) in the asteroid main belt (a). To clarify notation: T_B^A denotes a rigid transformation in $SE(3)$, in e.g., homogeneous coordinates, by describing the position and orientation of an orthonormal basis of A with respect to B, thus sending a vector v_A defined in A to B, when multiplied from the left, i.e., $v_B = T_B^A v_A$. (Figure (a) adopted from a figure by user *Mdf*, public domain, available at `https://commons.wikimedia.org/wiki/File:InnerSolarSystem-en.png`.)

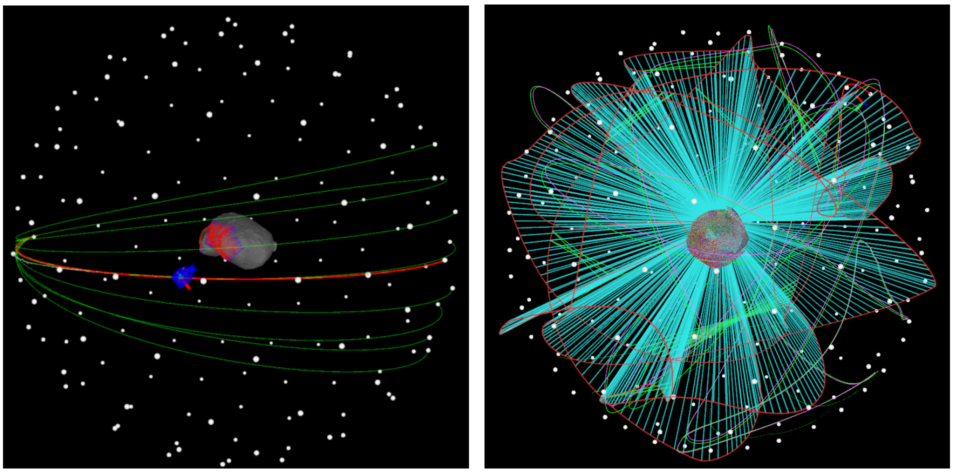


(a) S/C and lander shown as a rendering [67].

(b) S/C and lander visualized in RVIZ [40].

Fig. 2: Model of the Potential Target Characterization Module (PTCM) with attached lander.

scheme to determine where to look next in order to specifically reduce uncertainty. The latter can be defined with respect to different quantities like the trajectory or the map estimate. The map can in turn be used to deduct further information, like the coarseness of the surface of the asteroid, which serves as a foundation for discovering potential landing sites.



(a) White dots: evaluation points ξ_i, green lines: potential trajectories computed with WORHP, red line: chosen trajectory, blue box: model of S/C.

(b) Final graph structure, with the final maximum likelihood scan map, as obtained by Graph SLAM.

Fig. 3: Screenshots of simulation system.

1.1 Related Work

An ad-hoc approach to the task of obtaining the state and map distributions would be to estimate them separately. In [66], such an approach is presented, modeling the orbit selection for the mapping of small celestial bodies as a partially observable Markov decision process (POMDP). But the approach only considers mapping while the problem of trajectory estimation is omitted. On the other hand, [24] show the feasibility of a Flash-LiDAR-based relative navigation scheme around the asteroids Itokawa and Bennu, while the same authors present a robust solution in [25]. However, they assume knowledge about the asteroid in the form of a low-fidelity shape model.

In general, the independent estimation of these quantities typically leads to suboptimal results as the reciprocal correctional potential between the map and state distributions is not leveraged. The superior approach of their joint estimation is commonly known as SLAM [91, Chaps. 10–13]. Different solutions can be formulated for this kind of problem based on either filtering or smoothing techniques. The former approaches employ some kind of recursive state estimation, often based on either Kalman filters or particle filters, interpreting the problem as a dynamic Bayesian network. A popular filtering-based approach is the FastSLAM algorithm [56], which relies on a Rao-Blackwellized particle filter (RBPF) for factorizing the problem. For that reason, algorithms following this scheme are also referred to as RBPF-SLAM. FastSLAM has been enhanced by utilizing the measurements already in the prediction step resulting in FastSLAM 2.0 [55]. While both are based on landmark maps, the FastSLAM approach has been applied to grid maps as well [35]. Additionally, improvement techniques similar to FastSLAM 2.0 have been proposed in [32] for grid-map–based RBPF-SLAM. Recently, those approaches were extended to Evidential FastSLAM [17, 73] using the belief framework [86] to achieve an extended uncertainty representation [83] of the map distribution (cf. Figs. 5b and 7, as well as Sect. 3.2). Opposed to those approaches, the key idea behind graph-based SLAM is that purely recursive state estimation suffers from the first-order Markov assumption. The latter implies, that all information is accumulated in the last state x_{t-1} and thus all preceding data is discarded after processing. However, the posterior of the SLAM problem naturally forms a sparse graph [30], which allows for a smoothing approach. Thus, Graph SLAM approaches seek to optimize the entire system by interpreting the SLAM problem in terms of a graph structure representing spatial constraints [34] (see Figs. 3b and 6). These constraints are typically used as an error function within a least squares scheme (see e.g., [23, 29, 62]). The result is a globally corrected SLAM estimate, which can be computed efficiently using solvers exploiting the special properties of the sparse graph structure [49]. Throughout this paper, we will hence use a state-of-the-art Graph SLAM approach. In addition, we will employ the evidential grid map from [73, 17]. This way, we can draw on the extended uncertainty representation, offered by the latter, and at the same time benefit from a smoothing approach which globally optimizes over all measurements.

There already are SLAM approaches for relative navigation in the vicinity of small bodies in deep space. In [19, 20], a monocular RBPF-SLAM for autonomous navigation in the vicinity of asteroids is presented. The landmarks are SURF-features extracted from camera images of the surface. Instead of solely using camera-based features, it is also possible to employ LiDAR-aided crater tracking, relying on a model of the latter. This alternative landmark-based SLAM approach has been presented by [3]. However, the striking absence of impact craters on the Itokawa model renders this approach infeasible for this scenario.

Except for [17], all SLAM-approaches discussed so far are used for passive relative navigation and mapping. Active uncertainty reduction for robots has been introduced by [76] and the principle has been transferred to active deep space navigation in [60, 61]. Thus, a straightforward extension is to actively control the information intake process to improve the estimate of the state and/or map distribution. This idea is often referred to as *active* SLAM and was introduced in [28]. Further notable works in this field are [88, 53, 13, 12]. In [47], an interesting active SLAM approach for autonomous deep space navigation based on cross entropy optimization is presented. Within the scenario of relative pose estimation of a chaser satellite, control, planning, and estimation are jointly considered. However, the numerical simulation is conducted in 2D and there are only four landmarks on the target satellite.

In addition to active SLAM, we seek to autonomously discover safe landing sites based on the map estimate. By postprocessing the evidential grid map, we extract the surface of the asteroid and locally estimate corresponding normals. Based on that, we seek to identify large connected areas with coherent surface normals, considered as safe landing spots. A similar goal (based on images processing) has been pursued in [84] using the variance of pixel intensities as a roughness estimate. In addition, a visual landing approach for rocket re-entry and landing has been shown by [39, 57], but it has to rely on known 3D landmarks.

1.2 Contribution and Outline

The contribution of this paper comprises a real-time capable active 3D Graph SLAM estimating the trajectory of a S/C relative to a simultaneously established map estimate of an asteroid. To the knowledge of the authors, 3D active Graph SLAM has not yet been employed in a Flash-LiDAR-based deep space exploration approach in the vicinity of small celestial bodies. Furthermore, the numerical simulation is conducted in a realistic dynamic environment, where the S/C explores a rotating asteroid, which in turn is orbiting the Sun (see Fig. 1). The asteroid is actively explored using a novel twofold trajectory generation and

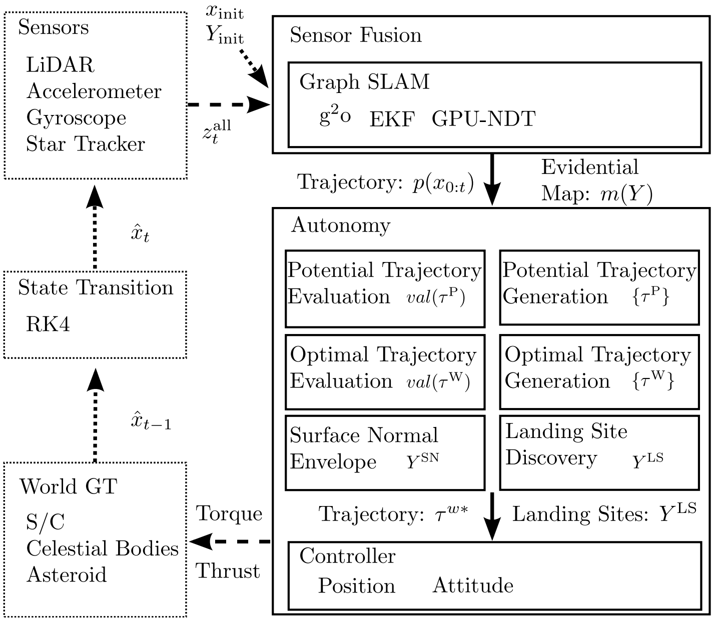


Fig. 4: Closed-loop overview of the proposed system.

evaluation approach. In addition, the uncertainty in the graph can be measured and used as basis for adaptive exploration. Finally, potential landing sites are discovered without prior knowledge by postprocessing the map-estimate.

The remainder of this paper is structured as follows. We begin with a brief but comprehensive description of our system in Sect. 2. It also serves as a guidepost for the sections describing the *Sensor Fusion* and *Autonomy* modules in detail as respective references are given. The subsequent Sect. 3 presents the joint basis of models the simulation environment and the SLAM algorithm draw on. Following that, the Graph SLAM approach for state and map estimation employed in the *Sensor Fusion* module is presented in detail in Sect. 4. Our in-depth module description finishes with the *Autonomy* module, presented in Sect. 5. In Sect. 6, we present the numerical evaluation of our approach in the scenario around the asteroid Itokawa using a custom-build ROS-based [70] simulation environment (see Fig. 3). The paper concludes with a summary and an outlook in Sect. 7. And finally, Appx. A lists the parameters used for the evaluation.

2 System Overview

The proposed system is shown in Fig. 4. On the right hand side, the internal modules of the S/C are shown, while the simulation modules (operating on ground truth data) are shown in dashed boxes on the left.

For initialization, the *Sensor Fusion* module (see Sect. 2.1) is provided with a state estimate x_{init} and an empty map Y_{init}. In the following steps, it draws on a Graph SLAM approach to yield a joint estimate $p(x_{0:t}, Y)$ of the trajectory $x_{0:t}$ of the S/C and the scan map Y of the asteroid, which is in turn used to derive an evidential grid map $m(Y)$. The nodes of the graph constructed in the frontend encode poses, i.e., position and attitude, while the edges represent relative transformations (translation and rotation or only rotation) between two nodes. We employ a tightly-coupled approach for graph construction, by setting the starting state of an Extended Kalman Filter (EKF) to the pose of a node v_i and let it estimate the differential pose to a subsequent node v_j. After a new node is inserted, the EKF is reset to that node. In addition, a GPU-accelerated Normal Distribution Transform (GPU-NDT) approach is used for scan matching, between point clouds obtained at the respective nodes v_i and v_j. The scan matching result is merged with the EKF estimate, to minimize the number of edges inserted into the graph. In addition, it is fed back to the EKF, to indirectly correct the velocity-part of the state estimate. The thus-constructed graph is then optimized in the backend resting on the g^{2}o framework.

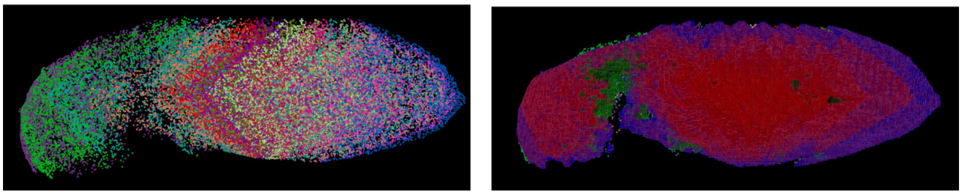
The resultant estimates are subsequently processed in the *Autonomy* module. It is implemented as a hierarchical state machine *(HCSM)*, allowing for the parallel execution of particular behaviors with different frequencies, while maintaining guaranteed pre- and post-conditions. (see Sect. 2.2). It establishes an evaluation structure around the asteroid, comprised of a set of evaluation nodes residing on a tessellated sphere. The evaluation nodes can be rapidly updated from the map estimate, thus providing values for the expected information gain (which quantifies the expected contribution to map improvement) and localization gain (which quantifies the extend to which it is expected to improve the localization accuracy). Subsequently, it generates and evaluates trajectories in a twofold scheme: In the initial step of the trajectory selection procedure, potential trajectories from the current pose to all other evaluation nodes are being computed. This set of trajectories is evaluated in the evaluation structure by considering the whole trajectory and accumulating the information gain and localization gain of the evaluation nodes, weighted by their respective distance. The k-best trajectories are picked, and their respective target positions are subsequently passed to the NLP solver WORHP to compute optimal trajectories to those destinations. The best trajectory is then selected based on the two measures stated above and an additional penalizing factor for frequent directional changes. Hence, a trajectory is assessed in terms of exploring new areas to reduce the map uncertainty or re-visiting well estimated areas providing global correctional potential for the SLAM algorithm. The resultant trajectory is subsequently passed to the *Controller* module. It computes thrust and torque vectors to keep the S/C on the desired trajectory. On demand, the *Autonomy* module preprocesses the map such that landing sites can be discovered.

In the simulation part of the loop, the controls are first applied to the *World Ground Truth (GT)* state with additive noise and a *State Transition* is triggered afterwards. The actual numerical integration is then conducted based on a 4th-Order Runge Kutta (RK4) method. Finally, noisy sensor measurements z_t^{all} are taken from the new ground truth world state $\hat{x}$, which are subsequently passed to the *Sensor Fusion* module of the S/C, thus providing the input for a new loop.

In the following, we provide an overview of the *Sensor Fusion* module and the *Autonomy* module, while they are described in detail in Sects. 4 and 5, respectively.

2.1 Sensor Fusion

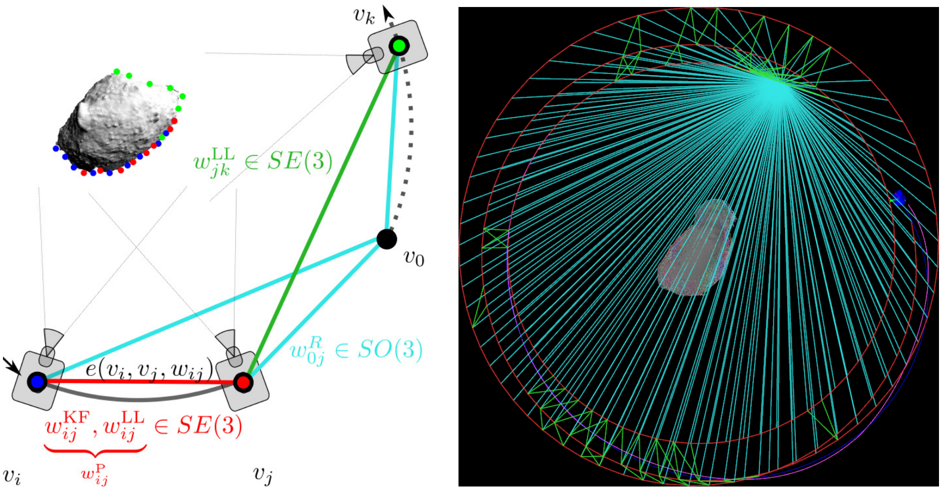
Within the *Sensor Fusion* module, we approach the SLAM problem in the following manner. From the joint estimation of the trajectory and map, we seek to extract a trajectory estimate $p(x_{0:t})$, where the single poses at time t are given in the asteroid frame $T_{ast_t}^{sc_t}$, and a map estimate $p(Y)$. The latter is given in the form of registered point clouds (see Fig. 5a) and is subsequently projected into an evidential grid map $m(Y)$ (see Fig. 5b). This projection



(a) Maximum likelihood scan map.

(b) Extracted evidential grid map.

Fig. 5: Evidential grid mapping, after obtaining a maximum likelihood estimate of the map *(a)* within Graph SLAM , the corresponding evidential grid map *(b)* is extracted afterwards.


v_k
$w_{jk}^{\mathrm{LL}} \in SE(3)$
v_0
$w_{0j}^{R} \in SO(3)$
$e(v_i, v_j, w_{ij})$
$\underbrace{w_{ij}^{\mathrm{KF}}, w_{ij}^{\mathrm{LL}}}_{w_{ij}^{\mathrm{P}}} \in SE(3)$
v_i
v_j

(a) Sketch of graph structure.

(b) Graph structure visualized in simulation.

Fig. 6: Graph structure: red edges denote the trajectory edges, with measurement w_{ij}^{P} potentially merged from w_{ij}^{KF} and w_{ij}^{LL}, all $\in SE(3)$, cyan edges encode the star tracker-based constraints $w_{0j}^{R} \in SO(3)$, and green edges represent constraints based on scan matching employed during loop closing, i.e., $w_{jk}^{\mathrm{LL}} \in SE(3)$.

is conducted with the inverse model, as described in Sect. 3.2, which draws on the trajectory estimate and the respective measurements in a mapping with known pose scheme [91, Chap. 9]. The joint estimation of a trajectory and a map may be formulated as

$$p(x_{0:t}, Y | z_{0:t}, u_{1:t}) \,, \tag{1}$$

where measurements for the single timepoints are denoted as $z_{0:t}$, while $u_{1:t}$ represent the respective control measurements.

Graph SLAM approaches can be roughly divided into a backend (see Sect. 4.1), where the optimization takes place and a frontend (see Sect. 4.2), where the structure of the graph $\mathcal{G}$ is constructed (see Fig. 6). The construction itself is conducted by inserting nodes v, each representing a pose estimate, into $\mathcal{G}$ and constrain them with edges w, which represent offset measurements. A star tracker–based offset measurement w_{0j}^{R} (see Sect. 3.4.2) constraints the attitude with respect to the navigation frame (see Sect. 4.2.1). In the graph, the latter is represented in terms of a node v_0 with aligned pose. In addition, it is common to employ some kind of recursive filtering technique to foster the convergence speed of the optimizer backend in Graph SLAM approaches. In a loosely coupled approach like, e.g., in Hector-SLAM [46], the filter operates outside the graph structure. It provides an initial guess for the pose of each new node and in return is updated with information from the graph. In contrast, we employ a tight-coupling in the spirit of [16], thus establishing an intertwined relation of the frontend and the backend. An Extended Kalman Filter (EKF) is used for the recursive prediction from a node v_i to the next node v_j, resulting in a relative offset measurement w_{ij}^{KF} (see Sect. 4.2.3). The latter is used to validate a scan matching attempt w_{ij}^{LL} between point clouds obtained at the nodes v_i and v_j, respectively (see Sect.4.2.4). Thus, we account for the scenario-specific observation that a successful scan matching attempt is more precise than the filter result, while a failed attempt may result in arbitrarily bad results. If validation succeeds, the scan matching result w_{ij}^{LL} is merged with the relative filter-estimate w_{ij}^{KF}, resulting in a pose constraint w_{ij}^{P} between v_i and v_j (see Fig. 6a). If validation fails, w_{ij}^{KF} is directly used as the constraint w_{ij}^{P} and scan matching is re-attempted in the next time step.

After the node v_j has been inserted into the graph, the EKF is reset by setting the position part r of the state vector to zero and the attitude part R to identity. In addition, the position and attitude related parts of the covariance Σ are set to zero as well. Thus, the latest node v_j effectively becomes the new reference pose x_0 of the filter. Accordingly, the backend is provided with a good initial guess by the filter and in return provides the filter operating in the frontend with an improved reference pose x_0 smoothed over all preceding measurements. However, leaving the velocity part of the state vector within the filter untouched would result in an accumulating dead-reckoning error (see Fig. 16f). Hence, upon successful scan matching, we additionally use the result w_{ij}^{LL} as a correctional measurement to indirectly correct the velocity. By that means, we obtain a bounded estimation error in the filter by feeding back additional information from the graph. Finally, explicit loop closing is employed to obtain additional pose constraints w_{jk}^{LL} (see Sect.4.2.5).

2.2 Autonomy

The *Autonomy* module is implemented as a set of behaviors, each related to a different aspect of the module's task. By implementing the system as a hierarchical state machine (HCSM) [1, 74], we ensure that multiple behaviors can be active at a given time, while maintaining a certain execution order. The latter is achieved by modeling multiple tracks of behavioral sequences with different pre-conditions and post-conditions and adapting this structure recursively, which results in a hierarchical sequence-tree.

The task of the *Autonomy* module is to explore, and thus establish, a consistent map of the asteroid's topology, while keeping a trade-off between localization performance and exploration speed. To this end, different trajectories are evaluated in a twofold approach within an evaluation sphere ν, which is discretized using evaluation nodes denoted by ξ (see Sect. 5.1). Based on the resultant map, the coarseness of the surface of the asteroid can be estimated, which allows for the discovery of landing sites. The latter are defined as larger, connected areas where the surface of the asteroid is smooth enough to land the S/C safely. To fulfill its tasks, the *Autonomy* module comprises the following sub-modules (see Fig. 4):

1. The *Potential Trajectory Generation* module computes a set of potential trajectories $\Upsilon = \{\tau^{\mathrm{P}}\}^{\mathcal{P}}$, with $1 \leq \mathrm{P} \leq \mathcal{P}$ (see Sect. 5.2).
2. The *Potential Trajectory Evaluation* module computes the respective values of a potential trajectory, i.e., $val(\tau^{\mathrm{P}})$ (see Sects. 5.3 and 5.4).

3. The *Optimal Trajectory Generation* module employs the nonlinear optimization problem (NLP) solver WORHP [80, 81, 11] to compute a set of optimal trajectories $\Psi = \{\tau^{\mathrm{W}}\}^{\mathcal{W}}$, with $1 \leq \mathrm{W} \leq \mathcal{W}$ (see also Sect.5.2).
4. The *Optimal Trajectory Evaluation* module determines the value of the respective optimal trajectories, i.e., $val(\tau^{\mathrm{W}})$ (again, see Sects.5.3 and 5.4).
5. The *Normal Envelope Generation* module generates a map Y^{SN} representing the coarseness c_c of a cell c of the asteroid surface calculated from smoothed local surface normals $\hat{n}_c$ (see Sect.5.6).
6. The *Landing Site Discovery* module processes the normal envelope Y^{SN}, resulting in a map Y^{LS} with annotated landing sites (again, see Sect.5.6).
7. Finally, the *Controller* module executes the pre-computed thrust controls from the chosen trajectory and compensates for derivations using a Ricatti-controller [97]. In addition, it computes the control torque for a nadir-pointing behavior [59], i.e., having the LiDAR sensor pointing to the center of mass of the asteroid.

Modules 1-4 work on demand, i.e., when a trajectory is about to be finished and has to be replaced with a new one. To this end, they generate a raw estimation of every possible trajectory from the current position estimate r_t to all possible target nodes $\xi_i \in \mathrm{E}'$ resulting in a set of potential trajectories Υ. E', in turn, is a subset of the evaluation nodes E residing on the evaluation sphere ν. Those potential trajectories are assessed by evaluating $val(\tau^{\mathrm{P}})$. The respective target nodes of the k best potential trajectories τ^{P^*} are then used to compute the same number of optimal trajectories Ψ. In a final evaluation step, the value $val(\tau^{\mathrm{W}})$ of those trajectories is assessed and the best optimal trajectory selected, which is denoted by $\tau^{\mathrm{W}*}$. In effect, the interplay of these modules establishes an active perception process. In addition, it can be adapted online based on the current pose uncertainty marginalized from the graph (see Sect. 5.5). The modules 5 and 6 are executed with a higher but fixed frequency, thus gradually re-computing the surface envelope Y^{SN} and, on top of the latter, updating the augmented map Y^{LS}. Finally, the controller module 7 is executed in every timestep as position and attitude have to be continuously controlled.

3 State, Motion, and Sensor Models

The simulation system as well as the internal modules of the S/C jointly draw on a model pool tailored to the scenario considered here. The pool comprises a manifold state representation, a map representation, a probabilistic motion model, as well as the single sensor models. The particular models are presented in detail in the following.

3.1 Manifold State Representation

For state estimation, a proper parametrization of the state space has to be chosen. Minimal parameterizations allow for the application of standard algorithms operating directly on the respective degrees of freedom, while overparametrized representations avoid singularities and dead-lock phenomena like, e.g., the gimbal lock [4]. We employ the so-called $\boxplus$-method[1] [36] to leverage the advantages of both options. On the one hand, a state is globally represented on a manifold space $\mathcal{X}$, while on the other hand, small state changes can be expressed in the tangent space $\mathbb{R}^n$, where n denotes the number of degrees of freedom in $\mathcal{X}$.

[1] pronounced “boxplus-method”

To yield an encapsulated manifold state representation, two operators $\boxplus$ and $\boxminus$ are defined. The operator $\boxplus : \mathcal{X} \times \mathbb{R}^n \to \mathcal{X}$ adds small state changes expressed in tangent space to the state defined on the manifold, while the operator $\boxminus : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^n$ expresses the difference between two states in $\mathcal{X}$ as a vector in $\mathbb{R}^n$.

The actual state space $\mathcal{S}$ is the Cartesian product of $SE(3)$, the special Euclidean group denoting a rigid transformation in three-dimensional Euclidean space, and $\mathbb{R}^3$, denoting the velocity of the S/C, which yields $\mathcal{S} = SE(3) \times \mathbb{R}^3$. $SE(3)$, in turn, is composed of a vector space $\mathbb{R}^3$ and the special orthogonal group $SO(3)$ to represent position and orientation, respectively. Accordingly, a state $x_t \in \mathcal{S}$ at time t is given by

$$x_t = \begin{bmatrix} r_t \\ R_t \\ v_t \end{bmatrix}, \tag{2}$$

where $r_t, v_t \in \mathbb{R}^3$ represent the position and the velocity, respectively, while $R_t \in \hat{\mathbb{R}}^{3\times 3}$ represents the orientation in the navigation frame represented by a Direction Cosine Matrix (DCM). All quantities are expressed in a heliocentric reference frame.

The operators defined above on arbitrary suitable manifolds can be applied to this specific state space by defining a compound operator comprising the single operators defined for the particular spaces in the state [36]. Specifically, for our state space $\mathcal{S}$, the compound operators are defined as $\boxplus_{\mathcal{S}} : \mathcal{S} \times \mathbb{R}^9 \to \mathcal{S}$ and $\boxminus_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^9$, which translates to

$$x \boxplus_{\mathcal{S}} d = \begin{bmatrix} x_r + d_{1:3} \\ x_R \boxplus_{\hat{\mathbb{R}}^{3\times 3}} d_{4:6} \\ x_v + d_{7:9} \end{bmatrix}, \qquad y \boxminus_{\mathcal{S}} x = \begin{bmatrix} y_r - x_r \\ y_R \boxminus_{\hat{\mathbb{R}}^{3\times 3}} x_R \\ y_v - x_v \end{bmatrix}, \tag{3}$$

where x, y are states in the manifold $\mathcal{S}$ and d represents a difference between them. The latter is expressed in the tangent space $\mathbb{R}^9$ of x and indicates the delta that has to be applied to x to reach y. For the position part and the velocity part of the state vector, the respective operators reduce to a simple vector operations, while $\boxplus_{\hat{\mathbb{R}}^{3\times 3}} : \hat{\mathbb{R}}^{3\times 3} \times \mathbb{R}^3 \to \hat{\mathbb{R}}^{3\times 3}$ and $\boxminus_{\hat{\mathbb{R}}^{3\times 3}} : \hat{\mathbb{R}}^{3\times 3} \times \hat{\mathbb{R}}^{3\times 3} \to \mathbb{R}^3$ are the $\boxplus$/$\boxminus$-operators for DCMs, which are based on Rodriguez' formula [58] and are derived in [36].

Finally, to represent the uncertainty of a state estimate in a manifold space $\mathcal{X}$, we use a normal distribution $\mathcal{N}_{\mathcal{X}}$ defined on that space. It can be represented in the tangent space as proposed in [36] with

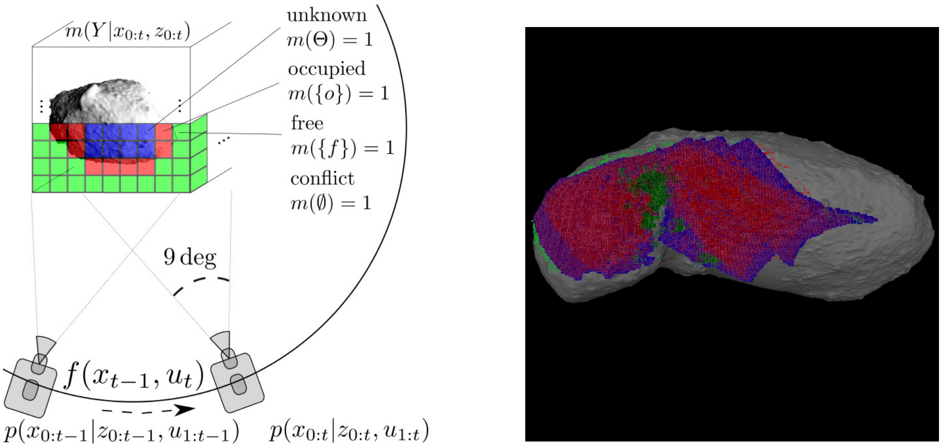
$$\mathcal{N}_{\mathcal{X}}(\mu, \Sigma) = \mu \boxplus \mathcal{N}(0, \Sigma), \tag{4}$$

where the mean μ is an element of the manifold $\mathcal{X}$ and $\mathcal{N}$ is an ordinary normal distribution defined on the vector space $\mathbb{R}^n$ with covariance $\Sigma \in \mathbb{R}^{n\times n}$.

3.2 Evidential Grid Map

The point cloud information acquired by the *Sensor Fusion* module is provided to the *Autonomy* module in the augmented form of an evidential grid map (see Figs. 5 and 7). To this end, the space to map is efficiently discretized into equally-sized cube-shaped cells, where the state of each grid cell can be either occupied (o) or free (f). The uncertainty about the estimated occupancy of a single grid cell is modeled by a belief function [83, 86] instead of probability distribution.[2]

[2] For a thorough elaboration of belief function theory, the interested reader is referred to [71, Chap. 2].



(a) General evidential SLAM setup.

(b) Partial evidential grid map of Itokawa.

Fig. 7: Evidential grid mapping for small celestial bodies.

As opposed to the relation $P(o) = 1 - P(f)$ for representing uncertainty in the probabilistic case, the evidential approach allows for a more distinguished representation of particular dimensions of uncertainty. They comprise the standard measures for occupied $m(\{o\})$ and free $m(\{f\})$, but additionally also feature measures for unknown $m(\Theta)$ and conflict $m(\emptyset)$. As a data-structure, we employ an octree, similar to the approach presented in [38].

For a brief introduction[3], the frame of discernment, i.e., the set of all mutual exclusive possibilities in the domain, is defined as $\Theta_Y = \{o, f\}$. The i-th grid cell is denoted as Y_i and it can take any value of the power set of Θ_Y, i.e., $Y_i \subseteq \Theta_Y$. Accordingly, the whole map is given by $Y \subseteq \Theta_Y^M$, where $M \in \mathbb{N}$ indicates the total number of cells. The mapping problem corresponds to estimating the mass function $m(Y|x_{0:t}, z_{0:t})$ of the map Y conditioned on the trajectory $x_{0:t}$ taken up to the current time t and a set of corresponding measurements $z_{0:t}$. By assuming conditional independence among the grid cells given the trajectory and the measurements, the joint mass function can be factorized according to

$$m(Y|x_{0:t}, z_{0:t}) = \prod_{i=0}^{M} m(Y_i|x_{0:t}, z_{0:t}) \,, \tag{5}$$

yielding M marginal mass functions. This factorization also applies to the inverse model, which provides a mass function $m(Y_i|x_t, z_t)$ for each cell Y_i, $1 \leq i \leq M$ given a state x_t and a corresponding measurement z_t [17]. We can make use of this property to update the single cells Y_i of the map recursively over time using the conjunctive rule of combination [73] denoted by $\textcircled{\cap}$:

$$\underbrace{m(Y_i|x_{0:t}, z_{0:t})}_{\text{updated grid cell}} = \underbrace{m(Y_i|x_{0:t-1}, z_{0:t-1})}_{\text{grid cell}} \textcircled{\cap} \underbrace{m(Y_i|x_t, z_t)}_{\text{inverse model}} \,. \tag{6}$$

Belief function theory allows to use other combination rules as well, a comparison of which in the context of SLAM can be found in [73]. However, the conjunctive rule is the only one that yields mass on $\emptyset$ and thus allows for a quantification of the conflict.

[3] For a comprehensive introduction to evidential grid mapping, the interested reader is referred to [17].

Based on these considerations, we can use the map as an interface by projecting the registered point clouds of the *Sensor Fusion* module recursively into the map and subsequently pass the result to the *Autonomy* module (cf. Figs. 4, 5, and 7). The latter makes use of the map distribution in a twofold way: it identifies (i) unknown areas in the map by computing an accumulated Θ_Y value as well as (ii) the relevant cells for localization and the landing site discovery by measuring the occupancy of the particular cells (see Sects. 5.3 and 5.6).

The unknown parts of a map are characterized by the absence of evidence for either being occupied or free. The amount of non-specificity of a mass function can be obtained by the generalized Hartley measure [44]. In the case of evidential grid maps, it reduces to [72]

$$NS(m) = m(\Theta_Y)\,. \tag{7}$$

The (normalized) occupancy is simply measured by the corresponding mass $m'(\{o\})$, where m' denotes the normalized mass function m.

3.3 Probabilistic Motion Model

A noisy state transition from a preceding state x_{t-1} to the current state x_t, both given in a heliocentric reference frame, is modeled by the distribution $p(x_t|x_{t-1}, u_t)$. It takes the specific form

$$x_t = g(x_{t-1}, u_t) \boxplus \epsilon_t\,, \qquad \epsilon_t \sim \mathcal{N}(0, Q_t^{\text{mm}}), \tag{8}$$

where $u_t = z_t^{\text{imu}}$ accounts for the IMU measurement, which is in turn comprised of $z_t^a \in \mathbb{R}^3$ and $z_t^\omega \in \mathbb{R}^3$ denoting the linear acceleration and the angular velocity measurement, respectively (see Sect. 3.4.1).

The actual state transition function is then given by

$$g(x_{t-1}, u_t) = x_{t-1} \boxplus \begin{bmatrix} v_{t-1} \\ z_t^\omega \\ (-a_{t-1}) + z_t^a \end{bmatrix} \Delta t\,, \tag{9}$$

where v_{t-1} denotes the velocity estimated in the preceding state x_{t-1}. Additionally, the accelerations a_{t-1} taken into account are caused by the Sun (a_{t-1}^{Sun}), the celestial bodies (a_{t-1}^{cb}), the solar pressure (a_{t-1}^{sp}), and the gravitation of the asteroid (a_{t-1}^{ast}) at time $t-1$, resulting in

$$a_{t-1} = a_{t-1}^{\text{Sun}} + a_{t-1}^{\text{cb}} + a_{t-1}^{\text{sp}} + a_{t-1}^{\text{ast}}. \tag{10}$$

In particular, the single accelerations are computed as

$$a_{t-1}^{\text{Sun}} = \frac{\mu_{\text{Sun}}}{||r_{t-1}||^3} r_{t-1}\,, \qquad a_{t-1}^{\text{cb}} = \sum_{p \in \mathcal{P}} \frac{\mu_p}{||p_{t-1} - r_{t-1}||^3}(p_{t-1} - r_{t-1})\,, \tag{11}$$

where μ_{Sun} denotes the gravitational constant of the sun, μ_p is the gravitational constant of planet p, and $p_{t-1} \in \mathbb{R}^3$ is the position of p at time $t-1$ from the set of planets P. The solar pressure a_{t-1}^{sp} [94] is approximated under the assumption of a perpendicular angle of incidence by

$$a_{t-1}^{\text{sp}} = \frac{S_E\, r_{t-1}^2}{c} C_R \frac{A}{m_{t-1}^{\text{sc}}} \frac{r_{t-1}}{|r_{t-1}|}\,, \tag{12}$$

where S_E denotes the solar constant (defined for the distance between Sun and Earth), c is the speed of light, C_R is the reflection coefficient, m_{t-1}^{sc} is the mass of the S/C at $t-1$, and finally A denotes the surface that is assumed to be affected.

The acceleration caused by the asteroid a_{t-1}^{ast} can be computed in two ways to introduce a modeling error with respect to the biggest acting force. The asteroid is considered to be a simple pointmass in the S/C's state estimation module. However, for ground truth computation the acceleration is derived from spherical harmonics [45] based on the parameters stated for Itokawa in [82]. Throughout this paper, we use a fourth order Runge Kutta (RK4) numerical integrator for the propagation of the asteroid state and the respective state of the S/C.

Finally, the covariance matrix for the motion model Q_t^{mm} in (8) is constructed from the state transition noise $A_t \in \mathbb{R}^{9\times 9}$ and the measurement noise $B_t \in \mathbb{R}^{9\times 9}$ according to

$$Q_t^{\text{mm}} = A_t + B_t = \begin{bmatrix} A_t^r & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & A_t^R & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & A_t^v \end{bmatrix} + \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & Q_t^{\omega}\Delta t & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & Q_t^{\text{a}}\Delta t \end{bmatrix} . \tag{13}$$

The blocks $A_t^r, A_t^R, A_t^v \in \mathbb{R}^{3\times 3}$ of the state transition noise denote the position, attitude, and velocity parts, respectively. The measurement noise is composed of the covariance matrices of the gyroscope $Q_t^{\omega} \in \mathbb{R}^{3\times 3}$ and the accelerometer noise $Q_t^{\text{a}} \in \mathbb{R}^{3\times 3}$ (see Sect. 3.4.1).

3.4 Sensor Models

The S/C considered in this paper features a sensor set for autonomous relative navigation as defined in [67, 68, 69]. The inertial measurement unit (IMU) provides information about the actually carried out controls, the star tracker provides an absolute attitude estimate, and overlapping LiDAR measurements provide pose offsets relative to the asteroid. In the following, the corresponding models for the particular sensors of the S/C are presented.

3.4.1 Inertial Measurement Unit

An IMU typically combines accelerometers and gyroscopes to measure the acceleration and the angular velocity of the body it is attached to. Thus, information about the actual impact of the spent torque and thrust is obtained and can be used directly for prediction in the motion model (see Sect. 3.3, in particular (8) and (9)). We model this device with the IMU measurement z_t^{imu}. It is comprised of a noisy measurement of the current ground truth linear acceleration $\hat{a}_t \in \mathbb{R}^3$ and a noisy measurement of the current ground truth angular velocity $\hat{\omega}_t \in \mathbb{R}^3$. In particular, they are given by

$$z_t^a = \hat{a}_t + \delta_t^{\text{a}} , \quad \delta_t^{\text{a}} \sim \mathcal{N}(0, Q_t^{\text{a}}) , \qquad z_t^{\omega} = \hat{\omega}_t + \delta_t^{\omega} , \quad \delta_t^{\omega} \sim \mathcal{N}(0, Q_t^{\omega}) , \tag{14}$$

where $\delta_t^{\text{a}} \in \mathbb{R}^3$ and $\delta_t^{\omega} \in \mathbb{R}^3$ denote additive, normally distributed noise with zero mean and the respective covariances $Q_t^{\text{a}} \in \mathbb{R}^{3\times 3}$ and $Q_t^{\omega} \in \mathbb{R}^{3\times 3}$.

As this is a rather simplified IMU model, further requirements like the estimation of the bias and scale factors would have to be taken into account for the application in an actual mission.

3.4.2 Star Tracker

The star tracker is basically an optical camera that observes the stars and compares them to a catalog (like *HIPPARCOS* [65] or Gaia DR2 [9]) listing their positions. By that means, an absolute attitude measurement $z_t^R \in SO(3)$ in the navigation frame can be obtained. The likelihood of the star tracker measurement is denoted by $p(z_t^R | x_t)$ and is defined as

$$z_t^R = \hat{R}_t \boxplus \delta_t^{\mathrm{R}} \,, \qquad \delta_t^{\mathrm{R}} \sim \mathcal{N}(0, Q_t^{\mathrm{R}}) \,, \tag{15}$$

where $\delta_r^R \in \mathbb{R}^3$ is normally distributed noise with zero mean, the covariance of which is given by $Q_t^{\mathrm{R}} \in \mathbb{R}^{3\times 3}$. As opposed to the measurement above, the star tracker measurement is defined on the manifold, and thus, rendering the sampling of noise a non-trivial procedure, which is described in detail in [60].

3.4.3 3D Flash LiDAR

A light detection and ranging (LiDAR) sensor is an active measurement device, which emits laser beams and senses the reflections in the outside world. Subsequently, a range value can be obtained from these observations by a time-of-flight computation. The 3D Flash-LiDAR technology [26] has been considered for numerous deep space applications, e.g., [2, 14]. Currently, NASA's OSIRIS REx mission is flying a 3D Flash LiDAR to be employed in the autonomous touch-and-go maneuver for sample retrieval [6]. The LiDAR device modeled here is coarsely inspired by the Advanced Scientific Concepts TigerEye 3D Flash LiDAR system [51], whose close related 3D space camera DragonEye has already been flying on NASA's STS 127 (Endeavor Orbiter) [14].

The LiDAR measurement z_t^{L} comprises points $z_{t,k}^{\mathrm{L}} \in \mathbb{R}^3$, $1 \leq k \leq N$ defined in the sensor frame $T_{sc_t}^{se}$, where N denotes the number of points. In addition, the distance component of every single point is affected by normally distributed noise according to

$$z_t^{\mathrm{L}} = \{\hat{z}_{t,k}^{\mathrm{L}} + [0, 0, \delta_{t,k}^{\mathrm{L}}]^\top\}^N \,, \qquad \delta_{t,k}^{\mathrm{L}} \sim \mathcal{N}(0, Q_t^{\mathrm{L}}), \qquad 1 \leq k \leq N. \tag{16}$$

The field of view is modeled as a pyramid frustum with an horizontal and vertical opening angle of 9 deg. Within the field of view, a resolution of 128 × 128 pixels is available. The measurement is affected by distance noise with covariance $Q_t^{\mathrm{L}} \in \mathbb{R}^{3\times 3}$. Furthermore, the depth image is obtained in one take by reading out the corresponding component of the fragment shader of a scene rendered on the GPU as described in the next section.

3.4.4 GPU-Based LiDAR Implementation

The computation and processing of point clouds is generally a time-consuming process. To significantly speed it up, we use a GLFW-based GPU implementation for fast computation of the respective LiDAR point clouds. The implementation is based on the GLSL shader language [75], using an approach similar to [95]. The main idea is to utilize the depth-component z, calculated in the fragment shader fs, instead of the z-buffer. We hence create a 32 bit RBG floating point type frame buffer fb with the desired dimensions (width x, height y) of the LiDAR sensor array and calculate the corresponding distances z.

In the vertex shader, we initially calculate the position of the i-th vertex $ve_{ast_t}^i \in \mathbb{R}^4$ of the asteroid model relative to the sensor frame of reference using

$$ve_{se}^i = T_{se}^{Sun}\, T_{Sun}^{ast_t}\, ve_{ast_t}^i \,, \tag{17}$$

where $T_{se}^{Sun} \in \mathbb{R}^{4\times 4}$ denotes the view matrix and $T_{Sun}^{ast_t} \in \mathbb{R}^{4\times 4}$ the current asteroid pose in homogeneous coordinates. As we calculate asteroid-centric, only the rotational component of this matrix changes. We interpret the entries $\hat{f}b_{x,y}$ in the frame buffer as the single ground truth LiDAR measurements $\hat{z}_{t,k}^{\mathrm{L}}$. To this end, we compute the particular pixel values in the sensor frame using the fragment shader interpolating the position between two vertices ve_{se}^{i} and v_{se}^{j} according to

$$\hat{f}b_{x,y} = [x, y, -z]^T = fs(ve_{se}^{i}, v_{se}^{j})_{se}\,, \tag{18}$$

where the sign of the z-axis is flipped to express the distance in terms of positive numbers.

Finally, the noise introduced in (16) is applied in a separate shader pass (and a separate shader) to yield a noisy measurement $fb_{x,y}$. In addition, not only the generation of the point clouds but also its counterpart the scan matching approach is run on the GPU (see Sect. 4.2.4).

4 Graph SLAM

A detailed description of the Graph SLAM approach as employed in the *Sensor Fusion* module is given in this section. We start by briefly describing the optimization backend in Sect. 4.1 and then proceed to a thorough discussion of the graph construction frontend in Sect. 4.2. The latter comprises a description of the pose and attitude edges in Sects. 4.2.1 and 4.2.2, respectively, and of the Extended Kalman Filter (EKF) in Sect. 4.2.3. Additionally, the scan matching approach is presented in Sect. 4.2.4, which is used for scan-to-scan correction and loop closing as discussed in Sect. 4.2.5.

4.1 Graph SLAM Backend

The posterior of the SLAM problem naturally forms a sparse graph [30], which is constructed as described in Sect. 4.2. As backend implementation, we use the g^{2}o-Framework [49], which makes efficient use of the sparse structure of the graph. An edge encodes a constraint between two vertices v_i and v_j, which is imposed by a differential measurement w_{ij} and weighted by a respective information matrix $\mathit{\Omega}_{ij}$ obtained by the inverse of the measurement covariance, i.e., $\mathit{\Omega}_{ij} = \mathit{\Sigma}_{ij}^{-1}$. A corresponding error function quantifies the tension between two vertices by computing an expected delta $\bar{w}(v_i, v_j)$ between the nodes and subsequently comparing it to the actually measured delta w_{ij} [34], which results in

$$e(v_i, v_j, w_{ij}) = w_{ij} - \bar{w}(v_i, v_j)\,. \tag{19}$$

If the vertices are defined on a manifold, the error function has to be locally linearized for optimization. Thus, we adapted g^{2}o with individual edges (see Sects. 4.2.1 and 4.2.2) in order to fit our scenario and the $\boxplus$-based EKF (see Sect. 4.2.3). A $\boxplus$-based error function additionally avoids a covariance transformation as needed for the original $\oplus$-based edges [87, 34] (cf. (4)).

This interpretation of Graph SLAM results in a set of nonlinear quadratic constraints, which can be used in an objective function $F : \mathcal{S}'^{n} \rightarrow \mathbb{R}$ of the form

$$F(x) = \sum_{\langle i,j\rangle \in C} e(v_i, v_j, w_{ij})^{\top} \mathit{\Omega}\, e(v_i, v_j, w_{ij}) = \sum_{\langle i,j\rangle \in C} e_{ij}(v_i, v_j)^{\top} \mathit{\Omega}\, e_{ij}(v_i, v_j)\,, \tag{20}$$

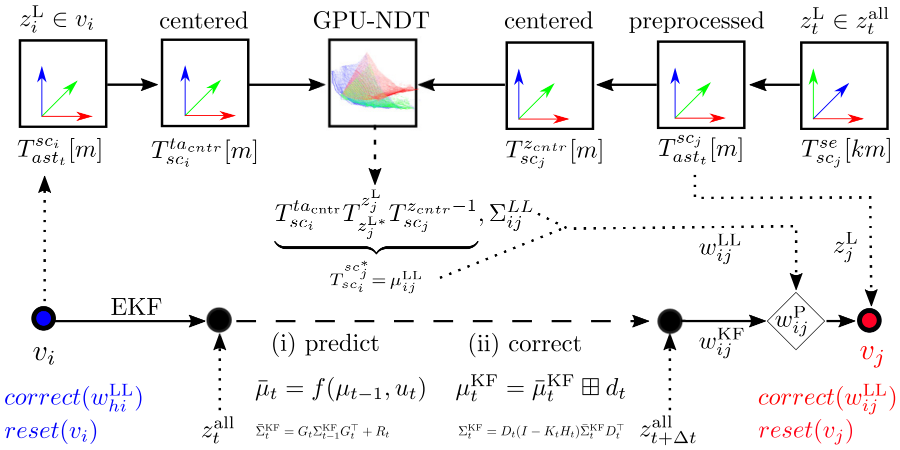

$z_i^{\mathrm{L}} \in v_i$
centered
GPU-NDT
centered
preprocessed
$z_t^{\mathrm{L}} \in z_t^{\mathrm{all}}$
$T_{ast_t}^{sc_i}[m]$
$T_{sc_i}^{ta_{cntr}}[m]$
$T_{sc_j}^{z_{cntr}}[m]$
$T_{ast_t}^{sc_j}[m]$
$T_{sc_j}^{se}[km]$
$\underbrace{T_{sc_i}^{ta_{\mathrm{cntr}}} T_{z_j^{\mathrm{L}*}}^{z_j^{\mathrm{L}}} T_{sc_j}^{z_{cntr}-1}}_{T_{sc_i}^{sc_j^*} = \mu_{ij}^{\mathrm{LL}}}, \Sigma_{ij}^{LL}$
w_{ij}^{LL}
z_j^{L}
EKF
w_{ij}^{P}
v_i
(i) predict
(ii) correct
w_{ij}^{KF}
v_j
$\bar{\mu}_t = f(\mu_{t-1}, u_t)$
$\mu_t^{\mathrm{KF}} = \bar{\mu}_t^{\mathrm{KF}} \boxplus d_t$
$correct(w_{hi}^{\mathrm{LL}})$
$reset(v_i)$
z_t^{all}
$\bar{\Sigma}_t^{\mathrm{KF}} = G_t \Sigma_{t-1}^{\mathrm{KF}} G_t^\top + R_t$
$\Sigma_t^{\mathrm{KF}} = D_t(I - K_t H_t)\bar{\Sigma}_t^{\mathrm{KF}} D_t^\top$
$z_{t+\Delta t}^{\mathrm{all}}$
$correct(w_{ij}^{\mathrm{LL}})$
$reset(v_j)$

Fig. 8: Tightly-coupled graph construction between a node v_i and v_j resulting in an edge with a constraining measurement w_{ij}^{P}, which is a pose offset, merged from a recursive state estimate w_{ij}^{KF} and a scan matching result w_{ij}^{LL}.

where C denotes the set of constrained index pairs, $\mathcal{S}'$ denotes the state space $\mathcal{S}$ without the velocity part (see Sect. 3.1), and with $e(v_i, v_j, w_{ij}) := e_{ij}(v_i, v_j)$. This function can be minimized in order to obtain a set of optimized poses and, as the transformations for the respective LiDAR measurements are known, a corresponding map. Specifically, the minimization yields an optimized vector of parameters of the particular vertices $x^* = [v_0^\top, \cdots, v_n^\top]^\top \in \mathcal{S}'^n$. Hence, we get

$$x^* = \underset{x}{\operatorname{argmin}} \, F(x)\,. \tag{21}$$

For optimization, we use the csparse numerical optimizer [22] employing the Levenberg-Marquardt algorithm on the sparse system of equations.

4.2 Graph SLAM Frontend

In the Graph SLAM frontend, the graph structure $\mathcal{G}$ is constructed such that it can be optimized in the backend (see Sect.4.1). To this end, the current measurements z_t^{all} are processed within a tightly-coupled graph construction approach as introduced in Sect. 2.1. For initialization, we insert a fixed node v_0 into the graph, whose pose is aligned with the navigation frame to serve as a reference for the attitude edges encoding star tracker measurements (see Sect. 4.2.1). Afterwards, Alg. 1 is executed in a processing loop. The algorithm performs the following steps.

1. Every loop starts in line 1 by feeding the asteroid state as a measurement z_t^{ast}, to update the pose of the asteroid in the integrator used for prediction in the EKF (cf. Sects. 3.3 and 4.2.3).
2. Subsequently, the IMU measurement z_t^{imu} is processed in line 2 as u_t in the state transition (see Sect.4.2.3, (28)). It thus incorporates the measured effect of the controls, which comprises torque and thrust applied to the S/C (see Sect. 3.4.1).

Algorithm 1: Tightly-Coupled Asteroid Graph SLAM Frontend

```
   Input: $\mathcal{G}_{t-1}, \mu^{\text{KF}}_{t-1}, \Sigma^{\text{KF}}_{t-1}, z^{\text{all,}}_t$ ;   // graph, EKF estimate, measurements
1  $x^{\text{ast}}_t \leftarrow z^{\text{ast}}_t$ ;   // update asteroid state, used in integrator
2  $\bar{\mu}^{\text{KF}}_t \leftarrow g(\mu^{\text{KF}}_{t-1}, u_t)$ ;   // EKF mean prediction
3  $\bar{\Sigma}^{\text{KF}}_t \leftarrow G_t \Sigma^{\text{KF}}_{t-1} G^{\top}_t + R_t$ ;   // EKF covariance prediction
4  $K^R_t = \bar{\Sigma}^{\text{KF}}_t H^{R\top}_t (H^R_t \bar{\Sigma}^{\text{KF}}_t H^{R\top}_t + Q^R_t)^{-1}$;   // compute Kalman gain
5  $\mu^{\text{KF}}_t \leftarrow \bar{\mu}^{\text{KF}}_t \boxplus K^R_t(z^{\text{R}}_t \boxminus h(\bar{\mu}^{\text{KF}}_t))$ ;   // EKF mean correction w/ star tracker
6  $\Sigma^{\text{KF}}_t \leftarrow D_t(I_{9\times 9} - K^R_t H^{\text{R}}_t)\bar{\Sigma}^{\text{KF}}_t D^{\top}_t$ ;   // EKF covariance correction
7  if $|r_t| > \epsilon_{trans} \vee |R_t| > \epsilon_{rot} \vee \#steps > \epsilon_{steps} \vee do\_scan\_matching$ then
8      $\mu^{\text{LL}}_t, \Sigma^{\text{LL}}_t \leftarrow$ GPU_NDT($z^{\text{L}}_i, z^{\text{L}}_t$);   // scan matching
9      $scan\_validated \leftarrow \big(|r^{\text{LL}}_t - r^{\text{KF}}_t| \leq \epsilon_{\text{validateTrans}}\big) \wedge \big(|R^{\text{LL}}_t \boxminus R^{\text{KF}}_t| \leq \epsilon_{\text{validateRot}}\big)$ ;
10     if $scan\_validated$ then
11         $K^{LL}_t = \Sigma^{\text{KF}}_t H^{LL\top}_t (H^{LL}_t \Sigma^{\text{KF}}_t H^{LL\top}_t + \Sigma^{LL}_{ij})^{-1}$;   // compute Kalman gain
12         $\mu^{\text{KF}}_t \leftarrow \mu^{\text{KF}}_t \boxplus K_t(\mu^{\text{LL}}_{ij} \boxminus \mu^{\text{KF}}_t)$ ;   // EKF correction with $\mu^{\text{LL}}_{ij}$
13         $\Sigma^{\text{KF}}_t \leftarrow D_t(I_{9\times 9} - K^{LL}_t H^{\text{LL}}_t)\Sigma^{\text{KF}}_t D^{\top}_t$ ;   // EKF covariance correction
14         $do\_scan\_matching \leftarrow$ False ;   // no scan matching in next step
15     end
16     else
17         $do\_scan\_matching \leftarrow$ True ;   // scan matching in next step
18     end
19     $\mu^{\text{P}}_t \leftarrow \mu^{\text{KF}}_t, \quad \Omega^{\text{P}}_t \leftarrow \big(\Sigma^{\text{KF}}_t\big)^{-1}$ ;   // constraint $w^{\text{P}}_{ij}$: (corr.) EKF estimate
20     $\mathcal{G}_t \leftarrow$ add_node($\mathcal{G}_{t-1}, j, \mu^{\text{P}}_t$) ;   // insert node $v_j$ with pose $\mu^{\text{P}}_t$
21     $\mathcal{G}_t \leftarrow$ add_edge $\Big(\mathcal{G}_t, 0, j, z^R_t, \big(Q^{\text{R}}_t\big)^{-1}\Big)$ ;   // insert attitude edge $w^{\text{R}}_{0j}$
22     $\mathcal{G}_t \leftarrow$ add_edge($\mathcal{G}_t, i, j, \mu^{\text{P}}_t, \Omega^{\text{P}}_t$) ;   // insert pose edge $w^{\text{P}}_{ij}$
23     $r_t \leftarrow [0,0,0]^{\top}, \qquad R_t \leftarrow I_{3\times 3}$ ;   // reset $\mu^{\text{KF}}_t$ relative to $v_j$
24     $\Sigma^{\text{KF}}_t \leftarrow \begin{bmatrix} 0_{6\times 6} & 0_{3\times 6} \\ 0_{6\times 3} & \Sigma^{\text{KF}}_{t,7:9,7:9} \end{bmatrix}$ ;   // reset $\Sigma^{\text{KF}}_t$ relative to $v_j$
25     if $scan\_validated$ then
26         $\mathcal{G}_t, loop\_closed \leftarrow$ validateNearest($\mathcal{G}_t, v_j$);   // attempt loop closing
27     end
28     else
29         $loop\_closed \leftarrow$ False;
30     end
31     if $last\_optimization > \epsilon_{optimize} \vee loop\_closed$ then
32         $\mathcal{G}^*_t \leftarrow$ optimize($\mathcal{G}_t$);   // call optimization in backend
33         $last\_optimization \leftarrow t$;
34     end
35 end
36 $\#steps \leftarrow \#steps + 1$;
37 return $\mathcal{G}^*_t$ ;   // updated and potentially optimized graph
```

3. In every step, lines 5 and 6 perform an attitude correction of the EKF by means of the star tracker measurement z^{R}_t (see Sect.4.2.3, (30)–(33)).
4. The subsequent line 7 checks if the S/C traveled a certain distance $> \epsilon_{\text{trans}}$, or changed its attitude $> \epsilon_{\text{rot}}$ or has not been scan matching for a number of steps $> \epsilon_{\text{steps}}$, or the scan matching result could not be validated in the last loop (cf. lines 9, 14, and 17):
   (a) Initially, scan matching is carried out between the point cloud z^{L}_i stored in the preceding node v_i and the current point cloud z^{L}_t in line 8 (see Sect.4.2.4 and cf. Figs.8 and 9). This results in a relative pose constraint w^{LL}_{ij} comprised of the transformation μ^{LL}_t and a corresponding covariance Σ^{LL}_t.

(b) If the scan is validated in line 9, i.e., the translation and rotation deltas are smaller than their respective thresholds $\epsilon_{\text{validateTrans}}$ and $\epsilon_{\text{validateRot}}$:
   i. The Kalman gain for the subsequent correction of the EKF with w_{ij}^{LL} is computed in line 11 (see Sect.4.2.3, (30)–(34) and (36)).
   ii. The actual correction is carried out in lines 12 and 13 to attain two goals in one step. First, the scan matching result w_{ij}^{LL} is merged with the EKF-estimate w_{ij}^{KF} resulting in the pose measurement w_{ij}^{P}, as obtained in line 19. Second, the velocity part of the state estimate is indirectly corrected to yield a bounded velocity error, as only the pose of the EKF is reset with respect to the last node in lines 23 and 24.
   iii. Finally, a flag is set in line 14, indicating that the scan matching procedure was successful.

(c) Else a flag is set in line 17, resulting in another scan matching attempt in the subsequent time step.

(d) In line 19 the constraint w_{ij}^{P} is obtained from the EKF estimate. Whether it incorporates the scan matching result depends on the successful validation of the scan matching attempt in line 9.

(e) The lines 20–22 insert a new node v_j, with the pose μ_t^{P}, into $\mathcal{G}$. Additionally, an attitude edge w_{0j}^{R} between v_0 and v_j (see Sect. 4.2.1), as well as a pose edge w_{ij}^{P} between v_i and v_j (see Sect.4.2.2) are added to the graph.

(f) Afterwards, the EKF is reset to the pose of v_j in lines 23 and 24, respectively (see Fig.8).

(g) In the final step after successful scan matching, a loop closing attempt with respect to the new node v_j is triggered in line 26 (see Sect.4.2.5).

5. Finally, if a loop was closed or the optimization has not been carried out since some $\epsilon_{\text{optimze}}$, the optimization backend is called (see Sect.4.1 and cf. lines 26, 32, and 33).

4.2.1 $SO(3)$ Attitude Edge

The measurement $z_t^R \in SO(3)$ is taken by the star tracker and is represented as a DCM (see Sect.3.4.2). It is used in the filtering as well as in the smoothing step. In the former, it serves as a correctional measurement in the Kalman filter (see Sect.4.2.3), while in the latter it serves as a differential measurement w_{0j}^{R} for $SO(3)$ edges constraining the attitude part of two vertices defined in $SE(3)$ (see the cyan edges in Fig.6a). However, as the EKF estimate is treated as a relative transformation it does not reflect the absolute nature of the star tracker measurements. For that reason and due to the fact that only a small portion of the star tracker measurements are actually added to the graph, the negative effects of considering them twice can be neglected.

For graph construction, we initially insert a fixed node v_0, which can not be altered in the optimization process and whose pose is aligned with the navigation frame. A $SO(3)$ attitude edge from v_0 to v_j is then inserted each time a new $SE(3)$ pose edge (see Sect.4.2.2) between v_i and v_j is inserted into the graph. For the attitude, the error function $e : \hat{\mathbb{R}}^{3\times3} \times \hat{\mathbb{R}}^{3\times3} \times \hat{\mathbb{R}}^{3\times3} \rightarrow \mathbb{R}^3$ is then defined as

$$e(v_0, v_j, w_{0j}^{\text{R}}) = w_{0j}^{\text{R}} - \bar{w}_{0j}^{\text{R}}(v_0, v_j) = \left(R_{v_j}^{-1}\, R_{v_0}\right) \boxminus_{\hat{\mathbb{R}}^{3\times3}} z_t^{\text{R}}\,, \tag{22}$$

where, again, the attitude of v_0 is aligned with the navigation frame, and thus $R_{v_0} = I$. The $\boxminus$-operator employed here is the one defined for DCMs (see Sect.3.1). It maps the error into the vector space $\mathbb{R}^3$ and thus enables the local linearization of the error in the optimizer backend (see Sect.4.1).

4.2.2 $SE(3)$ Pose Edge

The pose edge $w^{\text{P}}_{ij} \in SE(3)$ fully constrains the relation of the vertices $v_i, v_j \in SE(3)$. (see red edge in Fig.6a). The error function $e : \mathcal{S}' \times \mathcal{S}' \times \mathcal{S}' \rightarrow \mathbb{R}^6$ for poses is thus defined as

$$e(v_i, v_j, w^{\text{P}}_{ij}) = w^{\text{P}}_{ij} - \bar{w}^{\text{P}}_{ij}(v_i, v_j) = \left(\begin{bmatrix} R_{v_j} & r_{v_j} \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} R_{v_i} & r_{v_i} \\ 0 & 1 \end{bmatrix} \right) \boxminus_{\mathcal{S}'} \mu^{\text{P}}_t \,, \tag{23}$$

where $\boxminus_{\mathcal{S}'}$ denotes the compound $\boxminus_{\mathcal{S}}$-operator defined on the manifold but without the velocity part of the state (cf. Sect. 3.1). Again, to enable a local linearization of the error, it is mapped to vector space $\mathbb{R}^6$ by this operator. $R_{v_\circ}$ and $r_{v_\circ}$ denote the attitude and position part of the i-th and j-th node, respectively.

If scan matching fails and only one measurement constraints the vertices, we directly use the pose estimated by the Kalman filter as the virtual measurement (see Sect. 4.2, Step 4c). If scan matching is successful, we simply could add two edges connecting the same pair of nodes, one for the EKF estimate and one for the scan matching result. However, as every additional edge would slow down the optimization process, we merge both edges weighted by their respective uncertainty as introduced in [16]. More precisely, the Kalman filter estimate is merged with the scan matching result. If we formulate this merging process recursively using Bayes rule and additionally note that both measurements are normally distributed, we can draw on the basic Kalman filter update equations [16, 91, Chap. 3.2.1]. As the mean is defined on a manifold, we use a $\boxplus$-Kalman filter for its update [18, 60], which gives us

$$\mu^{\text{P}}_t = \mu^{\text{KF}}_t \boxplus_{\mathcal{S}'} \left(K_t \left(\mu^{\text{LL}}_t \boxminus_{\mathcal{S}'} \mu^{\text{KF}}_t \right) \right) \,, \tag{24}$$

where K_t denotes the Kalman gain, which can be obtained by

$$K_t = \varSigma^{\text{KF}}_t H^{\text{LL}\top}_t (H^{\text{LL}}_t \varSigma^{\text{KF}}_t H^{\text{LL}\top}_t + \varSigma^{\text{LL}}_t)^{-1} \,, \tag{25}$$

where the Jacobian of the measurement function, which maps a state from the state manifold to measurement space, is denoted by $H^{\text{LL}}_t = \frac{\partial h}{\partial \bar{\mu}_t} \in \mathbb{R}^{6\times 9}$ (see (36)). For the merger of the covariances into the information matrix $\varOmega^{\text{P}}_t$, we also employ the corresponding equation from the $\boxplus$-Kalman filter [18, 60], which yields

$$(\varOmega^{\text{P}}_t)^{-1} = \varSigma^{\text{P}}_t = D_t (I_{9\times 9} - K^{LL}_t H^{\text{LL}}_t) \varSigma^{\text{KF}}_t D^{\top}_t \,, \tag{26}$$

where the Jacobian D_t is as stated in (34). In practice, however, we update the EKF with a valid scan matching result anyway (see Sect.4.2.3 and Alg. 1, lines 12 and 13). Accordingly, we can skip the explicit merging here and use the updated EKF estimate $(\mu^{\text{KF}}_t, \varSigma^{\text{KF}}_t)$ instead.

4.2.3 Extended Kalman Filter

To recursively estimate the pose offset between two nodes v_i and v_j we use an Extended Kalman Filter (EKF) operating on the $\boxplus$-manifold as presented in [18, 60]. It estimates the posterior $p(x_t | u_{1:t}, z^{\text{all}}_{0:t})$ by assuming it to be normally distributed according to

$$x_t | u_{1:t}, z^{\text{all}}_{0:t} \sim \mathcal{N}(\mu^{\text{KF}}_t, \varSigma^{\text{KF}}_t) \,, \tag{27}$$

where the parametrized result is composed of $\mu^{\text{KF}}_t \in \mathcal{S}$ and $\varOmega^{\text{KF}}_t = (\varSigma^{\text{KF}}_t)^{-1} \in \mathbb{R}^{9\times 9}$. In general, the filter process consists of a prediction and a correction step.

The prediction of the mean $\bar{\mu}_t$ and the linearized prediction of $\bar{\Sigma}_t$ take the following form on the manifold

$$\bar{\mu}_t = g(\mu_{t-1}^{\mathrm{KF}}, u_t)\,, \qquad \bar{\Sigma}_t = G_t \Sigma_{t-1}^{\mathrm{KF}} G_t^\top + R_t\,, \tag{28}$$

where the predicted values are defined in the same spaces as their counterparts, i.e., $\bar{\mu}_t \in \mathcal{S}$ and $\bar{\Sigma}_t \in \mathbb{R}^{9\times 9}$. The state transition function g is defined as stated in (9). For the covariance prediction, this function is linearized around μ_{t-1} using its Jacobian $G_t \in \mathbb{R}^{9\times 9}$, which is given by

$$\frac{\partial g}{\partial \mu_{t-1}} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & I_{3\times 3}\Delta t \\ 0_{3\times 3} & \exp -\Delta R_t & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3} \end{bmatrix}\,, \tag{29}$$

where ΔR_t denotes the attitude delta (see (9)).

In the following, we state the general definition of the correction step of an EKF on a $\boxplus$-manifold [18, 60] and subsequently provide further specifications for the actual correctional measurements z_t^R and z_t^{LL}. Let $z_t \in \mathcal{Z}$ be a correctional measurement with measurement space $\mathcal{Z}$, whose degrees of freedom are given by n and $Q \in \mathbb{R}^{n\times n}$ being its covariance. Then, the Kalman gain $K_t \in \mathbb{R}^{9\times n}$ can be obtained by

$$K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}, \tag{30}$$

where $H_t = \frac{\partial h}{\partial \bar{\mu}_t} \in \mathbb{R}^{n\times 9}$ denotes the Jacobian of the measurement function $h : \mathcal{S} \to \mathcal{Z}$, which maps from the state space manifold to measurement space. To obtain the state change $d_t \in \mathbb{R}^9$ induced by the correction, we can weigh the difference between the expected measurement $h(\bar{\mu}_t)$ and the actual measurement z_t with the Kalman gain, which gives us the innovation

$$d_t = K_t(z_t \boxminus h(\bar{\mu}_t))\,. \tag{31}$$

With the innovation of the measurement, we can apply an update to the predicted mean $\bar{\mu}$ on the manifold using

$$\mu_t^{\mathrm{KF}} = \bar{\mu}_t \boxplus d_t\,. \tag{32}$$

Finally, the covariance is updated and rotated relative to the new μ_t^{KF} [60] by applying

$$\Sigma_t^{\mathrm{KF}} = D_t(I_{9\times 9} - K_t H_t)\bar{\Sigma}_t D_t^\top\,, \tag{33}$$

where the Jacobian D_t contains the partial derivatives for (32) developed around $\bar{\mu}_t$. Thus, we have

$$D_t = \frac{\partial \bar{\mu}_t \boxplus d_t}{\partial \bar{\mu}_t} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \exp -d_{t;4:6} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3} \end{bmatrix}\,, \tag{34}$$

where $d_{t;4:6}$ denotes the elements in d_t describing the change in attitude.

In addition, two particular Jacobians H_t have to be specified for the correction steps. Namely, these are H_t^{R} for the correction based on the star tracker measurement z_t^R and H_t^{LL} for the correction carried out with the scan matching result considered as a measurement w_{ij}^{LL}. As the sensor model for the former is linear with respect to the attitude of some state x_t (see (15)), the corresponding Jacobian $H_t^{\mathrm{R}} \in \mathbb{R}^{3\times 9}$ is simply the identity matrix, i.e.,

$$H_t^{\mathrm{R}} = \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} & 0_{3\times 3} \end{bmatrix}\,. \tag{35}$$

The same holds for position and attitude part of $H_t^{\mathrm{LL}} \in \mathbb{R}^{6\times 9}$, which results in

$$H_t^{\mathrm{LL}} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & I_{3\times 3} & 0_{3\times 3} \end{bmatrix}\,. \tag{36}$$

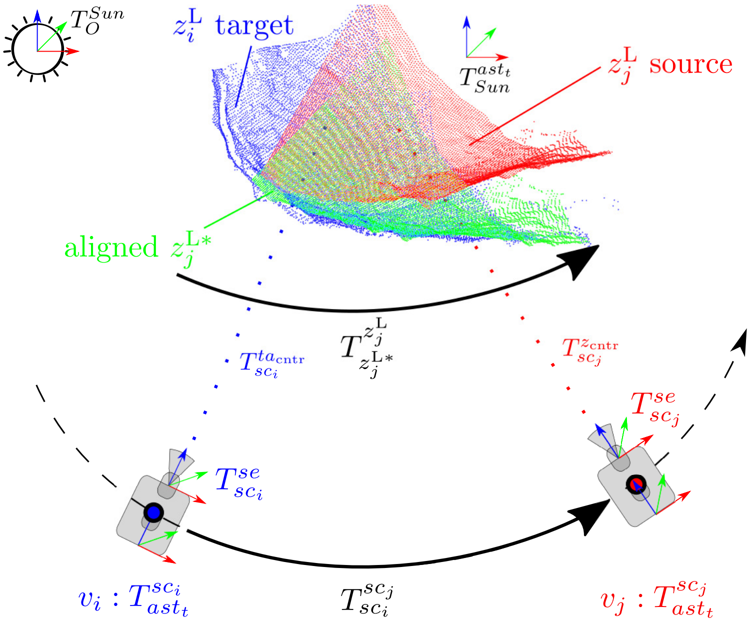

T_O^{Sun}
z_i^{L} target
$T_{Sun}^{ast_t}$
z_j^{L} source
aligned $z_j^{\mathrm{L}*}$
$T_{z_j^{\mathrm{L}*}}^{z_j^{\mathrm{L}}}$
$T_{sc_i}^{ta_{\mathrm{cntr}}}$
$T_{sc_j}^{z_{\mathrm{cntr}}}$
$T_{sc_j}^{se}$
$T_{sc_i}^{se}$
$v_i : T_{ast_t}^{sc_i}$
$T_{sc_i}^{sc_j}$
$v_j : T_{ast_t}^{sc_j}$

Fig. 9: Scan matching geometry showing a scan matching process for measurements z_i^{L} and z_j^{L}, to obtain a relative pose offset $T_{sc_i}^{sc_j}$ between the nodes v_i and v_j.

4.2.4 Scan Matching

Scan matching is a technique for aligning overlapping point clouds, stemming from two different LiDAR measurements, with each other. It is crucial for the approach presented in this paper as it is used for scan-to-scan alignment, the result of which is used as a pose constraint between two nodes v_i and v_j in the graph (see Sect. 4.2.2) and to correct the Kalman filter itself (see Sect. 4.2.3 and cf. Sect. 2.1). In addition, it is essential for explicit loop closing (see Sect. 4.2.5), which is a prerequisite for a globally consistent map within a Graph SLAM algorithm.

In general, the point cloud registration problem is posed as finding an alignment between a target cloud z_i^{L} and a source cloud z_j^{L} (see Fig. 9). As introduced in Sect. 3.4.3, each of the clouds contain a set of points $z_{t,k}^{\mathrm{L}} \in \mathbb{R}^3$ to represent the x-, y-, and z-coordinate of an underlying surface sample. Specifically, we will employ pairwise alignments to find a transformation $T_{z_j^{\mathrm{L}*}}^{z_j^{\mathrm{L}}}$ that minimizes some error function directly or indirectly defined on the points $z_{i,k}^{\mathrm{L}}$ and $z_{j,k}^{\mathrm{L}}$ when applied to the source cloud z_j^{L}.

We evaluated various ICP and GICP pipelines using implementations from the Mobile Robot Programming Toolkit (MRPT) [15] as well as from the well-established Point Cloud Library (PCL) [77, 78]. However, we could not reach the desired scan matching quality in a distance of 1 km from the asteroid (see Fig. 1b). We finally obtained satisfying results using Normal Distribution Transform (NDT), which was introduced in [8] as a compact representation of surfaces using Gaussians. For actual scan matching, a mixture model of a normal distribution and a uniform distribution is used to account for outliers in scans [7]. This approach was extended to 3D-NDT [52], which is a similar approach to correlative scan matching [64, 63]. The former discretizes the 3D space occupied by the target cloud z_i^{L} into a grid and estimates the parameters for the mixture model in each grid cell. The log-likelihood of being drawn from one of these distributions can now be computed for each of the scan points from the source cloud $z^{\mathrm{L}}_{j,k}$. By approximating the log-likelihood with a Gaussian $\tilde{p} \sim \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$ [52], an optimization problem with simple derivatives can be formulated according to

$$T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}*}} = \operatorname*{arg\,min}_{x,y,z,\alpha,\beta,\gamma} \; -\sum_{k=0}^{K} \tilde{\phi}\left(T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}\prime}} \, z^{\mathrm{L}}_{j,k}; \tilde{\mu}, \tilde{\Sigma} \right) , \tag{37}$$

which minimzes a score function by choosing a parametrization $[x, y, z, \alpha, \beta, \gamma]^\top \in \mathbb{R}^6$ for the optimal source cloud pose $T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}*}}$. To this end, the single points of a cloud are first transformed with a candidate pose $T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}\prime}} \, z^{\mathrm{L}}_{j,k}$. Subsequently, the likelihood is obtained by evaluating $\tilde{\phi}$, which is the probability density function of $\tilde{p}$.

However, while the results were satisfying, the computational demands made the approach infeasible. Taking into account the advances in GPU computing onboard of S/C [48], we choose a GPU-accelerated version of NDT as provided by [41, 42]. In addition, we hardened the GPU-NDT pipeline by enabling it to recover from errors resulting from the versatile inputs during a run.

For Graph SLAM, we want to compute a virtual measurement and a respective information matrix. To obtain the former, we transform the point clouds to their respective centers before the actual scan matching step and compute the transformation $T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}*}}$ as described above using GPU-NDT. Subsequently, we can compute the offset between the poses sc_i and sc_j (cf. Figs. 8 and 9) by

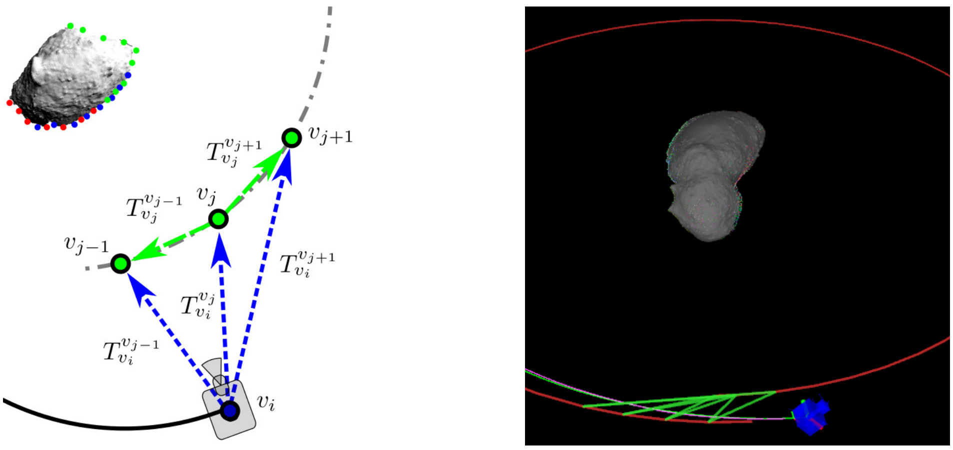
$$\underbrace{T^{sc_j^*}_{sc_i}}_{\mu^{\mathrm{LL}}_{ij}} = T^{ta_{\mathrm{cntr}}}_{sc_i} \, T^{z_j^{\mathrm{L}}}_{z_j^{\mathrm{L}*}} \, {T^{z_{\mathrm{cntr}}}_{sc_j}}^{-1} . \tag{38}$$

In addition, we pass the corresponding filtered estimate μ^{KF}_{ij} between the nodes v_i and v_j as an initial guess to GPU-NDT to improve the convergence speed. Afterwards, it is also used for validation by ensuring that the difference between the scan matching result and the EKF estimate is below a certain threshold in terms of attitude and translation (cf. Sect. 4.2, Step 4b and Alg. 1, line 9). If the validation is not successful, the scan matching result is rejected and scan matching is repeated in the next step (see Sect. 4.2, Step 4c).

The scan matching covariance $\Sigma^{\mathrm{LL}}_{ij}$ is estimated in the form of a normal distribution following [32]. However, we omit the evaluation of the motion model and replace the matching of z_j^{L} against an entire map m by scan matching a point cloud z_i^{L} from some node v_i. To evaluate the sensor model around the scan matching result small-pose offsets are iterated. This is done by computing a fixed number of samples K and letting x_k be the k-th evaluation of μ^{LL}_{ij}, which is given by $x_k = \mu^{\mathrm{LL}}_{ij} \boxplus_{\mathcal{S}'} [x_k, y_k, z_k, \alpha_k, \beta_k, \gamma_k]^\top$. Then, we can obtain the mean by

$$\tilde{\mu}^{\mathrm{LL}}_{ij} = \frac{1}{\eta} \sum_{k=1}^{K} x_k \, p(z_j^{\mathrm{L}} | z_i^{\mathrm{L}}, x_k) . \tag{39}$$

The measurement likelihood $p(z_i^{\mathrm{L}} | z_j^{\mathrm{L}}, x_k)$ is pre-calculated as a likelihood field [91, Chap. 6.4] and subsequently stored in a lookup table (LUT) [90, 54]. Thus, it can be efficiently retrieved per scan point with $\mathcal{O}(1)$. The value $\tilde{\mu}^{\mathrm{LL}}_{ij}$ computed here is considered only for the covariance



(a) Sketch of approach between a node v_i and some candidate node v_j. The blue, dashed lines are determined by scan matching, while the green lines are defined in the graph $\mathcal{G}$.

(b) Resulting loops in simulation.

Fig. 10: Validate-nearest loop closing.

computation in the following equation (40), while the mean actually used in the constraint w_{ij}^{LL} is the scan matching result μ_{ij}^{LL}. The covariance to be used as the information matrix $\varOmega_{ij}^{\text{LL}} = \left(\varSigma_{ij}^{\text{LL}}\right)^{-1}$ is obtained by

$$\varSigma_{ij}^{\text{LL}} = \frac{1}{\eta} \sum_{k=1}^{K} p(z_j^{\text{L}} | z_i^{\text{L}}, x_k) \left(x_k \boxminus \tilde{\mu}_{ij}^{\text{LL}}\right) \left(x_k \boxminus \tilde{\mu}_{ij}^{\text{LL}}\right)^\top . \tag{40}$$

Finally, the normalization factor is simply given by the sum over the likelihoods of the particular poses as given by the LUT, i.e., $\eta = \sum_{k=1}^{K} p(z_j^{\text{L}} | z_i^{\text{L}}, x_k)$.

4.2.5 Validate-Nearest Loop Closing

Loop closing is a very important step in SLAM, as it *ties* loose ends of the trajectory to already known parts after having been traveling through a previously unknown area. This is a prerequisite for a globally consistent map estimate. We use a *validate-nearest* loop closing approach in this paper. Following [64], it tries to validate the nearest node by computing transformations ranging over its respective neighbors (see Fig. 10). This approach can be implemented as shown in Alg. 2, which performs the following steps.

1. Lines 2 to 10 determines the nearest node v_j, from the set of nodes which are at least $\epsilon_{\text{vnSteps}}$ simulation steps apart from the current node v_i and closer than ϵ_{vnDst} in terms of Euclidean distance.
2. For the candidate node v_j (line 12) we first determine the transformation $T_{v_i}^{v_j}$ in terms of the current node v_i, using GPU-NDT (line 13).
3. Subsequently, we pick two validation nodes using a graph-index offset os in line 14 and do for each of them:
   (a) Determine transformation $T_{v_i}^{v_{j+os}}$ in terms of current node v_i using GPU-NDT (line 15).

Algorithm 2: Validate-Nearest Loop Closing

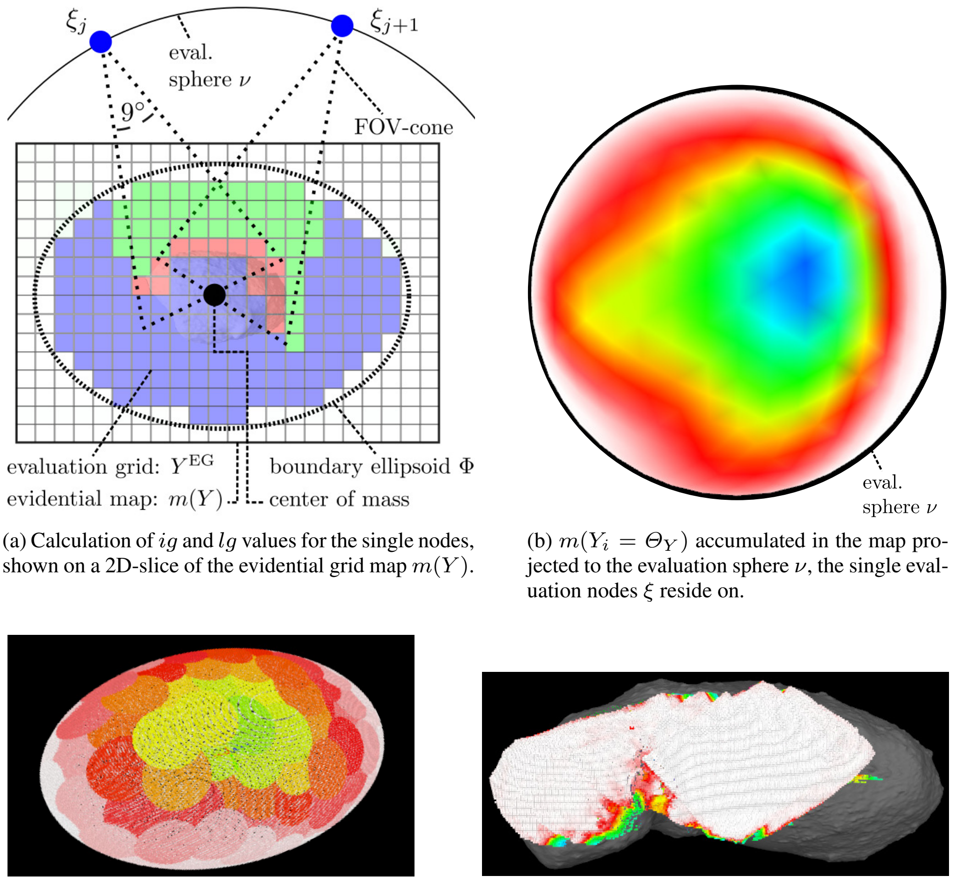
```
   Input: $\mathcal{G}_t, v_i$ ;                    // current graph, last node in graph
1  $minDst \leftarrow \infty$;
2  for $v_j \in \mathcal{G}_t$ do                          // determine nearest node
3      $curDst \leftarrow dst(v_i, v_j)$;
4      if $(i - j > \epsilon_{vnSteps}) \wedge (curDst < \epsilon_{vnDst})$ then
5          if $curDst < minDst$ then
6              $v_{min} \leftarrow v_j$;
7              $minDst \leftarrow curDst$;
8          end
9      end
10 end
11 $loop\_closed \leftarrow$ False;
12 $v_j \leftarrow v_{min}$;
13 $T_{v_i}^{v_j} = \mu_{ij}^{\mathrm{LL}}, \Sigma_{ij}^{\mathrm{LL}} \leftarrow$ GPU_NDT($v_i, v_j$) ;   // NDT: transformation & covariance
14 for $os \in \{-1, 1\}$ do               // iterate validation nodes $j-1$ and $j+1$
15     $T_{v_i}^{v_j+\mathrm{os}} = \mu_{i(j+os)}^{\mathrm{LL}}, \Sigma_{i(j+os)}^{\mathrm{LL}} \leftarrow$ GPU_NDT($v_i, v_{j+\mathrm{os}}$) ;
16     $T_{v_i}^{v_i'} = T_{v_i}^{v_{j+os}} \left(T_{v_j}^{v_{j+os}}\right)^{-1} \left(T_{v_i}^{v_j}\right)^{-1}$ ;
17     if $r_{v_i}^{v_i'} < \epsilon_{vnTrans} \wedge R_{v_i}^{v_i'} < \epsilon_{vnAtt}$ then   // validate translation and attitude
18         $\mathcal{G}_t \leftarrow$ add_edge$\left(\mathcal{G}_t, v_i, v_j, \mu_{ij}^{\mathrm{LL}}, \left(\Sigma_{ij}^{\mathrm{LL}}\right)^{-1}\right)$ ;
19         $\mathcal{G}_t \leftarrow$ add_edge$\left(\mathcal{G}_t, v_i, v_{j+\mathrm{os}}, \mu_{i(j+os)}^{\mathrm{LL}}, \left(\Sigma_{i(j+os)}^{\mathrm{LL}}\right)^{-1}\right)$;
20         $loop\_closed \leftarrow$ True;
21         break
22     end
23 end
24 return $\mathcal{G}_t$, loop_closed ;        // graph w/ loop closing edges, if validated
```

(b) Compute the transformation from v_i to v_i', i.e., from v_i to itself, over the candidate node v_j and the current validation node v_{j+os} (see line 16).

(c) If the offset in terms of translation $r_{v_i}^{v_i'}$ and attitude $R_{v_i}^{v_i'}$ is smaller than the threshold values $\epsilon_{\mathrm{vnTrans}}$ and $\epsilon_{\mathrm{vnAtt}}$, the loop is considered as closed, respective edges are added to the graph, and the for-loop is left (line 17ff.).

5 Autonomy Modules

The following sections will provide a detailed description of the sub-modules of the *Autonomy* module, starting with an explanation of the generation and assessment of the evaluation points (see Sect. 5.1). Following that, the trajectory generation (see Sect. 5.2) and evaluation (see Sect. 5.3) are described. The results of the latter assessment are used for active perception in Sect. 5.4, which can be carried out adaptively as shown in Sect. 5.5. Finally, the postprocessing of the map and subsequent extraction of landing sites are described in Sect. 5.6.



(a) Calculation of ig and lg values for the single nodes, shown on a 2D-slice of the evidential grid map $m(Y)$.

(b) $m(Y_i = \Theta_Y)$ accumulated in the map projected to the evaluation sphere ν, the single evaluation nodes ξ reside on.

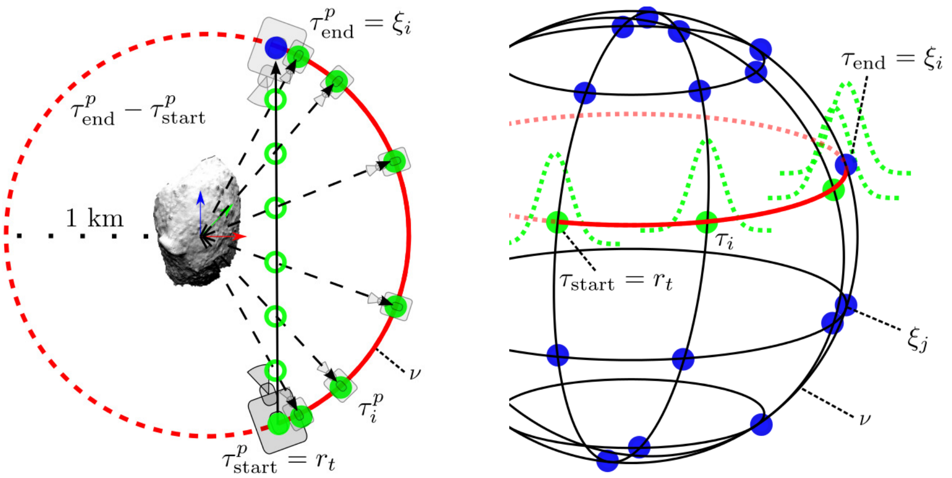
(c) Footprint of the FOV-cones of the evaluation nodes ξ on the boundary ellipsoid Φ.

(d) Accumulated $m(Y_i = \Theta_Y)$ in the map Y^{EG} shown on the asteroid.

Fig. 11: (a) Model for approximate computation of localization gain (lg) and information gain (ig) in the FOV-cones. Projections of non-specificity (i.e., $m(Y_i = \Theta_Y)$) to other representations (b)-(d). Please note the evidential grid map color-coding for (a) is blue = unknown, red = occupied, and green = free, while for (b), (c), and (d) it is a gradient ranging from white corresponding to $\max_i m(Y_i = \Theta_Y)$ to blue corresponding to $\min_i m(Y_i = \Theta_Y)$.

5.1 Evaluation Point Generation and Assessment

To establish a model for rapid approximate evaluation of the asteroid surface properties, we initially assume limited knowledge about the asteroid, including an estimate of its position as well as a rough estimate of its dimensions. In addition, we assume that the asteroid approximately has a convex, elliptical shape–or put in other words–it has no major cavities (cf. Figs. 7b and 11a). The latter assumption allows us to speed-up the whole process by simply computing mean values of the cells inside the FOV-cone attached to the evaluation nodes (see Sect. 5.3). Of course, it is generally possible to use an–albeit slower–raycasting approach to exactly assign map cells Y_i^{EG} to the corresponding evaluation nodes ξ_i.


$\tau^p_{\text{end}} = \xi_i$
$\tau^p_{\text{end}} - \tau^p_{\text{start}}$
1 km
ν
τ^p_i
$\tau^p_{\text{start}} = r_t$
$\tau_{\text{end}} = \xi_i$
τ_i
$\tau_{\text{start}} = r_t$
ξ_j
ν

(a) Plan view of potential trajectory generation, by projecting equidistant points from a vector directly connecting start and end of a trajectory onto a sphere.

(b) 3D view of trajectory evaluation by accumulating values of ξ_j in τ_i, with exponential decay with respect to their distance.

Fig. 12: Generation of potential trajectories τ^{P} and evaluation of trajectory nodes τ_i.

Based on those preliminaries, we create a boundary ellipsoid $\varPhi$ enclosing the asteroid with an additional safety margin. Inside $\varPhi$ an evaluation grid Y^{EG} corresponding to a subset of the evidential map $m(Y)$ is initialized (see Fig. 11a). To approximate the impact of the estimated topology of the asteroid on the expected measurements, we create an evaluation sphere ν with a radius of 1km (see Figs. 11a and 12b). We furthermore assume, that the measurements will be taken from poses distributed on it. To implement this, we subsequently conduct an icospheric tessellation, thus generating a set of distinct points E on that sphere, which we call evaluation nodes ξ_j (see Figs. 11a and 12b). The attitude of each node is chosen such that the LiDAR's 9° FOV-cone points to the asteroid's expected center of mass (COM). This model mimics the footprints of the actual LiDAR-based information intake from each ξ_j (see Fig. 11c). To yield another speed-up, we precalculate the indices of the evaluation grid cells Y^{EG}_i inside the FOV-cone of each evaluation node ξ_j and store them in a lookup table (LUT) for online-usage. The latter is possible, as ν is defined in the asteroid frame and thus is static with respect to its body-fixed coordinate frame. Afterwards, the particular cells Y^{EG}_c just have to be updated with the relevant values, namely information gain (ig) and localization gain (lg), from the evidential grid map $m(Y)$ (see Sect. 5.3).

5.2 Trajectory Generation

Two distinct types of trajectories are generated (i) the potential trajectories $\tau^P \in \varUpsilon$ and (ii) the optimal trajectories $\tau^W \in \varPsi$. They are computed from the current S/C position estimate r_t to an evaluation node ξ_i, thus we have: $\tau_{\text{start}} = r_t$ and $\tau_{\text{end}} = pos(\xi_i)$.

The former trajectories $\varUpsilon$ serve as a rapid approximation of the optimal trajectories and are generated as follows (see Fig. 12a). First, the direct connection between $\tau^{\text{P}}_{\text{start}}$ and $\tau^{\text{P}}_{\text{end}}$ is computed. Subsequently, equidistant points are distributed on the resulting vector.

Finally, they are projected on the geodesic connection between $\tau^{\mathrm{P}}_{\mathrm{start}}$ and $\tau^{\mathrm{P}}_{\mathrm{end}}$, coinciding with the sphere ν. In this way, we can mimic a LiDAR sampling of the asteroid surface from distinct points on an approximated trajectory. The resulting distribution of the intermediate trajectory points τ^{P}_{i} accounts for the fact, that the S/C initially accelerates and then gradually decelerates on the trajectories. For, e.g., a constant velocity, normally distributed points may be spread on the vector and projected afterwards.

The optimal trajectories Ψ are computed by the ESA NLP solver WORHP [11] (see Fig.3a). It is a is a software framework combining sequential quadratic programming (SQP) and interior point methods to solve large-scaled, sparse nonlinear problems. SQP methods are a class of algorithms most frequently used for solving practical optimization problems due to their robustness and good convergence properties. By exploiting the sparse structure of the derivative matrices, WORHP is especially efficient for solving high-dimensional problems like those resulting from discretized optimal control problems. A comprehensive description of the specific trajectory planning and optimization approach employed here can be found in [79, 80, 81].

The resulting optimal trajectories τ^{w} additionally comprise optimal controls in every trajectory node τ^{W}_{i}. And, as already mentioned, both types of trajectories share a common basic structure, consisting of a starting point τ_{start}, intermediate points τ_i, and an end point τ_{end}. Thus, they can be evaluated with the same approach, which is presented in the next section.

5.3 Trajectory Evaluation

The trajectories τ are evaluated for their accumulated information gain $ig(\tau)$ and localization gain $lg(\tau)$. Within the evaluation grid Y^{EG}, the former is measured in terms of non-specificity (7), while the latter is measured in terms of normalized occupancy (see Sect.3.2). Thus, the value of a trajectory is quantified in terms of exploring new areas or re-visiting well estimated areas providing global correctional potential for the SLAM algorithm.

Specifically, $ig(\tau)$ and $lg(\tau)$ are separately computed by considering the accordingly weighted influences of all evaluation nodes ξ_j on each trajectory node τ_i, respectively given by

$$ig(\tau) = \sum_{i=1}^{I}\sum_{j=1}^{J} \exp\left(-\frac{dst(\tau_i,\xi_j)}{\sigma}\right) ig(\xi_j)\,, \text{ and} \tag{41}$$

$$lg(\tau) = \sum_{i=1}^{I}\sum_{j=1}^{J} \exp\left(-\frac{dst(\tau_i,\xi_j)}{\sigma}\right) lg(\xi_j)\,, \tag{42}$$

where σ is a scaling factor for the exponential decay and $dst(\circ,\circ)$ a function which determines the Euclidean distance between a trajectory node and an evaluation node. The information gain $ig(\xi_j)$ and localization gain $lg(\xi_j)$ from the single evaluation nodes is obtained by

$$ig(\xi_j) = \frac{1}{|\mathcal{C}_j|}\sum_{\langle c\rangle \in \mathcal{C}_j} m(Y^{\mathrm{EG}}_c = \Theta_Y)\,, \qquad lg(\xi_j) = \frac{1}{|\mathcal{C}_j|}\sum_{\langle c\rangle \in \mathcal{C}_j} m'(Y^{\mathrm{EG}}_c = \{o\})\,, \tag{43}$$

where $\mathcal{C}_j$ is the set of grid cell indices of the evaluation grid Y^{EG}_c lying in the FOV-cone of ξ_j. In the case of the optimal trajectories, the cost of a trajectory is furthermore measured as the accumulated directional change $dc(\tau^{\mathrm{W}})$. It is expressed as the mean pairwise ratio of orthogonality of the subsequent controls encoded in the single trajectory nodes τ_i^{W}

$$dc(\tau^{\mathrm{W}}) = \frac{1}{I-1} \sum_{i=1}^{I-1} 1 - \left(\frac{ctr(\tau_i^{\mathrm{W}})\, ctr(\tau_{i+1}^{\mathrm{W}})}{\|ctr(\tau_i^{\mathrm{W}})\|\, \|ctr(\tau_{i+1}^{\mathrm{W}})\|} \right) , \tag{44}$$

where the function $ctr(\circ)$ returns the control vector for some trajectory node τ_i^{W} and I denotes the number of nodes in a trajectory.

5.4 Active Perception

For active perception using a SLAM approach, we want to control the information intake process by weighting between the conflicting aims to decrease uncertainty either in the map estimate or the trajectory estimate. The former can be reached by inspecting areas, which have not been mapped before. On the contrary, the latter can be reached by re-visiting explored areas of the map and giving the SLAM algorithm the chance to additionally incorporate correctional information by loop closing. This trade-off can be quantified for the particular trajectories drawing on the metrics stated in Sect. 5.3 above. Based on those, the value of a potential trajectory $val(\tau^{\mathrm{P}})$ is then defined as

$$val(\tau^{\mathrm{P}}) = (1-\kappa)\, ig(\tau^{\mathrm{P}}) + \kappa\, lg(\tau^{\mathrm{P}}) , \tag{45}$$

where κ is a gain factor in the closed interval $[0, 1]$ to trade-off between maximum information gain ig (0.0) and maximum localization gain lg (1.0). If we now let p be a penalizing factor in the closed interval $[0, 1]$, to weigh the accumulated directional change dc, the value of an optimal trajectory can be defined as

$$val(\tau^{\mathrm{W}}) = (1-\kappa)\, ig(\tau^{\mathrm{W}}) + \kappa\, lg(\tau^{\mathrm{W}}) - p\, dc(\tau^{\mathrm{W}}) . \tag{46}$$

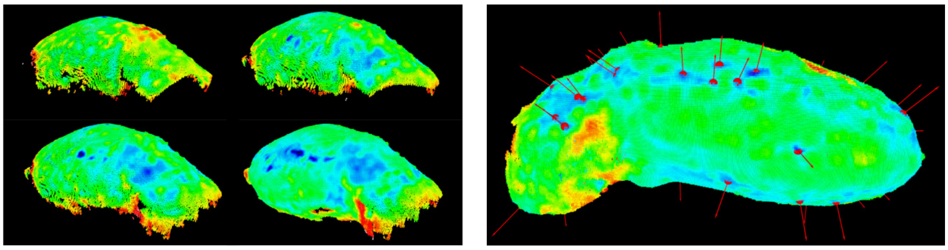
In the end, we want to select the k-best trajectories from the set of potential trajectories Υ and, following that, the best trajectory $\tau^{\mathrm{W}*}$ from the set of optimal trajectories Ψ. Furthermore, we induce an inhibition of return behavior, by removing an evaluation node from the set of potential target nodes E after it has been picked as a target.

5.5 Adaptive Parameters

After the SLAM approach has become controllable through the introduction of active perception in the preceding Sect. 5.4, the gain factor κ can be adapted online based on the position estimation uncertainty Σ_r as encoded in the graph structure [34]. Thus a dynamic trade-off can be established between the exploration of the map and the localization uncertainty by constantly updating the adaptive gain factor κ^* [85].

To this end, we define the mapping $\phi : \Sigma_r \rightarrow \mathbb{R}$ on the marginalized position covariance of the current vertex v_i as a measure of the uncertainty of state estimate [34]. Specifically, we use the D-optimality criterion, as defined in [43], as it is proportional to the uncertainty ellipse and invariant to re-parametrizations as well as linear transformations [13]. It is obtained in logarithmic space to avoid round-off errors caused by small uncertainties in single dimensions [13], according to

$$\phi(\Sigma_r) = (\det \Sigma_r)^{1/l} = \exp\left(\frac{1}{l} \sum_i \log(\lambda_i) \right) , \tag{47}$$



(a) Development of Y^{LS} over time.

(b) Final map Y^{LS} with landing sites (red spheres) and corresponding normals (red arrows).

Fig. 13: Surface normal estimation for landing site discovery.

where λ_i denotes the i-th eigenvalue of Σ_r and $l = 3$ is its size. The latter follows from $\Sigma_r \in \mathbb{R}^{3\times3}$, as we only consider the position. Finally, the adaptive gain factor κ^* is normalized and clamped to the closed interval $[\epsilon_{\text{min}}, \epsilon_{\text{max}}]$ according to

$$\kappa^* = \min\left(\epsilon_{\text{max}}, \max\left(\epsilon_{\text{min}}, \frac{\phi - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}\right)\right) , \tag{48}$$

where V_{max} and V_{min}, are factors to be empirically determined based on the value range of ϕ. Furthermore, ϵ_{min} and ϵ_{max} are set such that they match the expected value range for κ in (45).

5.6 Landing Site Discovery

The landing site discovery module seeks to identify locations on the asteroid that are suitable for the S/C to land and subsequently mark them in the augmented map Y^{LS}. A location is deemed suitable if it is flat and has a sufficient size to allow for the S/C to land.

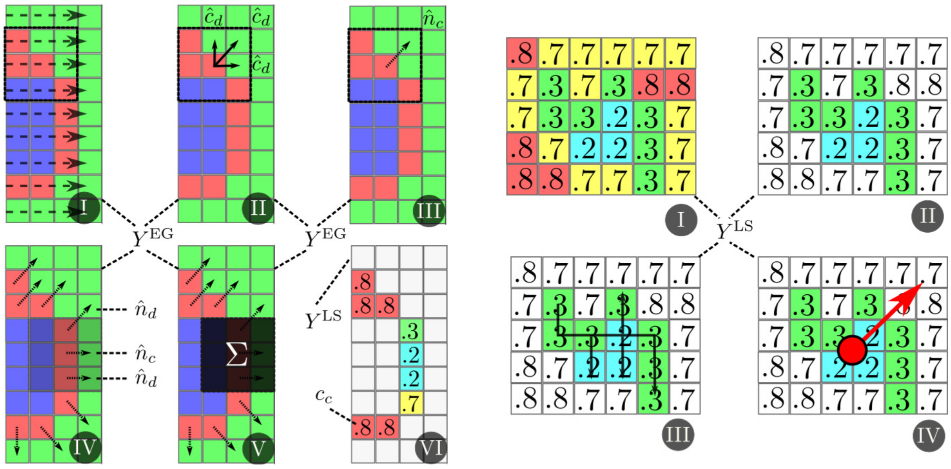
To compute the surface normals, we initially calculate a local normal estimation for each cell Y_c^{EG}, expressed as a unit normal vector $\hat{n}_c$ (see Fig. 14a (I)). First, while iterating through all cells Y_c^{EG}, we consider each of the respective 26 direct neighbors Y_d^{EG} in 3D (see Fig. 14a (II)). For all empty neighbors $en = \{Y_d^{\mathrm{EG}} | m(\{o\}_{Y_d^{\mathrm{EG}}}) < \epsilon_{\text{empty}}\}$, we add a unit vector $\hat{c}_d$ pointing from the center of Y_c^{EG} to the center of Y_d^{EG}. Finally, we yield an interpolated normal vector $\hat{n}_c$ by computing the mean value of all $\hat{c}_d$ (see Fig. 14a (III)). In addition, if a cell has free neighbors, it is considered a surface cell and flagged accordingly.

Based on the surface normals, we calculate the cell-wise coarseness value c_c. To this end, we first determine the neighbors of each cell which are surface cells sn, using $sc(\circ)$ which is simply an indicator function for the surface cell flag, i.e.,

$$sn = \{Y_d^{\mathrm{EG}} | sc(Y_d^{\mathrm{EG}}) == 1\} . \tag{49}$$

If we consider the pairwise dot products between $\hat{n}_c$ and the neighboring normal vectors $\hat{n}_d$, we can measure the coarseness of a cell c_c (see Fig. 14a (IV)). We define c_c such that 0 indicates a smooth surface and 1 coarse one. It can be derived from the local surface normals $\hat{n}_c$

$$c_c = \frac{1}{|sn|} \sum_{d=1}^{|sn|} 1 - \hat{n}_c \hat{n}_d \, , \text{where } \hat{n}_c = \frac{1}{|en|} \sum_{d=1}^{|en|} \hat{c}_d \, . \tag{50}$$



(a) Surface normal estimation, and computation of cell-wise coarsness value c_c sketched in 2D.

(b) Floodfill algorithm to identify connected smooth areas and their respective centers (red dot) and normals (red arrow).

Fig. 14: Landing site discovery in landing site map Y^{LS} extracted by postprocessing the evaluation grid Y^{EG}.

Afterwards we employ a 3D smoothing kernel Σ to create a gradient of the coarseness (see Fig. 14a (V)). Finally, if the coarseness c_c of a cell is below some threshold-value ϵ_{sl}, it is considered suitable for landing, and is accordingly annotated in Y^{LS} (see Fig. 14a (VI)).

In the last step, we discover connected flat regions to be considered as landing sites. To this end, a flood fill algorithm is run on the suitable landing cells annotated in Y^{LS} (cf. Fig. 14b (I), (II)). We iterate all cells, once a cell is suitable for landing, i.e., $c_c < \epsilon_{\mathrm{sl}}$, a new landing patch is initialized (see Fig. 14b (III)). All subsequent neighbors, for which again it holds that $c_c < \epsilon_{\mathrm{sl}}$ are then added to the patch. If no more cells complying with the threshold are found, the landing site discovery process is considered completed.

In a final pass, we determine the landing site locations, calculating the center of gravity of the patches comprised of connected cells. A corresponding normal for the connected patch is computed as well using (50). Finally, the result is annotated in Y^{LS}, resulting in a coarseness map of the surface with flagged safe landing sites (see Fig. 13).

6 Evaluation

We conduct a numerical evaluation to investigate the performance of the active Graph SLAM approach under eight different conditions E1–E8 (see Tab. 1). To this end, we vary the tradeoff factor κ systematically from 0.1 to 0.9 to measure the impact on localization and exploration performance in E1–E5. We omitted the extreme values 0.0 and 1.0, which tended to yield unstable results. In E6, we evaluated the adaptive parameter κ^*. And in the orbit condition E7, the S/C starts from a given state x_{init}, while in the following steps it only controls its attitude. Finally, to show the correctional potential of the scan matching result w_{ij}^{LL}, E8 is a single run in the orbit condition, where the scan matching-based EKF-correction is omitted. Thus, a dead-reckoning error propagates in the velocity dimensions of the state vector x.

Table 1: Evaluation codes and conditions.

Evaluation Code	Condition	Evaluation Code	Condition
E1	$\kappa = 0.1$	E5	$\kappa = 0.9$
E2	$\kappa = 0.25$	E6	$\kappa = \kappa^*$
E3	$\kappa = 0.5$	E7	orbit
E4	$\kappa = 0.75$	E8	No EKF scan matching correction

6.1 Setup

For evaluation, we use a custom-build simulation system using ROS [70] for communication and RViz [40] for visualization. All evaluation conditions run in the same general setup, which is described in the following.

We integrate the asteroid on an orbit, whose starting position and velocity is defined by orbital elements with a 2.8 AU semi-major axis, while the remaining parameters are set to 0. The asteroid rotates about its z-axis with an angular velocity of -0.0082 deg/s [93]. We simulate the local dynamic environment around Itokawa [82]. As stated in Sect. 3.3, we use spherical harmonics to compute the ground truth gravitational force, while we use a point mass approximation on board the S/C. The S/C is then integrated from a starting state $\hat{x}_{\text{init}}$ computed from orbital elements around Itokawa. Here, the semi-major axis is set to 1 km, while the remaining parameters are also set to 0. The ground truth state transition is computed with an RK4 integrator and the subsequent sensor measurement generation is conducted according to the models stated in Sect. 3.4. Initially, we assume to know the state of the S/C and pass this information to the *Sensor Fusion* module as x_{init} (see Fig. 4). Furthermore, we assume to know the position of the asteroid as well as its attitude with respect to the Sun at every time t and pass this information to the *Sensor Fusion* module as a measurement z_t^{ast}.

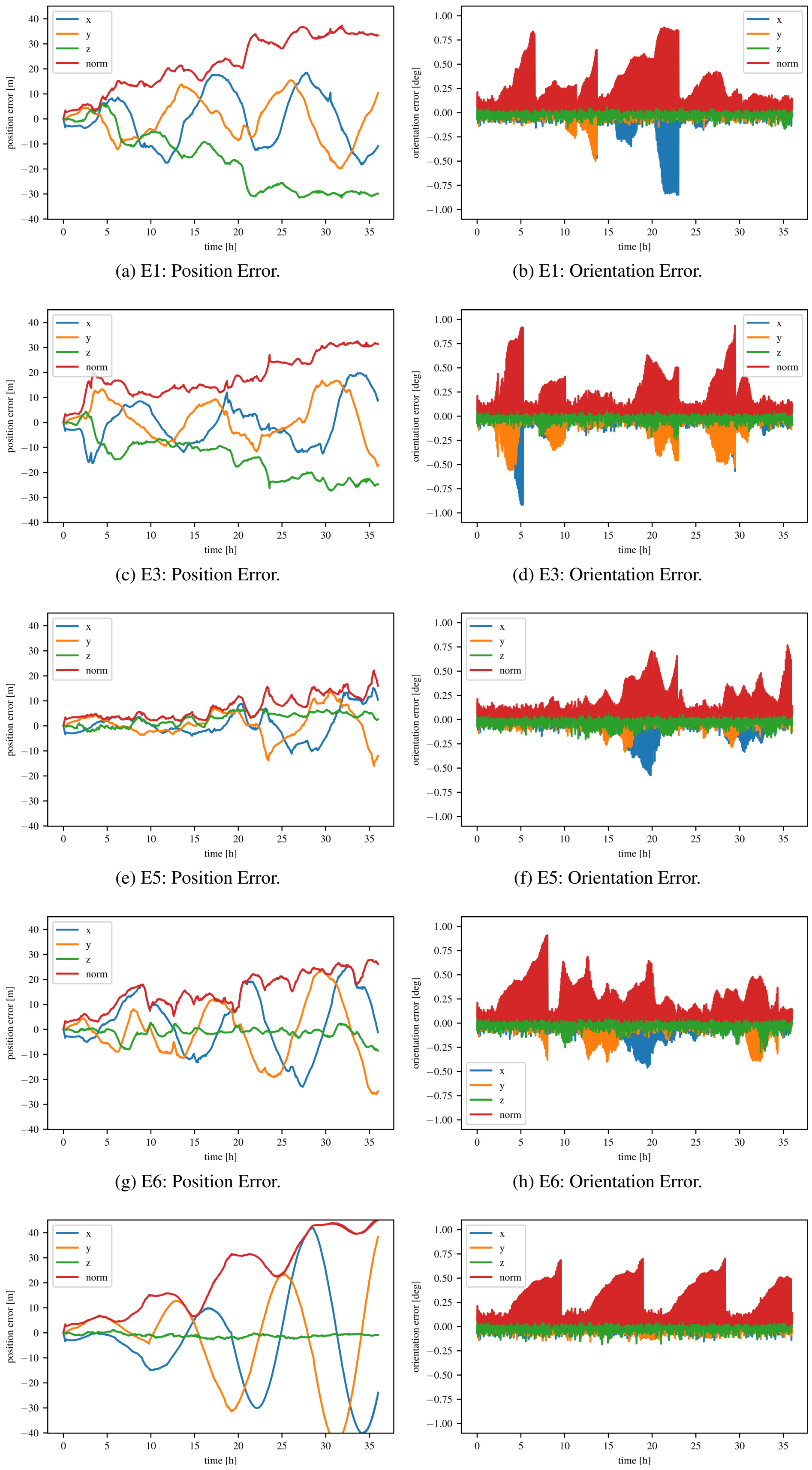
The S/C has an initial mass m_{init}^{sc} of 3000 kg, 649.29 kg of which are propellant for the thrusters denoted as $m_{\text{init}}^{\text{fuel}}$. We assume an independent application of torque and thrust, i.e., position and attitude can be controlled separately. In this setup, we simulate 1.5 days in 1 minute steps resulting in an absolute number of 2160 simulation steps, which allows for a complete exploration of the asteroid.

The accumulated non-specificity for exploration is computed as seen from the FOV cones of the single evaluation nodes (see Fig. 11a). This implies that there is always remaining non-specificity due to the fact that it is not possible to look *into* the asteroid, which always results in residual unexplored areas of the map.

6.2 Results

In Figs. 15 and 16, we present single run plots over time, showing exemplary position, orientation, and velocity errors for the evaluation conditions E1, E3, E5, E6, and E7.

The position error is strongly affected by the parameter κ resulting in a constant decline from E1 (see Fig. 15a) over E3 (see Fig. 15c) to E5 (see Fig. 15e). The adaptive parameter in E6 settles between E3 and E5 (cf. Figs. 15c, 15e, and 15g). The orbit condition E7, where the position is not controlled, results in an increased error (see Fig. 15i). In addition, the position z-error is orbit dependent. This becomes obvious in E7 (Fig. 15i). Here, the S/C orbits around the asteroid without changing its inclination. Accordingly, a small error in z



(a) E1: Position Error.

(b) E1: Orientation Error.

(c) E3: Position Error.

(d) E3: Orientation Error.

(e) E5: Position Error.

(f) E5: Orientation Error.

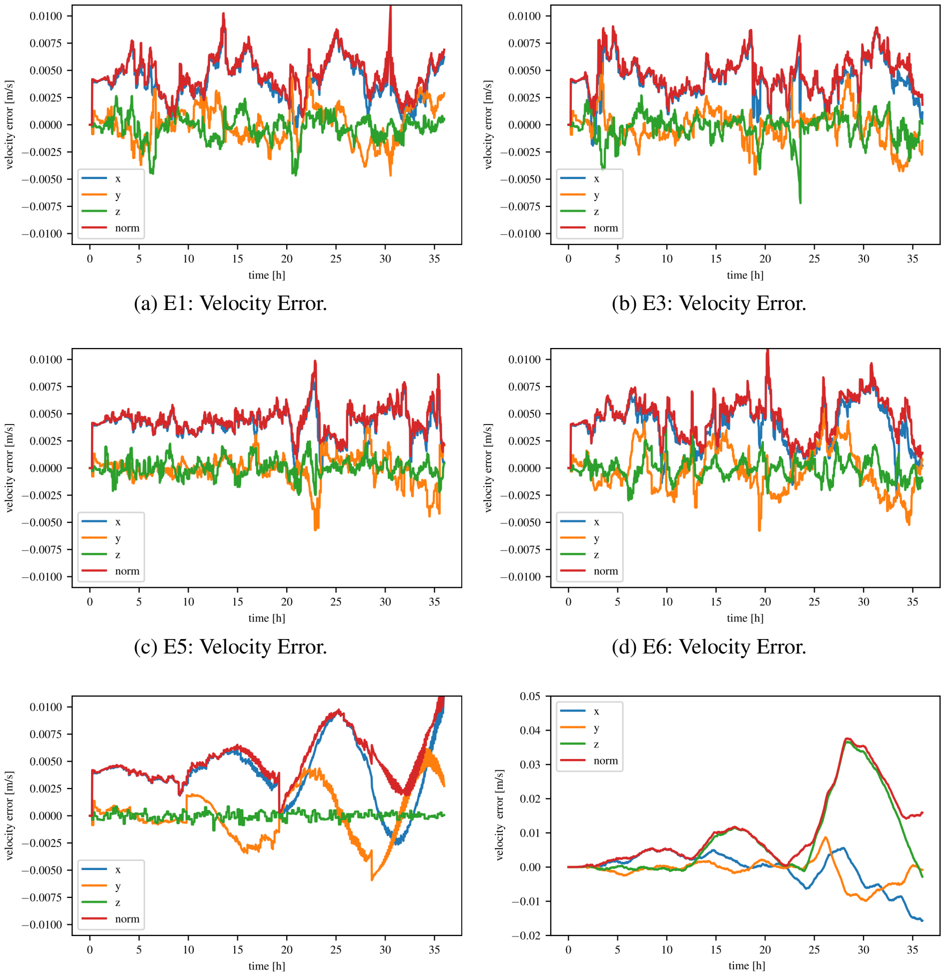
(g) E6: Position Error.

(h) E6: Orientation Error.

(i) E7: Position Error.

(j) E7: Orientation Error.

Fig. 15: Exemplary position and orientation errors of single runs.



(a) E1: Velocity Error.

(b) E3: Velocity Error.

(c) E5: Velocity Error.

(d) E6: Velocity Error.

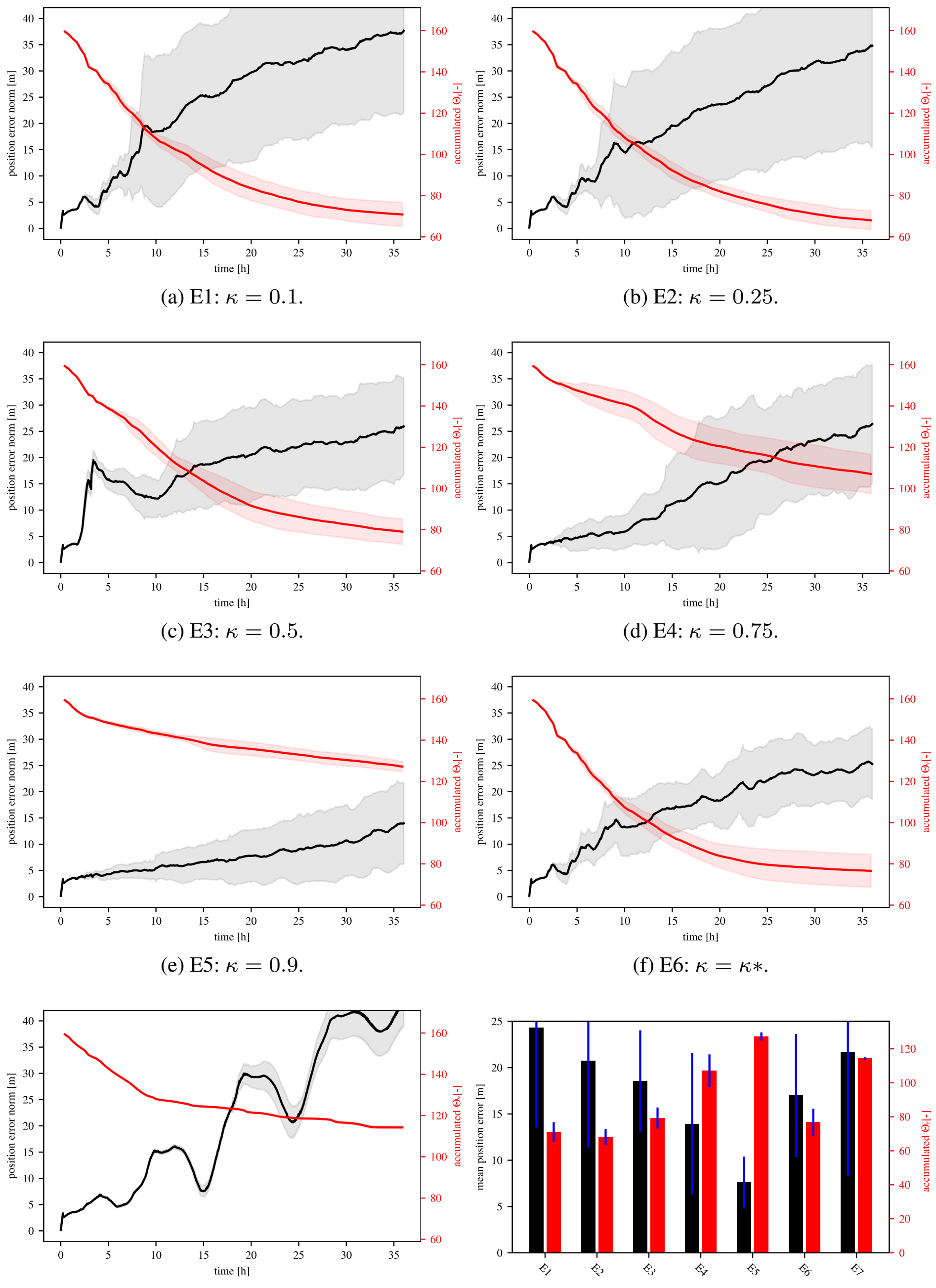
(e) E7: Velocity Error.

(f) E8: Velocity Error.

Fig. 16: Exemplary velocity errors of single runs. In (a)–(e) the EKF is corrected with the scan matching result, in (f) this correction is omitted. Please note the different scaling of the y-axis between the former (a)–(e) and the latter condition (f) plots.

is yielded. The chosen trajectories, in turn, depend on the exploration strategy. In E1 the S/C focuses on exploration and thus exhibits huge orbit changes in order to explore the asteroid quickly, which results in a higher z-error. In contrast, the focus of E5 (Fig. 15e) is on localization. Thus, the S/C does not rapidly change the orbit and the inclination, which results in a lower z-error. In E3 (Fig. 15c) the intermediate effect can be observed. Finally, the generally smoother behavior of the adaptive parameter (c.f. Figs. 17c and 17f) can also be observed in the development of the respective z-error (Fig. 15g).

The orientation error is small (its norm over all axes never exceeds 1 deg) in all evaluation conditions (cf. Figs. 15b, 15d, 15f, 15h, and 15j). This is due to the star tracker, which provides additional precise attitude information independent of the measurements relative



(a) E1: $\kappa = 0.1$.

(b) E2: $\kappa = 0.25$.

(c) E3: $\kappa = 0.5$.

(d) E4: $\kappa = 0.75$.

(e) E5: $\kappa = 0.9$.

(f) E6: $\kappa = \kappa*$.

(g) E7: Orbit.

(h) Average position error and final Θ_Y.

Fig. 17: Evaluation results for information gain vs. localization gain parameter variations κ averaged over 60 runs. Please note the different scales for the position error norm (over x, y, and z) on the left (black) and accumulated non-specificity (mass on Θ_Y) on the right (red). In addition, please note the varying scaling between the y-axes of (a)–(g) and (h).

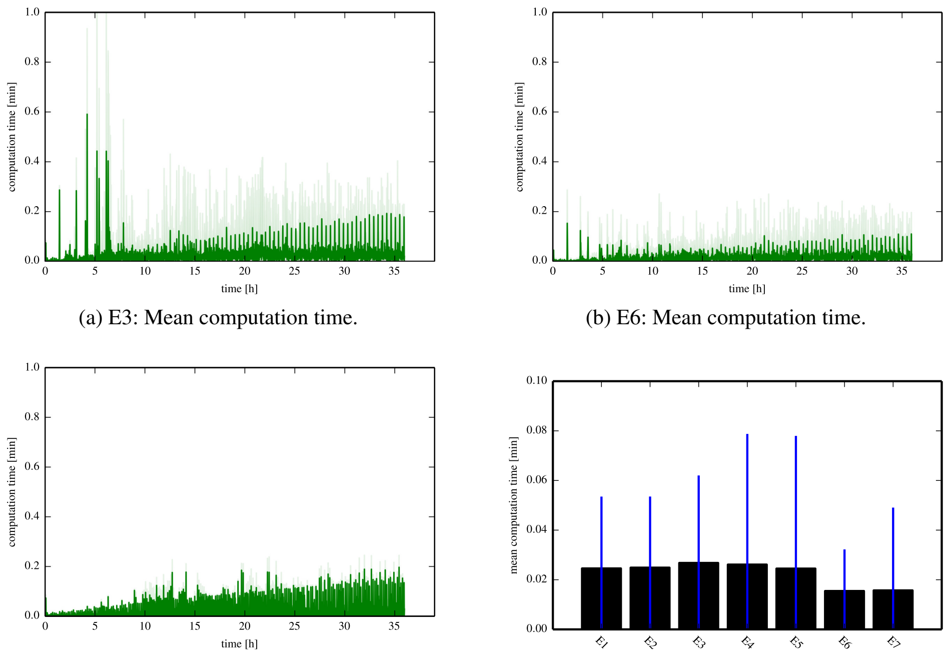
to the asteroid. In addition, the information is used in the EKF for correctional purposes, as well as to constrain the graph. Hence the orientation error is only weakly correlated with the position error, except for condition E7. However, the latter only holds in terms of structure and not in terms of magnitude (cf . Figs. 15i and 15j).

The velocity errors are mildly affected by the chosen trajectories, thus they exhibit the same behavior as the position errors–albeit on a minor scale. This results in a decline from E1 over E3 to E5 (cf. Figs. 16a–16c). And again E6 settles in between E3 and E5, as can be seen comparing Figs. 16b–16d. Additionally, the velocity error of E7 reflects the error growth of the position estimate (cf. Figs. 15i and 16e). Finally, Fig. 16f shows condition E8: a single run without taking into account the scan matching measurement w_{ij}^{LL} as correctional information in the EKF. Comparing this to the other runs, shown in Figs. 16a–16e, the impact is clearly visible (please note the different scaling of the y-axis). The unbounded velocity error growth, in turn, has a corresponding negative influence on the position estimate. In general however, the velocity estimate in the other conditions only has a minor influence on the position estimate. This is due to the fact that this quantity is only relevant in the EKF, which provides an initial pose estimate for new nodes as well as for the scan matching process. Within the graph structure, the quality of the latter is more important for the trajectory estimate.

In Fig. 17, evaluation results averaged over 60 runs are presented. The black line describes the mean of the localization error norm, while the gray shaded areas show the respective standard deviations. The red lines show the accumulated non-specificity (mass on Θ_Y) as seen from the evaluation nodes ξ and again, the red shaded areas indicate the respective standard deviations. This time, κ ranges from full information gain $\kappa = 0.1$ (E1, Fig. 17a) over $\kappa = 0.25$ (E2, Fig. 17b), $\kappa = 0.5$ (E3, Fig. 17c), and $\kappa = 0.75$ (E4, Fig. 17d) to full localization gain $\kappa = 0.9$ (E5, Fig. 17e). The system reacts to the trade-off between information and localization gain as expected. The position error increases more slowly when the weight of the localization gain is increased. With the exception of the E1 ($\kappa = 0.1$) condition, the opposite holds for the residual accumulated non-specificity: it increases accordingly, as the weight of the information gain is decreased. When κ is additionally adapted in condition E6, the system explores better than in E3, while showing a slower increase in the positioning error as well as a smaller variance throughout the runs (cf. Figs. 17c and 17f).

Fig. 17h depicts the mean position error over time of the single conditions as black bars and indicates the respective standard deviation as a blue line. In addition, the plot shows the *final* (i.e., from the last simulation step) accumulated non-specificity and its respective standard deviation again as a blue line. Again, the plot shows a decline in the position error and an accordingly increasing residual Θ_Y-value, when κ is increased from E1 to E5. The adaptive parameter condition E6 again has a similar appearance to E3, while exhibiting a better localization performance as well as resulting in a lower residual Θ_Y-value.

The orbit condition E7 (cf. Figs. 17g and 17h) performs worse than all other conditions in terms of exploration and position estimation. Surprisingly, the variance is the smallest over the 60 runs. This condition reveals that not only re-visiting areas is of importance but also the quality of the obtained information. The development of the exploration measure (non-specificity) shows that a lot explored areas have been re-visited but the localization performance does not improve. As the position of the S/C is not controlled, the information advantage is outweighed by an increasing velocity and a varying resolution of the point clouds, as the S/C gradually approaches the asteroid. In addition, the counter-rotation of the asteroid can result in a higher relative pose-offset, rendering the scan matching procedure more involved.



(a) E3: Mean computation time.

(b) E6: Mean computation time.

(c) E7: Mean computation time.

(d) Mean Computation times.

Fig. 18: Step-wise computation time averaged over 60 runs. In (a)–(c) the mean step-wise computation time over time is shown for evaluation conditions E3, E6, and E7. The mean is plotted in solid green, while the standard deviation is indicated in light green. In (d) the mean step-wise computation time averaged over time is shown for all evaluation conditions, the respective standard deviation is indicated by the blue lines.

The evaluation has been conducted mostly single-threaded on a workstation which features an Intel i7-4770 CPU at 3.40GHz, 16GB RAM, and a Nvidia TitanX GPU. In Fig. 18 the computation time per simulation step of our approach is evaluated, which comprises the world simulation, the sensor fusion as well as the autonomy module. In Figs. 18a–18c the mean step-wise computation times are plotted over time. In the first simulation steps depicted in Figs. 18a and 18b some peaks can be noticed. They stem from the fact that system executes similar computational expensive operations (like optimal trajectory computation and evaluation) in the first steps. In Fig. 18c the mean computation time without trajectory computations can be observed. In addition, the application of the adaptive factor κ^* appears to be beneficial in terms of mean computation time (c.f. Figs. 18a, 18b, and 18d). As seen in the evaluation over all conditions, depicted in Fig. 18d, the mean computation time remains well in the time-window of 1 minute as posed by the simulation stepsize.

In all runtime plots, a linear increase of the computation time can be observed. This results from the fact that the number of edges and thus the computational complexity of Graph SLAM grows with the path length. However, this effect can be mitigated by merging constraints obtained at places that have been visited multiple times, thus bounding the complexity to the size of the map [33]. As the evaluation is designed to explore the whole asteroid, the observed computation times can be considered to pose such an upper bound.

The fuel consumption never exceeds 25 kg of propellant and is similar over all conditions, except of course for E7 and E8 where no controls are applied. This is due to passing the same parameter set, controlling the time/fuel optimality, to WORHP in the *Optimal Trajectory Generation* module throughout all conditions. Thus, the proposed approach can be used for the sequential exploration of the basic properties of about 25 asteroids complying with the mission concept stated in [67, 68, 69].

7 Conclusion

In this paper, we propose an active Graph SLAM-approach for relative autonomous deep space navigation, mapping, and landing site discovery. We present a system, tailored to a specific scenario, where a S/C operates in the vicinity (1 km) of an asteroid in 2.8 AU distance from the Sun. The local conditions are based on a real 3D model of Itokawa and the corresponding inhomogeneous gravity field is described by the actual spherical harmonics. The system comprises a *Sensor Fusion* module, which draws on a Graph SLAM approach, tightly-coupled to an EKF, to jointly estimate the trajectory of the S/C and the map of the asteroid. The latter is projected into an evidential grid map to enable an extended uncertainty representation based on belief-functions. In addition, the *Autonomy* module actively controls the Graph SLAM approach by evaluating potential trajectories for their respective exploration or localization value. Based on those measures, a trade-off can be established between reducing uncertainty in either the trajectory estimate or the map estimate. To this end, suitable optimal trajectories are obtained in a twofold evaluation structure, to save computation time by using an approximate pre-evaluation stage. Finally, the resultant map can be post-processed for landing site discovery.

In a numerical simulation, we evaluated the active Graph SLAM approach. We chose a runtime of 1.5 days with a 1 minute simulation stepsize, which allows for a complete exploration of the asteroid. The results show in general that the position error norm over all axes never exceeds 40 m in the 1 km vicinity of the asteroid. For comparison, the relative navigation approach using SURF-based monocular SLAM in the range of 700 m around Itokawa presented in [19, 20] has deviations in a range up to 100 m in one axis using 400 particles. In addition, we show that the system can trade-off between localization performance and exploration speed. An adaptive parameter can additionally be computed online from a measure defined on the current pose uncertainty encoded in the graph structure. The adaptive parameter yields an improvement in localization and exploration performance at the same time. We can show that a simple approach, which relies on an initial state while omitting any controls afterwards, results in an instable localization performance. This is due to the inhomogeneous gravity field and a diminished informational quality. Finally, a comparison between the latter condition and a condition, where the scan matching result is not used for EKF correction shows, that this leads to an unbounded growing velocity error. The latter, in turn, results in an increasing positioning error and thus emphasizes the importance of the scan matching based velocity correction of the EKF.

An interesting problem, which has not been tackled here, is the estimation of the flight path of the celestial object itself, by using heliocentric and relative navigation approaches at the same time. The observations can be processed in a least squares parameter estimation scheme. In addition, the non-trivial task of online mass distribution estimation using the S/C itself as an acceleration sensor would have to be carried out in order to establish a complete model of the asteroid on the fly.

In general, a combination with the landmark based approaches proposed in [3, 19, 20] might improve robustness in appropriate scenarios. An interesting step in that direction would be to introduce salient features extracted from camera images aligned with the LiDAR. They can come in handy by improving the initial guess for scan matching or in the form of an additional constraint between two poses. However, to keep such an approach stable at all times, the camera should always point to illuminated areas of the asteroid. This may be additionally considered in the value functions of a trajectory using an asteroid shape model and an additional illumination model.

Another interesting approach would be to predict expected values of the unknown areas of the map. This could be done by applying some probabilistic regression scheme to the map and using the already explored values as support for the model. Thus, the system would not only be able to explore unknown areas, but unknown areas expected to be *valuable*.

The approaches presented here can be applied to other problems in the field of autonomous robotic navigation and exploration as well. Within the context of relative deep space navigation, one next step is the implementation on space-qualified hardware, while the long-term goal is the use of the algorithm in an actual space mission to a minor celestial body like an asteroid or a comet.

References

[1] James S Albus, Ronald Lumia, and H McCain. "Hierarchical control of intelligent machines applied to space station telerobots". In: *IEEE Transactions on Aerospace and Electronic Systems* 24.5 (1988), pp. 535–541.

[2] Farzin Amzajerdian et al. "Utilization of 3D imaging flash LiDAR technology for autonomous safe landing on planetary bodies". In: *Quantum Sensing and Nanophotonic Devices VII*. Vol. 7608. International Society for Optics and Photonics. 2010, p. 760828.

[3] Franz Andert, Nikolaus Ammann, and Bolko Maass. "Lidar-aided camera feature tracking and visual SLAM for spacecraft low-orbit navigation and planetary landing". In: *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2015, pp. 605–623.

[4] Marcelo H Ang and Vassilios D Tourassis. "Singularities of Euler and roll-pitch-yaw representations". In: *IEEE Transactions on Aerospace and Electronic Systems* 3 (1987), pp. 317–324.

[5] Viorel Badescu. *Asteroids: Prospective energy and material resources*. Springer Science & Business Media, 2013.

[6] Kevin Berry et al. "OSIRIS-REx touch-and-go (TAG) mission design and analysis". In: *36th Annual AAS Guidance and Control Conference; 1-6 Feb. 2013; Breckenridge, CO; United States*. American Astronautical Society; Rocky Mountain Section; San Diego, CA, United States. 2013.

[7] Peter Biber, Sven Fleck, and Wolfgang Straßer. "A probabilistic framework for robust and accurate matching of point clouds". In: *Joint Pattern Recognition Symposium*. Springer. 2004, pp. 480–487.

[8] Peter Biber and Wolfgang Straßer. "The normal distributions transform: A new approach to laser scan matching". In: *IROS*. Vol. 3. 2003, pp. 2743–2748.

[9] Anthony GA Brown et al. "Gaia Data Release 2. Summary of the contents and survey properties". In: *arXiv preprint arXiv:1804.09365* (2018).

[10] Frank Budnik, Trevor Morley, and Ruaraidh Mackenzie. “ESOC’s System for Interplanetary Orbit Determination: Implementation and Operational Experience”. In: *18th International Symposium on Space Flight Dynamics*. Vol. 548. 2004, pp. 387–392.

[11] Christof Büskens and Dennis Wassel. “The ESA nlp solver WORHP”. In: *Modeling and optimization in space engineering*. Springer, 2012, pp. 85–110.

[12] Luca Carlone et al. “Active SLAM and exploration with particle filters using Kullback-Leibler divergence”. In: *Journal of Intelligent & Robotic Systems* 75.2 (2014), pp. 291–311.

[13] Henry Carrillo, Ian Reid, and José A Castellanos. “On the comparison of uncertainty criteria for active SLAM”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 2080–2087.

[14] John A Christian and Scott Cryan. “A survey of LIDAR technology and its use in spacecraft relative navigation”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2013, p. 4641.

[15] Jose Luis Blanco Claraco. “Development of scientific applications with the mobile robot programming toolkit”. In: (2008).

[16] Joachim Clemens. “Multi-robot in-ice localization using graph optimization”. In: *Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on*. IEEE. 2017, pp. 36–42.

[17] Joachim Clemens, Thomas Reineking, and Tobias Kluth. “An evidential approach to SLAM, path planning, and active exploration”. In: *International Journal of Approximate Reasoning* 73 (2016), pp. 1–26.

[18] Joachim Clemens and Kerstin Schill. “Extended Kalman filter with manifold state representation for navigating a maneuverable melting probe”. In: *Information Fusion (FUSION), 2016 19th International Conference on*. IEEE. 2016, pp. 1789–1796.

[19] Cedric Cocaud and Takashi Kubota. “Autonomous navigation near asteroids based on visual SLAM”. In: *Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, California*. 2012.

[20] Cedric Cocaud and Takashi Kubota. “SURF-based SLAM scheme using octree occupancy grid for autonomous landing on asteroids”. In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. 2010, pp. 275–282.

[21] David W Curkendall and James S Border. “Delta-DOR: The one-nanoradian navigation measurement system of the deep space network—History, architecture, and componentry”. In: *The Interplanetary Network Progress Report* 42 (2013), p. 193.

[22] Timothy A Davis. *Direct methods for sparse linear systems*. Vol. 2. Siam, 2006.

[23] Frank Dellaert and Michael Kaess. “Square Root SAM: Simultaneous localization and mapping via square root information smoothing”. In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203.

[24] Ann Dietrich and Jay W McMahon. “Orbit Determination Using Flash Lidar Around Small Bodies”. In: *Journal of Guidance, Control, and Dynamics* 40.3 (2016), pp. 650–665.

[25] Ann B Dietrich and Jay W McMahon. “Robust Orbit Determination with Flash Lidar Around Small Bodies”. In: *Journal of Guidance, Control, and Dynamics* (2018), pp. 1–22.

[26] Rich Dissly et al. “Flash lidars for planetary missions”. In: *Workshop on Instrumentation for Planetary Missions*. 2012.

[27] Christopher D'Souza. *Autonomous Deep-Space Optical Navigation Project*. Technical Report. NASA Johnson Space Center, Houston, TX, United States, 2014.

[28] Hans Jacob S Feder, John J Leonard, and Christopher M Smith. "Adaptive mobile robot navigation and mapping". In: *The International Journal of Robotics Research* 18.7 (1999), pp. 650–668.

[29] Udo Frese, Per Larsson, and Tom Duckett. "A multilevel relaxation algorithm for simultaneous localization and mapping". In: *IEEE Transactions on Robotics* 21.2 (2005), pp. 196–207.

[30] Matteo Golfarelli, Dario Maio, and Stefano Rizzi. "Elastic correction of dead-reckoning errors in map building". In: *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*. Vol. 2. IEEE. 1998, pp. 905–911.

[31] Graciela González Peytaví et al. "Autonomous Orbit Navigation for a Mission to the Asteroid Main Belt". In: *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation. 2015.

[32] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters". In: *IEEE Transactions on Robotics* 23.1 (2007), pp. 34–46. ISSN: 1552-3098. DOI: 10.1109/TRO.2006.889486.

[33] Giorgio Grisetti et al. "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent". In: *Robotics: Science and Systems (RSS)*. 2007.

[34] Giorgio Grisetti et al. "A tutorial on graph-based SLAM". In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43.

[35] Dirk Hahnel et al. "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements". In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 1. IEEE. 2003, pp. 206–211.

[36] Christoph Hertzberg et al. "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds". In: *Information Fusion* 14.1 (2013), pp. 57–77.

[37] Tra-Mi Ho et al. "MASCOT – the mobile asteroid surface scout onboard the HAYABUSA2 mission". In: *Space Science Reviews* 208.1-4 (2017), pp. 339–374.

[38] Armin Hornung et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34.3 (2013), pp. 189–206.

[39] Andrew E Johnson et al. "A general approach to terrain relative navigation for planetary landing". In: *AIAA Aerospace@ Infotech Conf., Rohnert Park, CA*. 2007.

[40] Hyeong Ryeol Kam et al. "Rviz: a toolkit for real domain data visualization". In: *Telecommunication Systems* 60.2 (2015), pp. 337–345.

[41] Shinpei Kato et al. "An Open Approach to Autonomous Vehicles". In: *IEEE Micro* 35.6 (2015), pp. 60–68. ISSN: 0272-1732. DOI: 10.1109/MM.2015.133.

[42] Shinpei Kato et al. "Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems". In: *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. ICCPS '18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 287–296. ISBN: 978-1-5386-5301-2. DOI: 10.1109/ICCPS.2018.00035. URL: https://doi.org/10.1109/ICCPS.2018.00035.

[43] Jack Kiefer. "General equivalence theory for optimum designs (approximate theory)". In: *The annals of Statistics* (1974), pp. 849–879.

[44] George J Klir. *Uncertainty and information: foundations of generalized information theory*. John Wiley & Sons, 2005.

[45] Steven M Klosko and Carl A Wagner. "Spherical harmonic representation of the gravity field from dynamic satellite data". In: *Planetary and Space Science* 30.1 (1982), pp. 5–28. ISSN: 0032-0633. DOI: 10.1016/0032-0633(82)90068-X.

[46] Stefan Kohlbrecher et al. "Hector open source modules for autonomous mapping and navigation with rescue robots". In: *Robot Soccer World Cup*. Springer. 2013, pp. 624–631.

[47] Michail Kontitsis, Panagiotis Tsiotras, and Evangelos Theodorou. "An information-theoretic active localization approach during relative circumnavigation in orbit". In: *AIAA Guidance, Navigation, and Control Conference*. 2016, p. 0872.

[48] Fisnik Kraja, Georg Acher, and Arndt Bode. "Designing Spacecraft High Performance Computing Architectures". In: *Advanced Computing*. Springer, 2013, pp. 137–156.

[49] Rainer Kümmerle et al. "g2o: A general framework for graph optimization". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 3607–3613.

[50] Rob Landis and Lindley Johnson. "Advances in planetary defense in the United States". In: *Acta Astronautica* (2018).

[51] Thomas Laux. "ASC's 3D Flash LIDAR$^{\text{TM}}$ Camera: The Science behind ASC's 3D Depth Imaging Video Camera". In: *SMPTE International Conference on Stereoscopic 3D for Media and Entertainment*. 2010, pp. 1–10. DOI: 10.5594/M001404.

[52] Martin Magnusson. "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection". PhD thesis. Örebro Universitet, 2009.

[53] Ruben Martinez-Cantin et al. "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot". In: *Autonomous Robots* 27.2 (2009), pp. 93–103. ISSN: 1573-7527. DOI: 10.1007/s10514-009-9130-2.

[54] Michael Montemerlo, Nicholas Roy, and Sebastian Thrun. "Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit". In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 3. IEEE. 2003, pp. 2436–2441.

[55] Michael Montemerlo et al. "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges". In: *IJCAI*. 2003, pp. 1151–1156.

[56] Michael Montemerlo et al. "FastSLAM: A factored solution to the simultaneous localization and mapping problem". In: *Aaai/iaai*. 2002, pp. 593–598.

[57] Anastasios I Mourikis et al. "Vision-aided inertial navigation for spacecraft entry, descent, and landing". In: *IEEE Transactions on Robotics* 25.2 (2009), pp. 264–280.

[58] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[59] David Nakath, Joachim Clemens, and Carsten Rachuy. "Rigid body attitude control based on a manifold representation of direction cosine matrices". In: *13th European Workshop on Advanced Control and Diagnosis (ACD)*. Vol. 783. Journal of Physics: Conference Series. 2016. DOI: 10.1088/1742-6596/783/1/012040. URL: https://doi.org/10.1088/1742-6596/783/1/012040.

[60] David Nakath, Joachim Clemens, and Kerstin Schill. "Multi-Sensor Fusion and Active Perception for Autonomous Deep Space Navigation". In: *2018 21st International*

Conference on Information Fusion (FUSION) (FUSION 2018. Cambridge, United Kingdom (Great Britain), July 2018.

[61] David Nakath et al. “Optimal Rotation Sequences for Active Perception”. In: *Proc. SPIE: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*. Ed. by Jerome J. Braun. Vol. 9872. SPIE Press, 2016, pp. 987204–1–987204–13. DOI: `10.1117/12.2223027`. URL: `http://dx.doi.org/10.1117/12.2223027`.

[62] Edwin Olson, John Leonard, and Seth Teller. “Fast iterative alignment of pose graphs with poor initial estimates”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE. 2006, pp. 2262–2269.

[63] Edwin B Olson. “Real-time correlative scan matching”. In: *Ann Arbor* 1001 (2009), p. 48109.

[64] Edwin B. Olson. “Robust and Efficient Robotic Mapping”. AAI0821013. PhD thesis. Cambridge, MA, USA, 2008.

[65] Michael AC Perryman et al. “The HIPPARCOS catalogue”. In: *Astronomy and Astrophysics* 323 (1997), pp. L49–L52.

[66] Vincenzo Pesce, Ali-akbar Agha-mohammadi, and Michèle Lavagna. “Autonomous navigation & mapping of small bodies”. In: *2018 IEEE Aerospace Conference*. 2018, pp. 1–10. DOI: `10.1109/AERO.2018.8396797`.

[67] Alena Probst and Roger Förstner. “Spacecraft design of a multiple asteroid orbiter with re-docking lander”. In: *Advances in Space Research* (2017).

[68] Alena Probst et al. “KaNaRiA: Identifying the Challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies”. In: *Proceedings of the 66th International Astronautical Congress, Jerusalem, Israel*. International Astronautical Federation. 2015.

[69] Alena Probst et al. “Mission concept selection for an asteroid mining mission”. In: *Aircraft Engineering and Aerospace Technology: An International Journal* 88.3 (2016), pp. 458–470.

[70] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[71] Thomas Reineking. “Belief functions: Theory and algorithms”. PhD thesis. University of Bremen, 2014.

[72] Thomas Reineking and Joachim Clemens. “Dimensions of uncertainty in evidential grid maps”. In: *International Conference on Spatial Cognition*. Springer. 2014, pp. 283–298.

[73] Thomas Reineking and Joachim Clemens. “Evidential FastSLAM for grid mapping”. In: *Information Fusion (FUSION), 2013 16th International Conference on*. IEEE. 2013, pp. 789–796.

[74] Max Risler. “Behavior control for single and multiple autonomous agents based on hierarchical finite state machines”. PhD thesis. tuprints, 2010.

[75] Randi J Rost et al. *OpenGL shading language*. Pearson Education, 2009.

[76] Nicholas Roy and Sebastian Thrun. “Coastal navigation with mobile robots”. In: *Advances in Neural Information Processing Systems*. 2000, pp. 1043–1049.

[77] Radu Bogdan Rusu and Steve Cousins. “3d is here: Point cloud library (pcl)”. In: *Robotics and automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1–4.

[78] Radu Bogdan Rusu and Steve Cousins. “3D is here: Point Cloud Library (PCL)”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011.

[79] Anne Schattel. "Dynamic Modeling and Implementation of Trajectory Optimization, Sensitivity Analysis, and Optimal Control for Autonomous Deep Space Navigation". PhD thesis. Center for Industrial Mathematics (ZeTeM), University of Bremen, 2018. URL: `http://nbn-resolving.de/urn:nbn:de:gbv:46-00106726-14`.

[80] Anne Schattel, Mitja Echim, and Christof Büskens. "Low Thrust Trajectory Optimization for Autonomous Asteroid Rendezvous Missions". In: *6th International Conference on Astrodynamics Tools and Techniques (ICATT), Darmstadt, Germany*. 2016.

[81] Anne Schattel et al. "Optimization and Sensitivity Analysis of Trajectories for Autonomous Small Celestial Body Operations". In: *Progress in Industrial Mathematics at ECMI 2016*. Ed. by Peregrina Quintela et al. Vol. 26. ECMI 2016. Springer, 2017.

[82] Daniel Scheeres et al. "The actual dynamical environment about Itokawa". In: *AIAA / AAS Astrodynamics Specialist Conference and Exhibit*. 2006, p. 6661.

[83] Glenn Shafer. *A mathematical theory of evidence*. Vol. 42. Princeton university press, 1976.

[84] Yu-Fei Shen et al. "A vision-based automatic safe landing-site detection system". In: *IEEE Transactions on aerospace and electronic systems* 49.1 (2013), pp. 294–311.

[85] Robert Sim and Nicholas Roy. "Global a-optimal robot exploration in slam". In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE. 2005, pp. 661–666.

[86] Philippe Smets and Robert Kennes. "The transferable belief model". In: *Artificial intelligence* 66.2 (1994), pp. 191–234.

[87] Randall Smith, Matthew Self, and Peter Cheeseman. "Estimating uncertain spatial relationships in robotics". In: *Autonomous robot vehicles*. Springer, 1990, pp. 167–193.

[88] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. "Information gain-based exploration using Rao-Blackwellized particle filters". In: *Robotics: Science and Systems*. Vol. 2. 2005, pp. 65–72.

[89] Paul van Susante and Leslie Gertsch. "Minerals from Space: Terrestrial and Extraterrestrial Perspectives". In: *Conference: Earth and Space 2018 At: Cleveland, OH*. IEEE. 2018.

[90] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping". In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*. Vol. 1. IEEE. 2000, pp. 321–328.

[91] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[92] Yuichi Tsuda et al. "System design of the Hayabusa 2—Asteroid sample return mission to 1999 JU3". In: *Acta Astronautica* 91 (2013), pp. 356–362.

[93] Brian D Warner, Alan W Harris, and Petr Pravec. "The asteroid lightcurve database". In: *Icarus* 202.1 (2009), pp. 134–146.

[94] James R Wertz, David F Everett, and Jeffery J Puschell. *Space mission engineering: the new SMAD*. Microcosm Press, 2011.

[95] John O Woods and John A Christian. "Glidar: an OpenGL-based, real-time, and open source 3D sensor simulator for testing computer vision algorithms". In: *Journal of Imaging* 2.1 (2016), p. 5.

[96] Makoto Yoshikawa, Akira Fujiwara, and Jun'ichiro Kawaguchi. "Hayabusa and its adventure around the tiny asteroid Itokawa". In: *Proceedings of the International Astronomical Union* 2.14 (2006), 323–324. DOI: 10.1017/S174392130701085X.

[97] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*. Vol. 40. Prentice hall New Jersey, 1996.

A Parameters

For the covariance matrices A, Q the standard deviation (STD) is given to foster readability. As the noise is modeled to affect all axes in the same way, those values can be converted into the respective matrices by, e.g., $A = (STD)^2\, I$.

Table 2: Parameters used.

Parameter	Value	Unit	Description
$m_{\text{init}}^{\text{sc}}$	3000	kg	initial mass of S/C
$m_{\text{init}}^{\text{fuel}}$	649.29	kg	initial fuel of S/C
S_e	1366.1	kg/s^3	solar constant
c	299792485.0	m/s	speed of light
C_R	1.5	$1/m^2$	reflection coefficient
A	5.0	m^2	solar pressure surface area
A_t^r	0.1	m	position transition noise STD
A_t^R	0.001	deg	attitude transition noise STD
A_v^r	0.001	m/s	velocity transition noise STD
Q_t^ω	0.075	$\deg/s$	noise of the gyroscope noise STD
Q_t^a	0.000001	m/s^2	accelerometer noise STD
Q_t^R	0.1	deg	star tracker noise STD
Q_t^{L}	0.001	m	LiDAR distance noise STD
ϵ_{trans}	60	m	translation threshold for considering scan matching
ϵ_{rot}	5	deg	threshold for rotation for considering scan matching
ϵ_{steps}	1	–	threshold for #steps for scan matching
$\epsilon_{\text{validateTrans}}$	10	m	translation threshold for validating scan matching
$\epsilon_{\text{validateRot}}$	1	deg	rotation threshold for validating scan matching
$\epsilon_{\text{optimize}}$	10	–	maximum number of steps without optimization
$\epsilon_{\text{vnSteps}}$	20	–	minimum number of steps away for being a loop closing candidate
ϵ_{vnDst}	500	m	minimum Euclidean distance for being a loop closing candidate
$\epsilon_{\text{vnTrans}}$	20	m	threshold for translation in loop closing validation
ϵ_{vnAtt}	2	deg	threshold for rotation in loop closing validation
V_{max}	300	–	upper normalization threshold for ϕ
V_{min}	30	–	lower normalization threshold for ϕ
ϵ_{min}	0.1	–	lower clamping value for ϕ
ϵ_{max}	0.9	–	upper clamping value for ϕ
ϵ_{empty}	0.0	–	threshold value for a cell to be considered empty
ϵ_{sl}	0.01	–	threshold value for cell to be considered safe for landing
p	1.0	–	penalizing factor for accumulated directional change

Sensorimotor integration using an information gain strategy in application to object recognition tasks

SPATIAL COGNITION

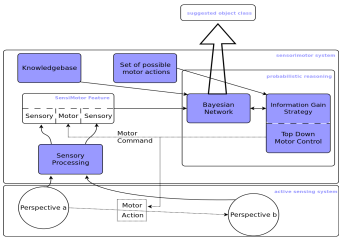
A5-[ActionSpace]

Tobias Kluth, David Nakath, Thomas Reineking, Christoph Zetzsche, Kerstin Schill

Introduction

Humans can recognize objects robustly and accurately. There is evidence that in natural settings this competence involves not only sensory processing but also motor components. This is not only true for the recognition act itself but also for the representation. However, while we have powerful models for pure sensory processing (hierarchical feed-forward networks [1]), models for a sensorimotor approach to object recognition are rare, and do often address only part of the problems. In particular, it is not yet clear what the specific relations between motor states and sensor information are, and how they enter into the underlying representation. Here we developed and implemented a probabilistic model for object recognition which combines motor states and bottom-up processes of feature extraction in an integrated sensorimotor architecture. Only this combination makes the investigation of optimizing strategies regarding motor controls possible. The top-down process computing the next movement of the robot then is modeled by an information gain strategy which uses a sensorimotor knowledge base to obtain the most informative motor action [2]. In a training phase the knowledge base is learned from real data to obtain the sensorimotor representation. We show how the integration of motor actions effects task performance in comparison to the modeling approach which only takes sensor information into account.

Model



The complete sensorimotor system consists of multiple subsystems:

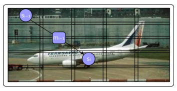
- *Active sensing system:* Can obtain sensor information and perfom motor actions.
- *Sensory processing system:* The raw sensor information is processed (clustered GIST features; 100 clusters) and fed into the SensoriMotor Feature (SMF).
- *Knowledge base:* Provides the learned joint distribution of SMF and object class Y.
- *Probabilistic reasoning system:* This systems uses a Bayesian network to infer the object class and to provide the top-down motor control obtained by an information gain strategy.

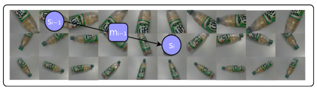
Knowledge Base

The knowlege base used provides a full joint probability distribution over all occourences of all possible SensoriMotor Features while looking at an object of class Y. This implies that with the support of the knowledge base every probabilistic query relevant in the context of the system can be answered. The first-order SensoriMotor Feature is defined as a tripel

$$SMF_i := \{s_{i-1}, m_{i-1}, s_i\},$$

where m_{i-1} is the relative movement between the sensor information s_{i-1} and s_i at step $i-1$ and i.





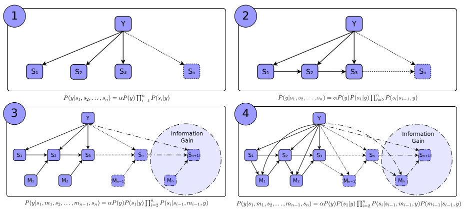
In the learning process every possible SMF is generated, which means that from every possible view on a known object every possible relative move is performed. The knowledge base is then defined as the joint probability distribution

$$kb := P(S_{i-1}, M_{i-1}, S_i, Y).$$

Bayesian Inference / Information Gain

We designed 4 Bayesian networks (BN) differing in assumptions of conditional independence. We can distinguish between two classes of networks:

- *Sensor network:* Only sensor information is considered for object recognition (BN1). The incorporation of an information gain strategy regarding movements is not possible but additionally statistical dependencies between sensor information are taken into account (BN2).
- *Sensorimotor network:* Sensor and motor information is considered such that the system is able to choose an optimal motor command by an information gain strategy (BN3, BN4).



$P(y|s_1, s_2, \ldots, s_n) = \alpha P(y) \prod_{i=1}^{n} P(s_i|y)$

$P(y|s_1, s_2, \ldots, s_n) = \alpha P(y) P(s_1|y) \prod_{i=2}^{n} P(s_i|s_{i-1}, y)$

$P(y|s_1, m_1, s_2, \ldots, m_{n-1}, s_n) = \alpha P(y) P(s_1|y) \prod_{i=2}^{n} P(s_i|s_{i-1}, m_{i-1}, y)$

$P(y|s_1, m_1, s_2, \ldots, m_{n-1}, s_n) = \alpha P(y) P(s_1|y) \prod_{i=2}^{n} P(s_i|s_{i-1}, m_{i-1}, y) P(m_{i-1}|s_{i-1}, y)$

The Information Gain Strategy employed relies on maximizing a utility function which calculates the difference between the current entropy $H(Y)$ and the conditional entropy $H(Y|S_{n+1}, m_n)$ (all probabilities are conditioned by $s_{1:n}, m_{1:n-1}$).

$$IG(Y, (S_{n+1}, m_n)) = H(Y) - H(Y|S_{n+1}, m_n)$$

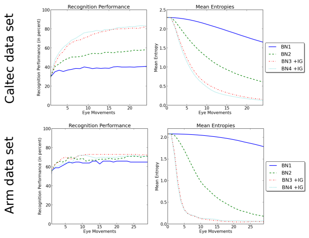
This equation can be rewritten to calculate the next move through minimizing the expected entropy resulting from that move:

$$m^* = \arg\min_{m_n}(E_{S_{n+1}}[H(Y|s_{1:n}, S_{n+1}, m_{1:n})])$$

Results

Ten fold cross validation was conducted on a subset of the Caltec256 data set [3] as well as on a data set made by a robotic arm with a camera attached. The properties of the data sets are:

- *Caltec data set:* 10 object classes, 100 objects each, 25 absolute positions, 300 SMFs per object, 80 possible relative movements.
- *Arm data set:* 8 object classes, 10 objects each, 30 absolute positions, 435 SMFs per object, 94 possible relative movements.



Including relative movements obtained by an Information Gain strategy (BN3, BN4) improves the performance over networks which just incorporate visual information (BN1, BN2).

Conclusion

We find that the information in the data is not only encoded in the sensory part. Including statistical dependies within the sensory space improves the performance but the use of an Information Gain strategy gives a much higher perormance improvement. The sensorimotor representation with its higher dimensionality thus has the potential to make powerful strategies possible.

References

[1] Serre T, Oliva A, Poggio T (2007). A feedforward architecture accounts for rapid categorization. PNAS 15(105): 6424-6429.

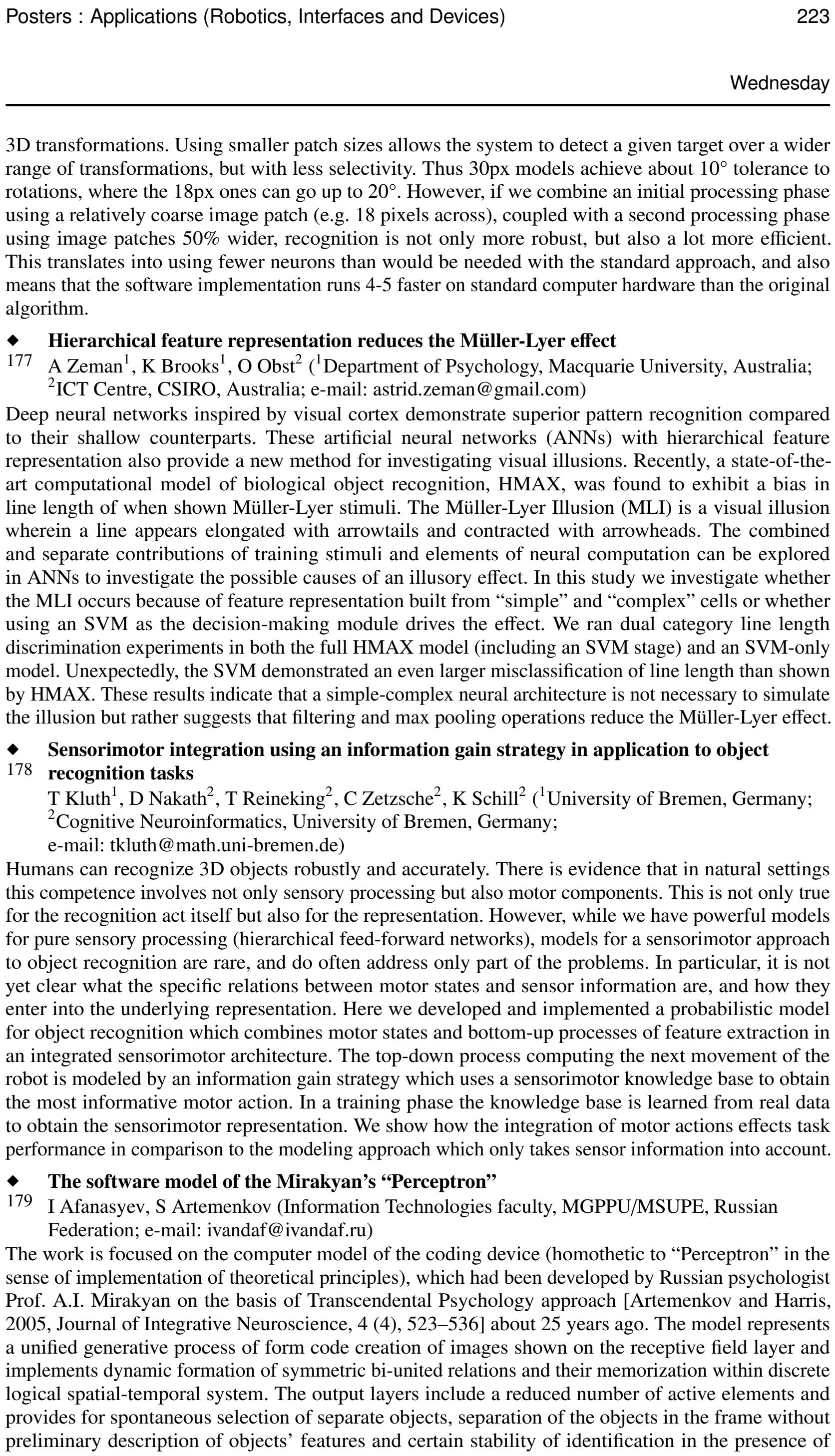
[2] Schill K, Umkehrer E, Beinlich S, Krieger G, Zetzsche C (2001). Scene analysis by saccadic eye-movements: Top-down and bottom-up modeling. J. Electron. Imaging. 10(1): 152-160.

[3] Griffin G, Holub A, Perona P (2007). Caltech-256 object category dataset.

Universität Bremen

DFG Deutsche Forschungsgemeinschaft

UNI FREIBURG



3D transformations. Using smaller patch sizes allows the system to detect a given target over a wider range of transformations, but with less selectivity. Thus 30px models achieve about 10° tolerance to rotations, where the 18px ones can go up to 20°. However, if we combine an initial processing phase using a relatively coarse image patch (e.g. 18 pixels across), coupled with a second processing phase using image patches 50% wider, recognition is not only more robust, but also a lot more efficient. This translates into using fewer neurons than would be needed with the standard approach, and also means that the software implementation runs 4-5 faster on standard computer hardware than the original algorithm.

◆ **Hierarchical feature representation reduces the Müller-Lyer effect**
177 A Zeman[1], K Brooks[1], O Obst[2] ([1]Department of Psychology, Macquarie University, Australia; [2]ICT Centre, CSIRO, Australia; e-mail: astrid.zeman@gmail.com)

Deep neural networks inspired by visual cortex demonstrate superior pattern recognition compared to their shallow counterparts. These artificial neural networks (ANNs) with hierarchical feature representation also provide a new method for investigating visual illusions. Recently, a state-of-the-art computational model of biological object recognition, HMAX, was found to exhibit a bias in line length of when shown Müller-Lyer stimuli. The Müller-Lyer Illusion (MLI) is a visual illusion wherein a line appears elongated with arrowtails and contracted with arrowheads. The combined and separate contributions of training stimuli and elements of neural computation can be explored in ANNs to investigate the possible causes of an illusory effect. In this study we investigate whether the MLI occurs because of feature representation built from "simple" and "complex" cells or whether using an SVM as the decision-making module drives the effect. We ran dual category line length discrimination experiments in both the full HMAX model (including an SVM stage) and an SVM-only model. Unexpectedly, the SVM demonstrated an even larger misclassification of line length than shown by HMAX. These results indicate that a simple-complex neural architecture is not necessary to simulate the illusion but rather suggests that filtering and max pooling operations reduce the Müller-Lyer effect.

◆ **Sensorimotor integration using an information gain strategy in application to object**
178 **recognition tasks**
T Kluth[1], D Nakath[2], T Reineking[2], C Zetzsche[2], K Schill[2] ([1]University of Bremen, Germany; [2]Cognitive Neuroinformatics, University of Bremen, Germany; e-mail: tkluth@math.uni-bremen.de)

Humans can recognize 3D objects robustly and accurately. There is evidence that in natural settings this competence involves not only sensory processing but also motor components. This is not only true for the recognition act itself but also for the representation. However, while we have powerful models for pure sensory processing (hierarchical feed-forward networks), models for a sensorimotor approach to object recognition are rare, and do often address only part of the problems. In particular, it is not yet clear what the specific relations between motor states and sensor information are, and how they enter into the underlying representation. Here we developed and implemented a probabilistic model for object recognition which combines motor states and bottom-up processes of feature extraction in an integrated sensorimotor architecture. The top-down process computing the next movement of the robot is modeled by an information gain strategy which uses a sensorimotor knowledge base to obtain the most informative motor action. In a training phase the knowledge base is learned from real data to obtain the sensorimotor representation. We show how the integration of motor actions effects task performance in comparison to the modeling approach which only takes sensor information into account.

◆ **The software model of the Mirakyan's "Perceptron"**
179 I Afanasyev, S Artemenkov (Information Technologies faculty, MGPPU/MSUPE, Russian Federation; e-mail: ivandaf@ivandaf.ru)

The work is focused on the computer model of the coding device (homothetic to "Perceptron" in the sense of implementation of theoretical principles), which had been developed by Russian psychologist Prof. A.I. Mirakyan on the basis of Transcendental Psychology approach [Artemenkov and Harris, 2005, Journal of Integrative Neuroscience, 4 (4), 523–536] about 25 years ago. The model represents a unified generative process of form code creation of images shown on the receptive field layer and implements dynamic formation of symmetric bi-united relations and their memorization within discrete logical spatial-temporal system. The output layers include a reduced number of active elements and provides for spontaneous selection of separate objects, separation of the objects in the frame without preliminary description of objects' features and certain stability of identification in the presence of

Active sensorimotor object recognition in three-dimensional space

David Nakath, Tobias Kluth, Thomas Reineking, Christoph Zetzsche, and Kerstin Schill

Cognitive Neuroinformatics, University of Bremen, Enrique-Schmidt-Straße 5, 28359 Bremen, Germany,
dnakath@informatik.uni-bremen.de,
WWW home page: http://www.informatik.uni-bremen.de/cog_neuroinf/en

Abstract. Spatial interaction of biological agents with their environment is based on the cognitive processing of sensory as well as motor information. There are many models for sole sensory processing but only a few for integrating sensory and motor information into a unifying sensorimotor approach. Additionally, neither the relations shaping the integration are yet clear nor how the integrated information can be used in an underlying representation. Therefore, we propose a probabilistic model for integrated processing of sensory and motor information by combining bottom-up feature extraction and top-down action selection embedded in a Bayesian inference approach. The integration of sensory perceptions and motor information brings about two main advantages: (i) Their statistical dependencies can be exploited by representing the spatial relationships of the sensor information in the underlying joint probability distribution and (ii) a top-down process can compute the next most informative region according to an information gain strategy. We evaluated our system in two different object recognition tasks. We found that the integration of sensory and motor information significantly improves active object recognition, in particular when these movements have been chosen by an information gain strategy.

Keywords: sensorimotor, object recognition, Bayesian inference, information gain

1 Introduction

The capabilities of artificial systems are easily exceeded by humans when it comes to perception-based interaction with the environment. With this in mind, it seems reasonable to take a closer look at the main principles of the human visual perception process. Especially the reciprocal advantageous interplay of motion and sensory information, which was early recognized by Gibson [1] and Neisser [2], should be considered here. Based on analog arguments, an "active perception" approach was proposed by [3,4,5]. The strong interrelation between movements in space and corresponding sensory perceptions can foster the even stronger concept of a *sensorimotor representation* [6,7,8,9]. In this concept, the classic notion of separate cognitive processing stages for sensory and motor information does not hold. In fact, they are integrated into one sensorimotor coding. This is a precondition for a sensorimotor representation which is established from the specific pattern of alternating sensory perceptions and spatial motor actions [6,10]. The constant checking and confirmation of sensory and motor information against an internal cognitive model then constitutes a scanpath, and thus the perception of a particular object [11,12].

To be able to check such an internal sensorimotor model of an object, the next motor action has to be chosen accordingly by an object recognition system. Generally, the problem of action selection can be solved in numerous ways, but as information gathering should be the purpose of motor actions it seems reasonable to choose an information-theoretic criterion. Prior research in this area often found that the principle of *information gain* is well suited to select an appropriate next action. This has been shown by [13] in the context of decision trees, where information gain was used to decide which attributes are the most relevant ones. Robotics also proofed to be a suitable domain, as information gain can be used there to actively reduce the uncertainty of the robot regarding its position and spatial environment [14,15]. Additionally, information gain was not only used to explain human selection behavior [16,17] but also to mimic it: Both in the form of human-like expert systems [18,19] and with a modeled sensorimotor loop in a saccadic eye movement control system [20,21].

Based on the preceding considerations the basic sketch of a sensomotoric object recognition system becomes apparent. Building upon the research of Schill, Zetzsche, and coworkers [20,21], we propose a sensomotoric probabilistic reasoning system for active object recognition integrating sensory perceptions and motor actions. Our system is inspired by the human perception process and therefore should model a sequential pattern of actions controlled by a top-down and a bottom-up process. Evidence suggests that sensory perception and motor actions partly share the same cognitive processing stage which makes it reasonable to integrate them into one single sensorimotor feature (SMF) [22,23,24]. Through this integration two improvements come into effect: (i) The accuracy of the recognition process is improved through the additional motor information which encodes spatial relations and (ii) the next motor action can be chosen according to the maximum information gain principle, thus supplying the sensors with an optimized input in the next recognition step.

The basic architecture of the system we propose is outlined in Sect. 2. In Sect. 3, we describe the implementation of the system. Subsequently, Sect. 4 shows the results of the evaluation in two different scenarios: Optimized control of 3D movements of a camera mounted on a robotic arm and simulated sensor movements on images from the Caltech 256 dataset. The paper is concluded with a discussion of the specific advantages offered by the proposed sensorimotor architecture.

2 Sensomotoric Object Recognition System

The sensorimotor system described in the following is a generic architecture (see Fig. 1). In the case of visual object recognition, the senorimotor loop starts out with a particular pose of the active sensor which passes its raw sensor data to the sensory processing module. After processing, the sensory data becomes part of a new sensorimotor feature, which is then fed into the probabilistic reasoning module. The Bayesian inference module calculates the new posterior distribution based on a previously-learned sensorimotor representation. The posterior distribution constitutes the current belief of the system. This belief is used by the information gain strategy to choose an optimal next movement from the set of possible motor actions. The selected movement then also becomes part of the sensorimotor feature and is subsequently executed by the active sensor. The whole process results in a new sensor pose, which in turn delivers new raw sensory data to enter the next cycle of the sensorimotor loop.

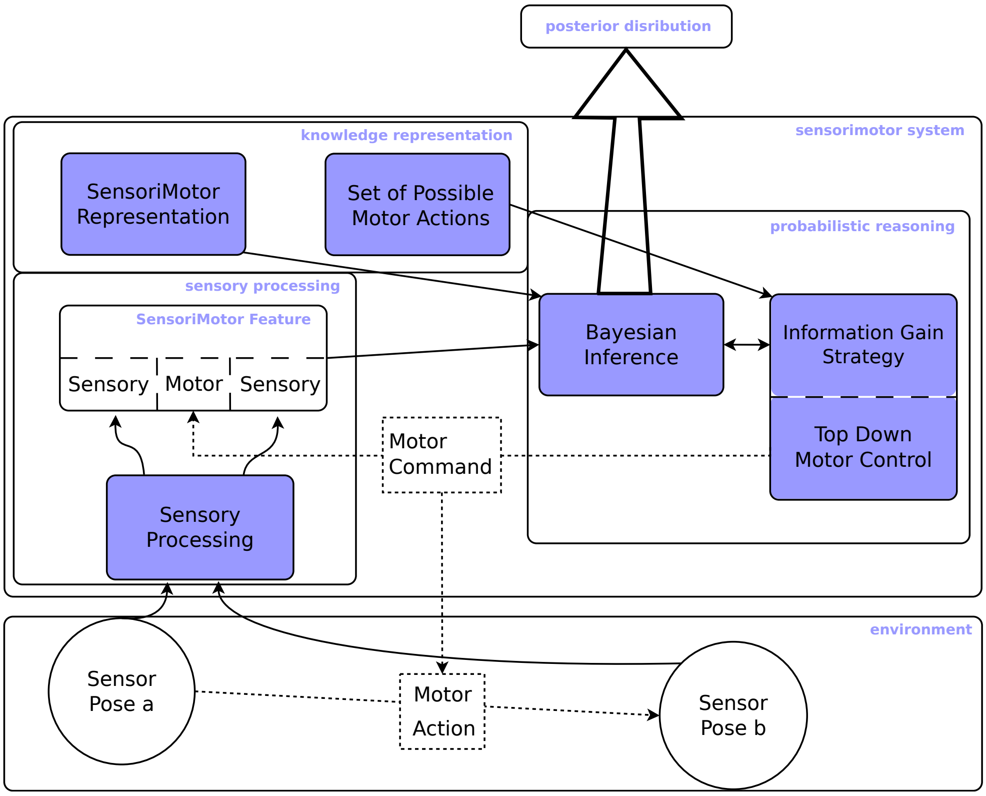


Fig. 1: Sensorimotor Information Processing System.

More formally spoken, the system depends on an *active sensor* (AS), which can be controlled such that it delivers information about a specific aspect of the world. In Fig. 1, the two arrows pointing from the sensor poses to the sensory processing module correspond to the mapping $A : U \times X \to R$, where U is the space of all motor actions which are currently possible, X is the state space of the active sensor, and R is the raw sensor data space.

The system has no knowledge about the actual state of the AS, instead it is only informed about the currently possible motor actions U. These are of course dependent on the state in the sense of $U : X \to \mathcal{P}(\Omega_U)$, where Ω_U is the set of all possible motor controls and $\mathcal{P}$ denotes the power set. Assuming that the output of the function U is given, we write U instead of $U(x),\ x \in X$, for convenience. Considering the state-agnostic behavior, the formal representation of the AS can be redefined to $A_x : U \to R$ where the index x recalls the dependency on the state

$$A_x(u) := A(x, u) = r,\ x \in X,\ u \in U(x),\ r \in R. \tag{1}$$

The only time-dependent variables are the sensor state x and the relative motor control u. In contrast, the world is assumed to be static which implies no dynamic changes in the raw sensor data $r \in R$.

This data is fed into the *sensory processing* (SP) which mainly extracts the relevant features belonging to a feature space F, i.e., $SP : R \to F$. Subsequently, the quantization operation $Q_S : F \to S$ maps the features to a specific feature class in the finite and countable space S. The possible motor actions are mapped with $Q_M : \Omega_U \to M$ to the finite countable set of actions M to yield a manageable product space of sensory and motor information. The results of these quantizations then become part of a sensorimotor feature (SMF). The single quantizations are represented in Fig. 1 by the arrows from the sensory processing and the motor command to the first-order sensorimotor feature which is defined as the triple

$$SMF_i := \{s_{i-1}, m_{i-1}, s_i\}, \tag{2}$$

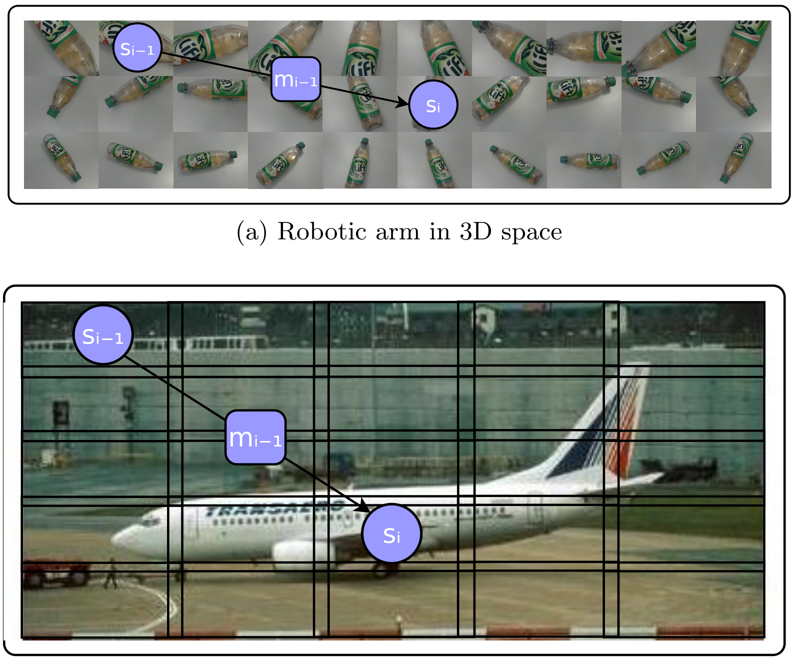
where $m_{i-1} := Q_M(u_{i-1})$ is the intermediate motor action between the sensor information s_{i-1} and s_i at time step t_{i-1} and t_i (see Fig. 2). The whole chain of operations to obtain the sensor information at a time step t_i can be described by

$$s_i := (Q_S \circ SP \circ A_x)(u_{i-1}). \tag{3}$$

The *knowledge representation* is comprised of the currently available motor actions U and the learned sensorimotor representation (SMR), which is a full joint probability distribution of $SMFs$ and the classes represented by the discrete random variable Y. Every possible SMF is generated on a set of known objects in a training phase. This means that, from every possible state x, every possible motor action u is performed, resulting in

$$SMR := P(SMF, Y) = P(S_{i-1}, M_{i-1}, S_i, Y). \tag{4}$$

The *probabilistic reasoning* module consists of a Bayesian inference approach accompanied by an information gain strategy. They rely on bottom-up sensory data and top-down information from the knowledge representation. This design enables the Bayesian inference system to take into account motor actions, thus



(a) Robotic arm in 3D space

(b) Simulated active sensor in 2D space

Fig. 2: Exemplary sensorimotor features (SMF) drawn on the discretized views on an object, both for the 3D case shown in (a) and 2D case shown in (b). Here, s_{i-1} denotes the preceding sensory input, m_{i-1} denotes the preceding movement, and s_i denotes the current sensory input.

improving the posterior distribution over the object classes Y. Furthermore, the information gain strategy can choose an optimal next motor action for the active sensor, thus improving the input of the following Bayesian inference step.

3 Model Implementation

Based on the schematic outline presented above, we applied our system in the field of active object recognition. We consider both the case of an active sensor moving in 3D space and a simulated active sensor moving in 2D space (see Fig. 2).

3.1 Active Sensor Implementation

For the 3D case, we used a discrete set M of movements of a camera mounted on a robotic arm (see Fig. 2a), which resembles an observer actually moving around an object. For the 2D case, we used simulated active sensor movements on images of a reduced version of the Caltech 256 [25] dataset.[1] Here, M consists of all possible relative movements between the individual cells of a 5×5 grid (see Fig. 2b). The latter setup mimics eye movements of a stationary observer. Hence, in both cases holds $\Omega_U = M$ and the quantization Q_M is an identity operation.

Although the implemented sensors are of a fundamentally different nature, the following basic learning and recognition principles can be applied to both of them: In the learning phase, features are extracted from the training data (i.e., images from every reachable state of the active sensor), which corresponds to the mapping SP introduced above. As the robotic arm relies on views showing the entire object, GIST-features [26,27] are used, while the more local image patches of the 2D case are described by the SURF-feature [28] with the highest score on that patch. The quantization Q_S is then learned by performing a k-means clustering on the extracted features ($k = 15$).[2] In order to build the individual $SMFs$, features are extracted (see SP) and the results are assigned to clusters with the aid of the previously defined mapping Q_S. These labels are combined with the corresponding intermediate movement resulting in a set of $SMFs$. Finally, all generated $SMFs$ are stored in a Laplace-smoothed SMR.

3.2 Probabilistic Reasoning

The probabilistic reasoning is comprised of a Bayesian inference module in the form of a dynamic Bayesian network (BN) and a corresponding information gain strategy. Four of these probabilistic reasoning modules were implemented to examine the difference between *sensor networks*, which only take into account sensory information (which also implies that no information gain strategy is used), and *sensorimotor networks*, which take into account integrated sensory and motor information. The object recognition in the sense of machine vision then takes place by classification which is performed by choosing the class with the maximum posterior probability.

The first representative of the *sensor networks* is Bayesian network 1 (BN1) (see Fig. 3a), which resembles a naive Bayes model only taking into account the current sensory input s_i. Thus, the inference can be performed by

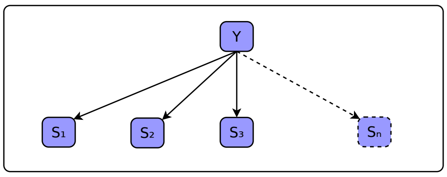
$$P(y|s_{1:n}) = \alpha P(y) \prod_{i=1}^{n} P(s_i|y), \tag{5}$$

where α is a normalizing constant guaranteeing that the probability function properties are satisfied and $s_{1:n}$ is a short notation for the n-tuple $(s_1, \ldots, s_n)$.

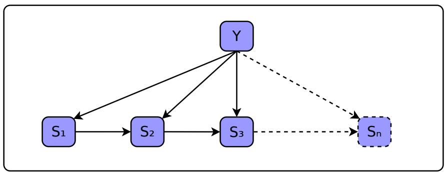
The second representative of the *sensor networks* is Bayesian network 2 (BN2) (see Fig. 3b), which assumes additional statistical dependencies between

[1] The reduced dataset consists in each case of 100 randomly-selected images from the object classes: *airplanes*, *cowboy-hat*, *faces*, *motorbikes*, *swan*, *breadmaker*, *diamond-ring*, *ketch*, *self-propelled-lawn-mower*, and *teapot.*

[2] We use only a small number of clusters in order to limit the number of model parameters and to prevent overfitting.



(a) BN1



(b) BN2

Fig. 3: The dynamic Bayesian sensor networks process only sensory information. BN1, which is shown in (a), represents a naive Bayes approach where the current sensory input s_i depends only on the object hypothesis Y. BN2, which is shown in (b), assumes statistical dependencies between the object hypothesis Y and the preceding sensory input s_{i-1} for every sensory input s_i.

the preceding and the current sensor information, s_{i-1} and s_i, resulting in

$$P(y|s_{1:n}) = \alpha P(y)P(s_1|y)\prod_{i=2}^{n} P(s_i|s_{i-1}, y). \tag{6}$$

Bayesian network 3 (BN3) (see Fig. 4a) uses the full information of the SMF and therefore belongs to the *sensorimotor networks*. The assumption that the current sensory input s_i depends on the preceding sensory input s_{i-1} and the intermediary motor action m_{i-1} integrates motor and sensor information in the recognition process and permits the application of the information gain strategy to choose the next optimal movement. The inference can then be conducted by

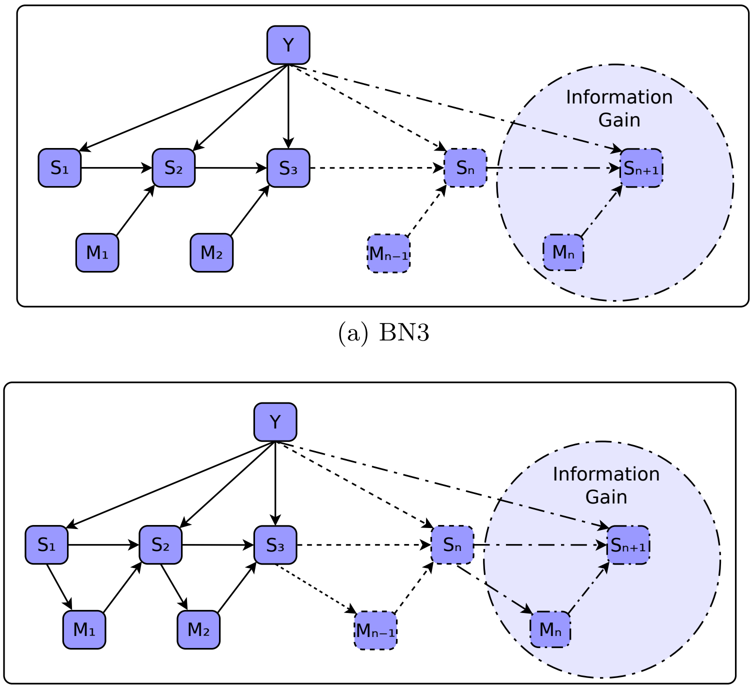
$$P(y|s_{1:n}, m_{1:n-1}) = \alpha P(y)P(s_1|y)\prod_{i=2}^{n} P(s_i|s_{i-1}, m_{i-1}, y). \tag{7}$$

Bayesian network 4 (BN4) (see Fig. 4b) mainly resembles BN3, but additionally allows statistical dependencies between the preceding sensory input s_{i-1} and the motor action m_{i-1}. The inference can thus be conducted by

$$P(y|s_{1:n}, m_{1:n-1}) = \alpha P(y)P(s_1|y)\prod_{i=2}^{n} P(s_i|s_{i-1}, m_{i-1}, y)P(m_{i-1}|s_{i-1}). \tag{8}$$

3.3 Information Gain

The strategy for action selection should satisfy two main properties: (i) The strategy should adapt itself to the current belief state of the system and (ii) the



(a) BN3

(b) BN4

Fig. 4: The dynamic Bayesian sensorimotor networks process integrated sensorimotor information. As motor information is taken into account, the next movement m_n can be chosen by the information gain strategy. BN3, which is shown in (a), assumes statistical dependencies between the current visual input s_i, the preceding movement m_{i-1}, the visual input s_{i-1}, and the hypothesis y. BN4, which is shown in (b), allows for the same dependencies as BN3 and, in addition, allows for a dependency of the preceding movement m_{i-1} on the preceding visual input s_{i-1}.

strategy should not make decisions in an heuristic fashion but tightly integrated into the axiomatic framework used for reasoning. The information gain strategy presented in this paper complies with both of these properties.

The information gain IG of a possible next movement m_n is defined as the difference between the current entropy $H(Y)$ and the conditional entropy $H(Y|S_{n+1}, m_n)$, i.e.,

$$IG(m_n) := H(Y) - H(Y|S_{n+1}, m_n), \tag{9}$$

where all probabilities are conditioned by $s_{1:n}, m_{1:n-1}$. This is equivalent to the mutual information of Y and (S_{n+1}, m_n) for an arbitrary m_n. As the current entropy $H(Y)$ is independent of the next movement m_n the most promising motor action m^* can be calculated by minimizing the expected entropy with respect to S_{n+1}, i.e.,

$$m^* = \underset{m_n}{\arg\min}(\underset{S_{n+1}}{E}[H(Y|s_{1:n}, S_{n+1}, m_{1:n})]). \tag{10}$$

Because the sensory input s_{n+1} is not known prior to executing m_n, the expected value over all possible sensory inputs S_{n+1} is taken into account. Subsequently, the so chosen motor action $m^* \in M$ can be integrated into the sensorimotor feature. The inverse mapping of Q_M can then be used to obtain a top-down motor command $u \in U$, which is then executed by the active sensor.

4 Evaluation

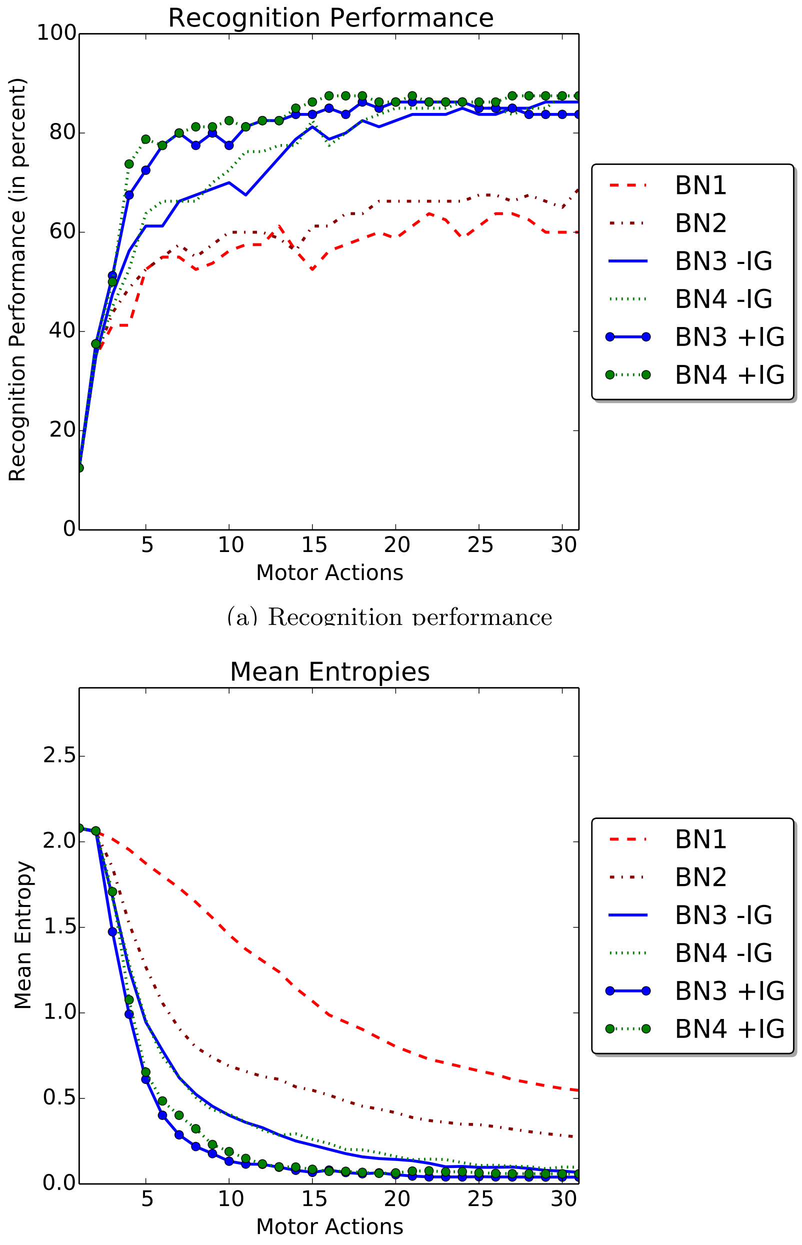
Both active sensor implementations were evaluated on two datasets based on a k-fold cross validation scheme ($k = 10$ for the 3D case and $k = 5$ for the 2D case). The case of 3D movements can be seen as a realistic test for robustness with noisy movements and sensor data. This realistic setting only allows a small dataset, consisting of 8 object classes, each containing 10 objects from 30 different points of view (see Fig. 2a). The case of simulated 2D movements allows for a larger dataset, as movements are simulated on a 5×5-grid on images. Therefore, the Caltech 256 dataset was chosen to serve as a scalability test. Our aim here is not to compete with state of the art recognition approaches but rather to investigate the effects of taking motor information into account while relying on a larger data basis consisting of 10 object classes, each with 100 samples.

Figure 5 depicts the results of the 3D case. The integration of information-gain-guided motor actions in the sensorimotor networks (BN3 +IG, BN4 +IG) proves to be beneficial in terms of recognition performance (see Fig. 5a). The sensor networks (BN1, BN2) perform worse, which holds true for the recognition performance as well as for the mean entropy reduction (see Fig. 5b). To illustrate the effect of the information gain strategy, the sensorimotor networks performing information-gain-guided movements (BN3 +IG, BN4 +IG) and random movements (BN3 -IG, BN4 -IG) were compared to each other. The sensorimotor networks with information-gain-guided movements perform better within the first 15 movements (see Fig. 5a), which is reflected by a steeper reduction in entropy (see Fig. 5b).

In the Caltech 256 evaluation (see Fig. 6) the advantage of using sensorimotor networks with information-gain-guided movements (BN3 +IG, BN4 +IG) for recognition persists over time (see Fig. 6a). This holds true compared to the sensorimotor networks with random movements (BN3 -IG, BN4 -IG) as well as compared to the sensor networks (BN1, BN2). This persisting advantage is also shown by the corresponding evolution of the mean entropies plotted in Fig. 6b.

5 Discussion

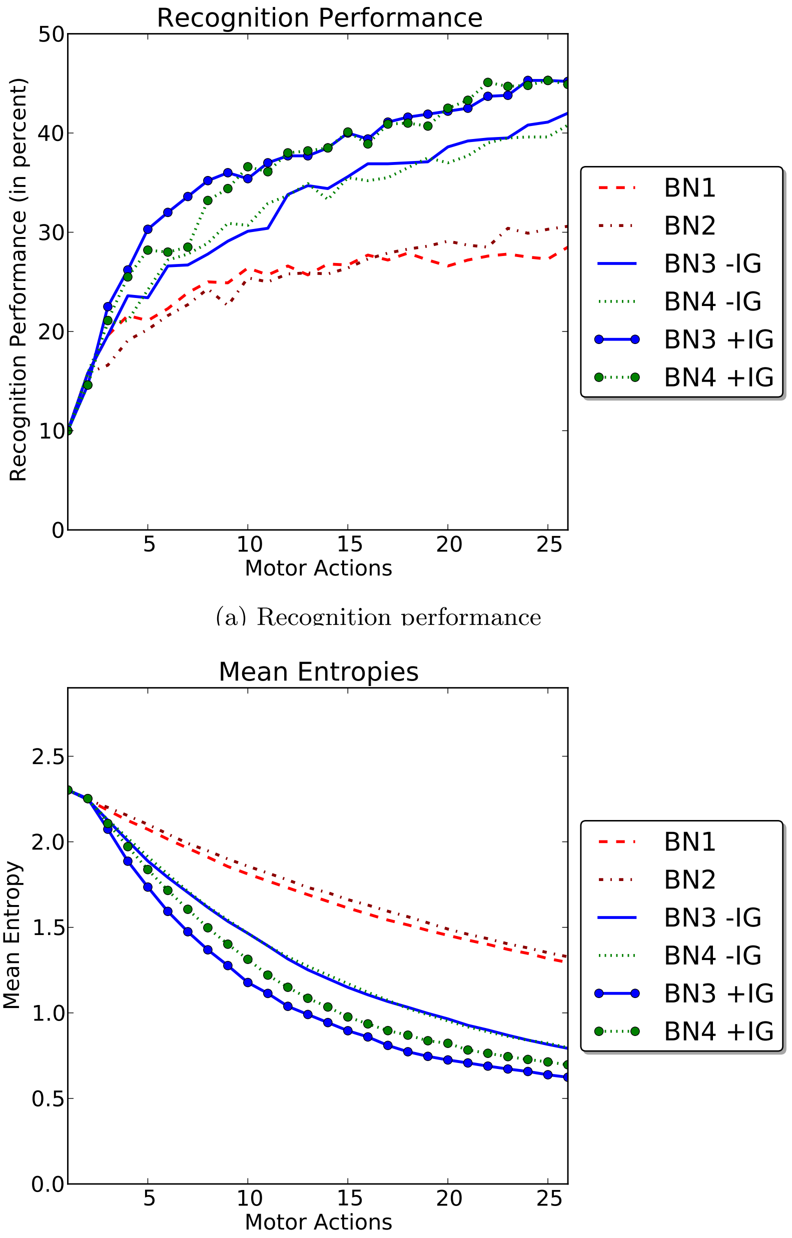
We have examined a sensorimotor object recognition system which chooses the next perspective on an object according to the principle of maximum information gain. The underlying sensorimotor representation improved the recognition



(a) Recognition performance

(b) Mean entropies

Fig. 5: Results of the robotic arm evaluation. BN 1, 2, 3 -IG, and 4 -IG executed random movements while BN 3 +IG and BN4 +IG executed information-gain-guided movements (GIST features, 15 clusters, 94 possible relative movements, inhibition of return). Recognition performance shown in (a) and mean entropy of the posterior distribution shown in (b) are both plotted against the number of performed motor actions.



(a) Recognition performance

(b) Mean entropies

Fig. 6: Results of the Caltech 256 subset evaluation. BN 1, 2, 3 -IG, and 4 -IG executed random movements while BN 3 +IG and BN4 +IG executed information-gain-guided movements (SURF features, 15 clusters, 80 possible relative movements, inhibition of return). Recognition performance shown in (a) and mean entropy of the posterior distribution shown in (b) are both plotted against the number of performed motor actions.

performance and enabled the system to optimize its selective serial information intake. It could be shown that the proposed information gain strategy is well suited to control such a selection process.

In this paper, we restricted our focus to the recognition rate and the information gain strategy. However, the criteria for the optimal next step in a selective information intake process may vary in other contexts, e.g., the amount of time or energy required to perform individual actions. The system could be adapted to different contexts on the basis of multicriteria optimization approaches [29].

In principle, our system is able to cope with situations where it only has partial access to information about its environment at a given moment. To overcome this shortcoming, it can act in a sequential fashion to establish the full picture. This is often seen as a contradiction but we could show here that, by integrating sensory and motor information, the underlying sensorimotor contingencies become usable, thus improving the process of sequential information intake controlled by reasonable intermediate actions.

Acknowledgements This work was supported by DFG, SFB/TR8 Spatial Cognition, Project A5-[ActionSpace].

References

1. Gibson, J.: The ecological approach to visual perception. Boston: Houghton Mifflin (1992)
2. Neisser, U.: Cognition and reality: Principles and implications of cognitive psychology. WH Freeman/Times Books/Henry Holt & Co (1976)
3. Bajcsy, R.: Active perception. Proceedings of the IEEE **76**(8) (1988) 966–1005
4. Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. International Journal of Computer Vision **1**(4) (1988) 333–356
5. Ballard, D.H.: Animate vision. Artificial intelligence **48**(1) (1991) 57–86
6. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and brain sciences **24**(5) (2001) 939–972
7. Noë, A.: Action in Perception. MIT Press (2004)
8. Prinz, W.: A common coding approach to perception and action. Springer (1990)
9. Hommel, B., Müsseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): A framework for perception and action planning. Behavioral and brain sciences **24**(05) (2001) 849–878
10. O'Regan, J.K.: What it is like to see: A sensorimotor theory of perceptual experience. Synthese **129**(1) (2001) 79–103
11. Noton, D., Stark, L.: Scanpaths in saccadic eye movements while viewing and recognizing patterns. Vision research **11**(9) (1971) 929–IN8
12. Stark, L.W., Choi, Y.S.: Experimental metaphysics: The scanpath as an epistemological mechanism. In W.H. Zangemeister, H.S., Freksa, C., eds.: Visual Attention and Cognition. Volume 116 of Advances in Psychology. North-Holland (1996) 3–69
13. Quinlan, J.R.: Induction of decision trees. Machine learning **1**(1) (1986) 81–106
14. Cassandra, A.R., Kaelbling, L.P., Kurien, J.A.: Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In: Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems' 96, IROS 96. Volume 2., IEEE (1996) 963–972

15. Stachniss, C., Grisetti, G., Burgard, W.: Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In: Robotics: Science and Systems. Volume 2. (2005) 65–72
16. Oaksford, M., Chater, N.: Information gain explains relevance which explains the selection task. Cognition **57**(1) (1995) 97–108
17. Friston, K., Kilner, J., Harrison, L.: A free energy principle for the brain. Journal of Physiology-Paris **100**(1–3) (2006) 70–87 Theoretical and Computational Neuroscience: Understanding Brain Functions.
18. Schill, K., Pöppel, E., Zetzsche, C.: Completing knowledge by competing hierarchies. In: Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc. (1991) 348–352
19. Schill, K.: Decision support systems with adaptive reasoning strategies. In: Foundations of Computer Science, Springer (1997) 417–427
20. Zetzsche, C., Schill, K., Deubel, H., Krieger, G., Umkehrer, E., Beinlich, S.: Investigation of a sensorimotor system for saccadic scene analysis: an integrated approach. In: Proceedings of the 5th International Conference of Simulation of Adaptive Behaviour. Volume 5. (1998) 120–126
21. Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., Zetzsche, C.: Scene analysis with saccadic eye movements: top-down and bottom-up modeling. Journal of Electronic Imaging **10**(1) (2001) 152–160
22. Zetzsche, C., Wolter, J., Schill, K.: Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. Cognitive processing **9**(4) (2008) 283–297
23. Reineking, T., Wolter, J., Gadzicki, K., Zetzsche, C.: Bio-inspired Architecture for Active Sensorimotor Localization. In Hölscher, C., Shipley, T., Olivetti Belardinelli, M., Bateman, J., Newcombe, N., eds.: Spatial Cognition VII. Volume 6222 of Lecture Notes in Computer Science., Portland, Oregon, Springer Berlin/Heidelberg (2010) 163–178
24. Schill, K., Zetzsche, C., Hois, J.: A belief-based architecture for scene analysis: From sensorimotor features to knowledge and ontology. Fuzzy Sets and Systems **160**(10) (2009) 1507–1516
25. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Technical report, California Institute of Technology (2007)
26. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Prorgess in brain research **155** (2006) 23–36
27. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision **42**(3) (2001) 145–175
28. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding **110**(3) (2008) 346–359
29. Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In: Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on. Volume 1., IEEE (1999) 35–40

Affordance-based object recognition using interactions obtained from a utility maximization principle

Tobias Kluth, David Nakath, Thomas Reineking, Christoph Zetzsche, and Kerstin Schill

Cognitive Neuroinformatics, University of Bremen, Enrique-Schmidt-Straße 5, 28359 Bremen, Germany,
tkluth@math.uni-bremen.de

Abstract. The interaction of biological agents within the real world is based on their abilities and the affordances of the environment. By contrast, the classical view of perception considers only sensory features, as do most object recognition models. Only a few models make use of the information provided by the integration of sensory information as well as possible or executed actions. Neither the relations shaping such an integration nor the methods for using this integrated information in appropriate representations are yet entirely clear. We propose a probabilistic model integrating the two information sources in one system. The recognition process is equipped with an utility maximization principle to obtain optimal interactions with the environment.

Keywords: affordance, sensorimotor object recognition, information gain

1 Introduction

The ability of humans to reliably recognize objects in the environment is still a largely unsolved problem for artificial systems. Usually, object recognition is understood as a classification problem where a fixed mapping from feature vectors to object classes is learned. This is in line with the classical view of perception as the input from world to mind and of action as the output from mind to world [1], which implies a dissociation between the capacities for perception and action. However, there is strong evidence that object recognition cannot be understood independently of the interaction of an agent with its environment [2]. "Active perception" approaches [3, 4] take this partially into account, but actions are not merely means for acquiring new information, they also provide evidence themselves for the recognition [5].What an agent perceives is thus also determined by what it does or what it is able to do [2].

Research in the direction of affordances by Gibson [6] also provides evidence that affordances are key ingredients of the perceptual process. A variety of studies regarding object recognition show that the visual information of a manipulable object causes an activation of representations of actions which can typically be executed on the object [7].The advantageous interplay between sensory and action information, which was also recognized by Neisser [8], should be considered in the recognition process.

The strong interrelation between motor actions and sensory perceptions is basis for the concept of a sensorimotor representation [2, 9]. Similarly to the affordance point of view the processing stages for sensory and motor information are not separated. The approach including the actions in the representation gives the opportunity to choose the next action such that a specific objective is pursued. Generally, the problem of action selection can be solved in numerous ways, but as information gathering should be one major purpose of motor actions it is appropriate to consider an information-theoretic utility function. Prior research in this area often found that the principle of *information gain* is well suited to select an appropriate next action.

In this paper, we propose a system for object recognition which incorporates both the information gain principle from sensorimotor systems and the theoretical concept of affordances. Building upon the investigations in [10], we developed a sensomotoric probabilistic reasoning system for affordance-based object recognition. The design of our architecture is motivated by two main goals: i) to provide a clear relation to Bayesian inference approaches, and ii) to enable a comparison between the classic sensory approach and the sensorimotor, affordance-oriented approach within one common probabilistic framework.

2 Object Recognition System

The system described in the following is a generic architecture (see Fig. 1). The recognition loop starts out with a particular pose of an object which is perceived by a sensor. The sensor passes its raw data to the sensory processing module. After processing, the sensory data becomes part of a new sensorimotor feature, which is then fed into the probabilistic reasoning module. The processed sensory data are also used to obtain a set of possible interactions, i.e., the affordances offered by the sensory data related to the abilities of the agent. The Bayesian inference module calculates the new posterior distribution based on a previously-learned sensorimotor representation. This representation contains the learned perceptual consequences of an interaction in a given state for every object class. The posterior distribution constitutes the current belief of the system. This belief is used by the information gain strategy to choose an optimal next action from the set of possible interactions. The selected interaction then also becomes part of the sensorimotor feature and is subsequently executed by the agent. The whole process results in a new state, which in turn delivers new raw sensory data to enter the next cycle of the recognition loop.

More formally speaking, the system depends on an *agent*, which can be controlled such that it perceives information about a specific aspect of the world. In Fig. 1, the two arrows pointing from the states to the sensory processing module correspond to the mapping $A : U \times X \to R$, where U is the space of all interactions which are currently possible, X is the state space, and R is the raw sensor data space.

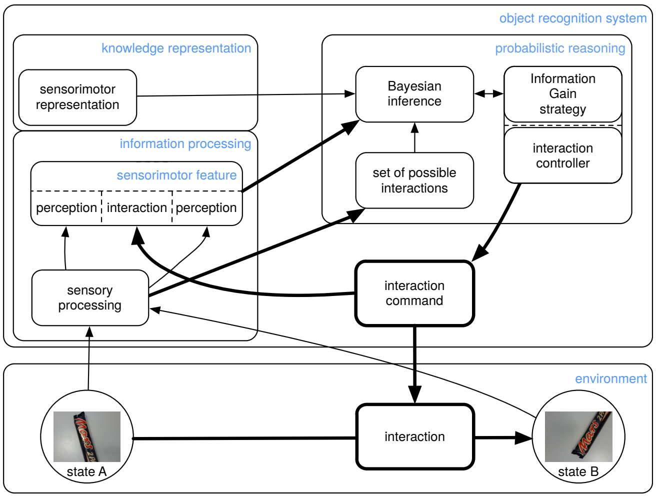


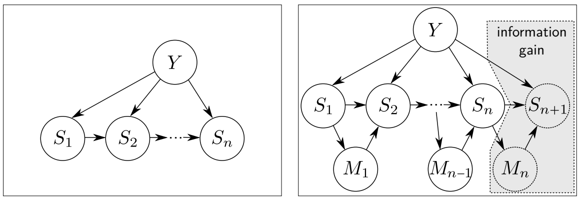
Fig. 1: Architecture of the object recognition system.

The system has no explicit knowledge about the actual state, and the currently possible interactions U. The possible interactions are of course dependent on the state but nevertheless both information must be obtained from the sensor data. The sensoric dependency on the state is formalized by the mapping $U : X \to \mathcal{P}(\Omega_U)$, where Ω_U is the set of all possible interactions and $\mathcal{P}$ denotes the power set. Note that U comprises the link from the state to the sensory processing module and the following link to the set of possible interactions in Fig. 1, i.e., the perceived affordances. Assuming that the output of the function U is given, we write U instead of $U(x),\ x \in X$, for convenience. Considering the state-agnostic behavior, the influence of the agent can be formally redefined to $A_x : U \to R$ with $A_x(u) := A(x, u) = r,\ x \in X,\ u \in U(x),\ r \in R$. The only time-dependent variables are the state x and the interaction u.

The raw sensor data $r \in R$ is fed into the *sensory processing* (SP) which mainly extracts the relevant features belonging to a feature space F, i.e., $SP : R \to F$. Subsequently, the quantization operation $Q_S : F \to S$ maps the features to a specific feature class in the finite space S. The possible interactions are mapped with $Q_M : \Omega_U \to M$ to the finite set of interactions M to yield a manageable product space of sensory information and actions. The results of these quantizations then become part of a sensorimotor feature (SMF). The single quantizations are represented in Fig. 1 by the arrows from the sensory processing module and the interaction command to the sensorimotor feature which is defined as the triple

$$SMF_i := \{s_{i-1}, m_{i-1}, s_i\}, \tag{1}$$

where $m_{i-1} := Q_M(u_{i-1})$ is the interaction between the sensor information s_{i-1} and s_i at time step t_{i-1} and t_i. The whole chain of operations to obtain the sensor information at a time step t_i can be described by $s_i := (Q_S \circ SP \circ A_x)(u_{i-1})$.



(a) BN1 (b) BN2

Fig. 2: In Bayesian network BN1 (a) sensory information S_n is processed only to obtain the object class Y. Bayesian network BN2 (b) is equipped with the information gain startegy which takes also the action M_n into account.

The *knowledge representation* is comprised of the learned sensorimotor representation (SMR), which is a full joint probability distribution of SMFs and the classes represented by the discrete random variable Y. Every possible SMF is generated on a set of known objects in a training phase. This means that, from every possible state x, the sensory consequence of every possible action u is perceived, resulting in

$$SMR := P(SMF, Y) = P(S_{i-1}, M_{i-1}, S_i, Y). \tag{2}$$

The *probabilistic reasoning* module consists of a Bayesian inference approach accompanied by an information gain strategy. They rely on bottom-up sensory data and top-down information from the knowledge representation. The information gain strategy can choose an optimal next interaction for the agent, thus improving the input of the following Bayesian inference step.

3 Model Implementation and Outlook

Based on the schematic outline presented above, we applied our system to object recognition using a robotic arm interacting with objects in 3D space. We used a discrete set of interactions M of a robotic arm with an object which comprise the relative position/pose of the visual sensor to the object ($\Omega_U = M$, $Q_M = Id$).

In the learning phase, features are extracted from the training data (images from every reachable state). GIST-features [11] are used to describe the sensory input, i.e., defining SP. The quantization Q_S is then learned by performing a k-means clustering on the extracted features. In order to build the individual SMFs, features are extracted and the results are assigned to clusters with the aid of the previously defined mapping Q_S. These labels are combined with the corresponding interactions in a set of SMFs. Finally, all generated SMFs are stored in a Laplace-smoothed SMR.

The probabilistic reasoning is comprised of a Bayesian inference module in the form of a dynamic Bayesian network (BN) and a corresponding information gain strategy. Two of these probabilistic reasoning modules were implemented to examine the difference between *sensor networks*, which only take into account sensory information (which also implies that no information gain strategy is used), and *affordance-based networks*, which integrate sensory perceptions and interactions. The object recognition in the sense of computer vision then takes place by classification which is performed by choosing the class with the maximum posterior probability.

The representative of the *sensor networks* is Bayesian network 1 (BN1) (see Fig. 2a), which resembles an extended naive Bayes model that additionally allows for statistical dependencies between the preceding and the current sensor information, s_{i-1} and s_i, resulting in

$$P(y|s_{1:n}) \propto P(y)P(s_1|y)\prod_{i=2}^{n} P(s_i|s_{i-1}, y), \tag{3}$$

where $s_{1:n}$ is a short notation for the n-tuple $(s_1, \ldots, s_n)$.

Bayesian network 2 (BN2) (see Fig. 2b) uses the full information of the SMF and therefore belongs to the *affordance-based networks.* The assumption that the current sensory input s_i depends on the action m_{i-1} integrates sensory perceptions and actions in the recognition process and permits the application of the information gain strategy to choose the next optimal interaction. Additionally, it is assumed that the action m_{i-1} statistically depends on the preceding sensory input s_{i-1}. The inference can then be conducted by

$$P(y|s_{1:n}, m_{1:n-1}) \propto P(y)P(s_1|y)\prod_{i=2}^{n} P(s_i|s_{i-1}, m_{i-1}, y)P(m_{i-1}|s_{i-1}). \tag{4}$$

The strategy for action selection should satisfy two main properties: (i) The strategy should adapt itself to the current belief state of the system and (ii) the strategy should not make decisions in an heuristic fashion but tightly integrated into the axiomatic framework used for reasoning. The information gain strategy presented in this paper complies with both of these properties.

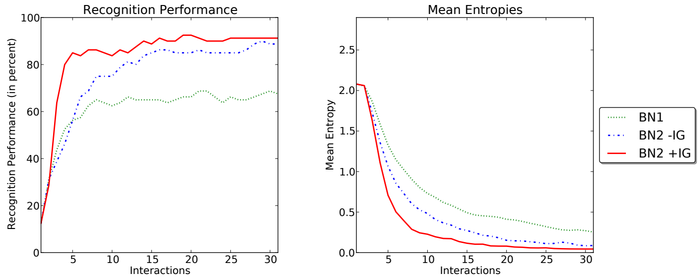
The information gain IG of a possible next action m_n is defined as the difference between the current entropy and the conditional entropy,

$$IG(m_n) := H(Y|s_{1:n}, m_{1:n-1}) - H(Y|S_{n+1}, m_n, s_{1:n}, m_{1:n-1}). \tag{5}$$

This is equivalent to the mutual information of Y and (S_{n+1}, m_n) for an arbitrary m_n. As the current entropy $H(Y|s_{1:n}, m_{1:n-1})$ is independent of the next action m_n the most promising action m^* can be calculated by minimizing the expected entropy with respect to S_{n+1},

$$m^* = \arg\min_{m_n}(\underset{S_{n+1}}{E}[H(Y|s_{1:n}, S_{n+1}, m_{1:n})]). \tag{6}$$

Because the sensory input s_{n+1} is not known prior to executing m_n, the expected value over all possible sensory inputs s_{n+1} is taken into account. The selected action $m^* \in M$ is integrated into the next sensorimotor feature. The inverse mapping of Q_M can then be used to obtain a top-down interaction command $u \in U$, which is then executed by the agent.



(a) Recognition performance (b) Mean entropies

Fig. 3: Results of the robotic arm evaluation (7 object classes, 10 objects per class, 30 discrete viewpoints). BN 1 and 2 -IG executed random movements while BN2 +IG executed information-gain-guided movements.

Preliminary results are shown in Fig. 3. In the future, we plan to conduct a more extensive evaluation of our approach (using different sensory features) by comparing it to established object recognition approaches on a larger data set. Furthermore we want to extend our approach by a saliency feature detector.

Acknowledgements This work was supported by DFG, SFB/TR8 Spatial Cognition, project A5-[ActionSpace], and DLR projects “EnEx” and “KaNaRiA”.

References

1. Hurley, S.L.: Consciousness in action. Harvard University Press (2002)
2. Noë, A.: Action in Perception. MIT Press (2004)
3. Bajcsy, R.: Active perception. Proceedings of the IEEE **76**(8) (1988) 966–1005
4. Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. International Journal of Computer Vision **1**(4) (1988) 333–356
5. Helbig, H.B., Graf, M., Kiefer, M.: The role of action representations in visual object recognition. Experimental Brain Research **174**(2) (2006) 221–228
6. Gibson, J.: The ecological approach to visual perception. Boston: Houghton Mifflin (1992)
7. Grèzes, J., Decety, J.: Does visual perception of object afford action? Evidence from a neuroimaging study. Neuropsychologia **40**(2) (2002) 212–222
8. Neisser, U.: Cognition and reality: Principles and implications of cognitive psychology. WH Freeman/Times Books/Henry Holt & Co (1976)
9. O’Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and brain sciences **24**(5) (2001) 939–972
10. Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., Zetzsche, C.: Scene analysis with saccadic eye movements: top-down and bottom-up modeling. Journal of Electronic Imaging **10**(1) (2001) 152–160
11. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Progress in brain research **155** (2006) 23–36

Adaptive Information Selection in Images: Efficient Naive Bayes Nearest Neighbor Classification

Thomas Reineking, Tobias Kluth and David Nakath

Cognitive Neuroinformatics, University of Bremen,
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
{trking,tkluth,dnakath}@cs.uni-bremen.de

Abstract. We propose different methods for adaptively selecting information in images during object recognition. In contrast to standard feature selection, we consider this problem in a Bayesian framework where features are sequentially selected based on the current belief distribution over object classes. We define three different selection criteria and provide efficient Monte Carlo algorithms for the selection. In particular, we extend the successful Naive Bayes Nearest Neighbor (NBNN) classification approach, which is very costly to compute in its original form. We show that the proposed information selection methods result in a significant speed-up because only a small number of features needs to be extracted for accurate classification. In addition to adaptive methods based on the current belief distribution, we also consider image-based selection methods and we evaluate the performance of the different methods on a standard object recognition data set.

Keywords: object recognition, classification, information selection, Bayesian inference, information gain

1 Introduction

Selecting relevant information from a high-dimensional input is a fundamental problem pertaining many different areas ranging from computer vision to robotics. An effective selection strategy uses only a small subset of the available information without negatively impacting the task performance. An example of a successful selection strategy is the processing of visual information in humans where eye movements are performed in order to extract the relevant information from a scene in a very efficient manner [11]. A key feature of this selection is its adaptivity because the selection is strongly influenced by the current belief about the scene [17].

In this paper, we follow the idea of an adaptive belief-based information selection and we investigate it in the context of object recognition. While object recognition is usually viewed as a static pattern recognition problem, we model the recognition as an information gathering process unfolding in time, which is more akin to visual processing in humans. In this case, recognition becomes a problem of Bayesian information fusion where the selection of relevant information is done adaptively with regard to the current belief distribution (in contrast to classical feature selection methods, e.g. [4,7]). We propose different criteria for optimal information selection and provide efficient algorithms for their application. In addition to belief-based selection methods, we also consider an image-based method that uses a saliency operator to identify relevant locations in an image.

We combine the information selection methods with the successful NBNN object recognition approach presented in [1]. We use NBNN because it is a probabilistic approach where local image features are sequentially processed in order to update a belief distribution over possible object classes.[1] For each extracted feature, multiple expensive nearest neighbor searches have to be performed, which is why selecting a small subset of relevant features greatly reduces the computational costs of NBNN classification (for making the nearest neighbor search itself more efficient, see [9]). Note that while we focus on object recognition in this paper, the proposed belief-based information selection methods are very versatile and could therefore also be applied in other contexts.

The paper is structured as follows. In the next section, the basics of the NBNN approach are introduced. In Sect. 3, the information selection methods are described in detail. In Sect. 4, the different selection methods are combined with the NBNN approach and compared empirically on a standard object recognition data set. The paper concludes with a short discussion of the proposed methods and possible extensions.

2 Naive Bayes Nearest Neighbor

For NBNN, a set of local image descriptors is extracted from the query image (e.g. SIFT descriptors [8]) which is then used to compute the posterior probability distribution over object classes. Let $\mathcal{C}$ denote the set of object classes, and let $d_{1:N}$ denote all descriptors extracted from the query image[2] where N is the total number of descriptors found in the image. By applying Bayes' rule and by making a naive Bayes assumption regarding the conditional independence of descriptors, the posterior is given by

$$P(c|d_{1:N}) \propto P(c) \prod_{i=1}^{N} p(d_i|c) \text{ with } c \in \mathcal{C}. \tag{1}$$

The likelihood $p(d_i|c)$ for the i-th descriptor is approximated using kernel density estimation (KDE). This avoids the severe errors caused by quantizing descriptors like in bag-of-words models [2]. To reduce computational complexity and in contrast to typical KDE, only the nearest neighbor (NN) of d_i in the training set is considered because the density contributions of descriptors that

[1] Other state-of-the-art classification approaches like deep networks [6] are not suited here because they do not allow for an incremental processing of features.

[2] We use the shorthand notation $d_{1:N} = d_1, \ldots, d_N$.

are farther away tend to be negligible. Using a Gaussian kernel, the likelihood is approximated by

$$p(d_i|c) = \frac{1}{|\mathcal{D}_c|} \sum_{d^{(j)} \in \mathcal{D}_c} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{||d_i - d^{(j)}||^2}{2\sigma^2}) \tag{2}$$

$$\approx \frac{1}{\sqrt{2\pi}\sigma|\mathcal{D}_c|} \exp(-\frac{||d_i - NN_c(d_i)||^2}{2\sigma^2}) \tag{3}$$

with

$$NN_c(d_i) = \underset{d^{(j)} \in \mathcal{D}_c}{\arg\min} ||d_i - d^{(j)}|| \tag{4}$$

where σ denotes the (class-independent) KDE bandwidth, $\mathcal{D}_c$ denotes the set of descriptors in the training set belonging to class c, and $NN_c(d_i)$ denotes the NN of d_i in $\mathcal{D}_c$. The posterior is thus given by

$$P(c|d_{1:N}) \propto P(c) \prod_{i=1}^{N} p(d_i|c) \propto P(c) \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{N} ||d_i - NN_c(d_i)||^2\right). \tag{5}$$

Note that we ignore the descriptor count $|\mathcal{D}_c|$ for the posterior because its influence is very limited and it simplifies the derivations below. Assuming a uniform class prior, the most probable class c^* can be found using the simple decision rule

$$c^* = \underset{c \in \mathcal{C}}{\arg\max} \log P(c|d_{1:N}) = \underset{c \in \mathcal{C}}{\arg\min} \sum_{i=1}^{N} ||d_i - NN_c(d_i)||^2. \tag{6}$$

Though the decision rule in Eq. (6) is independent of σ (it is therefore ignored in the original NBNN approach), the bandwidth turns out to be relevant for the selection of optimal descriptors in the next section. We determine the optimal bandwidth σ^* by maximizing the log-likelihood of all training set descriptors $\mathcal{D} = \cup_{c \in \mathcal{C}} \mathcal{D}_c$ according to

$$\sigma^* = \underset{\sigma}{\arg\max} \log p(\mathcal{D}|\sigma) = \sqrt{\frac{\sum_{c \in \mathcal{C}} \sum_{d^{(i)} \in \mathcal{D}_c} ||d^{(i)} - NN_c(d^{(i)})||^2}{|\mathcal{D}|}}. \tag{7}$$

3 Information Selection

For selecting the most relevant descriptors, we distinguish between belief-based selection methods and image-based ones. For belief-based selection, the probabilistic model introduced in the previous section is used to predict the effect of extracting a descriptor at a particular location in the image on the current belief distribution. In contrast, for image-based selection, the image information itself is used to determine which regions in the image are most relevant without considering the training data.

We model the information selection problem as one of finding the most promising *absolute* location in an image where the object is assumed to be depicted at the center of the image. This simplification allows us to ignore the problem of object detection, which would be necessary in case of more complex scenes with variable object locations. Let l_t denote the location of a descriptor d_{l_t} in an image at the t-th extraction step after already having extracted the first $t-1$ descriptors $d_{l_1:l_{t-1}}$. To select the next optimal location, we compute a score $S(l_t)$ for each location and choose the maximum

$$l_t^* = \operatorname*{arg\,max}_{l_t \in \mathcal{L}_t} S(l_t). \tag{8}$$

To limit the number of locations, we put a grid over each image where a location represents a grid cell. Because of the naive Bayes assumption, the likelihoods of the descriptors within a cell can simply be combined by multiplying them, i.e., each likelihood $p(d_{l_t}|c)$ represents a product of the likelihoods of individual descriptors located within the same grid cell.

In the remainder of this section, we first present two belief-based information selection methods and then an image-based one.

3.1 Maximum Expected Probability

For classification it is useful to select the descriptor that maximizes the expected posterior probability (MEP) of the true class. Because the value of the next descriptor is unknown prior to extracting it, it has to be modeled as a random variable D_{l_t}. The same applies to the value of the true object class of the query image, which is modeled as a random variable $C_{\text{true}} \in \mathcal{C}$. The score S_{MEP} is the conditional expectation of the true class posterior probability

$$\begin{aligned} S_{\text{MEP}}(l_t) &= E[P(C_{\text{true}}|d_{l_1:l_{t-1}}, D_{l_t})|d_{l_1:l_{t-1}}] && (9)\\ &= \int \sum_{c_{\text{true}} \in \mathcal{C}} p(c_{\text{true}}, d_{l_t}|d_{l_1:l_{t-1}})\, P(c_{\text{true}}|d_{l_1:l_t})\, dd_{l_t} && (10)\\ &= \int \sum_{c_{\text{true}} \in \mathcal{C}} p(c_{\text{true}}, d_{l_t})\, \frac{P(c_{\text{true}}|d_{l_1:l_{t-1}})}{P(c_{\text{true}})}\, P(c_{\text{true}}|d_{l_1:l_t})\, dd_{l_t} && (11)\\ &\approx \frac{1}{M} \sum_{i=1}^{M} \frac{P(c^{(i)}|d_{l_1:l_{t-1}})}{P(c^{(i)})}\, P(c^{(i)}|d_{l_1:l_{t-1}}, d_{l_t}^{(i)}) && (12) \end{aligned}$$

with respect to C_{true} and D_{l_t} given the previous descriptors $d_{l_1:l_{t-1}}$. Because the training samples are assumed to represent i.i.d. samples from the joint distribution $p(c_{\text{true}}, d_{l_t})$, the score can be approximated by a Monte Carlo estimate computed over the training set in Eq. (12) where $c^{(i)}$ denotes the class of the i-th image in the training set, $d_{l_t}^{(i)}$ denotes the descriptor in the i-th training image at location l_t, and M denotes the total number of images in the training set. All the posterior probabilities can be obtained using Eq. (5).

Computing the Monte Carlo estimate can be time-consuming because all descriptors in the training set have to be considered. However, the NN distances required for the likelihoods can be computed in advance so that the overall score computation is still significantly faster than having to process all descriptors from the query image. In addition, it would be possible to only use a subset of the training samples where each sample would be drawn with a probability given by the current belief distribution.

For the special case where no descriptors have been extracted ($t = 1$) or where one chooses to ignore previously extracted descriptors, we can compute a score that ignores the current belief distribution and only maximizes the normalized expected likelihood (MEL). Plugging in $P(c^{(i)})$ for the current belief distribution in Eq. (12) results in

$$\begin{aligned} S_{\mathrm{MEL}}(l_t) &= E[P(C_{\mathrm{true}}|D_{l_t})] && (13)\\ &\approx \frac{1}{M}\sum_{i=1}^{M} \frac{P(c^{(i)})}{P(c^{(i)})}\, P(c^{(i)}|d_{l_t}^{(i)}) && (14)\\ &= \frac{1}{M}\sum_{i=1}^{M} \frac{P(c^{(i)})\, p(d_{l_t}^{(i)}|c^{(i)})}{\sum_{c\in\mathcal{C}} P(c)\, p(d_{l_t}^{(i)}|c)}. && (15) \end{aligned}$$

Because this score is independent of previous descriptors, it can be computed offline and is thus extremely fast.

3.2 Maximum Expected Information Gain

A popular method for feature selection is the maximum expected information gain (MIG) [18]. Here we consider a "dynamic" information gain version that takes previous descriptors into account during the recognition process [12,15]. It is given by the expected uncertainty/entropy reduction resulting from observing a new descriptor d_{l_t}. The information gain score S_{MIG} is the conditional expectation of this reduction with respect to D_{l_t} given the previous descriptors $d_{l_1:l_{t-1}}$:

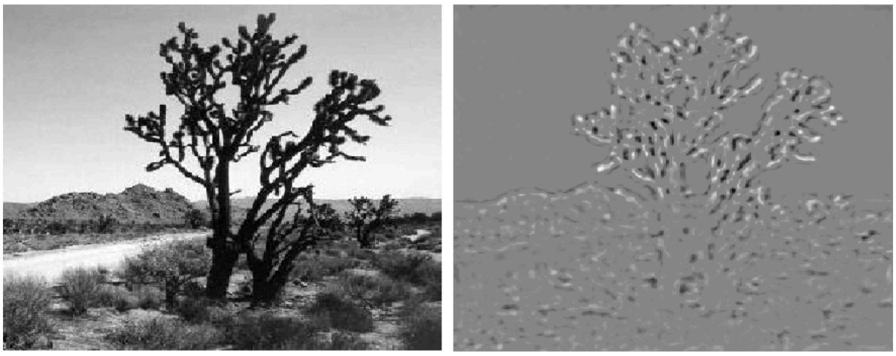
$$\begin{aligned} S_{\mathrm{MIG}}(l_t) &= H(C|d_{l_1:l_{t-1}}) - E[H(C|d_{l_1:l_{t-1}}, D_{l_t})|d_{l_1:l_{t-1}}] && (16)\\ &= H(C|d_{l_1:l_{t-1}}) - \int \sum_{c_{\mathrm{true}}\in\mathcal{C}} p(c_{\mathrm{true}}, d_{l_t}|d_{l_1:l_{t-1}}) H(C|d_{l_1:l_t})\, dd_{l_t} && (17)\\ &\approx H(C|d_{l_1:l_{t-1}}) - \frac{1}{M}\sum_{i=1}^{M} \frac{P(c^{(i)}|d_{l_1:l_{t-1}})}{P(c^{(i)})}\, H(C|d_{l_1:l_{t-1}}, d_{l_t}^{(i)}) && (18) \end{aligned}$$

with entropy

$$H(X) = -\sum_{x\in\mathcal{X}} P(x)\log P(x). \qquad (19)$$

Like for S_{MEP}, the expected value is approximated by a Monte Carlo estimate using samples from the training set in Eq. (18). Note that the information gain is independent of the true class, meaning that a high MIG score only requires the resulting posterior distribution to be “non-uniform”, thus completely ignoring how probable the true class is.

3.3 Intrinsically Two-Dimensional Signals



(a) Original

(b) $I2D$-saliency

Fig. 1: Extracted $I2D$-saliency (b) of the image shown in (a). The extracted $I2D$-score is the clipped eigenvalue computed with the following parameters: $n = 6$, $\sigma_r = 0.2$. Positive elliptically curved regions are light and negative elliptically curved regions are dark.

The following image-based selection method uses a saliency operator which detects intrinsically two-dimensional ($I2D$) signals [19]. The intrinsic dimensionality of a signal $u(x, y)$ is defined as $I0D$ for all signals that are constant and as $I1D$ for all signals that can be written as a function of one variable in an appropriately rotated coordinate system (e.g. an image of an oriented straight edge). In contrast, $I2D$-signals make full use of the two degrees of freedom (e.g. an image of a corner or crossing lines). The $I2D$-saliency also appears to play an important role in the control of saccadic eye movements [5,16] which motivates its use as a score function within the context of this work. In order to identify the interesting $I2D$-points, we make use of the generalized curvature operator introduced in [19]: The generalized curvature operator $T_n : C^2(\Omega) \to C(\Omega)$ with compact $\Omega \subset \mathbb{R}^2$ is defined for $n \in \mathbb{N}$ by

$$T_n(u)(x) = \frac{1}{4}\left((\Delta u)^2 - \epsilon_n(u)^2\right) = \frac{1}{4}\underbrace{(\Delta u + |\epsilon_n(u)|)}_{=\lambda_1(u)}\underbrace{(\Delta u - |\epsilon_n(u)|)}_{=\lambda_2(u)} \tag{20}$$

with eccentricity $\epsilon_n(u)^2 = (c_n * u)^2 + (s_n * u)^2$. The convolution kernels c_n and s_n are defined by their Fourier transform in polar coordinates ($x_1 = r\cos(\phi)$,

$x_2 = r\sin(\phi))$ by

$$\begin{aligned} \mathcal{F}(c_n)(r,\phi) &= (i)^n f(r)\cos(n\phi) \\ \text{and } \mathcal{F}(s_n)(r,\phi) &= (i)^n f(r)\sin(n\phi). \end{aligned}$$

f is a continuous function of the radius r given by $f(r) = 2\pi r^2 e^{\frac{1}{2}\frac{r^2}{\sigma_r^2}}$. λ_1 and λ_2 are the eigenvalues of the Hessian matrix of u in the case of $n = 2$ where the generalized curvature becomes the Gaussian curvature. The Gaussian curvature allows a distinction between elliptic, hyperbolic, and parabolic regions on the curved surface $\{(x, y, u(x, y))^T | (x, y)^T \in \mathbb{R}^2\}$. Using the eigenvalues, the clipped eigenvalue is defined by

$$CE(u) = |\min(0, \lambda_1(u))| - |\max(0, \lambda_2(u))|. \tag{21}$$

In contrast to directly using generalized curvature as a score function, the advantage of the clipped eigenvalue is that it can distinguish between positive elliptic and negative elliptic points, i.e., both eigenvalues are positive or negative. Furthermore, the clipped eigenvalue does not respond to hyperbolic regions. The latter is useful because hyperbolic regions are often found right next to elliptic ones, in which case the hyperbolic regions would only provide redundant information. The score function is then defined with respect to the luminance function u of the grid cell $\Omega(l_t)$ at location l_t by

$$S_{I2D}(l_t) = \frac{1}{|\Omega(l_t)|}\int_{\Omega(l_t)} |CE(u)(x)|\; dx. \tag{22}$$

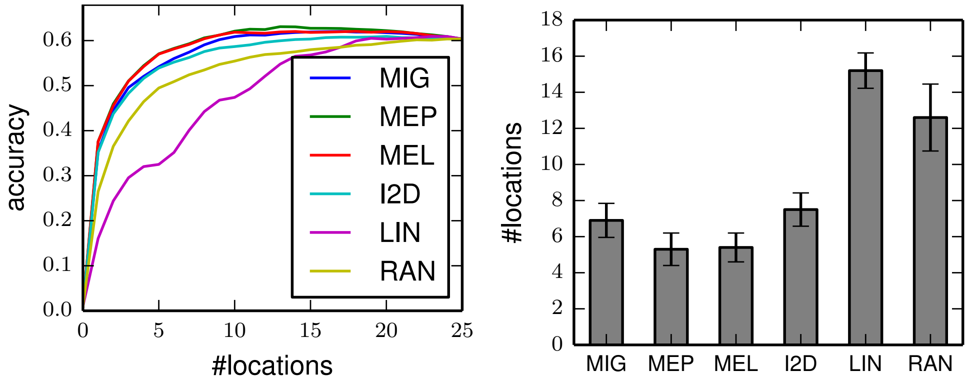
In contrast to belief-based score functions, the $I2D$-saliency is a purely image-based method. Consequently, it does not require any training data. The $I2D$-score function of an example image is illustrated in Fig. 1.

4 Evaluation

We evaluate the proposed information selection methods on the Caltech 101 data set [3]. We use 15 randomly selected images from each of the 101 object classes for training and 10 for testing. All images are scaled such that they have a maximum width or height of 300 pixels. Afterwards, densely-sampled SIFT descriptors are extracted (several thousands for each image depending on the size) and the NN distances are computed.[3]

Fig. 2 shows the mean accuracy over time for the different selection methods using a 5 × 5 grid and 10-fold cross validation. The MEP and MEL methods result in the quickest increase in accuracy and only require extracting descriptors from less than 6 grid cells on average for reliable classification (even though the MEL method ignores the current belief distribution). The MIG and I2D methods perform only slightly worse and all of the considered methods significantly

[3] We use the code provided at `https://github.com/sanchom/sjm` for SIFT descriptor extraction and the FLANN library [10] for fast NN matches.



(a) accuracy

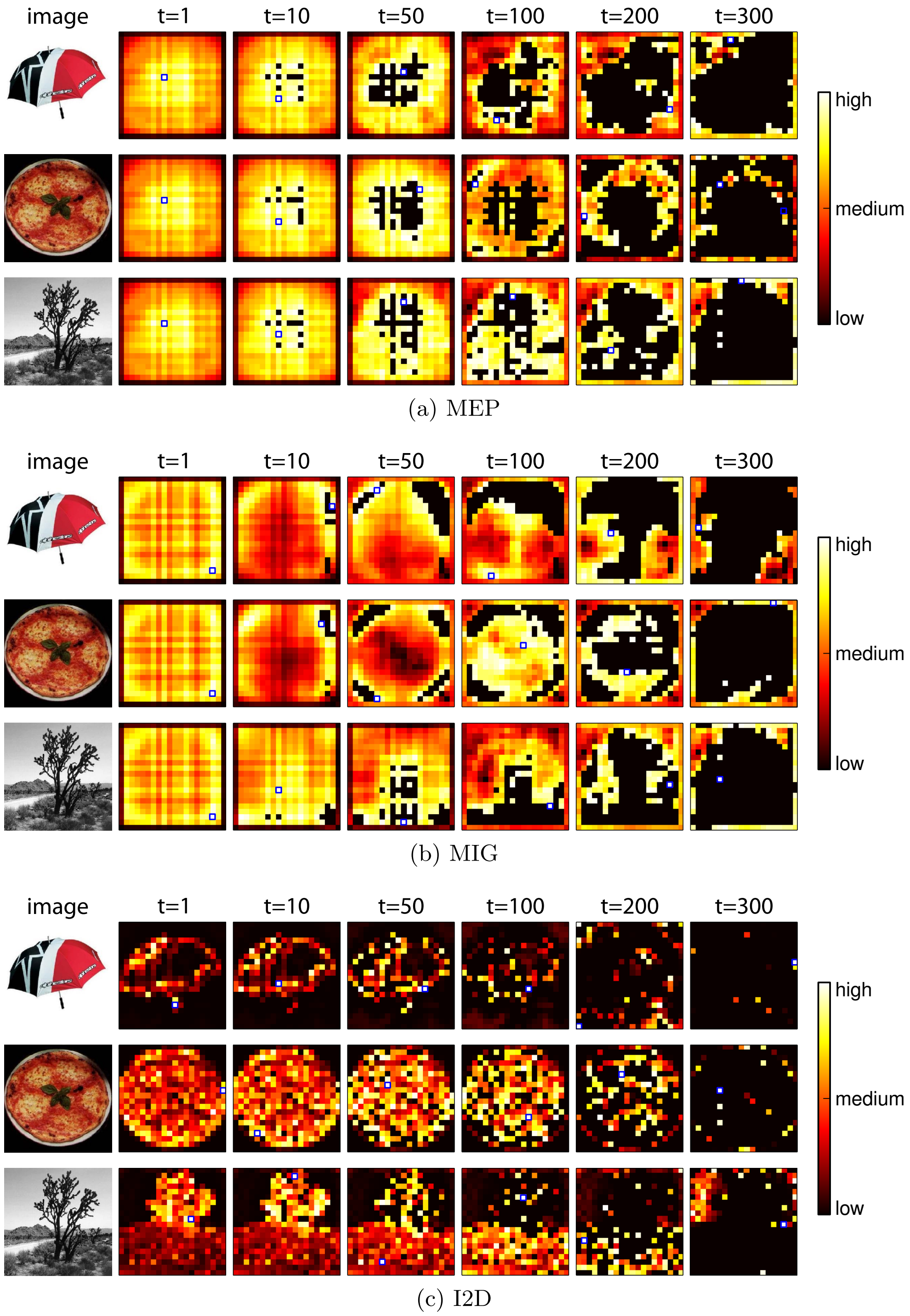
(b) location count

Fig. 2: (a) Mean accuracy on the entire Caltech data set plotted for different time steps/location counts using different selection methods. (b) Mean number of time steps/location counts required for reaching at least 90% of the final accuracy where all descriptors have been extracted. The indicated standard deviation is computed with respect to the different folds.

outperform the baseline methods where descriptors are either selected randomly ("RAN") or line by line starting at the top of the image ("LIN"). The final accuracy after having extracted *all* descriptors is identical for each method because the extraction order is irrelevant for the classification model. Interestingly, the accuracy is highest after having extracted about half of all descriptors (except for the baseline methods), showing that the remaining descriptors tend to only decrease the recognition performance.

To illustrate the process of sequentially selecting descriptors, Fig. 3 shows score distributions over time using a 20×20 grid for three example images. For the belief-based MEP and MIG selection methods shown in (a) and (b), the score distributions change significantly over time and adapt themselves to the query image based on the current belief distribution. The I2D score distribution remains constant over time aside from setting the score of previously selected locations to 0 (the apparent change in other locations is due to scaling in the visualization). At $t = 1$, both the MEP and the MIG scores are independent of the query image and only the I2D method uses the image information. Over time, the MEP and MIG scores adapt themselves to the current belief distribution over object classes, whereas the I2D score remains unchanged. The visible "grid pattern" (especially for $t \leq 10$) is an artifact resulting from some grid cells containing more descriptors than others (this could be avoided if all cells contained roughly the same number of descriptors).

Perhaps surprisingly, the MEP score is highest at the center while the MIG score is initially highest in the periphery. One possible explanation for this effect is that the MEP method can be interpreted as a "confirmation strategy" whereas the MIG method can be interpreted as a "discriminative strategy". For MEP,



(a) MEP

(b) MIG

(c) I2D

Fig. 3: Examples of score distributions over time using a 20 × 20 grid for different selection methods and query images. The small blue square indicates the cell with the highest score from which the next descriptor(s) are extracted. Cells that have already been selected have a score value of 0 (black).

extracting descriptors from the center of an object usually increases the probability of the true class without necessarily resulting in a unique classification (i.e. the overall belief distribution can still be very uniform). In contrast, the MIG method is agnostic with respect to the true class and only seeks to reduce uncertainty (e.g. by ruling out large numbers of classes). This could be accomplished by analyzing the "context" of objects, which is why the MIG method might first focus on the background.

5 Conclusion

We have proposed different methods for adaptive information selection from images where the current belief distribution directly determines which image locations should be considered next. In addition, we have also considered an image-based selection method that does not require any training data. Using these methods, we have extended the NBNN approach and we have shown that the selection methods make it possible to only consider a small subset of the available information while maintaining the original recognition performance. In particular for NBNN, where computing the NN distances for each descriptor is very time-consuming, the result is a significantly reduced computation time.

One of the problems not addressed in this paper is the fact that features in close proximity to each other are highly correlated. While the naive Bayes assumption can be justified for inference by the greatly reduced computational complexity, for the information selection it would be possible to use a more sophisticated model where correlations are explicitly considered. As a result, there would be a penalty for extracting features located very closely to each other, thus avoiding processing of redundant information.

In this paper, we have considered belief-based selection strategies (MEP, MIG) and image-based strategies (I2D) separately. A more promising approach could be a combination of both strategies [16] because the belief-based strategy completely ignores what is readily available in the image while a purely image-based strategy has difficulties selecting the relevant information because it ignores the training data. Due to the complementary nature of these strategies, a hybrid strategy could further improve the selection process.

We believe that the proposed selection methods can also be useful for problems beyond recognizing single objects. Especially for complex scenes containing many objects, an adaptive information selection strategy could predict the likely locations of objects and thereby facilitate understanding of the entire scene.

Finally, the general nature of the proposed information selection approaches allows for the application to systems which must perform actions to obtain new information from their environments (e.g. an autonomous spacecraft [14] or a melting probe [13]). These actions can cause high costs in terms of, for example, energy consumption or execution time. In these situations, it is thus highly desirable to avoid non-informative actions by using adaptive selection strategies.

Acknowledgments. This work was supported by the German Federal Ministry for Economic Affairs and Energy (DLR project "KaNaRiA", funding no. 50 NA 1318, and DLR project "CAUSE", funding no. 50 NA 1505).

References

1. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
2. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV. vol. 1, pp. 1–2 (2004)
3. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 106(1), 59–70 (2007), special issue on Generative Model Based Vision
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (Mar 2003)
5. Krieger, G., Rentschler, I., Hauske, G., Schill, K., Zetzsche, C.: Object and scene analysis by saccadic eye-movements: An investigation with higher-order statistics. Spatial vision 13(2-3), 201–214 (2000)
6. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
7. Liu, H., Sun, J., Liu, L., Zhang, H.: Feature selection with dynamic mutual information. Pattern Recognition 42(7), 1330–1339 (2009)
8. Lowe, D.G.: Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. vol. 2, pp. 1150–1157. Ieee (1999)
9. McCann, S., Lowe, D.G.: Local naive Bayes nearest neighbor for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3650–3656. IEEE (2012)
10. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (2014)
11. Najemnik, J., Geisler, W.S.: Optimal eye movement strategies in visual search. Nature 434(7031), 387–391 (2005)
12. Nakath, D., Kluth, T., Reineking, T., Zetzsche, C., Schill, K.: Active sensorimotor object recognition in three-dimensional space. In: Spatial Cognition IX, pp. 312–324. Springer (2014)
13. Niedermeier, H., Clemens, J., Kowalski, J., Macht, S., Heinen, D., Hoffmann, R., Linder, P.: Navigation system for a research ice probe for antarctic glaciers. In: IEEE/ION PLANS 2014. pp. 959–975. IEEE (2014)
14. Pavone, M., Acikmese, B., Nesnas, I.A., Starek, J.: Spacecraft autonomy challenges for next generation space missions (2013), http://goo.gl/nU8xG0, online, to appear in Lecture Notes in Control and Information Systems
15. Reineking, T., Schill, K.: Evidential object recognition based on information gain maximization. In: Cuzzolin, F. (ed.) Belief Functions: Theory and Applications, Lecture Notes in Computer Science, vol. 8764, pp. 227–236. Springer International Publishing (Sep 2014)

16. Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., Zetzsche, C.: Scene analysis with saccadic eye movements: Top-down and bottom-up modeling. Journal of Electronic Imaging 10(1), 152–160 (2001)
17. Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M.: Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychological review 113(4), 766 (2006)
18. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML. vol. 97, pp. 412–420 (1997)
19. Zetzsche, C., Barth, E.: Image surface predicates and the neural encoding of two-dimensional signal variations. In: SC-DL tentative. pp. 160–177. International Society for Optics and Photonics (1990)