Marine Vessel Inspection as a Novel Field for Service Robotics: A Contribution to Systems, Control Methods and Semantic Perception Algorithms.

Dipl. Inf. Markus Eich

Kumulative Dissertation

zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften

– Dr.-Ing. –

Vorgelegt im Fachbereich 3 (Mathematik und Informatik),

Universität Bremen

12. Dezember 2013

Datum des Promotionskolloquiums: 09.12.2013

1. Gutachter
Prof. Dr. Frank Kirchner (Universität Bremen)

2. Gutachter
Prof. Dr. Bernd Krieg-Brückner (Universität Bremen)

To Michaela, Tyra and my parents.

Zusammenfassung

Diese kumulative Doktorarbeit führt in ein neues Anwendungsfeld für die Service-Robotik ein: die Inspektion von Schiffen durch mobile Inspektionsroboter. In dieser Arbeit werden drei wissenschaftliche Beiträge vorgestellt und experimentell auf dem Gebiet der Schiffsinspektion verifiziert, sind jedoch nicht allein auf diese Anwendung beschränkt. Das Inspektionsszenario dient dabei lediglich als roter Faden, um die kumulativen wissenschaftlichen Ergebnisse dieser Doktorarbeit zu kombinieren.

Der erste Beitrag ist eine adaptiver, propriozeptiver Kontrollansatz für hybride Bein-Rad Roboter, wie den in dieser Arbeit beschriebenen Roboter ASGUARD. Aufgrund des in dieser Arbeit vorgestelten Steuerungskonzeptes ist der Roboter in der Lage, unwegsames Gelände und Treppen zu überwinden. Das vorgeschlagene System ist eine geeignete Plattform, um sich innerhalb von Frachträumen von Schüttguttransportern zu bewegen und so visuelle Daten aus dem Inneren der Frachträume zu liefern. Darüber hinaus hat das vorgeschlagene System auch die Fähigkeit Treppen zu steigen, so dass das System sich zusätzlich zwischen verschiedenen Decks bewegen kann. Basierend auf den propriozeptiven Daten der Gelenkmotoren und auf Basis der Nick- und Rollwinkeln des Körpers, welche während der Fortbewegung gemessen werden, adaptiert der Roboter dynamisch sein Laufmuster.

Der zweite wichtige Beitrag dieser Doktorarbeit ist ein eigenständiges System zur Inspektion von Schiffen, bestehend aus einem magnetischen Kletterroboter für die Inspektion von Schottwänden, einer Partikelfilter-basierte Methode zur Lokalisierung und einem *Spatial Content Management System* (SCMS) für die räumliche Darstellung und die Organisation von Inspektionsdaten. Das in dieser Arbeit beschriebene Gesamtsystem wurde sowohl durch mehrere Laborexperimente, als auch durch verschiedene Feldversuche auf zwei verschiedenen Schiffen evaluiert, jeweils in enger Zusammenarbeit mit Schiffsinspekteuren.

Der dritte wissenschaftliche Beitrag der Arbeit ist ein neuartiger Ansatz zur Klassifizierung von räumlichen Srukturen durch semantische Perzeptionsverfahren. Basierend auf der räumlichen Beziehung zwischen räumlichen Entitäten und räumlichen Merkmalen, können durch diese Verfahren strukturierte Umgebungen semantisch annotierten werden. Diese Methoden wurden im Bereich der Innenraum-Perzeption (für den Bereich Logistik und Haushalt), zur Klassifikation von Bodenprobenbehältern und zur Klassifizierung von Strukturteilen innerhalb von Schiffen verifiziert. Die vorgeschlagenen Verfahren ermöglichen die semantische Klassifikation der Strukturteile von Laderäumen, wodurch sowohl die Position von Inspektionsrobotern als auch Positionen von Schadstellen beschrieben werden können. Die in dieser Arbeit verwendeten Algorithmen basieren auf unorganisierten 3D-Punktwolken, welche von einem LIDAR innerhalb von Schiffsladeräumen erzeugt werden. In dieser Doktorarbeit werden dabei zwei verschiedene Methoden für die semantische Perzeption vorgestellt. Ein Ansatz basiert auf probabilistischen *Constraint Networks*, der zweite Ansatz basiert auf unscharfer Beschreibungslogik und *räumlichen Schlussfolgern* unter Verwendung einer räumlichen Ontologie über die Umgebung.

Zu jedem der drei genannten Kernbeiträge der Doktorarbeit befinden sich die entsprechenden Publikationen im Anhang.

Abstract

This cumulative thesis introduces a novel field for service robotics: the inspection of marine vessels using mobile inspection robots. In this thesis, three scientific contributions are provided and experimentally verified in the field of marine inspection, but are not limited to this type of application. The inspection scenario is merely a golden thread to combine the cumulative scientific results presented in this thesis.

The first contribution is an adaptive, proprioceptive control approach for hybrid leg-wheel robots, such as the robot ASGUARD described in this thesis. The robot is able to deal with rough terrain and stairs, due to the control concept introduced in this thesis. The proposed system is a suitable platform to move inside the cargo holds of bulk carriers and to deliver visual data from inside the hold. Additionally, the proposed system also has stair climbing abilities, allowing the system to move between different decks. The robot adapts its gait pattern dynamically based on proprioceptive data received from the joint motors and based on the pitch and tilt angle of the robot's body during locomotion.

The second major contribution of the thesis is an independent ship inspection system, consisting of a magnetic wall climbing robot for bulkhead inspection, a particle filter based localization method, and a spatial content management system (SCMS) for spatial inspection data representation and organization. The system described in this work was evaluated in several laboratory experiments and field trials on two different marine vessels in close collaboration with ship surveyors.

The third scientific contribution of the thesis is a novel approach to structural classification using semantic perception approaches. By these methods, a structured environment can be semantically annotated, based on the spatial relationships between spatial entities and spatial features. This method was verified in the domain of indoor perception (logistics and household environment), for soil sample classification, and for the classification of the structural parts of a marine vessel. The proposed method allows the description of the structural parts of a cargo hold in order to localize the inspection robot or any detected damage. The algorithms proposed in this thesis are based on unorganized 3D point clouds, generated by a LIDAR within a ship's cargo hold. Two different semantic perception methods are proposed in this thesis. One approach is based on probabilistic constraint networks; the second approach is based on Fuzzy Description Logic and spatial reasoning using a spatial ontology about the environment.

For each of the three mentioned key contributions of the thesis, the corresponding publications are provided in the Appendix.

Acknowledgments

First of all, I would like to thank my lovely wife Michaela Eich for her patience and moral support during the writing of this thesis. She was always able to make me go forward, step by step until the goal was finally reached.

Scientific work is usually not possible without the support of others. This is especially true in the research field of robotics. A lot of people have been involved in the projects and contributed especially in the design and physical development of all these fantastic robot platforms used in the experiments published in this thesis. From the DFKI Robotic Innovation Center I would like to thank Johannes Lemburg, Felix Grimminger, and Kristin Fondahl for building all the nice robots which were used, often damaged, and repaired again on time during the experiments. Without the physical systems for testing the algorithms and control methods, it would have not been possible to write this thesis. Additionally, I would like to thank my colleagues Ronny Hartanto, Sankaranarayanan Natarajan, and Steffen Planthaber for the fruitful and often entertaining conversations during lunch time.

Last but not least, I would like to thank my first supervisor Prof. Frank Kirchner for providing the opportunity of doing research in the challenging field of robotics at the DFKI Robotics Innovation Center, and Prof. Bernd Krieg-Brückner for being the second reviewer of this thesis.

Contents

List of Figures

List of Tables

1 General Introduction

1.1 The Application Domain of Marine Vessel Inspection

Seagoing vessels, such as bulk carriers, dry cargo ships, and tankers (cf. Figure 1.1), have to undergo regular inspections at survey intervals. This is performed by ship surveyors, using visual close-up surveys and thickness measurements, based on non-destructive testing methods (NDT) [51]. Vessel inspection is performed on a regular basis, depending on the requirements of the ship classification society, also called *classes*. For a close-up survey, the ship surveyor usually has to get within arms' reach of the inspection point. Structural damage, pitting, and corrosion are visually estimated based on the experience of the surveyor. Based on the visual inspection results, additional thickness measurements are performed. Documentation of the whole ship survey process is traditionally done manually by using cameras to take photos, chalk and pen for defect marking, and a clipboard for taking notes. Providing access to the

Figure 1.1: Two examples of marine vessels (Images Courtesy of Lloyd's Register and the MINOAS project)

inspection points of the marine vessel, e.g., inside a cargo hold, is the most time consuming, and therefore most cost intensive part of the inspection. Prior to the survey, staging (scaffolding) is often used to allow the surveyor to inspect bulkheads, beams, stiffeners and brackets. These parts are usually several meters above the bottom of the hold. Besides scaffolding, also "cherry-picking" methods are used, where the surveyor is inside a basket, transported by a tower crane or a crane with a hydraulic arm. This method is potentially risky for the surveyor because the basket can collide with the structural parts of the ship, causing injuries to the surveyor. A single section of a cargo hold of an average vessel can easily reach 15 meters in height and more than 20 meters across. Figure 1.2 shows different cargo holds of a bulk carrier and a container ship. The holds shown in the image were used during the field trials of the robot inspection system, described in Chapter 3.

(a) (b) (c)

Figure 1.2: (a), (b) Aspects of a cargo hold of a bulk carrier. This bulk carrier was also used for the first set of field trials with the robot inspection system, described in Chapter 3. (c) A cargo hold of a container vessel.

1.2 Service Robots for Marine Vessel Inspection

Using a service robot for marine vessel inspection imposes several requirements on the robotic system in terms of locomotion capabilities, localization approaches, and finally the representation of the acquired data to the surveyors. The inspection of marine vessels is a challenging area because every part of the vessel has different requirements for inspection robots. Inside a large cargo hold, as depicted in Figure 1.2, a flying inspection system is a suitable platform due to the large space within such a hold. UAVs which were recently developed for ship inspection purposes are described in [37] and [39]. Additionally, a variety of unmanned aerial vehicles (UAV) exist which do not depend on GPS for navigation. This is a mandatory requirement, because a GPS signal is usually unavailable or distorted due to the steel frame of a ship. Autonomous flying systems which are able to navigate inside a confined area are described in [14] and [18]. Flying systems are suitable for a quick overview of the structures inside a hull or a ship superstructure. In some cases, the surveyor needs a visual close-up survey in order to classify a defect. Besides the visual data provided by an inspection service robot, thickness measurements are usually taken. For such an application, a flying system is not suitable because it cannot approach a bulkhead or structure because of possible collisions. Inside ballast water tanks and inside narrow structures, a low-cost inspection system was introduced using a railway system [9]. Other inspection systems, especially for inspection inside tanks and ballast water tanks, were developed within the ROTIS II project [42]. Recently developed systems, such as the AUV Dagon [21], are usable for the underwater inspection of ship hulls under water and for the propeller section.

For dry inspection of cargo holds, climbing robots are the most suitable systems, because they can transport sensors, such as cameras or sonar thickness measurements, close to the inspected area. The inspection area is usually several meters up on a bulkhead. Traditionally, the inspection is done by surveyors using staging and “cherry-picking” methods, which are costly and time consuming, and sometimes dangerous due to the low safety standards in some countries. Because marine vessels are made of steel, magnetic climbing locomotion is a suitable way to move around the cargo hold of a vessel. Most climbing robots are heavy-weight systems and need an additional safety measure, such as a safety rope. In some scenarios, a safety rope is not usable due to the dangers of entanglement. A more detailed overview of the related

research concerning walking and climbing robots is presented in Chapter 3.

In this thesis, two systems are introduced to the area of inspection systems and control methods. One system is a hybrid leg-wheel robot which is able to climb stairs and is also able to move on rough terrain, such as sand, rock, and uneven terrain. Such a system is suitable to inspect the cargo holds of a bulk carrier that can be loaded with a variety of goods, such as gravel, coal, or rocks. The system and its proprioceptive control approach are described in Chapter 2. The focus in the corresponding chapter is on the novel control approach, using proprioceptive data to adapt the walking pattern dynamically without a pre-defined gait pattern.

The second inspection system introduced in this thesis is a lightweight magnetic crawler system which is able to climb bulkheads and structures within a marine vessel. The climbing system and its evaluation in several field trials on two different vessels are described in Chapter 3. Additionally, the localization approach and an interface for the spatial representation of the acquired inspection data are presented in the same chapter.

1.3 Robot Localization and Inspection Data Representation

In order to be usable for the domain of ship inspection, a service robot has to be localized. This is not only a mandatory step towards system autonomy, but the acquired data, such as inspection images and thickness measurements, have to be localized within a model of a ship. This is especially mandatory for the ship surveyor and for the report of the damage, blistering, and corrosion. The report has to contain the information as to where the data was reported in order to inform the office about further steps. Traditionally, defect marking is performed using chalk or special pens to mark the defects directly on the structure. There are a large number of localization methods for mobile robots, ranging from probabilistic, metric pose estimation using an internal map [52, 17], to topological methods based on the semantics of the environment [34]. Other methods use tracking methods, such as [40]. In Chapter 3, a more thorough literature review is given on this subject.

The localization method for a robot is highly dependent on the application. For the domain of ship inspection, a complete mapping of the environment is time consuming and inefficient. A more usable approach, as was identified during the MINOAS project, was to use a 3D CAD model of the vessel and use methods to localize the robot within this CAD model. The localization method chosen for the lightweight crawler, presented in Chapter 3, is based on external tracking using a custom developed tracking unit. The localization is based on visual tracking using a particle-filter based approach. The tracking methodology and the localization system is not limited to the lightweight magnetic crawler, but can be applied to other robot systems as well, if a line of sight can be provided between the optical tracking system and the inspection robot.

Beside the localization of the inspection data, which is based on the local reference frame of the robot, it is a mandatory requirement to represent the acquired survey data to the surveyor. Robots represent the position mainly in world coordinates which are not usable for a practical application. Within this thesis, two methods are proposed for representation of the inspection data to a surveyor. In Section 3.4, a spatial content management system is provided, which

was developed during the MINOAS project. The representation of the inspection data is based on a 3D CAD model of the ship and topological nodes and graphical representations of ship sections are used to access the data content. This approach is based on a metrical representation of the ship environment and needs a 3D CAD model of the ship in order to represent it.

Another approach towards classifying the location of defects and the robot within a cargo hold is the use of semantic perception methods. This allows a robot-based inspection system a more natural interaction with the surveyor. For the domain of marine vessel inspection, a closed-world assumption can be made, because all the structural parts which are named in the domain are standardized. This is a mandatory fact, because different surveyors around the world have to communicate with each other and it has to be ensured that the same semantic labels are used for the structural parts. Within this thesis, two novel approaches are introduced using semantic perception methodologies in order to classify the structural parts of a ship. One method is based on spatial reasoning based on spatial features and spatial relationships using probability propagation within a constraint network. The second method uses a spatial ontology using Fuzzy Description Logic in order to classify the structural parts. The last two methods were developed and published in another application domain using different environments. Within the corresponding Chapter 4, it is further shown by experimental results that the same approach can be applied to the domain of vessel inspection by adapting the domain knowledge about the environment.

1.4 Structure of the Thesis

The structure of the thesis is shown in Figure 1.3. The introductory part, described in this chapter, gives an overview of the domain of marine vessel inspection and gives a motivation for the contributions presented in this thesis.

Not all the publications mentioned in this thesis were published in the domain of vessel inspection. The scientific contributions that are covered by the publications in the Appendix are given in Chapters 2, 3, and 4. The control approach and the methods for semantic perception described in this thesis were developed and implemented during different projects at the DFKI. The domain of ship inspection serves as a golden thread throughout this thesis in order to combine the scientific results achieved in different fields of service robotics.

Chapter 2 introduces a proprioceptive control approach for a leg-wheel robot. The system is highly mobile and can cover rough terrain or stairs, and is therefore an ideal sensor carrier for vessel inspection. The ASGUARD system and the corresponding control approach described in this chapter were primarily developed for the purpose of search and rescue applications, financed by the SentryBot project (Project Number, DFKI-EP 9518).

Chapter 3 describes a complete marine inspection system which was developed during the MINOAS project (EU-SCP8-GA-2009-233715). The system consists of several modules, including a magnetic climbing robot for bulkhead inspection, a localization approach using a custom designed localization unit, and a methodology for presenting the inspection data to a ship surveyor. The complete system has been evaluated in several field trials, including a usability evaluation of the complete system.

The semantic perception approaches presented in Chapter 4 were partly developed within the SemProM project (BMBF-01IA08002) and partly within the IMPERA project (DLR-50RA1111). The domain of the corresponding publications about perception was primarily perception of indoor structures and the detection of soil sample containers for a sample return mission, respectively. Within Chapter 4, the application of the published semantic perception methods are verified in the domain of marine vessel inspection. The methods and algorithms correspond to the attached publications, the ontology of the domain and the reasoning rules were adapted in this thesis to give evidence that the algorithms are transferable to other application domains. Chapter 5 gives a conclusion about the work presented and gives some outlook on future research stemming from the results presented in this thesis. The Appendix contains the accumulated publications.

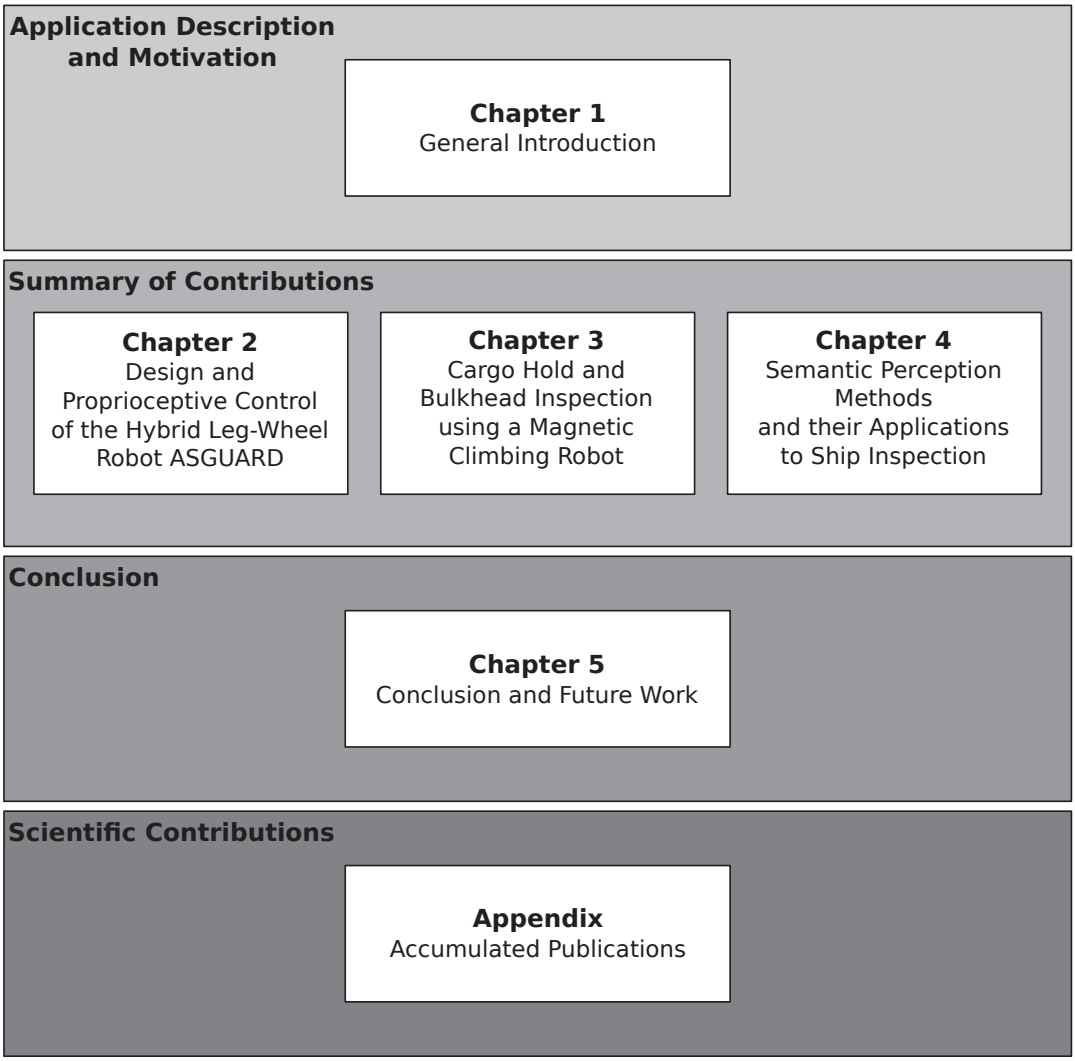


Figure 1.3: Structure of the thesis

1.5 Thesis Contribution

Within this thesis, three main contributions are provided. The contributions are not bound to the application domain described in this thesis. The topic of the thesis serves as a golden thread throughout the thesis, in order to combine several publications which have been published by the author in several domains, projects, and applications. The service robotics domain of marine vessel inspection is a good example of how different methodologies can be combined to provide robot based vessel inspection as a new application field. All contributions provided in the thesis are applicable to this domain. The three main scientific contributions include:

- **Development of novel control approaches:** An adaptive compliance control approach for hybrid leg-wheeled mobile systems. The method is able to optimize the gait of a hybrid leg-wheel system by using proprioceptive data. With the described approach, the system is able to sense single steps during stair climbing. The approach is verified using the mobile robot ASGUARD.

- **Development of new tools for vessel inspection:** A stand-alone vessel inspection system, consisting of a magnetic wall climbing robot, a particle-filter based localization technique, and a spatial content management system for representation of the inspection data. The system has been evaluated in several field trials on marine vessels. Beside the empirical data given in the corresponding publications, also qualitative evaluation results have been provided by the ship surveyor during the trials.

- **Development of novel perception methodologies:** Two methods for the semantic classification of structural parts are described in this thesis. Each method uses a different approach for scene reconstruction based on a 3D point cloud and spatial feature extraction. One method uses a probabilistic constraint network. By using probability propagation within this network, the structural parts of a ship hold can be classified. The second approach towards semantic structure classification uses a Fuzzy Description Logic in order to describe spatial features and spatial relationships. The relevant publications included in this thesis have been published in the domain of indoor perception and object classification using a mobile robot. Within the corresponding chapter of this thesis, additional evidence is given that the same approaches can be applied to the domain of marine vessel inspection. This is achieved by adapting the knowledge base to the domain of marine vessels.

1.6 Bibliographical Remarks

Many publications in this thesis have been done including also contributions from other authors. This section briefly describes what the main contribution of the author of this thesis was in each publication. The paper [EGBS08] was written by the author except for Section 3 of the paper. The paper [EGK08] was written by the author. The second author contributed the CAD drawings and designed the robot ASGUARD. This paper won the Best Paper Award at the conference. [EGK09] was mainly written by the author of this thesis. The second author was credited for the hardware designed of ASGUARD and the images provided. This journal publication was an extended version of a conference contribution [EGK08] which was

recommended by the reviewers for publication in the *Industrial Robot Journal*.

The work published in [EBPGF+14] was a joint paper between several partners of the MINOAS project. The author of the present thesis wrote the paper except for the sections about the UAV, the corrosion detection, the thickness measurements, and the heavy weight crawler. All contributions concerning the lightweight crawler, the particle-filter based localization approach, the spatial content management system, and the usability evaluation, were made by the author of this thesis. The work [EV11] was written by the author, except for the introductory section. The same applies to the paper published in [VEGF10]. In [FEWK12], the author provided the main contribution to the laboratory experiments section and to the overall system concept.

The paper [EDK10] was written entirely by the first author. The second author, who was a student at that time, contributed to the experimental data provided in the results section. The paper [Eic13] was published as a single author publication.

1.7 Additional Peer-reviewed Publications by the Author

Additional papers related to the thesis topic have been published by the author and are not included as part of this thesis. Research dealing with the proprioceptive control of hybrid leg-wheel robot has been published as [EGK08, EGK09]. Other research dealing with robot-based ship inspection was published as [BBC+11, OBPG+10, OBG+10, CRB+10]. Additional work on semantic perception done by the author can be found in [EK10, LDE+11, EDK10, EHK+14].

Other peer-reviewed publications which were written by the author, but are not directly related to the topic of this thesis, were published as [EKF04, KES07, RAME11, SEN+13, VESK09, KWE13].

2 A Proprioceptive Control Method for Hybrid Leg-Wheel Robots

2.1 Introduction

The locomotion capabilities required of an inspection robot for marine vessel applications depend on the type of inspection and the inspection scenario. For the survey of a marine vessel, robots serve basically as sensor carriers which provide visual and other data of parts of the vessel that are difficult for the surveyor to access. One example of a ship survey is the visual inspection of cargo holds. Usually, cargo holds have to be cleaned in order to provide access for the surveyor. Bulk carriers are used for transporting sand, rocks, coal, grain or scrap metal. Even if the holds are unloaded, the bottom plating of the cargo hold is usually covered with remains of the load, making it difficult for an inspection robot to move inside the hold. Using a robot with suitable locomotion capabilities can provide access to data inside a partly loaded cargo hold. The locomotion requirements inside cargo holds are comparable with those of rough outdoor environments. A mobile robot, equipped with suitable sensors, such as a camera for visual inspection, has to be able to traverse sandy, rocky or uneven terrain inside the hold. Some examples of cargo holds are given in Figure 2.1. For this scenario, the locomotion requirements for an inspection robot are very demanding. Additionally, seagoing

Figure 2.1: Non-empty cargo holds of two bulk carriers, loaded with sand. (Images Courtesy of RINA group and the MINOAS project)

vessels usually have separate decks and levels which are connected by stairs, therefore stair climbing capabilities of an inspection robot increase the usability of such a system. Ideally, an inspection robot should have the ability to climb stairs and be able to move over demanding terrain, such as gravel or sand.

Naturally, the most common locomotion principle for mobile robots is based on wheels. Wheeled robots have the ability to move at higher speed but are usually not able to climb

stairs or to deal with harsh environments or uneven, obstructed terrain. When using wheels, the locomotion capability of a robot is strongly dependent on the type of suspension. Some suspensions allow a high grade of mobility in rough terrain such as a rocker–bogie suspension [28] or a highly adaptive suspension system as described in [47].

Tracked robots provide high locomotion capabilities and are able to deal with loose soil, such as pebbles, sand or rocks. The locomotion capability of tracked robots is highly dependent on the clearance of the robot chassis. Tracked systems are basically not able to overcome stairs or obstacles higher than one-half of the front wheel diameter. Some exceptions exist, which make use of an additional flipper attached to the front wheels of the tracks which are able to lift the robot over an obstacle higher than the wheel diameter [7].

In contrast to wheeled systems, multi-legged systems, such as the robots Scorpion [48], Space-climber [2], and Lauron [16], use biologically inspired locomotion principles and are able to climb slopes and to move in very rough terrain. The drawback of multi-legged systems is the locomotion speed, which is usually much slower than that of wheeled systems. For some years, hybrid locomotion principles have been investigated for robots, combining the advantages of legs and wheels in the same system. The most frequent approach concerning hybrid locomotion is the use of spokes or legs which are attached to a rotating axis [41]. This system is a purely mechanical system which uses an adaptive spring for the spoked leg-wheels. Other systems, such as the hexapod robot RHex [33], make use of rotating C-shaped legs around six different axes. The same principle is applied to the stair-climbing robot Loper [20]. These systems are suitable for moving in rough outdoors environments and are able to cope with stairs. From the control perspective, the robot RHex makes use of three discrete, pre-defined walking patterns which are triggered according to the locomotion requirements, e.g., flat terrain, stairs, or rough terrain.

Within this chapter, a novel control approach for hybrid leg-wheel robots is introduced using a Central Pattern Generators (CPG) and a control approach based on proprioceptive data which is provided directly by the driving motors and by an internal IMU. The control architecture is verified using the hybrid leg-wheel robot ASGUARD (cf. Figure 2.2). The key idea is to use the

Figure 2.2: The leg-wheel robot ASGUARD is a suitable sensor carrier for rough terrain. The robot was primarily developed for harsh environments in search and rescue scenarios. The high mobility of the robot makes it also a suitable platform for sensors for cargo hold inspections of marine vessels.

CPGs to generate walking patterns for the system in terms of frequency, direction, and phase shift for each wheel axis. The patterns are generated in time–angle space, defining an angle

FPGA	FPGA Xilinx Spartan III XC3S1000
H-Bridges	6
Encoder Inputs	6
Interfaces	RS 232
AD channels	8
Voltage	24V DC

Table 2.1: MotCon6 Specification

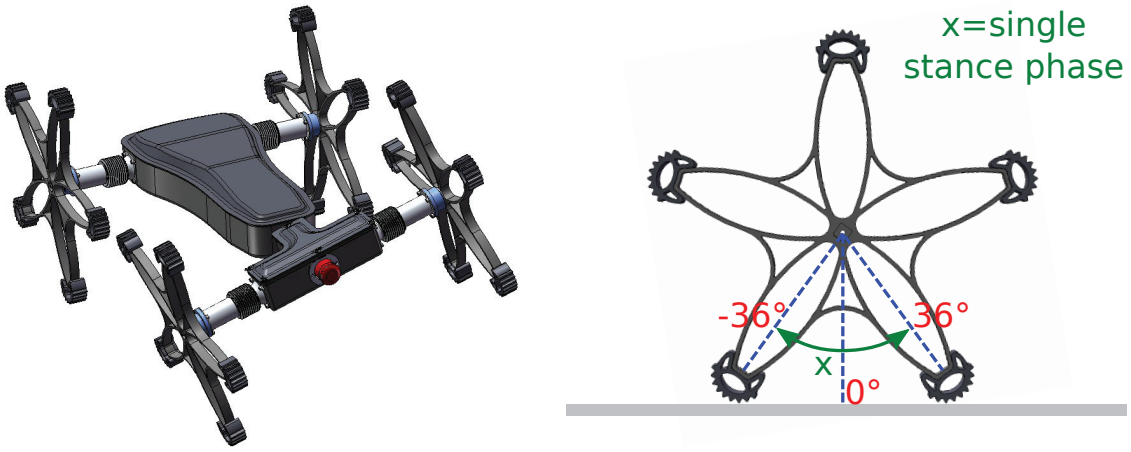
for a joint at a certain time. The resulting trajectory for each leg is adapted using a cascaded compliance controller which controls the stiffness of each leg depending on the proprioceptive feedback generated by the environment. The results of this chapter were collected in the papers [EGBS08, EGK08, EGK09].

At the time of writing of this thesis, an improved version of the robot ASGUARD exists, which is able to navigate autonomously using a 3D representation of the environment [44]. This newer version of the robot uses a different control approach than the one described in this chapter, but uses proprioceptive data in order to improve the localization accuracy in unstructured outdoor environments.

2.2 ASGUARD: A Quadruped System Design

The hybrid leg-wheel robot ASGUARD is a quadruped system with high locomotion capabilities. A schematic view of the robot ASGUARD (Version 2) is shown in Figure 2.3a. The locomotion is based on four motors with encoders, driving the four leg-wheels independently. Each leg-wheel consists of five separate legs which are oriented around a hip shaft with an angular distance of 72°. A schematic view of the leg-wheel is shown in Figure 2.3b. The legs (five legs on each of the four wheels) of the robot rotate around one axis, resembling a quadruped motion while moving. Each wheel is driven by an 83 W motor with a planetary gear providing a rate of 46:1. For the proprioceptive feedback, which is needed by the control approach, quadrature encoders and the applied torques are used. The torques are estimated by measuring the motor current. Additionally, an IMU provides the tilt and roll angle of the robot's body. The proprioceptive data is measured by a custom designed FPGA control board (MotCon6). The low-level architecture of the MotCon6 Board is described in [EGBS08]. A short overview of the board is given in Figure 2.5.

The first two versions of the ASGUARD system are described in detail in [EGBS08, EGK08]. The contribution within this chapter is the adaptive low-level locomotion approach based on a central pattern generator (CPG) and an adaptive controller suitable for autonomous forward climbing of rough terrain and an autonomous behavior for stair climbing. The robot has an embedded Intel-based embedded PC which is connected to the FPGA based low-level control board via an RS232 serial connection. The commands provided to the low-level controller are velocity commands and the desired rotational speed. Motion commands can be provided by an OCU (Operator control unit) or by a path planning module, using the same interface. The MotCon6 board and the block diagram are shown in Figure 2.5. The technical data of the board is given in Table 2.1.



(a) (b)

Figure 2.3: (a) Schematic view of the robot ASGUARD. The locomotion is based on four star-shaped wheels with five legs on each wheel. Each leg-wheel is actuated independently. A passive joint in the middle of the body adds another degree of freedom to the system. This ensures that all leg-wheels have ground contact, similar to a spine. (b) shows a schematic of a single leg-wheel.

An overview of the hardware modules and their interaction is displayed in Figure 2.4. The main control loop of the locomotion system is distributed between the embedded CPU board running Linux and the FPGA-based motor controller. An inertial measurement unit provides the pitch and roll angle of the robot's body and directly influences the adaptability of the controller. The core hardware module is the MotCon6 board, which houses the trajectory generator for each leg-wheel, the stiffness controller, and the position controller. Each trajectory generated is based on a saw tooth pattern (cf. Section 2.3) and is parameterized by the controller. The parameters include the frequency of each pattern, the direction, and a phase shift which is used to change the gait of the robot. Additionally, the position error is fed into the CPG generator, triggering a synchronization between the generated patterns with the current leg position if the error is above a threshold. This is an important feature for the adaptability of the motion pattern, described in the next section.

In addition to the motion pattern generator, a position controller is implemented on the MotCon6 Board. The position controller controls the position of each individual leg in the interval $[-36°, 36°]$. (Only this interval is relevant for the leg control.) For each axis, the stiffness is controlled by the impedance parameter of the position controller. The position controller is implemented as a PID-controller where the P-value controls the stiffness of each leg and is updated within a fixed frequency on the embedded CPU. The constant update for the CPG parameters (phase, frequency and direction), the impedance for each leg as well as the CPG synchronization signal is provided by the embedded CPU and is send to the MotCon6 Board at a constant frequency. These values are calculated by the embedded CPU based on the four individual position errors of each leg, the current measured for each motor, and the average current for all motors. Additionally, the pitch of the body is measured in order to detect whether the system is climbing a slope or stairs, which directly affects the control approach. In the following Section 2.3, the low-level controller, based on the values introduced above, will be described in detail.

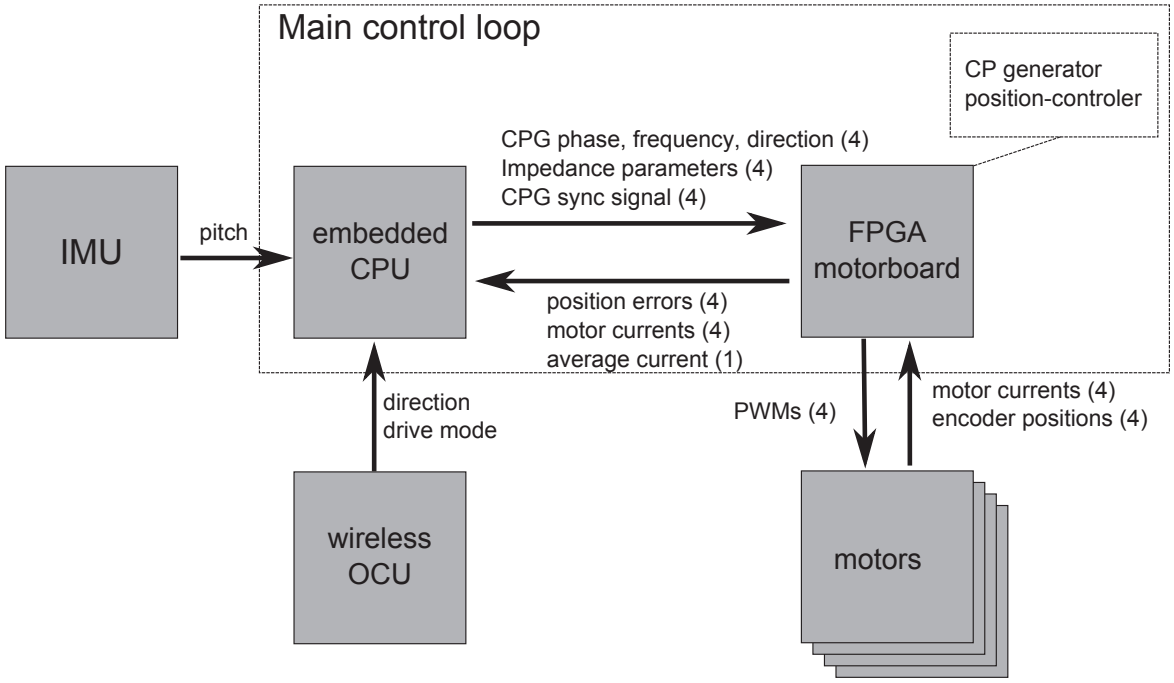


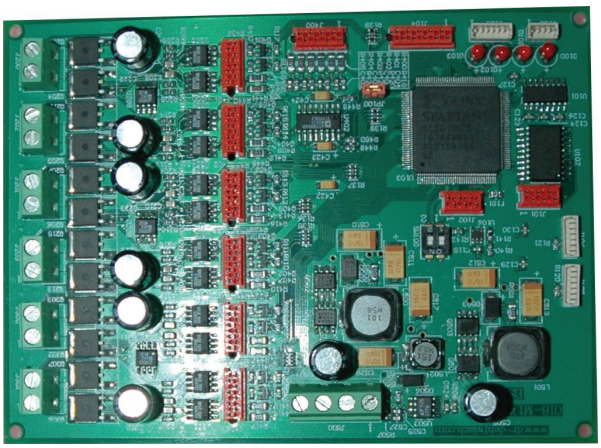
Figure 2.4: The interaction of the different modules used for the leg-wheel control. The internal control loop is distributed between the embedded CPU and the MotCon6 board which is responsible for the generation of the motion trajectories (CPGs). The embedded CPU interacts with the FPGA board using serial communication.

2.3 A Proprioceptive Control Approach for Hybrid Leg-Wheel Robots

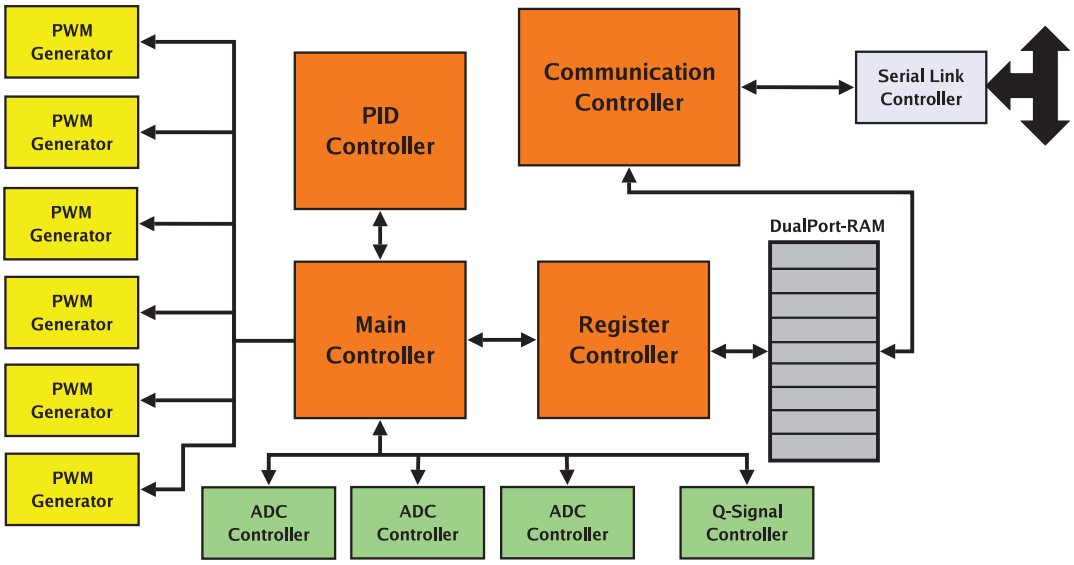
The basis for the low-level control is the MotCon6 board, described in the previous section. This module is used to generate four independent trajectories, one for each leg-wheel. The internal control loop on the FPGA board follows the generated pattern, using a runtime-configurable position controller.

The locomotion principles of walking systems in general consist of a swing and a stance phase [48, 2]. In contrast, the robot ASGUARD has only a continuous stance phase. For the locomotion control principle it is important to divide the motion of the leg-wheel into phases, between $[-36°, 36°]$ ($[-\frac{1}{5}\pi, \frac{1}{5}\pi]$). This is the phase between the points when the robot has contact with the ground with two legs. For each single step, a trajectory pattern between the phase $-\frac{1}{5}\pi$ and $\frac{1}{5}\pi$ is generated in time–angle space. The sample pattern resembles a saw-tooth pattern (cf. Figure 2.6). The trajectory is generated on the FPGA board in real time, independently for each wheel. The use of Central Pattern Generators for robots has been proven to work robustly in a variety of walking machines, such as the robots Scorpion [48], Aramies[49], Spaceclimber[2] and also for two-legged robots [3].

The key advantage of using a motion pattern is that the position of each leg can be controlled in time–angle space. By the phase-shift of the patterns for each leg, different walking patterns are



(a) MotCon6 FPGA Board



(b) MotCon6 FPGA Block Diagram

Figure 2.5: The MotCon6 Board

generated. It is also possible to adapt the gait for different surfaces. For instance, the system moves smoothly if the phase of the pattern is asynchronous. If all legs are synchronized (i.e., every leg is in the same phase), the walking behavior approaches more a dynamic bounding behavior [30]. An energy efficient and stable gait type that moves the robot forward is highly dependent on the surface and the environment on which the robot is moving. For instance, while climbing stairs, the most stable walking pattern is when the front legs are synchronized from left to right. This will minimize the roll angle of the body and will cause the robot to move in a straight line, in line with the direction of the stairs. The same is true for the rear legs. The offset between the rear and the front pair depends on the size of the steps and the distance between the steps. An analysis of different walking patterns based on the synchronization and the offset of the ASGUARD legs is described in [1]. In [33], an approach is described using pre-defined walking patterns for a hexapod robot. The behavior is triggered as soon the stair is detected by the legs. The drawback of a pre-defined pattern is that the system cannot adapt to irregular environments.

The key idea of the approach described in this chapter is that the motion patterns are im-

plicitly adapted by synchronizing the motion pattern with the current leg position if the error between the motion pattern and the leg position is above a threshold. The adaptive compliance controller consists of two submodules. The internal control module (implemented in the MotCon6 FPGA board) generates the motion pattern for each leg. The second task of the MotCon6 board is the compliance control for each leg depending on the average torque produced by each of the four motors. For the robot ASGUARD, the locomotion control is based on the frequency of the saw tooth pattern, on the direction of the pattern, and on the phase shift with respect to the other patterns. The phase shifts between the patterns state how the legs are synchronized.

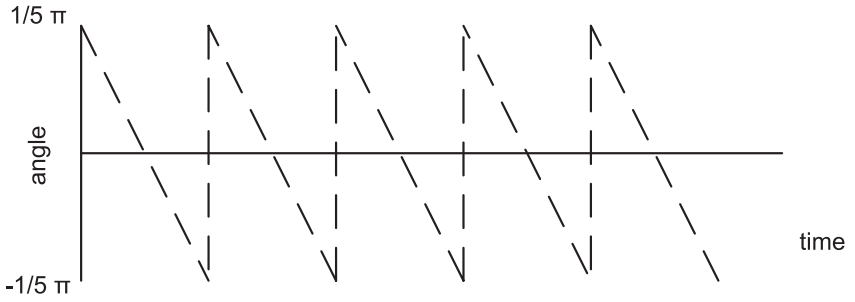


Figure 2.6: A sample pattern in joint-angle space.

The phase shift for each axis is relevant to the type of locomotion applied to the robot. Figure 2.7 shows a block diagram of the low-level controller. The following nomenclature is used in the diagram:

- An edge with a variable denoted by X_1^4 means that the variable has a cardinality of 4 and that the values X_1, X_2, X_3, X_4 are used as input or output.

- The inputs to the system are denoted by the phase offset Φ for each motion pattern generated by the pattern generator, the frequency f indicating the locomotion speed for each of the leg-wheels, and the direction being denoted by Dir. The direction for each leg-wheel can either be forward or backward.

- The variable Xd denotes the desired position of the leg on a wheel for one phase. The desired position of each leg is within the phase $[-36°, 36°]$. The error between the actual position of the leg on a wheel and the desired position is denoted by e. (The controller does not take the complete leg-wheel rotation into account because of the leg redundancy.)

- The four separate torques generated by the stiffness controller are denoted by τ. The different torques applied to the legs move the legs to the actual position, denoted as X.

- The environment creates a force on each leg. The force is measured indirectly by measuring the current of the motor which is denoted by I. A highly accurate determination of the exact forces from the motor current and the mechanical properties of the leg-wheel are not necessary for the proposed approach: it is important only to calculate the dis-

crepancy of the torques (estimated by the current measurement) for each motor. The individual discrepancies of the current for each motor are denoted by Δ.

- The inclination of the robot is denoted by Ψ. The inclination influences the overall compliance of the controller.

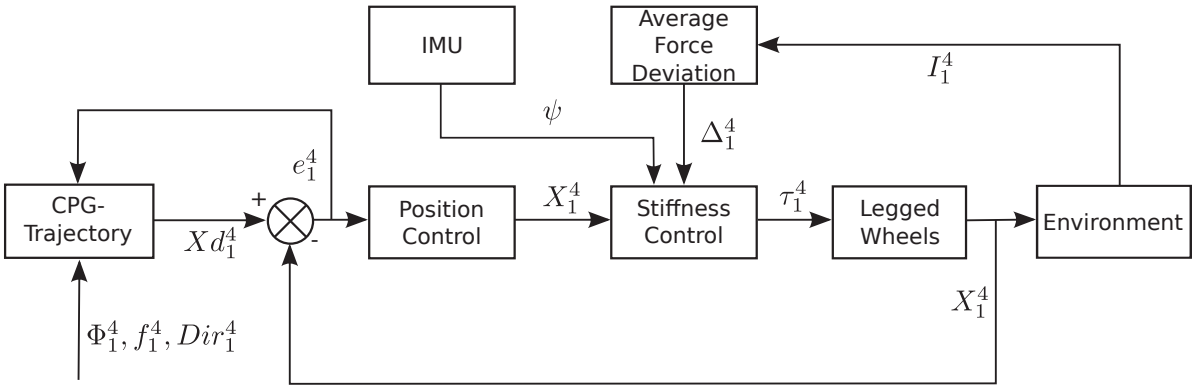


Figure 2.7: The low-level controller diagram of ASGUARD

The high level controller provides the input to the CPG pattern generator. The input includes the frequency f_1^4 for each pattern, the phase offset Φ_1^4, and the direction of the motion pattern. If the offset Φ_1^4 is 0, for the CPG, all legs are synchronized. The high-level controller takes the input from the operator or from a navigation algorithm and it is given as the velocity and rotational speed. The pattern generator provides the target position for each individual leg on the leg-wheel; the position controller controls the position of each leg in the interval of $[-36°, 36°]$.

The adaptability of the motion pattern is implicitly provided by the stiffness controller and the Average Force Deviation, denoted by Δ_1^4. The system balances the load for each leg by this approach. This means that a leg which is taking a higher load than the average is less stiff, i.e., the leg is more compliant than the other legs. This again causes a higher position error for legs with a higher motor torque. The physical effect of this approach is that the robot stops increasing the torque on a specific leg if the leg is completely blocked by the environment, causing an increase in the torque on the remaining legs to free the blocked leg from the blocking position.

The adaptability of the stiffness controller depends on the inclination. The inclination adaptability was introduced in [EGK08]. The reason behind this is that the ASGUARD system has the ability to move quickly on flat terrain, and has a high adaptability of its gait while moving slowly and climbing debris, rocks, and stairs. While moving on flat ground, the rear legs provide a higher load to the system while accelerating. An adaptive approach as described above would work against the ability of a quick acceleration because it would tend to balance the load, resulting in sub-optimal locomotion results. While climbing stairs and walking on inclined terrain, the robot has to be more adaptive and the locomotion speed is not relevant. To decide in which mode the robot is in automatically, the inclination of the robot body is measured by an IMU. The input from the IMU, denoted by Φ_1^4, has a direct effect on the stiffness control. If the inclination of the robot's body is zero, the stiffness controller is non-adaptive, i.e., the position controller is used without the compliance part, providing maximum torque

on each motor, as described in [EGK08]. The inclination of the body affects the compliance of the controller in a linear way, i.e., the more the robot is inclined, the more adaptive is the controller.

In order to adapt the walking pattern to the ground, the error feedback from the position controller is provided to the pattern generator. This approach was first introduced in [EGK09]. If the error of the position on a single leg, denoted by e_1^4, reaches a threshold, the phase of the internal motion pattern on that specific leg is synchronized with the current position of the leg, thus changing the overall walking pattern of the robot. The effect of this approach is similar to a physical torque wrench combined with a spring. Each single leg is compliant with a combination of the position controller and the stiffness controller. If the motor is moving the leg against an obstacle (or moving the robot forward), the torque is increased until the position error is greater than a pre-defined threshold. In this case, the motion pattern is synchronized within the pattern generator (cf. Figure 2.7). The combination of CPG synchronization and the compliance part of the controller act together like the mechanical clutch of a torque wrench where the clutch slips if the applied torque exceeds a threshold. In contrast to its mechanical counterpart, the maximum torque for each leg depends on the system's average load.

By the application of the controller described above, the overall system energy consumption is reduced, as is shown in [EGK09]. In this paper it is also shown that the analysis of the energy distribution for each leg pair (front and rear) and the change of the load can be used to estimate the type of the ground. If the front and rear legs are in the same phase on both sides (left and right) this corresponds to ASGUARD's climbing a flight of stairs or a similar regular structure. In the shift of the systems load (front to rear), the number of steps of a flight of stairs can be sensed by the robot without using exteroceptive information [EGK09].

2.4 Contribution of the Corresponding Publications

In this chapter, control methods for hybrid leg-wheel robots are proposed, enabling such robots to adapt their gait pattern based on proprioceptive feedback. The approach was verified using the hybrid system ASGUARD which was primarily developed for search and rescue applications in rough terrain. The locomotion principle makes the system an ideal sensor carrier for cargo hold inspection of a marine bulk carrier. Two versions of the ASGUARD robot were developed, each robot with a different design and controller. The first version of the robot and the first control principle were first published in [EGBS08].

An improved version of the control approach and of the robot was published in [EGK08]. The contribution in this paper was also the adaptability of the controller depending on the inclination. The paper won the Best Paper Award at the 2008 IEEE International Workshop on Safety, Security, and Rescue Robotics.

In [EGK09], it was shown that the number of steps of a flight of stairs could be estimated just with the observation of the load compared between the front and the rear actuators. The legs synchronize and adapt implicitly the walking gait based on the proprioceptive data measured by the load on each joint and by measuring the orientation of the system with respect to the gravity vector. The corresponding publication is an extended version of [EGK08] which was selected for the *Industrial Robot Journal* Innovation Award and was thereby recommended for

journal publication.

3 Robot-based Cargo Hold and Bulkhead Inspection

3.1 Introduction

In this chapter, a practical application of service robots in the field of ship surveying is described. The work has been carried out during the EU-funded MINOAS project. The goal of the project was to introduce a robot system to support the surveyor during the inspection process. The main purpose of a robot during a ship survey is to acquire data in areas which are hard for a surveyor to access. The main type of data the surveyor needs to estimate a ship's condition is based on visual feedback. The project's goal was to provide robotic solutions for ship inspection with a focus on the cargo holds of marine vessels. A survey of a ship cargo hold takes place every 2–5 years, depending on the requirements of the responsible ship classification societies (classes) and on the age of the vessel (cf. [51]). The procedure on how cargo holds are inspected is described in detail in [EBPGF+14]. Staging within a cargo hold and the deployment of "cherry-picking" cranes in the cargo hold is the most time consuming part. The main costs arise from the fact that a marine vessel incurs significant costs for every day the vessel is not in service. As described in [EBPGF+14], a robot inspection system has to comply with the following requirements:

- The robot has to be able to carry visual sensors to parts of the cargo hold.
- The robot should be able to mark defects on the operator's request.
- The inspection data acquired by the robot has to be localized in order to generate a history of data from the same spot.
- Inspection data has to be stored and represented to the surveyor for damage assessment in an intuitive way.

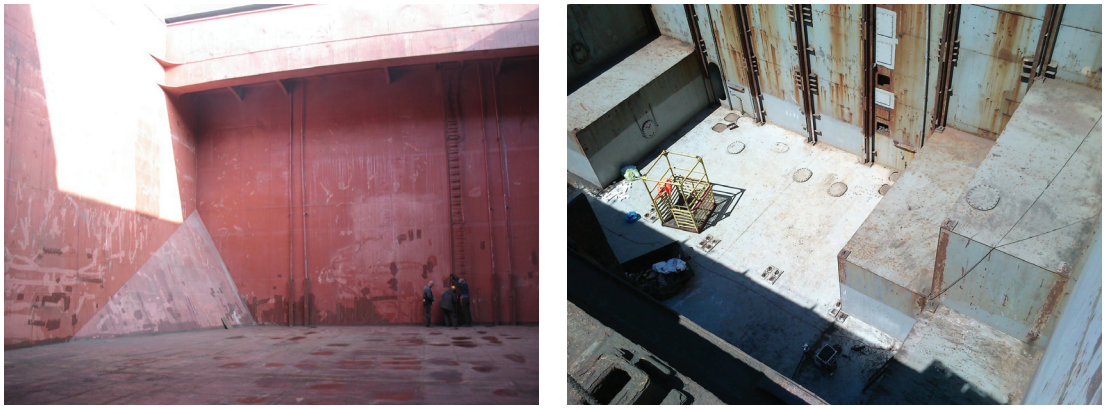
To cover all the above requirements, a novel robot-based inspection system was developed and verified in the laboratory environment as well as in two field trials on marine vessels. The contribution of this chapter of the thesis in the field of marine vessel inspection consists of hardware and algorithms which were co-developed within the project. The core of the system is a magnetic wall climbing robot for bulkhead inspection, which is described in Section 3.2.

The second part of the inspection system is the localization technique, which has a strong focus on the usability of the system in everyday environments. The selection of the localization method was based on the system's specification and on the requirement of easy usability and setup. The particle-filter based localization methods and the corresponding hardware design are introduced in Section 3.3.

The third contribution in this chapter is a method to provide the acquired inspection data to the surveyor and the inspection team in an intuitive way. The inspection data is organized within a Spatial Content Management System (SCMS) which represents all data aligned with a 3D model of the vessel to the surveyor. The SCMS is described in detail in Section 3.4.

3.2 Lightweight Magnetic Crawler for Ship Inspection

In this section the corresponding publications in the Appendix are summarized. For the inspection of bulkheads within a cargo hold, a lightweight climbing robot is proposed. The main use-case for the robot is the inspection of the cargo hold of a bulk carrier, shown in Figure 3.1. The main focus during the development of the inspection system for cargo holds



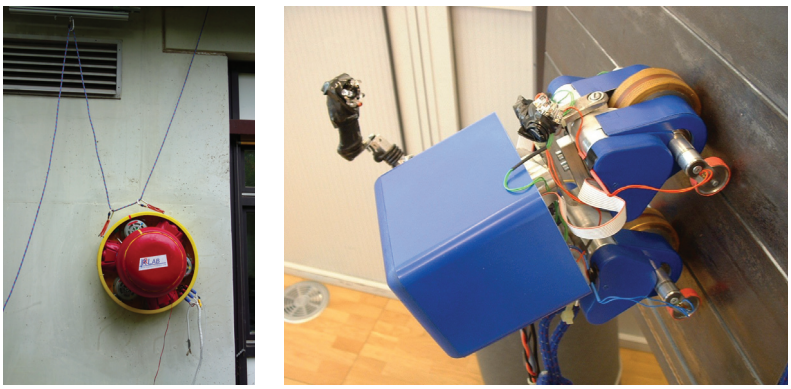
(a) (b)

Figure 3.1: The cargo holds of two different vessels. (a) shows the cargo hold of a bulk carrier. (b) shows the cargo hold of a container ship. These cargo holds were used for the system evaluation.

was to provide visual images of the parts of the hold without using scaffolding or "cherry-picking" cranes. Possible locomotion principles for climbing robots in this domain can be divided into the use of suction pads and vacuum, magnetic adhesion forces, sticking systems, or robots using active clinging.

The requirements for the vessel inspection are described in detail in [EBPGF+14]. Based on these requirements, the adequate locomotion principle for the inspection robot is based on magnets. Some magnetic climbing systems for ship inspection exist in research and some are already commercially available. One example of a magnetic inspection robot is the magnetic climbing robot MARC [5] (cf. Figure 3.2a). Another example of a wall climbing robot is given in [25] (cf. Figure 3.2b). The robot MARC has a weight of 50 kg and can only be operated as a tethered system because of safety issues. Both robots use magnetic tracks which provide sufficient adhesion force on metal surfaces. MARC is equipped with an additional manipulator for surface grinding and can obtain thickness measurements. Other robots, such as the wall climbing robot CROMSCI [24], use a vacuum-based locomotion concept. A vacuum motor is integrated into the system, removing the need for carrying a pressure hose. The magnetic climbing robot MAGNEBIKE is able to climb complex metallic structures. Marine vessel surveying using robots is a new research field and not many systems exist for this particular

(a) (b)



(c) (d)

Figure 3.2: Some examples of magnetic climbing robots which are suitable for ship inspection. The robots shown in (a) and (b) use magnetic tracks. The robot CROMSCI (c) uses an embedded vacuum motor to stick to the wall. The MAGNEBIKE (d) has an adaptive body and uses magnet wheels to climb metal surfaces.

application, although it is certainly true that there has been a lot of research in the field of wall climbing robots. To start with, a robotic system which was developed to inspect hot welding seams was introduced by [46]. This small-sized system has a weight of 30 kg and requires a safety tether during its operation. The system uses an infrared sensor to check the temperature of the hot weld seam and works in parallel with a welding robot. Heavyweight hull cleaning robots have already been used for ship surveying and repairs. These robots have a weight over 100 kg and are used to remove large areas of coating on the ship's hull using water jetting techniques or brushes [38]. Some other robots for marine inspection are already available, such as the robot steel-climber from Miko Marine [32] or the Magnet Crawler M250 from Jetstream Europe [23]. Both are heavyweight magnetic crawlers for blasting, ship cleaning, and inspection. Other systems which are usable for hull and underwater inspection are the hybrid system Roving Bat from ECA Robotics [11] and the ROV-based inspection systems by Video Ray [56]. These systems are primarily used for ballast water tank inspection and underwater hull surveys.

A more detailed review of the state of the art of climbing robots is presented in [EV11, FEWK12, EBPGF+14] in the Appendix of this thesis. The requirement for the MINOAS Project was to develop an inspection system which is easy to use, does not require installation setup, and is easy to use by the surveyor. For this task, a lightweight robotic system was

developed which is suitable for carrying a camera. This robot is used for online inspection of the cargo hold and for image acquisition. The design of the magnetic climbing robot is shown in Figure 3.3. The robot is a free-climber, meaning that no tether is used for communication,

Figure 3.3: The magnetic wall climbing robot. The system was used in several field trials in a marine vessel.

power supply, or safety. The safety of the robot is assured by a safety net which is carried by a person; its low weight, below 1 kg, makes it easy to catch if the system drops off the bulkhead. The advantage of using a free climbing system is the increased locomotion capability if the system is not tethered. Not using a tethered system also prevents entanglement of the system with structural parts of the cargo hold. The first approach using a lightweight robot for cargo hold inspection was published in [VEGF10]. At that early stage, the focus was mainly on the design of the magnetic wheels. The focus during the first design steps was on speed and on maneuverability. The locomotion concept of the robot is based on differential steering, using two actuated wheels, augmented with neodymium magnets. Several wheel designs for the magnetic crawler were proposed and tested in [VEGF10]. An improved version of the magnetic climbing robot was published in [EV11] where the first step was taken towards a localization system for the magnet crawler. The final design of the physical platform was published in [EBPGF+14]. In this publication the results of several field trials in the domain of vessel inspection are presented.

3.3 Localization Methods for Inspection Robots within Ship Cargo Holds

In order to compare inspection data, it is mandatory that the surveyor is able to localize the data within the cargo hold. Therefore, the robot's position has to be estimated reliably within a defined reference frame. Robot localization is a wide ranging field in robotics and many approaches exists. The use of a GPS system usually does not provide sufficient accuracy and does not work indoors or inside a cargo hold of a vessel. The localization approaches for mobile robots can be divided into methods using internal map representations and SLAM[1] methods, external beacons for relative navigation, and external localization using external tracking methods. The main differences and the advantages and disadvantages of these localization methods are discussed in the following sections.

[1]Self-localization and Mapping

3.3.1 SLAM-based localization methods

Localization using simultaneous localization and mapping methods is based on an internal map representation that is built by exteroceptive sensors such as LIDAR[2] systems or cameras. The basic principles of SLAM methods are based on continuous sensor registration of the features of the environment within a map reference frame. Usually the features are obstacles within the environment which are detected by a LIDAR. This means that the features are extracted from the LIDAR point cloud. The robot is moving within the reference frame of the map while correcting its own position based on the same LIDAR information. Other methods for SLAM include features extracted from stereo vision [27]. The drawback of SLAM approaches is the iterative error of the position estimate which accumulates over time. The accumulating error causes map inconsistencies especially if a loop in the trajectory is closed. This means that the robot revisits a pre-visited spot on the map. A method using a single-camera SLAM method is proposed in [10] where visual features are extracted from a mono camera. Tracking of the features is used to estimate the ego-motion of the camera without inertial or odometry feedback. Several methods have been developed to deal with loop-closures, also known as the revisiting problem.

These methods include the use of multiple maps which are kept simultaneously using a Rao–Blackwellised Particle Filter [17]. Recent methods make use of a graph based approach, where each scan is stored in the node of a graph representation. The edges of the graph represent the translation and the rotation between the scans. The graph structure is optimized when the loop of the robot path is closed. SLAM based methods have been proven to work in indoor and outdoor environments with ground based systems as well as flying robots, such as UAVs. The drawback of such methods is that usually a complete map of the environment has to be explored before a robot can be used in this environment. With respect to the application to ship inspection, this would include a time consuming mapping of the whole cargo hold. The other drawback is that the robot has to be able to carry additional sensors for pose estimation and a powerful processing unit for map building. For a light weight climbing robot, SLAM based methods are not feasible in the given application, therefore an external localization method was selected to work in the everyday environment of ship inspection.

3.3.2 Localization using External Beacons and Tracking

The most commonly known localization method using external beacons is the global positioning system (GPS). In this case, global-stationary satellites emit a radio signal. A GPS receiver uses the signal run-time, coded into the signal, to triangulate the global position. The inaccuracy of GPS-based positioning makes thie method inefficient for stand-alone positioning in robotics, especially in confined spaces. Some approaches exist which make use of a signal triangulation method similar to GPS. The approach described in [19] makes use of an external beacon to localize an indoor cleaning robot. A set of external beacons project an IR pattern on the ceiling which is used by the mobile robot to estimate the position using a method called vector field SLAM. The signal from the ceiling is received by an IR sensor on the robot and the map is estimated by the robot's processing unit. Other approaches make use of external beacons based on 2D QR Codes to improve the position estimation provided by a particle filter [31]. In this case, the QR code is used in combination with an adaptive Monte Carlo Filter and a

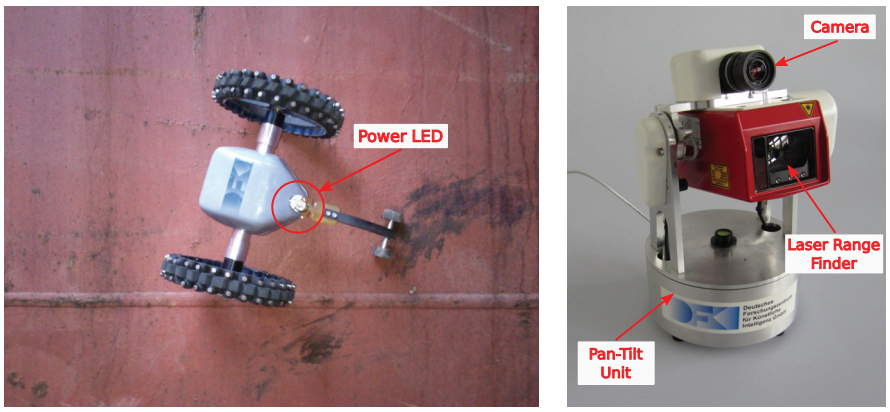
[2]Light detection and ranging

grid map. The external marks support the system in case the position is lost due to significant localization errors.

In contrast to robot localization based on external beacons, external tracking methods are based on external observation of the system using cameras or other sensors. For instance, in [40] a system is proposed which uses an external camera for robot and obstacle tracking. Commercially available tracking systems, such as the Vicon Tracker [55], provide a high resolution position resolution but are sensitive to sunlight and need a high installation effort to cover a whole cargo hold. Some works exist using optical tracking methods for free-moving objects, based on motion estimation [6]. External position tracking for robots is usually applied if the robot does not have the capacity to carry internal sensors, such as a small UAV [35]. External tracking methods can be applied if a line of sight to the system can be guaranteed and the system itself does not have enough processing power or sensor capabilities, which is the case for the lightweight climbing robot described in this chapter.

3.3.3 Tracking-based Localization Method for Vessel Inspection Robots

The approach used for the localization of the inspection robot is based on visual tracking of a lightweight crawler using a traceable target on the back of the robot (cf. Figure 3.4). The advantage of an external tracking system is the easy and fast deployment during the vessel inspection task, without the need of prior map building or additional installation. An external tracking system can be placed without much effort inside the cargo hold. The tracker is able to estimate the position of the climbing robot with respect to the tracker's reference frame. For global positioning within the hold, the tracker has to be calibrated. This is done by measuring several points on two different bulkheads. The position and the orientation of the tracker are marked by a weld seam on the ground plating. The development of the tracking unit and the



(a) (b)

Figure 3.4: (a) shows the LED target attached to the climbing robot. (b) shows the 3D tracker which is used to estimate the position of the climbing inspection robot during the ship's survey.

accuracy of the proposed method were first described in [FEWK12]. An enhanced version with a particle filter for the pose estimation was verified in field experiments in [EBPGF+14]. The robot position estimation approach is depicted in Figure 3.5. The core of the tracker control architecture is a position estimator based on a blob tracker combined with an Laplacian of

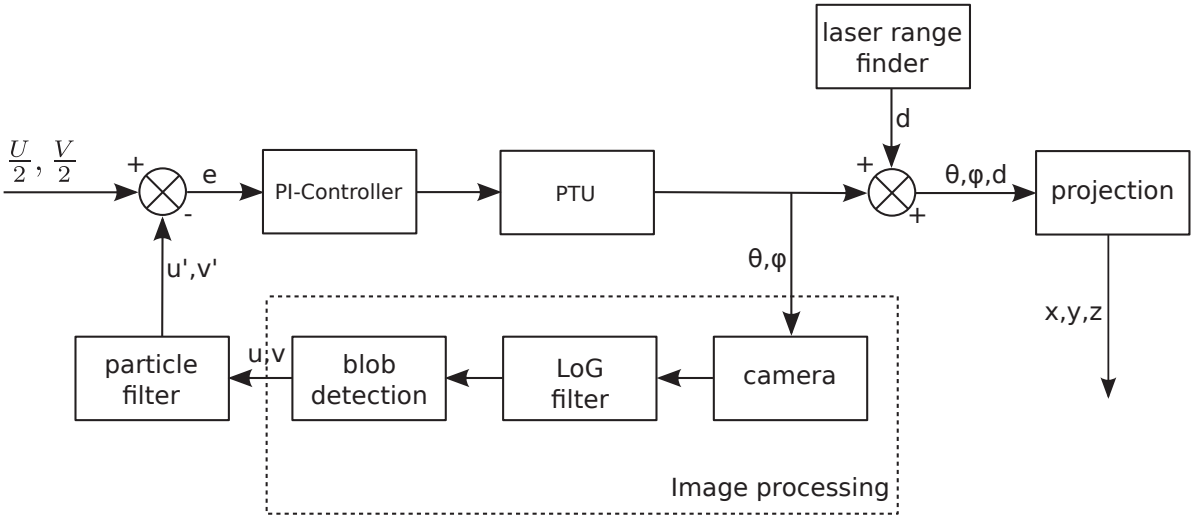


Figure 3.5: Block diagram of the localization approach.

Gaussian filter (LoG-Filter), given by

$$g(u,v) = -\frac{1}{\pi\sigma^4} e^{-\frac{u^2+v^2}{2\sigma^2}} \left(1 - \frac{u^2+v^2}{2\sigma^2}\right), \tag{3.1}$$

as described in [EBPGF+14]. A particle filter is applied to the 2D pose estimation of the robot within the 2D image plane, denoted by u, v. The particle filter ensures the reliability of the position estimation if other light sources and reflections are visible by the tracker and during temporary occlusions. The particle filter for the pose estimation of the climbing robot is given by

$$Bel(x_t) = \mu \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\|o_t - x_t\|}{\sigma}\right)^2} \int p(x_t|x_{t-1}, \mathcal{T}(x_{t-1}, x_{t-2}), g(x_{t-1}, \sigma)) Bel(x_{t-1}), \tag{3.2}$$

where the last motion at time x_{t-1} is based on the translation between the two prior estimated positions, $\mathcal{T}(x_{t-1}, x_{t-2})$, i.e., the last motion of the blob within the 2D image plane is projected as the motion model. The Gaussian noise is incorporated as the function $g(x_{t-1}, \sigma)$. The details of the position estimation filter are described in [EBPGF+14]. Based on the angles θ, ψ of the tracking unit and the distance d provided by the laser range finder, the position of the robot is estimated. The experimental setup and the localization results are published in [EBPGF+14]. Several trajectories generated by the robot on a metal wall are summarized in Figure 3.6 and Figure 3.7.

Prior to the field experiments, the tracking concept was verified within a controlled laboratory environment using a simpler tracking approach without a particle filter in the experiment [FEWK12]. This first experiment was used to estimate the error based on mechanical inaccuracies of the overall system. In [EV11] an older, less sophisticated tracking approach was described to prove the overall concept of an external tracking approach using a PTU-mounted camera in combination with a laser range finder.

3.4 Spatial Content Management System (SCMS)

The Spatial Content Management System (SCMS) is part of the robot-based marine inspection system, described in [EBPGF+14]. This central aspect of the overall architecture is responsible for the representation of the inspection data and allows the surveyor to access the inspection data provided by the robot. In order to compare inspection data consisting basically of visual data and thickness measurements, the data has to be tagged with a position relative to the ship's reference frame and with a time stamp. Some commercial software packages exist which are used by ship classification societies, such as the GL HullManager [15]. This tool provides data for assessing ship conditions and stores them for later use. Mobile tools for damage assessment include a mobile tool for the surveyor where the surveyor has to add inspection data manually on a portable device [45]. Beside the commercially available tool for inspection data storage, some publications exist which focus on the need for inspection data management tools [13, 12].

Currently, all available inspection data storage systems are based on manual input provided by the surveyor. Within the MINOAS project, an approach was investigated to integrate the robot and the surveyor as man-in-the-loop into one integrated system. In this proposed approach, the inspection data is acquired by a robot and evaluated by the surveyor. The damage assessment is supported by corrosion and crack detection, based on the approach developed by [36]. The inspection data is spatially allocated within a hybrid topological/metric reference model of the ship. A 3D interface allows the surveyor to compare inspection data over a time-line.

During the MINOAS project, software components were developed as part of the inspection robot software framework containing a module for organization and visualization of inspection data. In this module the actual content of the inspection data (such as images, videos, NDT and related documents) are allocated spatially to the structure of the ship. Naturally, it is not possible to define a metric reference frame for the whole ship, therefore a hybrid topological/-metric representation is chosen. This approach derived from the approach of how a surveyor would orient himself within the vessel. If the location of damage or corrosion is communicated between experts, a topological or symbolic annotation is chosen. In the spatial allocation system, the ship is organized within a topological structure as described in [EBPGF+14]. The ship surveyor selects the corresponding node within the ship structure, visualized within a 3D model of the ship. The localization data is organized in a hybrid topological/metrical structure. The sections of the ship are organized topologically. Each node has a local coordinate frame assigned in which the robot is localized metrically.

The robot is localized within the cargo hold's reference frame using the 3D tracking unit described in Section 3.3. A 3D model of the lightweight crawler can be visualized within the 3D environment. The raw inspection data are allocated within the 3D environment of the SCMS as data nodes. Each data node is represented as a 3D blob within the 3D ship model. The data for each blob is acquired by the robot and automatically added to the SCMS during an inspection run. The content of the inspection data (i.e., the actual data, such as video data and images) are stored in a workstation and the content is linked to the 3D environment. The surveyor can use the SCMS to select certain parts of interest of the ship and is able to navigate virtually within the 3D environment to access the content of the data which is spatially allocated to the 3D model of the ship.

3.5 Contribution of the Corresponding Publications

The contribution within this chapter is an integrated approach of several modules, including a magnetic wall climbing robot and a localization method for wall climbing robots with a strong focus on the system's usability for a ship survey. Additionally, a content management tool is introduced which allows retrieving inspection data from a robot and organizing and representing it in a user-specific way.

The three main components of the inspection system were developed, integrated, and tested during the MINOAS project. The goal of the project was to develop tools, architectures, and systems which are application driven. The evaluation of the complete system was performed in two ship trials on two different vessels. During these trials, experts from the field of ship inspection were able to evaluate the complete system. The first approach towards the application of ship inspection was published in [VEGF10] where the focus was on the system design and the control of a lightweight climbing robot. The focus in this paper is on the problem description and on the design of the locomotion principle. The need for a localization method and the need to represent the data to the user were pointed out in [EV11]. In [FEWK12], the first experiments using a defect marking system and a localization system were presented. The experiments were mainly done in a laboratory environment and partly on a marine vessel. In the field report published in [EBPGF$^+$14], the final results from the trials with the complete inspection system are described. In this paper, the combination of all three modules, i.e., the wall climbing robot, the localization method for the robot within the cargo hold, and the spatial content management system, was evaluated and tested together with ship surveyors.

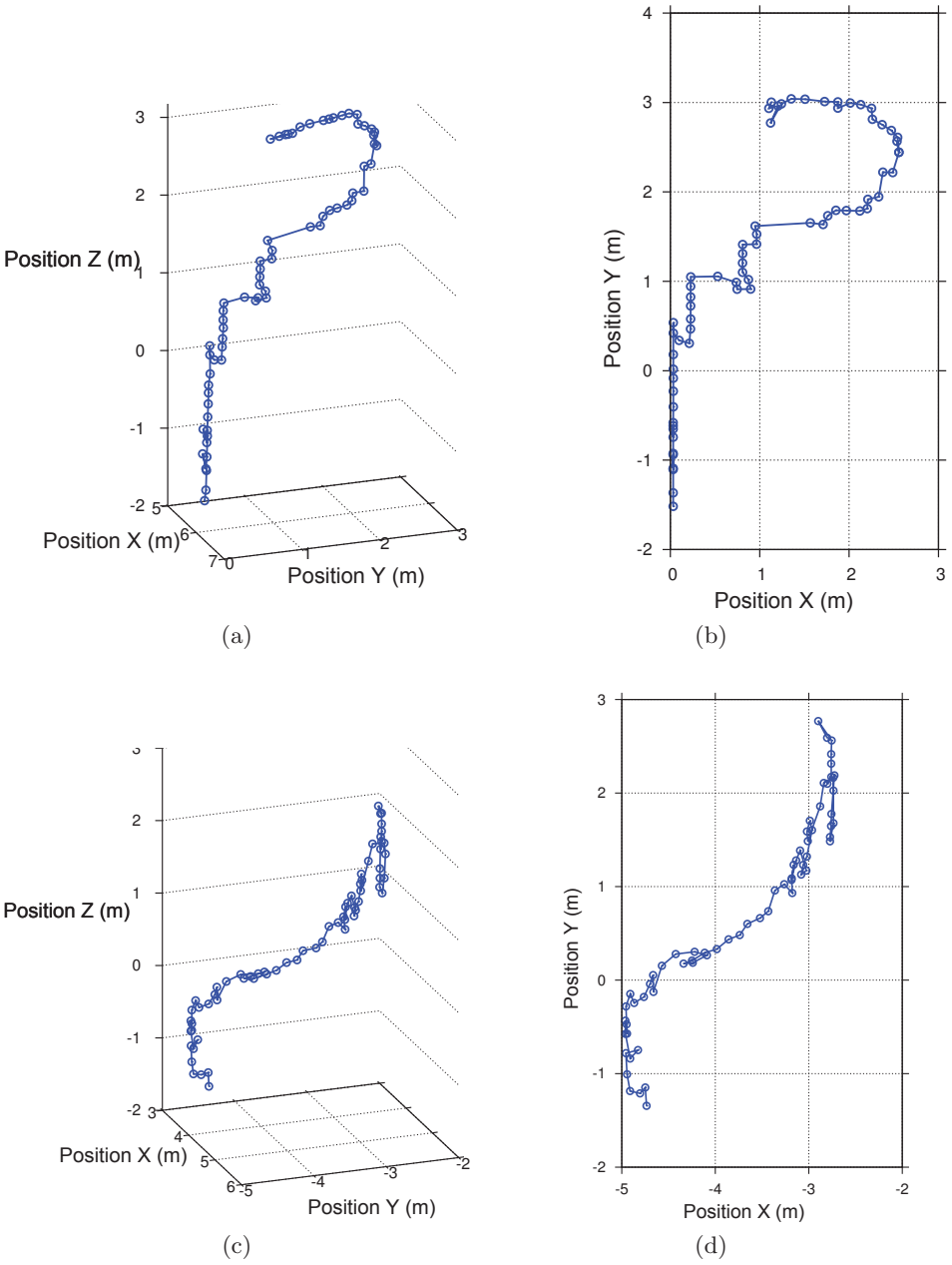
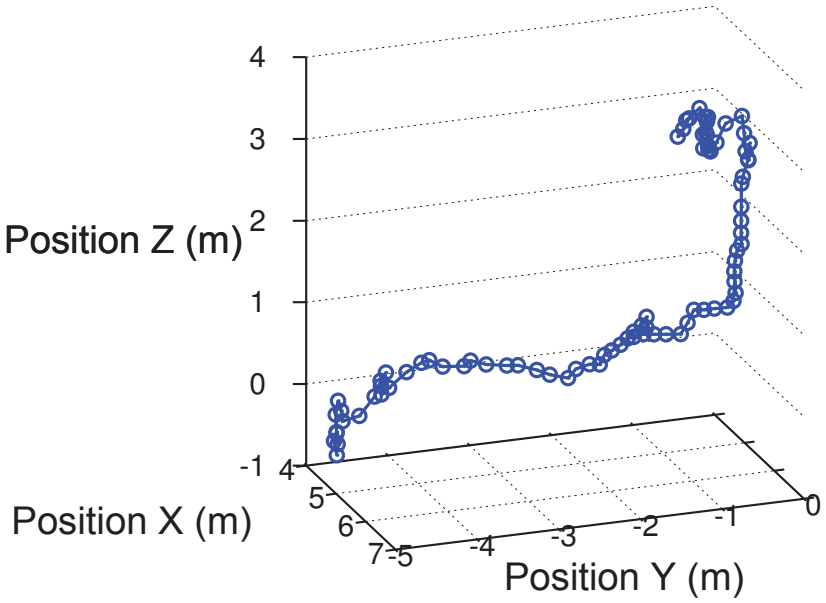
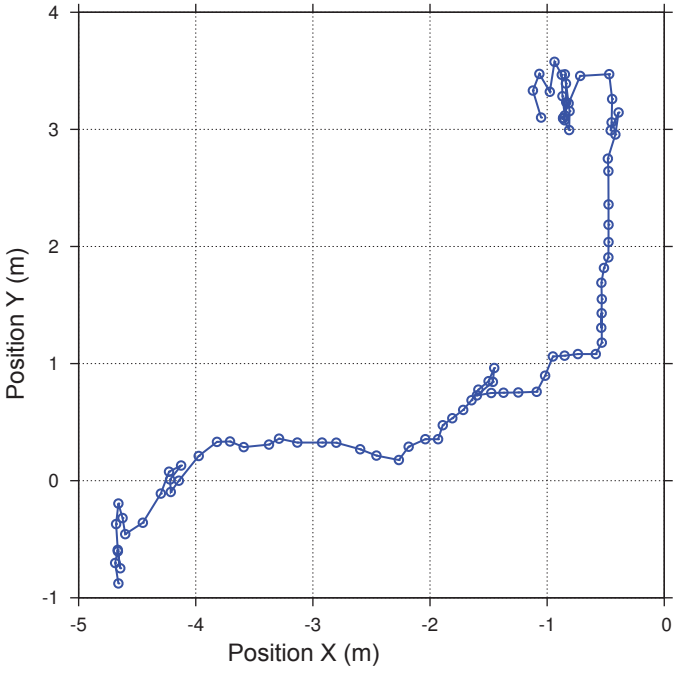


Figure 3.6: Trajectories estimated by the tracking approach using the proposed particle filter for position estimation. Two runs are shown, each in a 3D and 2D perspective.



(a)



(b)

Figure 3.7: An additional run, estimated with the tracking approach described in this section, again in a 3D and 2D perspective.

4 Semantic Perception Methods and their Applications to Ship Inspection

4.1 Introduction

As described in Section 3, it is a mandatory requirement to allocate the data within a marine vessel in order to compare and to organize the inspection data. It is important that the surveyor has access to the data also as offline data, after the actual inspection. Several methods can be used to represent data to the surveyor. In Section 3.4, a method was proposed and evaluated during field trials, where different sections of the ship and of a cargo hold are topologically organized. Within the proposed structure, the data are organized in a topological tree structure where the data is metrically allocated within a leaf of the tree representation. This is achieved using the spatial content management system described in Section 3.4. The drawback of this type of representation is the need for a 3D model in order to present the data to the surveyor.

Other methods of spatial data representation include semantic labels for the structural parts from which the inspection data was acquired. In order to accomplish this, the structural parts of the ship, e.g., of a cargo hold, have to be classified, i.e., the parts of the ship have to be labeled. This allows describing the damage and coating defects in a more natural way, such as "wastage on upper stool at frame 10," "wastage on transverse bulkhead at frame 5," or "blistering on the second guide rail on the port bulkhead." Semantic labels for the cargo hold of a bulk carrier are shown in Figure 4.1 and Figure 4.2 where the structural parts, such as the side shell frames, transverse bulkheads, and stiffeners, are displayed.

During the inspection process, the surveyor creates a report of the damage. In this report, the surveyor describes the locations of the damage (wastage). The naming of the structural parts of a ship is standardized, because different offices of the ship classification societies have to exchange reports. Some examples of surveyor reports are presented in Table 4.1 (by courtesy of Lloyd's Register) where a standardized nomenclature is used in order to inform the office about the wastage. The following nomenclature is used in the report: TM (thickness measurements), CoC (Condition of Class), meaning the repair is mandatory for the ship to be classified, H (sequential number of hull defects). In order to classify structural parts, using the methods proposed in this thesis, a 3D point cloud is generated by a LIDAR system. The basic approach for the classification of the structural parts is based on shape analysis and on spatial features, including the spatial relations between the objects. The description of the structural parts can be based on geometrical primitives, such as planes, rectangles, and circles (in the 2D case), or primitives such as boxes, spheres, and cylinders (in the 3D case). An example of a structural classification, based on spatio-semantic descriptions, is given in Figure 4.1, which uses the spatio-semantic descriptions given in Table 4.2.

The expressions in the corresponding table contain only shape descriptions, geometric mea-

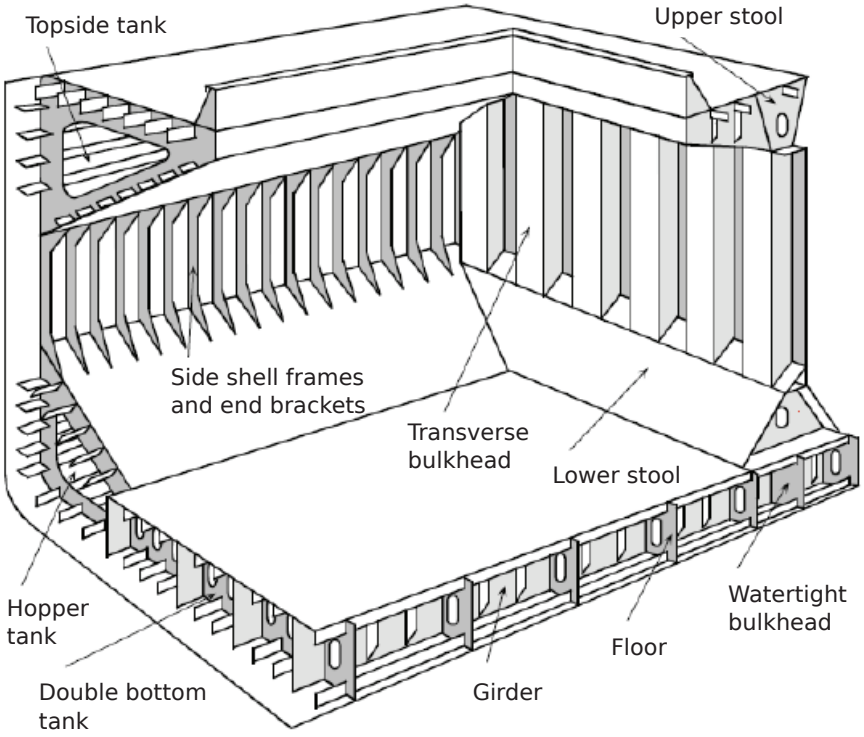


Figure 4.1: A scheme of a bulk carrier cargo hold including the semantic labels of the structural parts. Images by courtesy of Lloyd's Register, RINA, and the MINOAS project.

sures, and spatial relations, such as orthogonal to, parallel to, rising to, above, below, beside, etc. All spatial relations can be tied to semantic expressions that are also geometrically calculable in the spatial domain (cf.[EDK10] for details). Once these descriptions are given, it is possible to describe the structural parts to a person who is not an expert in the field of ship nomenclature, just by using the basic expressions for feature description and spatial relations between the structural parts. It is useful to start with a basis for the description to begin reasoning about the objects. These are referred to as spatial axioms (cf. [EDK10]), i.e., spatial entities which can be classified without having any dependencies on other entities. This is important for the reasoning approach (such as the constraint network described in (cf. [EDK10]) or the Fuzzy DL Reasoner used in [Eic13], because circular dependencies within the corresponding graph lead to undecidability. Cyclic dependencies are not allowed in a decision graph, which is the proposed classification method used in [EDK10].

Another example that shows the key idea behind spatio-semantic description and classification is given by Figure 4.3. The image shows the cargo hold of a container carrier vessel. Without having any knowledge of the nomenclature used by marine experts, the visible parts of the ship can be described as stated in Table 4.3, using only spatial relations, metrics, and shape description. Again, based on the description of the structure, someone not expert about ships can still identify the structural parts without having prior knowledge about the environment. Naturally, the domain of the environment has to be known. Within different domains (e.g., warehouses, factories, etc.), the same spatio-semantic descriptions would result in a mis-classification of the environment. The advantage of the description above is that the spatio-semantic features are measurable in the geometric domain (cf. [EDK10] for details).

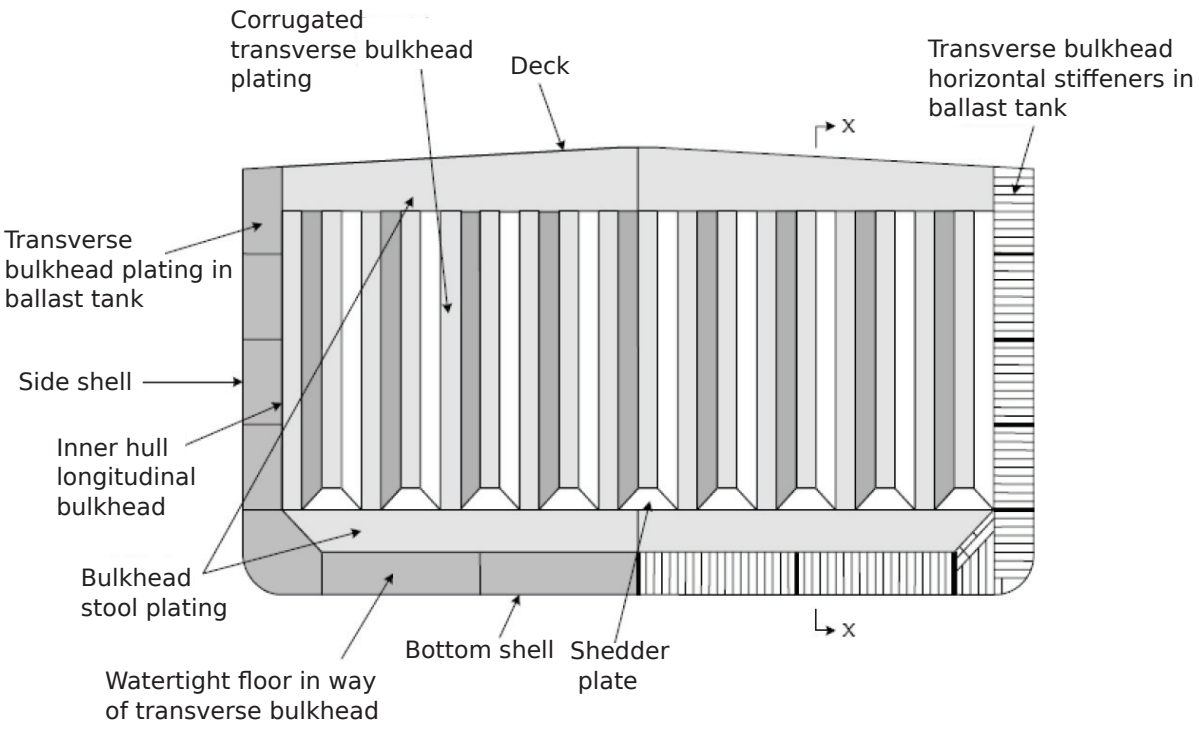


Figure 4.2: Structural parts of a transverse bulkhead. Images by courtesy of Lloyd's Register, RINA, and the MINOAS project.

The environment can be reconstructed using 3D point clouds, generated by a 3D LIDAR system. Individual shapes can be extracted as described in Section 4.2. Based on a knowledge representation of the environment, spatial entities (usually point clusters) can be extracted. Two different approaches for shape extraction and two methods for applying spatial reasoning to the extracted features are provided in the following sections.

4.2 3D shape recovery and spatial feature extraction

In order to semantically analyze the environment of a specific domain, 3D point clouds are used (cf. [EDK10, Eic13]). Three dimensional point clouds are generated using 3D LIDAR systems. The key idea in the published work is to describe the objects and spatial entities in terms of shapes, spatial features, and spatial relations. In the approach proposed in this chapter, the basic shape descriptions are based on, but are not limited to, primitives, such as cuboids, planes, and cylinders. In contrast to identifying object and parts of the environment directly by using model learning and ICP matching [4], the key contribution is to describe what the environment "looks like" and how its objects are related to each other. The perception methods described in this section are based on 3D unorganized point clouds. Unorganized in this context means that initially the neighbor relation between points is not known. Three-dimensional point clouds are usually generated using a 3D LIDAR system and, to a certain extent, stereo images. The latter provide a much lower spatial resolution of the environment and are therefore not considered for this method.

Stated cause: HULL-STRUCTURAL: WASTAGE Date reported: xxx MLIs affected: 3580 NO.1 FOREPEAK W.B. TANK SDN Defect/Damage 1.1 Defective MLIs: 3580 NO.1 FOREPEAK W.B. TANK Description: Local wastage found on various frames. Repairs on partly wasted structure to be carried out according to TM report, Office XXXX, kept in this office. Therefore a suitable CoC recommended to be imposed. Structure considered safe and efficient meanwhile. CoC imposed: H01	Stated cause: HULL-STRUCTURAL: WASTAGE Date reported: xxx MLIs affected: 2363 NO.28 W.B. TANK SDN Defect/Damage 2.1 Defective MLIs: 2363 NO.28 W.B. TANK Description: Local wastage found on frame no. 23 and tween deck plating at frame no. 27. Repairs on partly wasted structure to be carried out according to TM report, Office XXXX, kept in this office. Therefore a suitable CoC recommended to be imposed. Structure considered safe and efficient meanwhile. CoC imposed: H02

Table 4.1: Two reports generated by a surveyor during an inspection. The report is based on standard nomenclature which is used by classification societies. This assures reports are exchangeable and comparable.

• A bottom plate is the lowest horizontal plane.
• Bulkheads are vertical planes orthogonal to the bottom plate.
• Transverse bulkhead columns are vertical cuboids *approximately* 1 m wide.
• Lower stools are plates inclined *approximately* 45° between the bulkheads and the bottom plating.
• Side shell frames are vertical planes orthogonally oriented to a bulkhead.
• Side shell frames are parallel to all side shell frames.
• The distance between two side shell frames is *approximately* 1 m.
• Shedder plates are planes inclined *approximately* 45° between two columns of bulkheads.

Table 4.2: Spatio-semantic descriptions of bulk carrier holds. Such spatial relations are defined as imprecise knowledge about the environment and written in *Italics*.

As a first step, the generated 3D point cloud of the environment has to be analyzed for its geometric structure. This can be achieved by a variety of methods, such as model matching [43] or feature learning [29]. In the following sections a method for 2D shape and feature extraction from unorganized point clouds is described.

4.2.1 Region Growing and 2D Shape Analysis

In this section, the extraction of 2D planes from a 3D (or 2.5 D) point cloud is introduced. The 2D planes are then analyzed for shape using a convex and concave hull analysis. Based on the extracted alpha shape polygons, the 2D shape is classified. The methods and the results of this section were published in [EDK10].

After a 3D point cloud of the environment has been generated, the unorganized point cloud has to be analyzed for the occurrence of geometrical structures. The neighborhood relation of the points is significant for the neighbor search used in the algorithm proposed in this

• The bottom plate is the lowest horizontal plane and larger than 30 m^2.
• Bulkheads are vertical planes orthogonal to the bottom plate.
• Bulkheads have a minimum size of 30 m^2.
• Tanks are cuboids of *approximately* 2 m height and are *approximately* 8 m wide.
• Tanks are on the bottom plate or tanks are on tanks.
• Guide rails have a width smaller than 0.5 m and a height greater than 8 m.
• Guide rails are on bulkheads.
• Guide rails are parallel to guide rails.
• Manhole covers have a circular shape.
• Manhole covers are on the bottom plate or on tanks.

Table 4.3: Spatio-semantic descriptions of container holds. Note the imprecise expression *approximately* in the statements. This expression is also covered in the approach described in this chapter.

section: therefore a k-nearest neighbor (KNN) search is initiated in the point cloud [53]. The method for extracting 2D planes from 3D point clouds using iterative region growing is shown in Algorithm 1. The algorithm is an improvement on the region growing algorithm described in [54]. The proposed algorithm works on unorganized 3D point clouds and the k-nearest neighbor relation is used. For the sake of performance, the cost-intensive KNN search is performed once at the beginning of the process. The algorithm keeps track of the growing frontier of the plane, i.e., each plane has its assigned nearest neighbors which are the next candidate for the growing process. During the region growing process, the KNNs of each point added to the plane candidate are merged with the region frontier RC_{front}. The key principle of the algorithm is the extension of the frontier points by neighboring points if the plane property is still fulfilled.

Here, R denotes the set of regions that are identified during the process, while RC denotes the region candidate. Ψ denotes the set of unprocessed points. The priority queue $knn_i \in KNN$ denotes the list of the k-nearest neighbors of point $\psi_i \in \Psi$. All $knn \in KNN$ as well as RC_{front} are implemented as priority queues and therefore arranged according to their distance. In order to increase memory efficiency, knn_i contains only the pointer to $\psi \in \Psi$. The function $\Omega(RC_{front}, i)$ is used to select $\psi \in \Psi$ being the ith neighbor of the growing region front. The set of KNNs for each point ψ is computed once and stored in a priority queue, arranged according to their distance from the corresponding seed points (Lines 1–4). The first point ψ is selected from the remaining points of set Ψ and added to the region candidate RC (Lines 6–8). The set knn_ψ of ψ becomes the first region growing frontier RC_{front} (Line 9). The algorithm processes each point in RC_{front} and checks whether the merging criteria are met. This is done by checking the mean square error (MSE) of the plane including the new point as well as the orthogonal distance ($\|\|\|_\perp$) of the new point from the best fitting plane (BFP) of RC (Line 11). In the current implementation of the algorithm, the least squares fitting algorithm implemented in CGAL [8] is used. Eventually, the point selected by $\Omega(RC_{front}, 1)$ in Ψ is added to RC and removed from Ψ (Lines 12–13). The corresponding k-nearest neighbors of ψ are merged into the priority queue RC_{front} (Line 14). If the queue RC_{front} contains no unprocessed points, the region candidate is added to the set of classified regions if it contains sufficiently many points (θ), but is dropped otherwise (Lines 18–21). The algorithm terminates if no unprocessed points are left in Ψ.

To verify the algorithm, two models of a ship scenario were tested. One model was based

Figure 4.3: The cargo hold of a container vessel with structural parts visible. The nomenclature (i.e., the semantic labels) of the displayed structures is given in the text in order to illustrate the idea behind the spatio-semantic description.

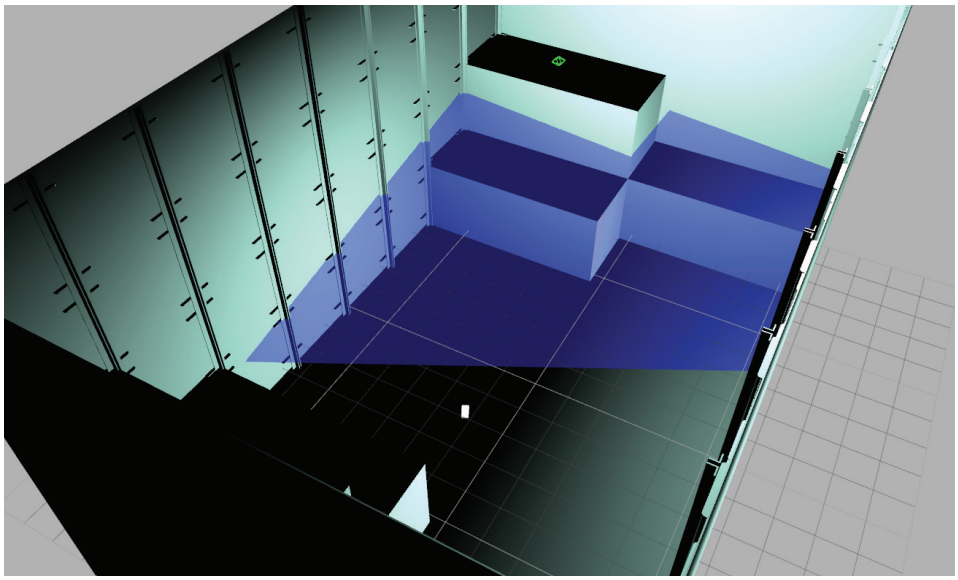
on a container cargo hold, shown in Figure 4.3. From the 3D CAD model of the hold, the 3D simulator Gazebo was used to simulate a 3D laser sweep within the scene [26]. The laser was simulated with an error deviation of 1 cm, corresponding to the error specification of the real Hokuyo UTM30-LX Laser scanner. The second experiment was done within a 3D model of a bulk carrier cargo hold. The two experimental environments used for the sample data acquisition are shown in Figure 4.4. The algorithm shown in Algorithm 1 was used to extract the segments of the structure of the cargo holds of a container ship and of a bulk carrier. The results of the extracted structure are given in Figure 4.5.

In order to analyze the extracted planes, several features are extracted from the structural parts of the holds. The key idea behind the spatio-semantic analysis is that form parameters are extracted and the spatial relations between the different objects are extracted. As described in [EDK10], the spatial features needed for the classification are given in Table 4.4.

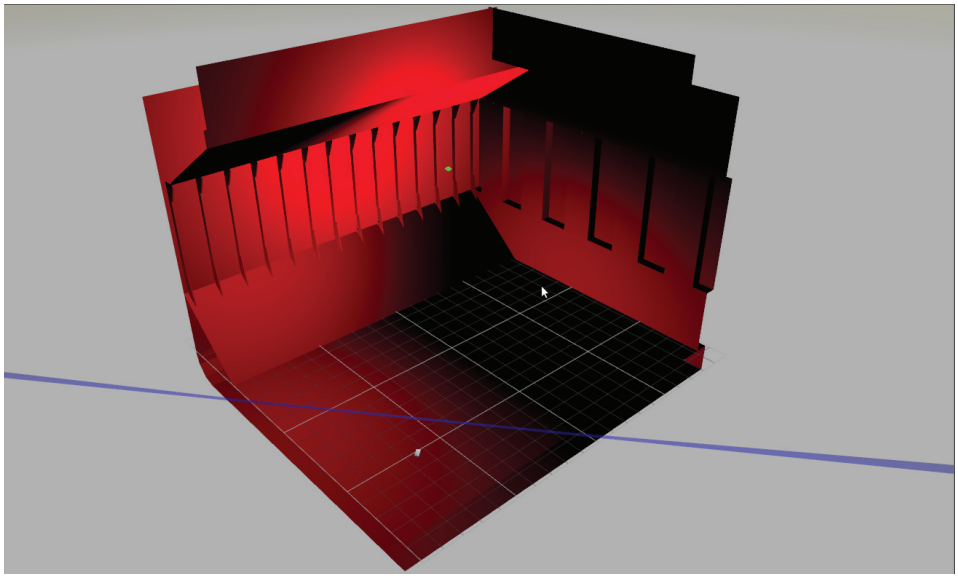
A	Area covered by the extracted plane square meters
E	Maximum extension of the rectangle.
R	Rectangularness $\in [0, 1]$ gives the matching results between the extracted shape and a perfect rectangle.
P	3D position of the shape (x, y, z). The position is set at the center of gravity of the shape.
N	3D orientation of the shape (ϕ, θ, ψ). The orientation is given by the plane normal pointing at the observer at (0,0,0) at the scene.

Table 4.4: The spatial features which are extracted from the unorganized point cloud.

In the case of the bulk carrier (cf. Figure 4.5 upper part), 41 possible spatial entities were



(a)



(b)

Figure 4.4: Two different ship holds used for the structural classification. (a) Hold of a container ship. (b) Hold of a bulk carrier. The data are based on the original 3D ship CAD data and are therefore to scale. The scanning laser was simulated using the specification of a Hokuyo UTM 30-LX laser. The blue plane shows the scanning area of the laser scanner during a sweep scan.

Algorithm 1 Region Growing Algorithm

$R \leftarrow \emptyset, RC \leftarrow \emptyset, RC_{front} \leftarrow \emptyset$
for all $\psi \in \Psi$ **do**
    calculate $knn_i \in KNN$ with $\|\psi_i - knn_i\| < \delta$
end for
while $\Psi \neq \emptyset$ **do**
    select random $\psi \in \Psi$
    $RC \leftarrow \psi$
    remove ψ from Ψ
    $RC_{front} \leftarrow knn_\psi$
    while $RC_{front} \neq \emptyset$ **do**
        if $MSE(RC \cup \Omega(RC_{front}, 1)) < \epsilon$ and $\|BFP(RC) - \Omega(RC_{front}, 1)\|_\perp < \gamma$ **then**
            $RC \leftarrow RC \cup \Omega(RC_{front}, 1)$
            remove $\Omega(RC_{front}, 1)$ from Ψ
            $RC_{front} \leftarrow RC_{front} \cup knn_\psi$
        end if
        pop RC_{front}
    end while
    if $size(RC) < \theta$ **then** drop RC
    else
        $R \leftarrow R \cup RC$
    end if
end while

Setup	γ	θ	ϵ	δ	KNN
Cargo Hold	0.1 m	300	0.02 m	0.1 m	5
Container Hold	0.08 m	800	0.03 m	0.2 m	5

Table 4.5: Parameters used for the point cloud segmentation.

extracted. The processing was based on the 271,923 points in an unorganized point cloud and took 109.489 seconds for the region growing and feature extraction. The extraction of the spatial entities shown in the cargo hold setup (cf. Figure 4.5 lower part) was based on 264,046 points and took 122.328 seconds. The segmentation algorithm was run on an Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz. From the container hold setup, 19 regions of interest were extracted. The parameters of Algorithm 1 used for the segmentation are given in Table 4.5. Some of the extracted spatial features of the segmented regions are shown in Table 4.6 for some spatial entities. The values *ortho* and *paral* give the score of the extent to which the region is orthogonal to the $x - y$ plane of the robot. Orthogonality and parallelism are two spatial relations which are used later to analyze the spatial relations between the different shapes (cf. [EDK10]). The relation features used are basically "soft features," where the score defines the likelihood of two shapes' being orthogonal or parallel to each other. This is due to the fact that the extracted relations can be erroneous due to sensor noise or structural inaccuracies. The values *ortho* and *paral* are both in the closed interval $[0, 1]$. The *rectangularness* is a feature which is extracted by analyzing the alpha shape of an extracted region and comparing the shape with a minimum spanning rectangle of the same segment. Figure 4.6 shows two extracted regions from the cargo hold scenario. The corresponding shapes are shown in Figure 4.6.

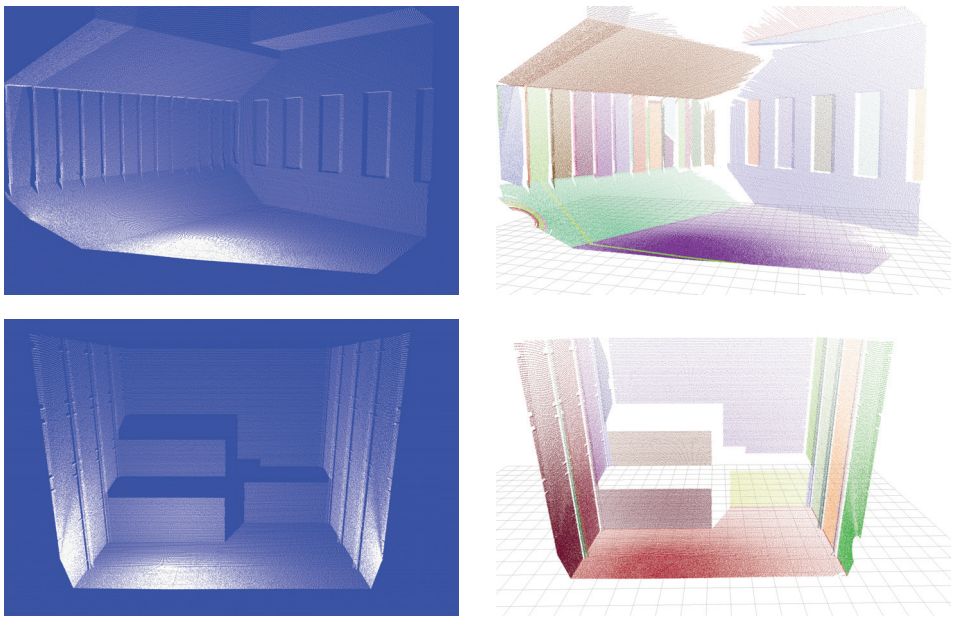


Figure 4.5: Classified planes, generated from unorganized 3D point clouds of a cargo hold and a container hold.

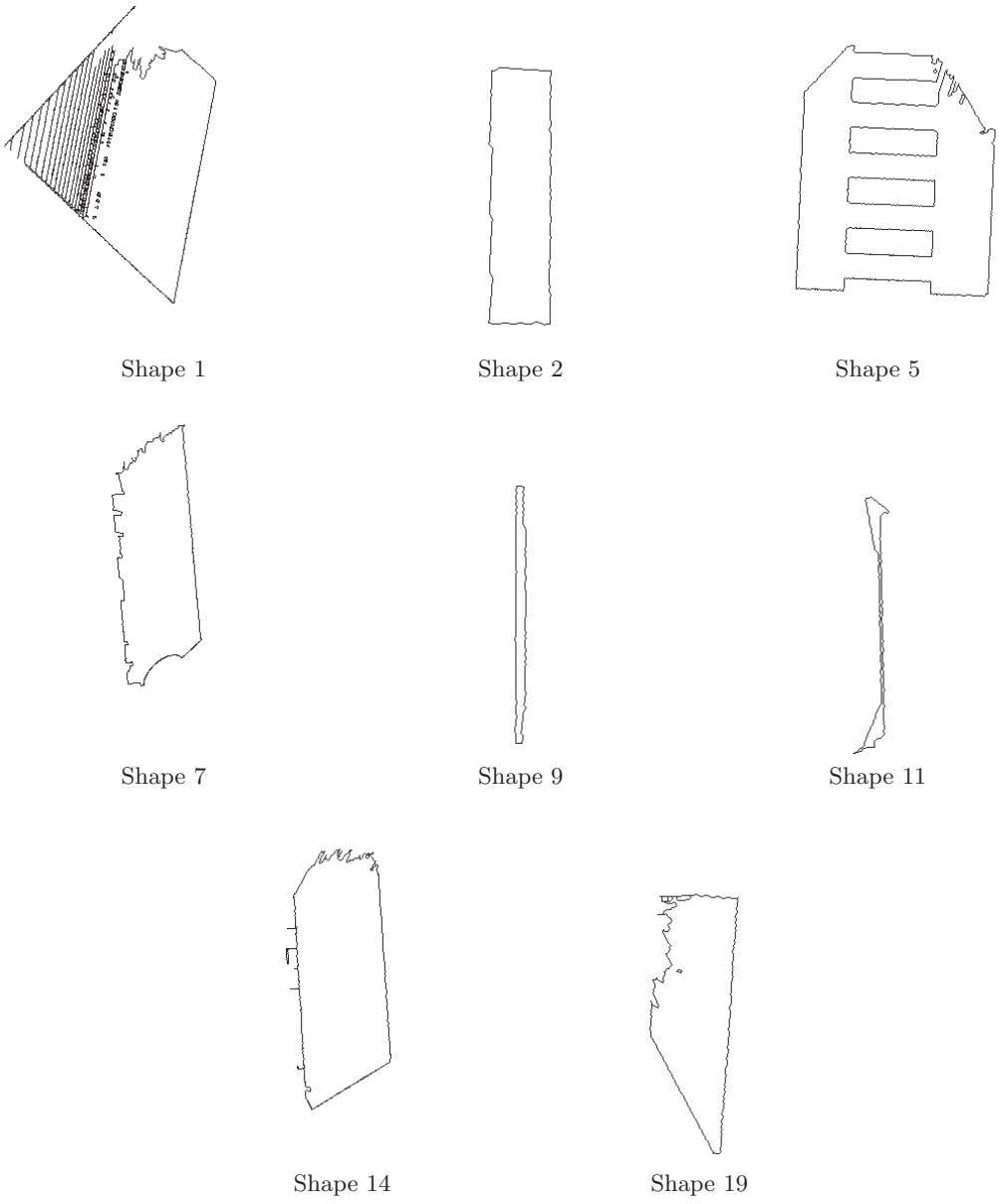
Shape ID	$A(m)$	$E(m)$	R	$Px(m)$	$Py(m)$	$Pz(m)$	$orthoG$	$paralG$
1	44.43	16.53	0.19	4.08	-3.25	0.01	0.00000	0.99999
2	8.28	5.96	0.96	15.48	-1.82	6.45	0.99921	0.00078
5	19.19	17.13	0.08	11.97	-6.19	6.86	0.99906	0.00093
7	58.75	17.72	0.62	6.82	5.41	1.83	0.24983	0.75017
9	1.18	5.58	0.90	5.42	8.27	6.50	0.99775	0.00225
11	1.03	6.44	0.20	1.59	10.49	6.54	0.99279	0.00720
14	72.53	18.56	0.55	6.81	4.23	10.9	0.15810	0.84189
19	23.67	11.72	0.55	8.91	-7.44	13.5	0.00000	0.99999

Table 4.6: Features extracted from the cargo hold scenario, shown in Figure 4.5, upper part. The features *orthoG* and *paralG* give the likelihood of the shape's being orthogonal or parallel to the $x - y$ plane.

The approach described in this section of the thesis uses spatial relations as well as shape analysis based on the analysis of the alpha shapes of the extracted spatial entities. The extracted features are used to classify the structural parts based on the constraint network described in the following section.

4.2.2 Spatio-Semantic Analysis of 3D Environments

The first approach described in this thesis for structural analysis is based on a probability propagation within a constraint network to classify spatial objects as described in [EDK10]. The paper gave evidence that spatial entities can be classified using a spatio-semantic description of an office environment. In this section, the same approach is transferred to the domain of robot based ship inspection. The approach used is comparable to the human way of describing objects. An important factor in describing objects is their shape and the structural



Shape 1 Shape 2 Shape 5

Shape 7 Shape 9 Shape 11

Shape 14 Shape 19

Figure 4.6: Alpha shapes extracted from the cargo hold

inter-dependency of structural parts, or spatial entities. The structure for the spatial reasoning approach is given as a constraint network, where every node is represented as a feature vector Φ defined as

$$\Phi \quad = \quad (A, E, R, \Theta)^T,$$

where $A \in \mathbb{R}^2$ is the area covered by the shape, $E \in \mathbb{R}$ describes the maximum extension of the detected entity, and $R \in [0, 1]$ describes the "rectangularness" as described in the previous section. $\Theta \in [0, 1]$ describes the relation to the other detected entities, and is given by

$$\Theta \quad = \quad P(\omega, \alpha),$$

where $\omega \in \Omega$ is the target of the relation and $\alpha \in A$ is the definition of the spatial relation. Ω is a set of targets, i.e., labels in the semantic domain; A holds the attribute space, mapping semantic attributes to geometric attributes. The labels in the semantic domain are defined in a model data base. In the domain of ship inspection, where the positions of the robots or the detected defects have to be allocated to things which are semantically labeled, the structural parts are given by ω. The attributes describe the spatial relation between the detected entities (i.e., the parent of the relation) and the target entity. Examples of such attributes are given in Table 4.7.

$\alpha \rightarrow$	$[0, 1]$
"$above$" $\rightarrow$	$(Pos\_Z_{parent} - Pos\_Z_{target}) < 0$
"$behind$" $\rightarrow$	$(Pos\_XY_{parent} - Pos\_XY_{target}) < 0$
"$orthogonal$" $\rightarrow$	$1 - N_{parent} \cdot N_{target}$
"$parallel$" $\rightarrow$	$1 - ((N_{parent} \cdot N_{target}) - (\lvert N_{parent} \rvert \cdot \lvert N_{target} \rvert))$
"$diagonal$" $\rightarrow$	$1 - \lvert orthogonal - parallel \rvert$
"$near$" $\rightarrow$	$closest\_point(parent, target) < \epsilon$

Table 4.7: Spatio-semantic relations which are mapped by the function α. The semantic symbol for the spatial relation is mapped into the geometrical domain. The mapping is a function which matches the symbol to $[0, 1]$, where 1 means the symbol is fully satisfied.

The constraint network is represented by nodes and interconnected by the relation between spatial entities. The network represents the knowledge about the domain. In the proposed method, the features of the model do not have to be precise. The measurements of a shape (e.g., the area covered or the spatial relation) can be roughly approximated while modeling the spatial domain. The modeled knowledge can contain facts like "roughly 20 m wide" or "roughly parallel." Such "soft features" are covered by the model matcher using a linear function. The latest approach in spatial domain modeling described in Section 4.2.4 uses a Fuzzy Reasoning approach that uses model matching based on Fuzzy Sets. This allows a more precise modeling of the spatial domain, the spatial features, and the spatial relations. As an example for the model of the bulk carrier cargo hold domain, the structural parts of a bulk carrier cargo hold are modeled as shown in Table 4.8.

The result is a network structure where the interconnection of the nodes is represented by (ω, α), where α is the spatio-semantical annotation of the relation given in Table 4.7 and

$$
\begin{aligned}
\Phi_{upper\_stool} =& (120, 20, 1, P(\text{“}diagonal,\text{”} \text{“}ground\_plating\text{”}) \\
& * P(\text{“}above,\text{”} \text{“}lower\_stool\text{”})) \\
\Phi_{lower\_stool} =& (100, 20, 1, P(\text{“}diagonal,\text{”} \text{“}ground\_plating\text{”}) \\
& * P(\text{“}below,\text{”} \text{“}upper\_stool\text{”})) \\
\Phi_{ground\_plating} =& (270, 20, 1, P(\text{“}parallel,\text{”} \text{“}XY - plane\text{”}) * P(\text{“}lowest\text{”}) \\
\Phi_{transverse\_bulkhead} =& (90, 17, 0.5, P(\text{“}orthogonal,\text{”} \text{“}ground\_plating\text{”}) \\
& * P(\text{“}above,\text{”} \text{“}lower\_stool\text{”}) \\
& * P(\text{“}orthogonal,\text{”} \text{“}sideshell\_plating\text{”}) \\
\Phi_{sideshell\_plating} =& (10, 10, 1, P(\text{“}above,\text{”} lower\_stool) \\
& * P(\text{“}above,\text{”} \text{“}lower\_stool\text{”}) \\
& * P(\text{“}orthogonal,\text{”} \text{“}ground\_plating\text{”}) \\
\Phi_{sideshell\_frame} =& (120, 20, 1, P(\text{“}orthogonal,\text{”} \text{“}ground\_plating\text{”}) \\
& * P(\text{“}near,\text{”} \text{“}sideshell\_plating\text{”}) \\
\Phi_{corrugated\_plating} =& (9, 5, 1, P(\text{“}parallel,\text{”} \text{“}transverse\_bulkhead\text{”}) \\
& * P(\text{“}behind,\text{”} \text{“}transverse\_bulkhead\text{”})
\end{aligned}
$$

Table 4.8: Reasoning network setup for the classification of structural parts in a bulk carrier cargo hold

ω is the target entity of the relation. Most spatial entities have to satisfy several spatial dependencies, e.g., side shell plating, which has to be above the lower stool, below the upper stool, and orthogonal to the ground plane. The entity “ground plating” serves as a spatial axiom, meaning that no direct dependency on other entities is needed to classify this entity. The ground plating can be detected directly from the robot coordinate frame (being the lowest plane parallel to the $x - y$ plane of the robot). The reasoning approach starts always with spatial axioms. Iteratively, each extracted entity from the point cloud is matched with every entity of the model.

The reasoning algorithm starts as follows: In a probability matrix PM, given by MxE, where M denotes the number of models and E denotes the number of spatial entities, the probability for each field is initialized to be 1. The rule set (RS) is initialized with all spatial rules Θ as described above in this section.

The initial probability matrix is given in Table 4.9. A spatial entity is classified if the corresponding row has only one entry greater than zero. The rules are then again applied to the identified entity, i.e., the spatial relation given in the constraint network. The algorithm is described in Algorithm 2. It is important to note that the model checking is based on a conjunctive application of the rules and feature matching. This means that the probability of one model/entity pair is eliminated if a single feature has a likelihood of zero.

The run time of the algorithm is given by $\mathcal{O}(|m| * |e| * |r|)$ where m is the number of models, e the number of entities, and r is the total number of rules (spatial relations).

Algorithm 2 Spatial Reasoning Algorithm

```
1: PM ← 1.0                                          ▷ Initializes probability matrix
2: Entity ← ExtratedShapes                           ▷ Initializes shape models from feature extractor
3: Model ← Models                                    ▷ Get Model from domain knowledge base
4: RS ← Rules                                        ▷ Initializes rule set
5: ARS ← AxiomRules                                  ▷ Initializes axiom rule set
6: AM ← AxiomModels                                  ▷ Initializes axiom models
7: for all amodel ∈ AM do
8:     for all entity ∈ Entity do
9:         for all ar ∈ ARS do
10:            p[amodel, entity] ← ar(entity,amodel)   ▷ Identifies spatial axiom.
11:        end for
12:    end for
13: end for
14: for all model ∈ Model do
15:    for all entity ∈ Entity do
16:        for all rule ∈ RS do
17:            if rule can be applied to entity then
18:                p[model, entity] ← rule(entity, model)
19:            end if
20:        end for
21:    end for
22: end for
```

Once a node is matched, i.e., the column of a model has only one entry greater than zero, this node serves as a reasoning basis for the next shape. The probability matrix after the classification of the axiom entity (i.e., the ground plane) is given in Table 4.10. The resulting classifications, after the reasoner is finished with the structural analysis, are given in Table 4.11.

In this section it was shown how the relevant publication about semantic perception (cf.[EDK10], which was primarily developed for indoor perception in an office environment, is applicable to the domain of marine vessel inspection, especially to the semantic labeling of the structural parts. This again is the basis for description of the spatial defects and semantic localization of a robot. Without semantic classification, the defects could only be described within a metrical model (cf. Section 3.3. This would need a full 3D model of the ship in order to make the data presentable to a ship surveyor. An example of a spatial data allocation system was presented in Section 3.4.

4.2.3 3D Plane Extraction and Clustering using Random Sample Consensus Methods

In the previous section, a semantic perception method was presented based on 2D shape extraction and 2D shape analysis. The spatial 3D relations between the extracted shapes were used to classify the objects. Another method of spatial analysis, published in [Eic13], is based on direct 3D shape extraction based on a RanSaC approach which classifies geometrical primitives within a 3D point cloud. The use of RanSaC-based model checking was first introduced

Shape ID	US	LS	GP	TB	SP	SF	CP
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	1.000	1.000	1.000	1.000	1.000	1.000	1.000
11	1.000	1.000	1.000	1.000	1.000	1.000	1.000
14	1.000	1.000	1.000	1.000	1.000	1.000	1.000
19	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 4.9: The likelihood is initialized with an equal distribution. US: upper stool, LS: lower stool, GP: ground plating, TB: Transverse bulkhead, SP: side shell plating, SF: side shell frame, CP: Corrugated plating.

Shape ID	US	LS	GP	TB	SP	SF	CP
1	1.000	1.0000	**0.833**	1.000	1.000	1.000	1.000
2	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
5	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
7	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
9	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
11	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
14	1.000	1.0000	0.000	1.000	1.000	1.000	1.000
19	1.000	1.0000	0.000	1.000	1.000	1.000	1.000

Table 4.10: The ground plane (GP) is eliminated first based on the spatial axiom rule.

by [43]. In the approach described in this section, planar surfaces are classified using the approach given in Algorithm 3.

The algorithm starts with a point cloud P and an empty set of candidates. For each point, the normal is calculated by taking into account the k-nearest neighbors. The candidate seed is defined by three initial points p_1, p_2, p_3, which are selected randomly. Every point in the remaining point cloud Ψ is checked as to whether it is a supporting point for the plane by taking into account the normal of the plane and the distance of the point from the plane. If the number of supporting points is above a threshold, the model is accepted and the points belonging to the plane model are removed from the point cloud Ψ. The algorithm terminates if the probability of detecting more shapes ($P(m_P, P)$) is below a threshold (τ). This is simply calculated if the number of remaining points in P (denoted as m_P) does not change over a pre-defined number of iterations. The remaining point cloud can then be processed further in order to detect non-planar shapes.

The remaining point cloud is clusterized using a point cloud cluster algorithm. Because the three-dimensional clusters representing the guide rails on the port and starboard bulkhead are not based on geometrical primitives as described in [43], features, such as extension, position and orientation, are directly extracted from the shape clusters without model matching. It will be shown in the next section that spatial relations in combination with the extracted spatial features are sufficient to classify guide rails.

Figure 4.7 shows a point cloud generated within a container hold. The visible structure consists

Shape ID	US	LS	GP	TB	SP	SF	CP
1	0.000	0.000	**0.833**	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	**0.910**
5	0.000	0.000	0.000	**0.284**	0.000	0.000	0.000
7	0.000	**0.434**	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	**0.498**	0.000	0.000
11	0.000	0.000	0.000	0.000	0.000	**0.598**	0.000
14	**0.273**	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4.11: The spatial reasoning results for the bulk carrier cargo hold. The following structures are detected. US: upper stool, LS: lower stool, GP: ground plating, TB: Transverse bulkhead, SP: side shell plating, side shell frame, CP: corrugated plating.

Algorithm 3 Extract shapes in the point cloud P

$\Psi \leftarrow \emptyset$ ▷ plane shapes
$\mathcal{C} \leftarrow \emptyset$ ▷ plane candidates
$P \leftarrow$ point cloud ▷ 3D point cloud
$N \leftarrow$ point normals
repeat
  $\mathcal{C} \leftarrow$ random point $p_1, p_2, p_3 \in P$
  for all $p \in P$ **do**
    if $|p - \mathcal{C}|) < \epsilon$ and $N_{\mathcal{C}} - N_p < \theta$ **then** ▷ Check if point support models.
      $\mathcal{C} \leftarrow \mathcal{C} \cup p$
    end if
  end for
  if $m_{\mathcal{C}} > \delta$ **then** ▷ check if candidate contains enough points
    $P \leftarrow P \backslash P_m$ ▷ remove points from the source
    $\Psi \leftarrow \Psi \cup \mathcal{C}$ ▷ add candidate to shape
  end if
  $\mathcal{C} \leftarrow \emptyset$ ▷ remove points from candidates
until $P(m_P, P) < \tau$
return Ψ

of the bulkheads (port, transverse and starboard), the cuboid-shaped tanks, and the container guide rails on the port and on the starboard side.

The planar shapes which can be extracted by the algorithm are those that belong to the bulkheads or to the tanks. The features which are extracted after the planar classification are the same as described in Table 4.4. The difference from the region growing algorithm is the performance of the algorithm. Compared to the region growing approach, the RanSaC approach only takes 25.4778 seconds for the successful extraction of all planes, tested on an Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz. The drawback of the RanSaC approach is that planar segments lying in the same plane are identified as one entity. This is because the RanSaC Algorithm takes the plane, defined by the three seed points, as a model and all points are checked against this model. In order to extract different sections lying on the same plane, an additional clustering of the segments has to be done.

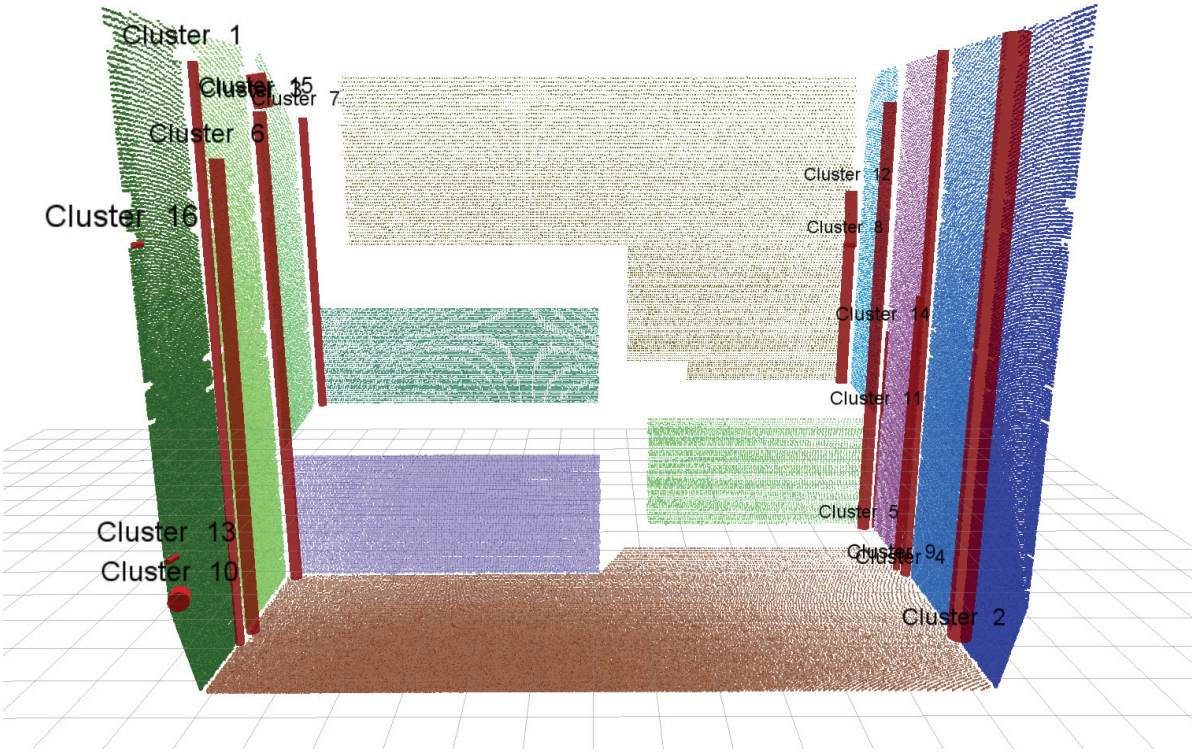


Figure 4.7: Container hold segmented using a 3D RanSaC algorithm. The planar shapes are marked in separate colors. The extension and the orientation of the remaining clusters are shown as columns, containing the clusters.

4.2.4 Spatial Reasoning using Fuzzy Description Logic

In this section, the results published in [Eic13] are transferred to structural classification in the domain of marine vessels. Similar to the section above, the basic algorithms published in the corresponding paper are used and verified with perceptual data from a marine vessel cargo hold. For the experiments presented, the features extracted by the RanSaC algorithm, described in the previous section, are used to classify the structural parts. In the approach described in this section, a reasoning approach based on Fuzzy Description Logic is proposed. The main difference between the spatial reasoning approach given in this section and that given in Section 4.2.2 is the way the features from the extracted spatial entities and the model are matched. In Section 4.2.2, the model matching was based on a linear function and the knowledge about the model was contained in a constraint network. For the approach described in this section, a Fuzzy Ontology is used and the reasoning approach is based on Fuzzy Logic reasoning, which makes use of Fuzzy Sets for the features and spatial relations. This allows a more precise reasoning approach, because for each relation or spatial feature, a different Fuzzy Set can be defined. Fuzzy Description Logic (FuzzyDL) and a corresponding reasoner were first introduced by [50]. Other work including spatial reasoning on Fuzzy Sets is given by [22] where spatial features are used to classify regions within MRI-images.

The approach used in this section is described in detail in the corresponding publication in the Appendix of this thesis [Eic13]. In this section, the approach will be used on data from the marine vessel inspection domain. The experimental results are included in this section.

This shows that the approach is usable in the service robotics domain of ship inspection. The RanSaC approach, described in Algorithm 3, extracted in the cargo hold scenario 12 planar spatial entities and 16 different clusters. The perception module generated implicitly 402 assertions for the A-Box. The amount of assertions results from the fact that for every relation detected by the perception layer results in a new assertion plus the spatial features for each individual such as *height*, *extension* and *rectangularness*.

In the nomenclature of the corresponding paper, the T-Box for the fuzzy classification approach is defined in Listing

Listing 4.1: The T-Box of the Knowledge Base

```
(define-fuzzy-logic lukasiewicz)

(functional hasPlanarArea)
(range hasPlanarArea *real* 0 200)

(functional hasRectangularness)
(range hasRectangularness *real* 0 1)

(functional hasExtension)
(range hasExtension *real* 0 100)

(functional isGroundPlane)
(range isGroundPlane *real* 0 1)

(functional hasHeight)
(range hasHeight *real* 0 20)

(functional isLeftMostPlane)
(range isLeftMostPlane *real* 0 1)

(functional isRightMostPlane)
(range isRightMostPlane *real* 0 1)

(define-fuzzy-concept GroundPlane linear(0,1,0.7,0.5))
(define-fuzzy-concept LeftMostPlane linear(0,1,0.7,0.1))
(define-fuzzy-concept RightMostPlane linear(0,1,0.7,0.1))
(define-concept GP (some isGroundPlane GroundPlane))
(define-concept LMP(some isLeftMostPlane LeftMostPlane))
(define-concept RMP(some isRightMostPlane RightMostPlane))
(define-concept PB (or (and (some isParallel LMP) (some isOnPlane ↪
    ↩ LMP)) (some isLeftMostPlane LeftMostPlane)))
(define-concept SB (or (and (some isParallel RMP) (some isOnPlane ↪
    ↩ RMP)) (some isRigthMostPlane RightMostPlane)))
(define-concept TB (and (some isOrthogonal GP) (some isOrthogonal ↪
    ↩ PB) (>= hasExtension 10)))
(define-concept TA (and (some isOrthogonal GP) (some isOrthogonal ↪
    ↩ PB) (<= hasExtension 10)))
(define-concept PG (and (some isParallel GP) (some isOnPlane PB) ↪
    ↩ (>= hasHeight 5)))
```

```
35 (define-concept SG (and (some isParallel GP) (some isOnPlane SB) ↪
     ↩ (>= hasHeight 5)))
36
37 (symmetric isOrthogonal)
38 (symmetric isParallel)
39 (symmetric isOnPlane)
```

Shape ID	GP	PB	SB	TB	TA	PG	SG
plane1	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
plane2	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
plane3	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000
plane4	**0.806**	0.000	0.000	0.000	0.000	0.000	0.000
plane5	0.000	0.000	0.000	0.000	0.000	0.000	0.000
plane6	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
plane7	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
plane8	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000
plane9	0.000	0.000	0.000	0.000	**0.792**	0.000	0.000
plane10	0.000	0.000	0.000	**0.800**	0.000	0.000	0.000
plane11	0.000	0.000	0.000	0.000	**0.798**	0.000	0.000
plane12	0.000	0.000	0.000	0.000	**0.798**	0.000	0.000
cluster1	0.000	0.000	0.000	0.000	0.000	**0.731**	0.000
cluster2	0.000	0.000	0.000	0.000	0.000	0.000	**0.75**
cluster3	0.000	0.000	0.000	0.000	0.000	**0.592**	0.000
cluster4	0.000	0.000	0.000	0.000	0.000	0.000	**0.698**
cluster5	0.000	0.000	0.000	0.000	0.000	0.000	**0.573**
cluster6	0.000	0.000	0.000	0.000	0.000	**0.622**	0.000
cluster7	0.000	0.000	0.000	0.000	0.000	**0.624**	0.000
cluster8	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster9	0.000	0.000	0.000	0.000	0.000	0.000	**0.660**
cluster10	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster11	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster12	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster13	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster14	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster15	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster16	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4.12: The spatial reasoning results from using the 3D Fuzzy perception approach. The following structures are classified. GP: Ground Plating, PB: port bulkhead, SB: starboard bulkhead, TB: transverse bulkhead, TA: tank, PG: port guide rail, SG: starboard guide rail.

4.3 Contribution of the Corresponding Publications

The section about semantic perception using a constraint network and region growing based feature extraction was published in [EDK10]. In that publication, the method was used to classify objects and spatial entities within an office environment. In the section above, evidence was given that the same method can also be applied to the structural classification of a ship's

cargo hold. In [Eic13] a method based on Fuzzy Description Logic and Fuzzy Reasoning was proposed to classify shapes for soil sample container classification. In the section above it was shown that this approach is also applicable to the ship inspection domain by adapting the corresponding A-Box and T-Box to the task of cargo hold inspection.

5 Conclusion and Outlook

5.1 Conclusion

In this thesis, the novel application of robots to marine vessel inspection was introduced. A collection of robot systems and underlying control architectures was laid out. A complete vessel inspection system was described, consisting of a lightweight magnetic climbing robot for bulkhead inspection, a localization system based on an external tracking unit, and a tracking approach using a particle filter. The inspection data are organized within a novel spatial content management (SCM) system for spatial data representation to the surveyor. This system was evaluated by several laboratory experiments as well as in several field trials. The spatial allocation is based on a metric positioning of the defect data acquired inside the cargo hold using a particle filter based 3D tracking system proposed in this thesis.

An additional allocation method for inspection robots and defect localization was introduced in this research, which labels the structural parts of a ship. This provides a more intuitive way for reporting the defects of a cargo hold to a ship surveyor. It has been shown that semantic labeling in a structured environment is solvable by applying reasoning approaches based on spatial features and spatial relations.

In this work, two methods were proposed for applying semantic labels to the structural parts. The corresponding publications, which are attached to this thesis, had been developed and verified in two application domains that are completely different from that of vessel inspection. It was shown in the corresponding sections that the same approaches are transferable to the domain of vessel inspection by changing the spatial ontology. Evidence has been provided that the same feature extraction methods can be applied to the domain of marine cargo holds without any need for changing their basic principles.

5.2 Outlook

This thesis has covered different aspects of the practical application of service robotics to the novel field of marine vessel inspection. Naturally, a thesis can only provide for a limited aspect of this field. In this thesis, three main contributions were made to the field of service robotics. For each of the contributions given in this thesis, future research directions are given in the sections below.

5.2.1 Proprioceptive and Adaptive Compliance Control

The approach described in this thesis has given evidence of the performance of a hybrid leg-wheel system using a control approach that adapts to the ground sensed by the robot. It has been shown how stairs can be sensed based on proprioceptive data only. It is an interesting research field to combine this locomotion approach with exteroceptive sensing on a higher level. The compliance is currently based only on the torque of the motors and the IMU inside the body of the robot. Perceiving the environment (such as stairs or a field of rock, gravel, or sand) could possible improve the adaptation of the walking gaits. If, for instance, difficult and rugged terrain is sensed by the robot, the robot could autonomously move more slowly and be more adaptive than on flat terrain. On the other hand, the proprioceptive data provided by the robot can enhance the perception of the environment, such as classifying the type of ground on which the system is moving.

5.2.2 Climbing Robots for Marine Vessel Inspection

Many improvements have already been discussed in the corresponding papers. An advantage for the system would be an increase in its autonomy. At present, the robot is guided manually while its position is tracked. The localization system proposed in this thesis provides already a key requirement for autonomy. The problem is its local sensing capabilities due to the weight limits. An improved version with higher payload could carry more processing power and also additional range finding sensors. Another option would be the use of external sensing capabilities which are placed inside the hold. Using 3D reconstruction of a hold in combination with the localization system could realize local path planning while climbing a vertical bulkhead. It would be possible to implement autonomous weld seam following, without high processing power. The localization unit could be significantly improved by using a zoom camera, providing higher localization precision at larger distances. Regarding the spatial content management system described, the automatic generation of 3D CAD models based on point cloud data would significantly improve the system. At present, the 3D representation has to be provided manually for the cargo hold.

5.2.3 Semantic Perception

The two perception approaches described in this thesis are both based on the extraction of spatial features and the analysis of the spatial relations between the extracted entities. Mainly planar features and cluster analysis have been used to describe the spatial features extracted by the region growing method and by the RanSaC method described in this thesis. A significant improvement to this approach would be the detection of free-form shapes. Free-form shapes, such as household objects (e.g., cups, bottles, and cutlery) as well as stones or rocks, cannot be described with geometric primitives. These features have to be trained in a separate classification module. This could be provided, for instance, by learning methods. Another challenge is the detection of compound objects. These objects are composed of basic shapes or geons which are organized in a defined spatial relation to form one object. Another research direction is the learning of a spatial ontology. At the current stage, the object database has to be set up by an expert with the appropriate domain knowledge. As an ontology-based approach has been used, semantic web information could be included in the ontology. This

would also allow sharing of domain knowledge between robots. On the other hand, knowledge about the domain could be updated by the robot. Finally, semantic reasoning with spatial objects can be significantly enhanced by taking into account not only the spatial domain, but also the spatial–temporal domain. This would contain any change in the spatial relations over time. For instance, a door could be classified not only by its shape and orientation towards a wall, but also by the change of angular orientation if the door is opened. This would also require inclusion of the interaction between the robot and the environment.

Bibliography

[1] A. Babu, S. Joyeux, J. Schwendner, and F. Grimminger. Effects of Wheel Synchronization for the Hybrid Leg-Wheel Robot Asguard. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, pages 370–377, Sapporo, Japan, 2010.

[2] S. Bartsch, T. Birnschein, F. Cordes, D. Kühn, P. Kampmann, J. Hilljegerdes, S. Planthaber, M. Römmermann, and F. Kirchner. Spaceclimber: Development of a six-legged climbing robot for space exploration. *ISR/ROBOTIK*, pages 1–8, June 2010.

[3] S. Bartsch and F. Kirchner. Robust control of a humanoid robot using a bio-inspired approach based on central pattern generators, reflexes, and proprioceptive feedback. In *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 1547–1552, 2006.

[4] P. J. Besl and H. D. Mckay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.

[5] M. Bibuli, G. Bruzzone, G. Bruzzone, M. Caccia, M. Giacopelli, A. Petitti, and E. Spirandelli. MARC: Magnetic Autonomous Robotic Crawler Development and Exploitation in the MINOAS Project. In *Proc. 11th International Conference on Computer and IT Applications in the Maritime Industries, COMPIT 2012*, 2012.

[6] O. Birbach and U. Frese. A Precise Tracking Algorithm Based on Raw Detector Responses and a Physical Motion Model. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2013)*, pages 4746 – 4751, Karlsruhe, Germany, 2013. IEEE.

[7] D. Calisi, D. Nardi, K. Ohno, and S. Tadokoro. A semi-autonomous tracked robot system for rescue missions. In *SICE Annual Conference, 2008*, pages 2066–2069, 2008.

[8] CGAL2013. CGAL: Computational Geometry Algorithms Library. http://www.cgal.org, 2013. [Date of access: 16/06/2013].

[9] L. Christensen, N. Fischer, S. Kroffke, J. Lemburg, and R. Ahlers. Cost-Effective Autonomous Robots for Ballast Water Tank Inspection. *Journal of Ship Production and Design*, 27(3):127–136, 2011.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):1052—-1067, 2007.

[11] ECA-Robotics. ROVING BAT - Hybrid free-flying ROV and crawler. http://www.eca-robotics.com/en/robotic-vehicle/robotics-naval-rov-roving-bat-hybrid-free-flying-rov-and-crawler/216.htm, 2013. [Date of access: 16/06/2013].

[12] J. Fletcher. Electronic inspection data management for salt water ballast tanks. In *Marine Coatings Conference 2008*, Hamburg, Germany, 2008. Elcometer.

[13] J. Fletcher and R. Kattan. A Ballast Tank Coating Inspection Data Management System. In *CORROSION 2009*, Atlanta, GA, 2009. NACE International.

[14] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, Lorenz, Meier, P. Tanskanen, and M. Pollefeys. Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV. In *Intl. Conf. Intell. Robots and Systems*, 2012.

[15] Germanischer Lloyd. GL Hull Manager. http://www.gl-maritime-software.com/gl-hullmanager.php, 2013.

[16] M. Goeller, A. Roennau, A. Gorbunov, G. Heppner, and R. Dillmann. Pushing around a robot: Force-based manual control of the six-legged walking robot LAURON. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on Conference on*, pages 2647–2652, 2011.

[17] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448, Barcelona, Spain, 2005. IEEE.

[18] S. Grzonka, G. Grisetti, and W. Burgard. A Fully Autonomous Indoor Quadrotor. *Trans. Robotics*, 28(1):90–100, 2012.

[19] J.-s. Gutmann, P. Fong, L. Chiu, and M. E. Munich. Challenges of Designing a Low-Cost Indoor Localization System Using Active Beacons. In *International Conference on Technologies for Practical Robot Applications (TePRA 2013)*, Boston, MA, 2013. IEEE.

[20] S. Herbert, A. Drenner, and N. Papanikolopoulos. Loper: A quadruped-hybrid stair climbing robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on Robotics and Automation*, pages 799–804, 2008.

[21] M. Hildebrandt, C. Gaudig, L. Christensen, S. Natarajan, P. M. Paranhos, and J. Albiez. Two years of Experiments with the AUV Dagon - a Versatile Vehicle for High Precision Visual Mapping and Algorithm Evaluation. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles. IEEE/OES Autonomous Underwater Vehicles (AUV-2012), September 24-27, Southampton, United Kingdom*. o.A., 2012.

[22] C. Hudelot, J. Atif, and I. Bloch. Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, 159(15):1929–1951, Aug. 2008.

[23] Jetstream. Jetstream Magnetic Crawler M250. http://www.jetstreameurope.co.uk/products/magnetic-crawler.html, 2013. [Date of access: 16/06/2013].

[24] M. Jung, D. Schmidt, and K. Berns. Behavior-Based Obstacle Detection and Avoidance System for the Omnidirectional Wall-Climbing Robot CROMSCI. In H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, editors, *Emerging Trends in Mobile Robotics*, pages 73–80, Nagoya, Japan, 2010. World Scientific Publishing Co. Pte. Ltd.

[25] L. P. Kalra, J. Guf, and M. Meng. A Wall Climbing Robot for Oil Tank Inspection. In *Proceedings of the International Conference on Robotics and Biomimetics (ROBIO)*, pages 1523–1528, 2006.

[26] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE Computer Society, 2004.

[27] K. Konolige, M. Agrawal, R. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor Mapping and Navigation Using Stereo Vision. In O. Khatib, V. Kumar, and D. Rus, editors, *Experimental Robotics*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 179–190. Springer Berlin Heidelberg, 2008.

[28] P. Kumar and P. M. Pathak. Dynamic Modeling, Simulation and Velocity Control of Rocker-Bogie Rover for Space Exploration. *Int. J. Intell. Mechatron. Robot.*, 1(2):27–41, Apr. 2011.

[29] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based Object Labeling in 3D Scenes. In *IEEE International Conference on on Robotics and Automation*, 2012.

[30] J. Machowinski. *Dynamisches Laufen mit dem System Asguard.* Diploma thesis, University of Bremen, 2009.

[31] E. McCann, M. Medvedev, D. J. Brooks, and K. Saenko. "Off the Grid": Self-Contained Landmarks for Improved Indoor Probabilistic Localization. In *International Conference on Technologies for Practical Robot Applications (TePRA 2013)*, pages 1–6, Boston, MA, 2013. IEEE.

[32] Miko. Steel-Climber. http://www.miko.no/steel-climber/, 2013. [Date of access: 16/06/2013].

[33] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler. Reliable stair climbing in the simple hexapod'RHex'. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2222–2227. IEEE, 2002.

[34] A. Nüchter. *Semantische dreidimensionale Karten für autonome mobile Roboter.* PhD thesis, Bonn, 2006.

[35] H. Oh, D.-Y. Won, S.-S. Huh, D. H. Shim, M.-J. Tahk, and A. Tsourdos. Indoor UAV Control Using Multi-Camera Visual Feedback. *J. Intell. Robotics Syst.*, 61(1-4):57–84, Jan. 2011.

[36] A. Ortiz and Bonnin-Pascual. Detection of Cracks and Corrosion for Automated Ves-

sels Visual Inspection. In *Proceedings of the Catalan Congress on Artificial Intelligence*, volume 220 of *Frontiers in Artificial Intelligence and Applications*, page 111, 2010.

[37] A. Ortiz, F. Bonnin-Pascual, and E. Garcia-Fidalgo. On the Use of UAVs for Vessel Inspection Assistance. In *Proceedings of the 1st Workshop on Research, Education and Development on Unmanned Aerial Systems, Seville (Spain), Nov 30th - Dec 1st*, pages 71–80, 2011.

[38] F. Ortiz, J. A. Pastor, B. Alvarez, A. Iborra, N. Ortega, D. Rodriguez, and C. Conesa. Robots for hull ship cleaning. In *2007 IEEE International Symposium on Industrial Electronics*, volume 00, pages 2077–2082. IEEE, June 2007.

[39] F. Ortiz Albertoand Bonnin-Pascual, E. Garcia-Fidalgo, and J. P. Beltran. A Control Software Architecture for Autonomous Unmanned Vehicles Inspired in Generic Components (I). In *Proc. 19th Mediterranean Conf. Control and Automation*, 2011.

[40] D. Pizarro, M. Marron, D. Peon, M. Mazo, J. C. Garcia, M. a. Sotelo, and E. Santiso. Robot and obstacles localization and tracking with an external camera ring. In *2008 IEEE International Conference on Robotics and Automation*, pages 516–521, Pasadena, CA, May 2008. IEEE.

[41] R. Quinn, D. Kingsley, J. Offi, and R. Ritzmann. Improved mobility through abstracted biological principles. In *IEEE Int. Conf. On Intelligent Robots and Systems (IROS'02)*, 2002.

[42] ROTIS2. ROTISII: Publishable Final Activity Report. http://cordis.europa.eu/documents/documentlibrary/124772341EN6.pdf. [Date of access: 10/12/2012].

[43] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[44] J. Schwendner and S. Joyeux. Self localisation using embodied data for a hybrid leg-wheel robot. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Phuket, 2011.

[45] Sertica. Inspection module, iPad. http://www.sertica.dk/da-DK/Forside-UK/Solutions/Marine/iPad.aspx, 2013. [Date of access: 03/05/2013].

[46] J. Shang, B. Bridge, T. Sattar, S. Mondal, and A. Brenner. Development of a climbing robot for inspection of long weld lines. *Industrial Robot: An International Journal*, 35(3):217–223, 2008.

[47] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet. Innovative design for wheeled locomotion in rough terrain. *Robotics and Autonomous Systems*, 40(2-3):151–162, 2002.

[48] D. Spenneberg and F. Kirchner. *The bio-inspired Scorpion robot: design, control & lessons learned*. I-Tech Education and Publishing, Wien, Austria, 2007.

[49] D. Spenneberg, A. Strack, H. Zschenker, and F. Kirchner. Reactive Leg Control for Four-

Legged Walking Based on CPG and Reflex Models. In *Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, 2006.

[50] U. Straccia. Towards spatial reasoning in fuzzy description logics. *2009 IEEE International Conference on Fuzzy Systems*, pages 512–517, Aug. 2009.

[51] K. Tanneberger and A. Grasso. MINOAS Deliverable (D1): Definition of the Inspection Plan / Definition of Acceptance Criteria. http://minoasproject.eu/excibition/Publications/PDF/D1.pdf, 2011.

[52] S. Thrun. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429, May 2006.

[53] P. M. Vaidya. An optimal algorithm for the all-nearest-neighbors problem. In *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 117—-122, Washington, DC, USA, 1986. IEEE Computer Society.

[54] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga. Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*, pages 1–6. IEEE Press, 2007.

[55] Vicon. Vicon tracking systems. http://www.vicon.com/products/vicontracker.html, 2013. [Date of access: 19/06/2013].

[56] VideoRay LLC. Video Ray. http://www.videoray.com/, 2013. [Date of access: 16/06/2013].

Additional Peer-reviewed Publications by the Author

[BBC+11] Marco Bibuli, Gabriele Bruzzone, Massimo Caccia, Alberto Ortiz, Markus Eich, and Efthymios Kolyvas. The MINOAS Project: Marine INspection Robotic Assistant System. In *Proc. 19th Mediterranean Conf. Control and Automation*, 2011.

[CRB+10] M. Caccia, R. Robino, W. Bateman, M. Eich, A. Ortiz, L. Drikos, A. Todorova, I. Gaviotis, F. Spadoni, and V. Apostolopoulou. MINOAS a marine inspection robotic assistant: System requirements and design. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles. IFAC Symposium on Intelligent Autonomous Vehicles (IAV-10)*, Lecce, Italy, 2010.

[EDK10] Markus Eich, Malgorzata Dabrowska, and Frank Kirchner. 3D Scene Recovery and Spatial Scene Analysis for Unorganized Point Clouds. In *Proceedings of CLAWAR'2010: 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 21–28, Nagoya, 2010. World Scientific Publishing Co. Pte. Ltd.

[EGK08] Markus Eich, Felix Grimminger, and Frank Kirchner. Adaptive Stair-Climbing Behaviour with a Hybrid Legged-Wheeled Robot. In L. Marques, A. De Almeida, M. O. Tokhi, and G. S. Virk, editors, *Advances in Mobile Robotics. 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 768–775, Coimbra, 2008. World Scientific Publishing Co. Pte. Ltd.

[EGK09] Markus Eich, Felix Grimminger, and Frank Kirchner. Proprioceptive control of a hybrid legged-wheeled robot. In *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008*, pages 774–779. Ieee, February 2009.

[EHK+14] Markus Eich, Ronny Hartanto, Sebastian Kasperski, Sankaranarayanan Natarajan, and Johannes Wollenberg. Towards Coordinated Multi-Robot Missions for Lunar Sample Collection in Unknown Environment. *Journal of Field Robotics*, to appear, 2014.

[EK10] Markus Eich and Frank Kirchner. Reasoning About Geometry: An Approach Using Spatial-Descriptive Ontologies. In Kerstin Schill, Bernd Scholz-Reiter, and Lutz Frommberger, editors, *In Workshop Artificial Intelligence and Logistics (AILog), (ECAI-10)*, Lisbon, Portugal, 2010.

[EKF04] Markus Eich, Markus Kemper, and Sergej Fatikow. A Navigation System Concept

for an Indoor Micro Air Vehicle. In *First European Micro Air Vehicle Conference and Flight Competition*, Braunschweig, Germany, 2004. Deutsche Gesellschaft für Ortung und Navigation.

[KES07] Frank Kirchner, Markus Eich, and Dirk Spenneberg. Surveillance of Security-Critical Compounds and Buildings Using a Team of Autonomous Robots. In Jürgen Beyerer, editor, *Future Security. 2nd Security Research Conference (FuSec-2007)*, pages 143–147, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.

[KWE13] Sebastian Kasperski, Johannes Wollenberg, and Markus Eich. Evaluation of cooperative exploration strategies using full system simulation. In *Proceedings of the 16th International Conference on Advanced Robotics (ICAR 2013).*, Montevideo, Uruguay, 2013. IEEE Press.

[LDE+11] Johannes Lemburg, José De Gea, Markus Eich, Dennis Mronga, Peter Kampmann, Andreas Vogt, Achint Aggarwal, Yuping Shi, and Frank Kirchner. AILA - Design of an autonomous mobile dual-arm robot. In *ICRA 2011*, 2011.

[OBG+10] Alberto Ortiz, F Bonnin, A Gibbins, P Apostolopoulou, William Bateman, Markus Eich, F Spadoni, Massimo Caccia, and Leonidas Drikos. First steps towards a roboticized visual inspection system for vessels. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–6. IEEE, 2010.

[OBPG+10] Alberto Ortiz, Francisco Bonnin-Pascual, Andrew Gibbins, Panagiota Apostolopoulou, William Bateman, Markus Eich, Francesco Spadoni, Massimo Caccia, and Leonidas Drikos. First Steps towards a Roboticized Visual Inspection System for Vessels. In *Proc. of the IEEE Emerging Technologies and Factory Automation conference*, 2010.

[RAME11] Marc Ronthaler, Achint Aggarwal, Dennis Mronga, and Markus Eich. AILA - ein Dual-Arm Roboter für die Logistik. *Industrie Management - Zeitschrift für industrielle Geschäftsprozesse*, 01:35–38, 2011.

[SEN+13] Peter Stephan, Markus Eich, Jörg Neidig, Martin Rosjat, and Roberto Hengst. Applying Digital Product Memories in Industrial Production. In Wolfgang Wahlster, editor, *SemProM: Foundations of Semantic Product Memories for the Internet of Things*, pages 283–304. Springer Berlin / Heidelberg, 2013.

[VESK09] Andreas Vogt, Markus Eich, Renato Samperio, and Frank Kirchner. A Practical Interaction Between Robots and RFID-based Digital Product Memories in a Logistic Scenario. In *TEPRA2009*, Woburn, Massachusetts, 2009.

List of Accumulated Publications

[EBPGF+14] Markus Eich, Francisco Bonnin-Pascual, Emilio Garcia-Fidalgo, Alberto Ortiz, Gabriele Bruzzone, Yannis Koveos, and Frank Kirchner. A Robot Application to Marine Vessel Inspection. *Journal of Field Robotics*, to appear, 2014.

[EDK10] Markus Eich, Malgorzata Dabrowska, and Frank Kirchner. Semantic Labeling: Classification of 3D Entities Based on Spatial Feature Descriptors. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, Alaska, USA, 2010. http://www.best-of-robotics.org/pages/BRICS-events/icra2010/SemanticLabeling2010Eich.pdf.

[EGBS08] Markus Eich, Felix Grimminger, Stefan Bosse, and Dirk Spenneberg. Asguard : A hybrid legged-wheel security and sar-robot using bio-inspired locomotion for rough terrain. In *Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment*, Benicàssim (Spain), January 2008. http://www.robot.uji.es/documents/rise08/reports/Eich.pdf.

[EGK08] Markus Eich, Felix Grimminger, and Frank Kirchner. A Versatile Stair-Climbing Robot for Search and Rescue Applications. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2008. SSRR 2008*, pages 35–40. IEEE, October 2008.

[EGK09] Markus Eich, Felix Grimminger, and Frank Kirchner. Adaptive compliance control of a multi-legged stair-climbing robot based on proprioceptive data. *Industrial Robot: An International Journal*, 36(4):331–339, 2009.

[Eic13] Markus Eich. An Application of Fuzzy DL-Based Semantic Perception to Soil Container Classification. In *International Conference on Technologies for Practical Robot Applications (TePRA 2013)*, pages 1–8, Woburn, Massachusetts, USA, 2013. IEEE.

[EV11] Markus Eich and Thomas Vögele. Design and Control of a Lightweight Magnetic Climbing Robot for Vessel Inspection. In *Proc. 19th Mediterranean Conf. Control and Automation*, 2011.

[FEWK12] Kristin Fondahl, Markus Eich, Johannes Wollenberg, and Frank Kirchner. A Magnetic Climbing Robot for Marine Inspection Services. In Volker Bertram, editor, *11th International Conference on Computer and IT Applications in the Maritime Industries*, pages 92–102, Liege, Belgium, April 2012. Technische Universität Hamburg-Harburg.

[VEGF10] Thomas Vögele, Markus Eich, Felix Grimminger, and Kristin Fondahl. A Hybrid Legged Wheel Climbing Robot for Marine Inspection. In *In Emerging Trends in Mobile Robotics, CLAWAR-2010*, 2010.

Individual Contribution to the Listed Publications

In this section the individual contributions of the author of this thesis are pointed out. A short summary of the contributions are given as well as an estimate of the personal share in each publication.

Publication	Individual Contribution	Personal Share
[VEGF10]	I wrote the paper, except for the introduction. I developed the idea of using lightweight robots for cargo hold inspection. The physical system was built by the 3rd and 4th authors.	75%
[EV11]	I wrote the paper, except for the introductory section. I designed the overall system concept of the lightweight inspection system and the localization and human interaction concept.	90%
[FEWK12]	I was mainly involved in the acquisition of the experimental data during the ship trials and the development of the system architecture. I also developed the idea of the spatial content management system and for the inspection data representation. The localization method was also part of my personal contribution	50%
[EBPGF+14]	I completely wrote Sections 1, 2, 3.1, 3.2.2, 3.4, 5, 6.2, 6.5, and 7. I developed all methods and performed the experiments of the mentioned sections. These corresponding sections deal with the lightweight crawler, the robot localization technique, and the spatial content management system for the representation of the inspection data.	50%

Table 1: Individual contributions related to *Marine Vessel Inspection*

Publication	Individual Contribution	Personal Share
[EGBS08]	I wrote the paper except for Section 3.3, where the FPGA layout is described. I developed the control software of the robot and conducted all experiments.	75%
[EGK08]	The paper is a significant extension of [EGBS08]. I wrote the entire paper, developed the control architecture, and conducted the experiments. My contribution was every part of the system which is software related. The second author developed the physical hardware used in the experiments. The paper won the Best Paper Award at the SSRR 2008 conference.	90%
[EGK09]	This paper is an extension of [EGK08] which was recommended at the CLAWAR 2008 conference to be published in the *Industrial Robot Journal*. I wrote the entire paper, developed the control architecture, and conducted the experiments. The second author developed the physical platform which was needed for the experiments.	90%

Table 2: Individual contributions related to *Proprioceptive Locomotion Control*

Publication	Individual Contribution	Personal Share
[EDK10]	I wrote the entire paper. I developed the concept of using spatial features for classification of the spatial entities. The second author was a student I was supervising at this time, who supported the paper by implementing parts of the region growing algorithm.	90%
[Eic13]	This work was written by me as a single author. I developed the approach of using FuzzyDL for spatial object classification, implemented the software, and conducted the experiments.	100%

Table 3: Individual contributions related to *Semantic Perception*

Accumulated Publications

ASGUARD: A Hybrid Legged-Wheel Security and SAR-Robot Using Bio-Inspired Locomotion for Rough Terrain

Markus Eich, Felix Grimminger, Stefan Bosse, Dirk Spenneberg, Frank Kirchner
Robotics Lab
German Research Center for Artificial Intelligence (DFKI)
28359 Bremen, Germany

Abstract

For outdoor locomotion of mobile robots, one has to cope with different requirements for such systems. These robots have to be highly agile and fast on flat ground and, at the same time, should be able to deal with very rough terrain, e.g. rubble, gravel, and even stairs. This is particularly true for robots which are used for surveillance and search and rescue missions (SAR) in outdoor environment as well as for robots for remote inspection, such as CBRNE detection in crises situations. Tracked robots are currently regarded as the best trade-off between high velocity and mobility in rough terrain. Those systems have the drawback of high energy consumption due to friction and are generally not able to climb stairs or very steep slopes. In this paper we present a hybrid legged-wheel approach which is used for our outdoor security robot ASGUARD. The robot has four rotating wheels in the shape of a five-pointed star. Each point of the star serves as a leg during locomotion and is controlled using bio-inspired central pattern generators (CPGs). We will show in this work that with our approach, the robot can handle high velocities, is able to climb stairs, and can cope with piles of rubble without changing the hardware configuration.

1 Introduction

In this paper we present our fast and highly agile quadruped robot ASGUARD[1](cf. Figure 1). The robot was designed to be used in harsh outdoor environment with a focus on security and outdoor surveillance as well as on disaster mitigation missions. For those applications, a robot has to transport a variety of mission-depending application sensors inside a difficult terrain. Those missions are commonly named "Three D" missions. "Three D" stands for dull, dirty, and dangerous and implies, e.g., tasks where rescue personnel must enter a collapse-endangered building in search for injured people, the acquisition of samples in contaminated areas or patrolling every day along the same fence of a security-relevant compound. For all those applications, the requirements for such an in-situ system are that it has to deal with obstacles or uneven and difficult outdoor terrain. Additionally, the robot has to move fast where the ground is leveled and easier to cross. In order to cope with those two requirements, the quadruped robot ASGUARD was designed, which makes use of multiple rotating legs along one hip shaft.

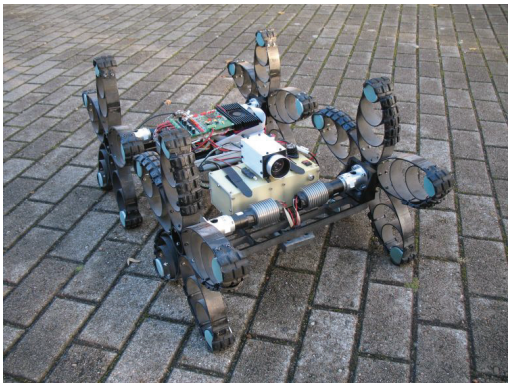


Figure 1: The quadruped robot ASGUARD

In the last few years, some work regarding hybrid legged wheel locomotion approaches has been done. Sometimes in literature referred to as compliant legs [11] or spoked wheel [10], this approach makes often use of a very simple and therefore very robust locomotion principle. The key idea is to use one rotating actuator for driving one or more simple legs around one axis.

[1] Advanced Security Guard

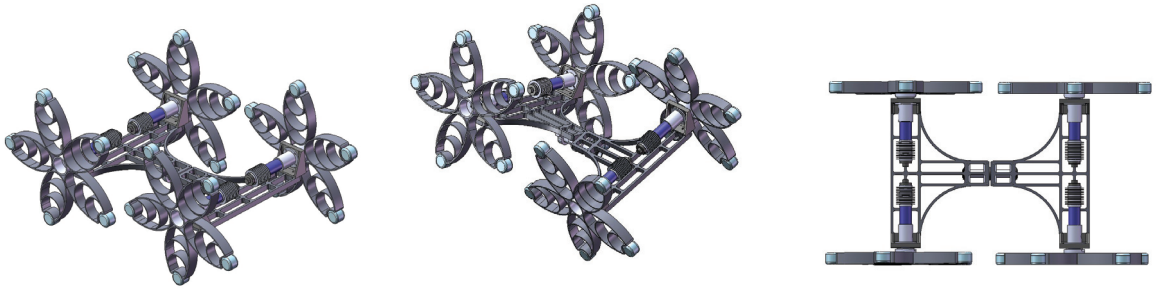
In [11] the hexapod RHex is described. The robot uses one rotating compliant leg per actuator and is able to ascend and descend stairs. RHex uses a fixed pattern for the trajectory of each leg. The locomotion is performed by a tripod gait, where the "retraction" and "protraction" phases are alternatingly triggered. For the stair climbing behavior of the RHex robot, six phases were defined, based on a fixed transition model [9]. The parameters for the single gait were defined, using empirical analysis. [8] describes an improvement on the prior RHex robot by introducing two additional behaviors based on proprioceptive sensing. One behavior is the adjustment of the compliant legs in direction of the gravity force by using an internal tilt-sensor. The behavior ensures an optimal position in the stance phase of the legs while climbing a slope. The second behavior introduced is the "pronking" controller, enabling a jumping behavior which can be found in hooved mammals, like the springbok. For synchronising, the trajectories, the proprioceptive data from six ground contact sensors are used. Another proprioceptive information of the robot is used in [4], where the motor current is used to detect the contact with a flight of stairs. In this case, the tripod gait changes to a metachronal wave gate. In contrast to our approach, the trajectories of the legs are based on a fixed, hand-adjusted gait configuration. The proprioceptive information is used to trigger the transition between those two gaits.
[10] and [1] use a design of a multi-spoked wheel for their hexapod Whegs, which comes closer to our design of our quadruped Asguard. The bio-inspired mechanical design is derived from an analysis of the cockroach gait. In contrast to other research regarding legged wheel approaches, Whegs uses no sensor information to adjust the tripod gait. It uses only the compliant legs design to adapt to different types of terrain. Whegs uses only one DC motor for locomotion and one servo for active steering.
To our knowledge, all locomotion concepts for hybrid legged wheel approaches are based on fixed motion patterns which are controlled in open-loop manner. In all research we found, only inclination and ground contact were used to change the pattern of the trajectory, which is then again done in open-loop control. In our research we mainly focus on a robust stair-climbing behavior based on a behavior-based closed-loop approach. We will use the information of the measured motor current and the shaft encoders in order to change the behavior of each leg.
The remainder of the paper is arranged as follows: The robot platform Asguard is described in Section 2. The control approach and its implementation in FPGA hardware design is described in Section 3. In Section 4 we present the results of our algorithm, with the robot climbing a flight of stairs. In Section 5 we will discuss our results and give some ideas about our future research direction.

2 Platform Description

The long-term goal of our research is to develop a robust outdoor platform which is suitable to be included in disaster mitigation as well as in security and surveillance missions. The platform should be able to transport application sensors to areas that are dangerous for humans to access, e.g. a collapse-endangered building or an industrial compound after a chemical accident. In those cases, before they enter, the rescue personnel might need some information about the air contamination or the whereabouts of people inside an area. The robot should be upgradeable with a variety of application sensors, e.g. cameras, thermal vision, or chemical sensors. To be usable in any search and rescue or security context, the robot has to be operational without changing batteries for at least two hours. All those requirements were specified with potential end users beforehand.
This defined the minimum size of Asguard, as well as the energy budget and the minimum payload. To be usable for a variety of missions, the robot has to be able to carry sensors to areas which are normally not accessible to wheeled and tracked robots.

2.1 The Physical Robot Asguard

The robot Asguard is a hybrid quadruped outdoor robot which was inspired by insect locomotion, as described in [10] and [1]. The first prototype of our system is driven by four directly actuated legs with one rotational degree of freedom. In Figure 2 three aspects of the robot frame are shown; in Table 2.1 the dimensions and other physical data are given. After testing the ground traction with a rigid corpus, we found out that we could increase ground contact by adding an additional rotational degree of freedom along the body axis, serving as an elastic spinal column (cf. Figure 2(b)). By this we could increase the ground traction significantly. For the low level control of the robot, a custom-designed FPGA motor control board (Motcon6) is used which controls the four motors in a closed-loop manner. The



(a) Isometric view (b) Isotropic view with rotated body (c) Top view

Figure 2: The CAD design of the quadruped robot ASGUARD

Height	44cm
Length	95cm
Width	50cm
Wheelbase	51cm
Weight	11.5kg
Motors	4x Faulhaber 24V DC motors with 46:1 planetary gear
Motor Power	4x 83W
Battery	10Ah/30V, Lithium Polymer Batteries
Battery Weight	1.6kg

Table 1: Physical dimensions of the robot

locomotion of the robot is performed by central pattern generators, which describe the trajectory of each leg within the phase of $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$. The detailed pattern generator approach is described in Section 3 and works as an independent low level controller. With the MOTCON6 board, we are also able to measure the power consumption as well as the position of each leg in real time, providing us with important proprioceptive information about the system.
With the actual motor-gear configuration and with the current weight of the robot, we measured a maximum speed of around $2m/s$ on flat ground. For each leg we can individually define the trajectory of a leg , allowing us to synchronise or asynchronize the legs with each other. We found out that the maximum speed is depending on the used gait and gait transitions during movement. An analysis of different gaits in terms of energy efficiency and velocity will be done in the near future.

2.2 The Multi-Legged Wheel Design

Our design of the legged wheel consists of five compliant legs which are assembled as shown in Figure 3(b). Each individual leg is designed as shown in Figure 3(a). They are arranged around the hip shaft,

(a) Single leg (b) Full legged wheel on stair (c) 3D sketch of the legged wheel

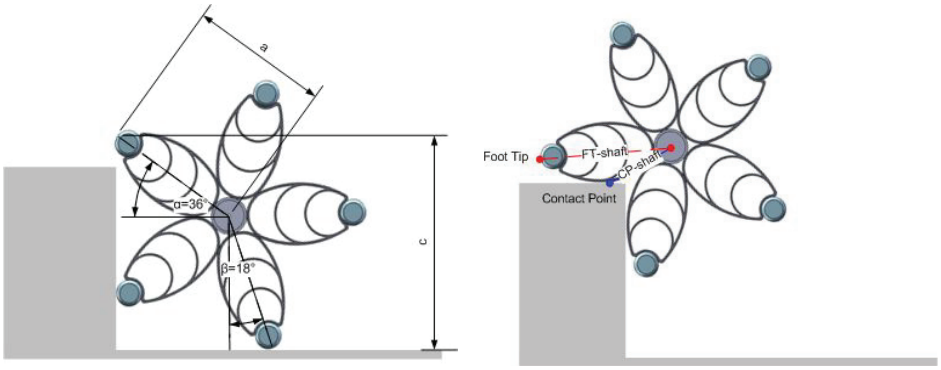
Figure 3: Design of the legged wheel

with an angular distance of $\frac{2\pi}{5}$. Because of the symmetry of the legs, we have only to consider the phase

between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$ (cf. Figure 3(c)). By this configuration we can assure that we have a minimum of four legs on the ground, which ensures a stable configuration of the robot. The outer radius of the legged wheel is $22cm$. The inner radius (i.e. the height of the hip joint shaft, if two legs have ground contact) of the legged wheel is $18cm$. In order to decrease the physical shock during locomotion, shock-absorbing leg tips were used.
As described already in [10], a compliant leg design is able to overcome obstacles which are higher than the wheel axis. A theoretical value for our legged-wheel approach can be calculated by

$$c = a * sin(\alpha) + a * cos(\beta) \tag{1}$$

with a naming the minimum height of the legged wheel. The angles α and β are shown in Figure 4(a). For our current wheel configuration the maximum height of an obstacle that can be overcome is $33.85cm$ for the values of $a = 22cm$, $\alpha = 36°$, $\beta = 18°$. Practically, Asguard can climb on a plateau with a height of $25cm$, which is well above the height of the hip shaft. The limit of $25cm$ is not depending on the wheel design but limited by the robot's overall physiognomy (i.e. wheelbase, center of gravity, etc.). In contrast to that, a wheeled robot would only be able to go on a plateau of a height which is much less than the height of the wheel shaft. While driving with high velocities, only the leg tips have direct



(a) Stair climbing with a hybrid legged wheel (b) Stair climbing with a hybrid legged wheel

Figure 4: Climbing an obstacle with a legged wheel

contact to the ground. In this case, Asguard behaves like a wheeled system, with the radius b (cf. Figure 3(c)), reaching velocities of around $2m/s$. If the robot climbs a flight of stairs (cf. Figure 4(b), an increase of leverage implies more power to climb it, because the ground contact point moves further to the axis of the leg. This is comparable to gear changing, but without any additional actuation.

3 Bio-Inspired Locomotion Control for Legged-Wheel Robots

3.1 Central Pattern Generation in Walking Robots

The concept of using CPGs (Central Pattern Generators) is well known and utilized in the area of ambulating robots. Central Pattern Generators (CPG) are the major mechanisms in animals to control and to produce rhythmic motion. They are characterized by the ability to produce rhythmic motion patterns via oscillation of neuronal activity without the need of sensory feedback [16]. However, sensory feedback is normally integrated into the CPG-control. Mostly load and position data of the controlled limb/joint are fed back into the CPG-network, which is used to implement a closed-loop control of the rhythmic motion of the system actuators. To modulate the controlled rhythmic patterns, the CPG can change its frequency, phase, and amplitude [12]. For the use in robots it is reasonable to develop an abstract CPG model which inherits only the basic principles of the CPG's functionality.
Many different ways to achieve this have been proposed and tested, e.g. [2, 7, 5, 17]. In [15] our approaches for controlling walking robots are described in detail. Our model consists of a controller module (using a PID-controller) and a unit to produce rhythmic trajectories in the joint angle space. To

produce the rhythmic trajectories, we describe a CPG-pattern as a function of part-wise fitted together third-order Bezier polynomial (for details see [13]). This CPG-approach allows very easy adaptation of rhythmic walking patterns as well as the overlaying of different CPG-patterns. For example, our approach enables the Scorpion robot to walk omni directional by overlaying a forward and a lateral rhythmic walking pattern[14]. Together with an implemented posture and reflex module, our walking robots are able to walk very robust in an adaptive way through rough terrain.
Thus, implementing this control approach on a hybrid legged wheel system looks very promising. First steps in that direction have already been done, which are described in the next section.

3.2 Using CPGs for Hybrid Legged Wheel Approach

In order to control our robot ASGUARD, we are facing two control requirements. On one hand, we have to control the velocity, i.e. the rotational speed of each of the legs. On the other hand, we have to control the position of each leg of the quadruped robot. For controlled obstacle and stair climbing we have to assure certain positions over time. From the CPG control methods, used in a variety of walking machines (cf. Section 3.1), we learned an efficient approach to control such systems by using trajectories in the time-angle space. In contrast to many pure legged robots, which have generally more than one degree of freedom for each leg, we have only one angle to control over time. As described in Section 2.2, we only have to consider the angular space between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$. The patterns are generated by our custom-designed FPGA-based MOTCON6 board. From our high-level controller we can modify the pattern parameters by changing the pattern frequency, the direction of the generated pattern as well as a phase offset. By this phase offset we can change the stance phase, i.e. the time in which a leg has ground contact. Figure 3.2 shows the typical trajectory of the front left leg in time-angle space. Because we use more than one leg per wheel, we can make use of a simple saw tooth pattern in time-angle space. The actual velocity of the robot is controlled by the frequency of the trajectory in the internal control loop. A P-Controller, which is implemented in MOTCON6 control board, is used to follow the trajectory generated by the internal pattern generator.
In Figure 3.2 PAR gives the actual trajectory, the variable PSR names the target trajectory. The figure shows also the error between PAR and PSR as well as the power consumption of the motor.
An advantage of our design is that we can modify the amplification factor of the proportinal control term of the inner control loop at runtime. By changing those parameters on-line, we allow a larger error between the current leg position and the target trajectory. This is an important factor because we are using the proprioceptive information of the applied force to change those parameters. For left/right control of the robot, we use a differential steering approach by individually controlling the speed and the direction of movement for each side of the robot. By this, we save additional actuators for steering, thus saving weight and energy.

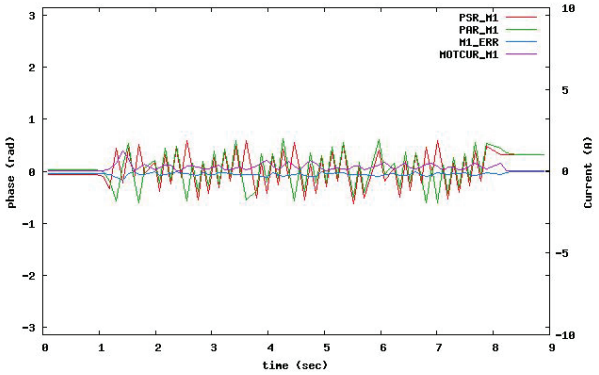


Figure 5: Proprioceptive data of the legs on flat ground

3.3 Implementing Timed Motion Pattern in FPGA

The low level control concept of the ASGUARD robot is centered around a newly developed electronic control board named MOTCON6. This control board features autonomous position and motion control for up to six independent DC actuators. The power electronics can drive DC motors within a voltage range of 12-35V and can deliver 4(8)A current continuously as well as 80A peak current for each motor with high efficiency around 99%. To control DC-motor driven legs, several functional blocks are required. These are (cf. Figure 6):

- Power electronics for driving the electrical motors
- Data acquisition of sensor signals
- Position control for each motor
- Limited protection (leg position, peak current values, power consumption, and temperature)
- Communication between the low level control and higher control levels

Several special purpose peripheral circuits are required to interface the power electronics, the data acquisition blocks, and the communication blocks. Generic microcontrollers deliver some of the required peripherals, but not all. Therefore, customized digital logic featuring system-on-chip architecture is preferred. For rapid prototyping, Field Programmable Gate Arrays (FPGA) are best suited for this

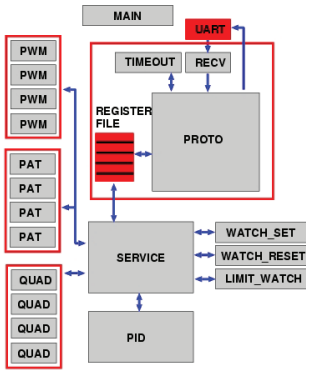


Figure 6: Functional blocks implemented with concurrent processes required for autonomous robot actuator control.

purpose. For reliability and for maintenance reasons, a motor control unit should be independent from a master controller, like a microcontroller or an embedded PC. Another important design aspect is power efficiency, regarding the power electronics and the power consumption of the analog and digital parts required for operation. The communication and the link technology must be independent of the target robot architecture and the main controller architecture, and must provide fault tolerance and reliability. The MOTCON6 board uses a register bank memory for data exchange between the controller and the host controller. The registers are read and modified by messages. All registers are mapped to controller parameters and sensor data tables. The control architecture is partitioned and modularized into different system blocks supporting independent leg actuator control, implemented with concurrent processes (see Figure 6):

1. Pulse-width-modulated (PWM) signal generators delivering drive signals for the H-motor-bridge power electronics with 8 bit pulse-width resolution
2. ADC device drivers providing an interface to three external ADC devices with 12 bit data resolution providing a total of 24 channels (not shown)
3. Quadrature signal decoder (rotational angle position measurement) with 12 bit resolution per turn and additional turn counter (QUAD)

4. Service controller for synchronizing processes and managing access to the 256 registers of (SERVICE). The register bank contains all parameters required for actuator control and all acquired sensor data.

5. Communication controller (PROTO) implementing a simple and fault-tolerant communication protocol stack, following a connectionless text-message based approach, independent of the underlying serial link controller.

6. Serial link controller unit (UART) attached to the communication controller.

7. The position controller for each actuator (PID), implemented using an optimized and fully developed Proportional-Integral-Differential term controller with 12 bit data resolution and 1kHz loop frequency.

8. Limit observation and regulation (LIMIT), for example over-current detection.

9. Parameterizable pattern generators (PAT) generating leg position patterns. The pattern leg frequency, direction, and a phase offset can be specified with parameters changeable during runtime.

All modules are connected using a System-On-Chip architecture and either shared register or queues or channels for data exchange, and mutexes or semaphores for interprocess communication.
The target technology used on the MOTCON6 board is a Xilinx Spartan-III FPGA with an estimated gate count of approx. 1M gates, with the highly optimized PID controller, the communication and main control blocks fitting well into the FPGA resources. The hardware design occupies 50 % of slice resources (about 300k equivalent gate count), longest path estimation predicts a maximal clock frequency of 140 MHz (20 MHz actually used), 25 FSMs were inferred. The hardware design was made using an imperative multi-process approach on system level (CONPRO silicon compiler, [3]), (about 1500 lines source code), synthesizing pure Register-Transfer-Logic and VHDL code (about 15000 lines), further synthesized to gate level with Xilinx ISE software.
The implemented leg position controller is a compact version of a traditional PID-controller. The input data is derived from the actual and past angular leg position sensor signal, and the scaled output signal U from the controller is fed directly into the PWM generators driving the motor voltage using highly efficient H-Bridge technology. The control function delivers three terms: a proportional, an integral, and a differential function term.
The actual measured leg position is P_A, the target leg position is P_D. The actual error value is the difference of these two values. An enhanced recursive and compact version of the controller algorithm was derived for efficient synthesis into digital logic concerning hardware resources and latency. Therefore, there is only one equation built from a linear superposition of the three terms. The control parameters K_P for the proportional term, K_I for the integral term, and K_D for the differential term must be transformed into the parameters K_0, K_1, and K_2 using the equation shown below, derived from [6].

$$U(n) = U(n-1) + \Delta U(n) \tag{2}$$
$$\Delta U(n) = K_0 E(n) - K_1 E(n-1) + K_2 E(n-2) \tag{3}$$
$$E(n) = P_D - P_A \tag{4}$$
$$K_0 = K_P + K_I + K_D \tag{5}$$
$$K_1 = K_P + 2K_D \tag{6}$$
$$K_2 = K_D \tag{7}$$

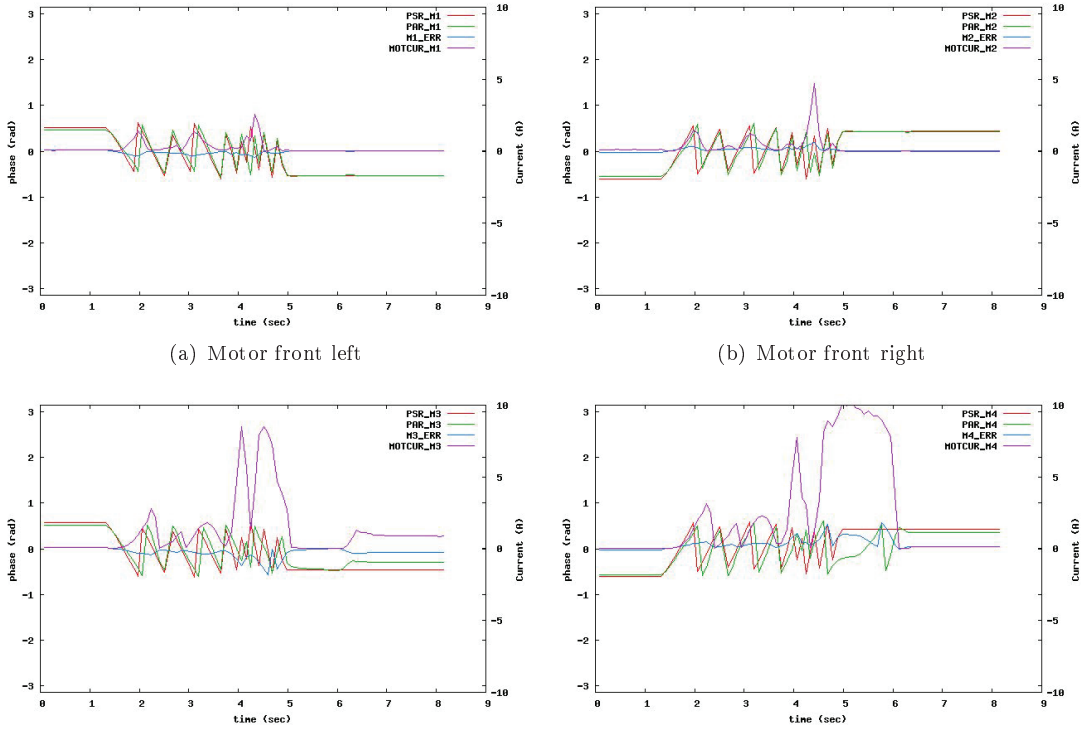
4 Using Proprioceptive Data for Changing Leg Trajectories on Stairs

When climbing a stair, we have to ensure in our control approach that the legs on the front axis are synchronized, i.e. that they have exactly the same pattern phase. The same is valid for the synchronization of the rear legs. An optimal behavior in climbing a stair would be to keep the tilt angle of the whole robot minimal and that all motors have more and less the same power consumption. By keeping the left and right wheels at the same phase through our CPG approach, on a stair, we can assume that the

energy consumption on the left and on the right side of each axis is about the same, given that we have the same motor configuration, mechanical friction within the system, and the weight distribution along the body axis.
In order to to distribute the torque of each actuator, which is directly related to the measured motor current, we use an approach to modify the amplification of the P-Factor within the internal control loop, which is responsible for following the trajectory of the generated pattern (cf. Equation 8).

$$O_i = (\kappa_i - (cur_i - \frac{\sum cur}{n}) * \iota_i) * (PAR_i - PSR_i) \tag{8}$$

O refers to the motor output of the controller; PAR and PSR name the actual position of the hip encoders and of the target position, respectively. The measured motor current in A (cur) we assume to be in the interval $[0, 10]$ for each motor. The constants κ and ι are used to map the P-Factor within the limits of our P-Controlled, i.e. the interval $[20, 100]$.
For our test setup, we empirically determined the value for $\kappa = 50$ and $\iota = 5.5$. To show the effect of our control approach, we logged several runs on a flight of stairs with a step hight of $18cm$, a step size of $27cm$, and an inclination of $34°$ (75%). Figure 7 shows the data of a stair climbing run without our adaptive controller. Figure 8 shows our approach with a current adaptive control, using Equation 8.



(a) Motor front left

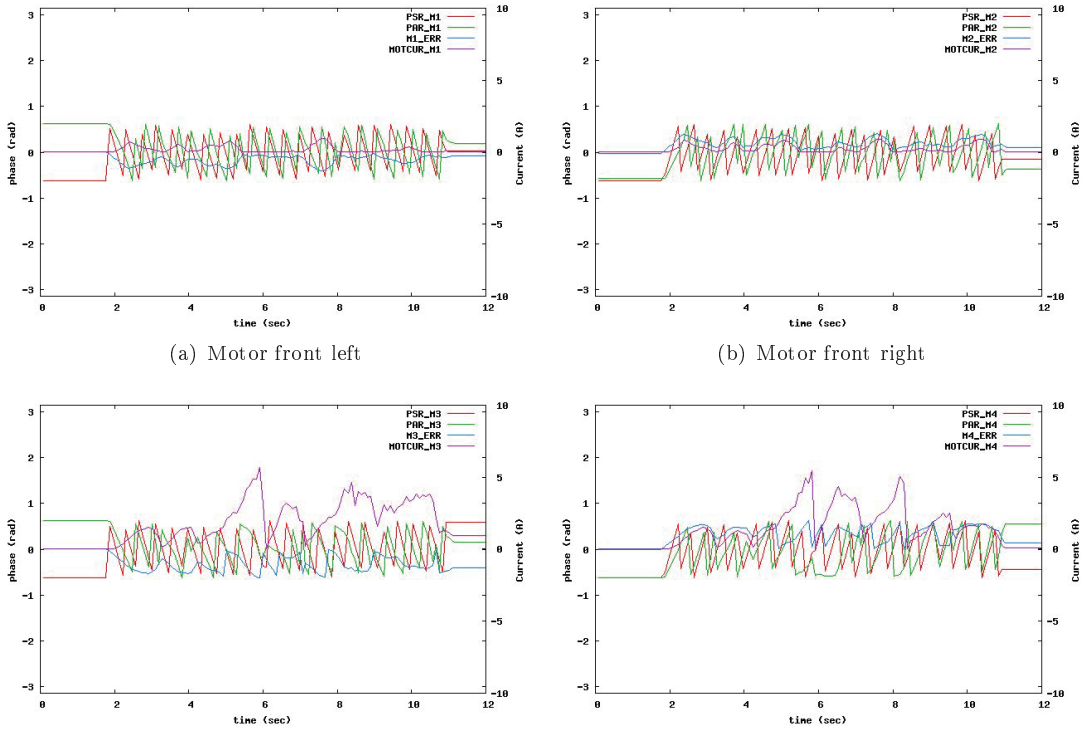
(b) Motor front right

(c) Motor rear left

(d) Motor rear right

Figure 7: Proprioceptive data of the legs on stairs without adaptive control. The robot did a backflip on the stairs before reaching the landing.

What can be seen in those figures is that we could reduce the load on the rear legs (motor 3 and 4) significantly. Without the adaptive control, the drawn current reached easily 8-10A. This occurs if the rear legs erect the whole robot on the stairs, causing it to flip backwards. With the adaptive controller, the peak values were around 5A in Figure 8. Because of the actual weight distribution of the robot, we were not able to properly balance the current on each motor, the rear legs had still to carry the main load.
To assess the robustness of our approach, we did ten runs on the same stairs without manual interference. The speed was set to 10% of the maximum speed. The phase offset for each leg was set to zero in the run. By dynamically changing the P-factor of the internal control loop , the rear legs went out of synchronization because a smaller P-factor allows a greater error within the phase. By using our



(a) Motor front left

(b) Motor front right

(c) Motor rear left

(d) Motor rear right

Figure 8: Proprioceptive data of the legs on stairs using adaptive P-factors. The robot reached the landing.

adaptive control approach, the robot was able to climb the stairs in 7 out of 10 trials. In two runs the robot skidded sidewards. This could not be prevented because we did not use any IMUs to measure the unintended sidward skidding. In just one run the robot flipped back while climbing. We are optimistic that we can cover this occasional case by additional sensing of the robot's inclinaton. This aspect will be included in the next step. Without an adaptive control the robot flipped back in almost every run.

5 Conclusion

In this paper we introduced for the first time our quadruped robot ASGUARD which will be usable for "Three D" missions. The robot uses a hybrid legged wheel approach for locomotion which is controlled using bio-inspired pattern generators. Similar pattern generators can be found in a variety of legged animals. We used the proprioceptive data from the actuators to implement an adaptive control mechanism. By this control approach, ASGUARD was able to climb a stair with an inclination angle of 75% at 70% reliability. This could be achieved without any additional exterioproceptive data, like cameras or laserscanners. Additionally, the power consumption of the legs, especially of the rear legs, could be significantly reduced.

In future research we will add more proprioceptive sensors, like IMUs[2] and magnetic field sensors in order to estimate the pose on the stairs. By this we intend to improve our performance and prevent backflips and side skidding on stairs. We showed in this work the potential of a hybrid legged wheel locomotion and that our system using this approach is able to run at high speed as well as to overcome obstacle and climb stairs.

Another focus regarding our quadruped robot is the analysis of different walking patterns in terms of energy consumption and forward speed. Also the gait transition between different patterns needs some deeper analysis because we are using differential control for left/right steering. Therefore we must change the frequencies of the CPGs while maintaining the phases of the legs. Regarding the mechanical design,

[2]Inertial Measurement Unit

we are currently working on an improved dust and waterproof version of ASGUARD in order to make the system usable under real outdoor conditions.

References

[1] T. Allen, R. Quinn, R. Bachmann, and R. Ritzmann. Abstracted biological principles applied with reduced actuation improve mobility of legged vehicles. In *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003), Volume 2*, pages 1370–1375, 2003.

[2] J. Ayers. A conservative biomimetic control architecture for autonomous underwater robots. In Ayers, Davis, and Rudolph, editors, *Neurotechnology for Biomimetic Robots*, pages 241–260. MIT Press, 2002.

[3] S. Bosse. Conpro: High level hardware synthesis with an imperative multi-process approach. *to be published*, 2007.

[4] G. C. Haynes and A. Rizzi. Gaits and gait transitions for legged robots. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, May 2006.

[5] A. Ijspeert and J. Kodjabachian. Evolution and development of a central pattern generator for the swimming of a lamprey, 1998.

[6] R. Isermann. *Digital Control Systems: Fundamentals, Deterministic Control*. Secaucus, New Jersey, U.S.A.: Springer Verlag, 1989.

[7] F. Kirchner, D. Spenneberg, and R. Linnemann. A biologically inspired approach towards robust real world locomotion in an 8-legged robot. In J. Ayers, J. Davis, and A. Rudolph, editors, *Neurotechnology for Biomimetic Robots*. MIT-Press, Cambridge, MA, USA, 2002.

[8] H. Komsuoglu, D. McMordie, U. Saranli, D. Moore, M. Buehler, and D. E. Koditschek. Proprioception based behavioral advances in hexapod robot. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, pages pp 3650 – 3655, Seoul, Korea, May 21-26, 2001.

[9] E. Moore and M. Buehler. Stable stair climbing in a simple hexapod robot, 2001.

[10] J. Quinn, R.D.and Offi, D. Kingsley, and R. Ritzmann. Improved mobility through abstracted biological principles. In *IEEE/RSJ International Conference on Intelligent Robots and System, Volume 3*, pages 2652–2657, 2002.

[11] U. Saranli, M. Buehler, and D. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, July 2001.

[12] K. T. Sillar, O. Kiehn, and N. Kudo. Chemical modulation of vertebrate motor circuits. In P. Stein, S. Grillner, A. I. Selverston, and D. Stuart, editors, *Neurons, Networks and Motor Behavior*, pages 183–193. MIT Press, Cambridge, 1997.

[13] D. Spenneberg. A hybrid locomotion control approach. In *In Proceedings of the CLAWAR 2005 Conference*, 2005.

[14] D. Spenneberg and F. Kirchner. Omnidirectional walking in an eight legged robot. In *Proc. of the 2nd Int. Symp. on Robotics and Automation (ISRA2000)*, Monterrey, N.L., Mexico, 2000.

[15] D. Spenneberg and F. Kirchner. *The Bio-Inspired Scorpion Robot: Design, Control & Lessons Learned, In Climbing & Walking Robots(Book)*, pages 197–218. Houxiang Zhang, I-Tech Education and Publishing, 2008.

[16] P. Stein. Motor systems, with specific reference to the control of locomotion. *Ann. Rev. Neuroscience*, 1:61–81, 1978.

[17] A. H. C. Y. Fukuoka, H. Kimura. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *International Journal of Robotics Research*, 22:187–202, 2003.

A Versatile Stair-Climbing Robot for Search and Rescue Applications

Markus Eich, Felix Grimminger, Frank Kirchner
DFKI Bremen
(German Research Center For Artificial Intelligence)
Robotics Group
28359 Bremen, Germany
{*first_name.last_name*}*@dfki.de*

Abstract — For disaster mitigation as well as for urban search and rescue (USAR) missions, it is often necessary to place sensors or cameras into dangerous or inaccessible areas to get a better situation awareness for the rescue personnel, before they enter a possibly dangerous area. Robots are predestined to this task, but the requirements for such mobile systems are demanding. They should be quick and agile and, at the same time, be able to deal with rough terrain and even to climb stairs. The latter is always required if the rescue personnel has to get access to higher floors inside a building. A rugged, waterproof and dust-proof corpus, and, if possible, the ability to swim, are only a few of many requirements for such robots. With those requirements in mind, the hybrid legged-wheeled robot ASGUARD was developed. This robot is able to cope with stairs, very rough terrain, and is able to move fast on flat ground. We will describe a versatile adaptive controller, based only on proprioceptive data. An additional inclination feedback is used to make the controller versatile for flat ground as well as for steep slopes and stairs. An attachable float provided, the robot is able to swim, using the same locomotion approach. By using twenty compliant legs, which are mounted around four individually rotating hip-shafts, we use an abstract model of quadruped locomotion. For the control design, four independent pattern generators are used. In contrast to many other hybrid legged-wheeled robots, we use the direct proprioceptive feedback in order to modify the internal control loop, thus adapting the model of the motion pattern. For difficult terrains, like slopes and stairs, we use a phase-adaptive approach which is using directly the proprioceptive data from the legs.

Keywords: *stair-climbing robot, adaptive walking, rescue robotics*

I. INTRODUCTION

The robot ASGUARD[1] was designed to be used in harsh outdoor environment with a focus on security and outdoor surveillance as well as on disaster mitigation missions. For those applications, a robot has to transport a variety of mission-depending application sensors inside a difficult terrain. Those missions are commonly named “Three D” missions. “Three D” stands for dull, dirty, and dangerous and implies, e.g., tasks where rescue personnel have to enter a collapse-endangered building in search for injured people, the acquisition of samples in contaminated areas or patrolling every day along the same fence of a security-relevant compound. For all those applications, an in-situ system has to deal with obstacles or uneven and difficult outdoor terrain. Additionally, the robot should be able to move fast where the ground is levelled and easier to cross. In order to cope with these two requirements, the quadruped robot ASGUARD was designed (cf. Figure 1). It makes use of multiple rotating legs along one hip shaft. The overall concept and the design is described in [1], [2].

This simple, but very robust, hybrid legged-wheel approach has already been analysed by several research teams. The key idea is to use one rotating actuator for driving one or more simple legs around one axis. Sometimes referred to as compliant legs [3] or spoked wheel [4], this approach makes often use of a very simple and therefore very robust locomotion principle.

[3] describes the hexapod RHEX. The robot uses one rotating compliant leg per actuator and is able to ascend and descend stairs. RHEX uses a fixed pattern for the trajectory of each leg. The locomotion is performed by a tripod gait, where the retraction and protraction phases are alternatingly triggered. For the stair-climbing behaviour of the RHEX robot, six phases were defined, based on a fixed transition model [5]. The parameters for the single gait were defined, using empirical analysis. Slope adaptation of RHEX, based on inertial sensors, is described in [6].

Fig. 1. The hybrid legged-wheeled robot ASGUARD

For synchronizing the trajectories, the proprioceptive data from six ground contact sensors are used. Another proprioceptive

[1]**A**dvanced **S**ecurity **Guard**

information of the robot is used in [7], where the motor current is used to detect the contact with a flight of stairs. In this case, the tripod gait changes to a metachronal wave gait. In contrast to our approach, the trajectories of the legs are often based on a fixed, hand-adjusted gait configuration. There exist some works about gait transitions but to our knowledge, no work exists on generating the gait on-line during locomotion without a pre-defined pattern which allows to adapt the gait directly according to the proprioceptive data of the legs.
[4] and [8] use a design of a multi-spoked wheel for their hexapod WHEGS, which comes closer to the design of our quadruped ASGUARD because they use more than one compliant leg per axis. The bio-inspired mechanical design is derived from an analysis of the cockroach gait. WHEGS uses no sensor information to adjust the tripod gait: it uses only the compliant legs design to adapt to different types of terrain. WHEGS uses only one DC motor for locomotion and one servo for active steering.
To our knowledge, all locomotion concepts for hybrid legged-wheeled approaches are based on fixed motion patterns. Inclination and ground contact are usually used to select from a range of predefined walking patterns. Our robust stair-climbing behaviour is based on an adaptive closed-loop approach, where the direct torque feedback is used to adapt the overall walking pattern. While running fast on flat ground, e.g. a paved road, the controller of our robot has not to be adaptive. Therefore we implemented two different controllers for the robot: one for flat ground and one for stair climbing and very rough terrain. Therefore we included an inclination sensor on which the output of the two different controllers are gradually merged.
The remainder of the paper is arranged as follows: a short summary of the physical platform ASGUARD is described in Section II. The general control concept and its latest extension is described in Section III. In Section IV we present the experimental results of our approach. In Section V we will discuss those and give some ideas about our future research direction.

II. PLATFORM

The long-term goal of our research is to develop a robust outdoor platform which is suitable to be included in disaster mitigation as well as in security and surveillance missions. The platform should be able to transport application sensors to areas that are dangerous for humans to access, e.g. a collapse-endangered building or an industrial compound after a chemical accident. In those cases, before they enter, the rescue personnel might need some information about the air contamination or the whereabouts of people inside an area. The robot should be upgradeable with a variety of application sensors, e.g. cameras, thermal vision, or chemical sensors. To be usable in any search and rescue or security application, the robot has to be operational without changing batteries for at least two hours. All these requirements were specified in cooperation with potential end users, like fire fighters and rescue personnel. The robot is also waterproof and can move in the water if a float is attached to the robot's body. For swimming, the legged wheels serve as paddles, allowing a swim speed of around 0.5 body-lengths per second.
This defined the minimum size of ASGUARD, as well as the energy budget and the minimum payload. To be usable for a variety of missions, the robot has to be able to carry sensors to areas which are normally not accessible to wheeled and tracked robots.

A. The Physical Design of ASGUARD

The robot ASGUARD is a hybrid quadruped outdoor robot which was inspired by insect locomotion, as described in [4] and [8]. The first prototype of our system is driven by four directly actuated legs with one rotational degree of freedom. In Figure 2, a CAD model of of the robot frame with the four actuators is shown; in Table I the dimensions and other physical data are given. After testing the ground traction with a rigid corpus, we found out that we could increase ground contact by adding an additional rotational degree of freedom along the body axis, serving as an elastic spinal column. By this we could increase the ground traction significantly.

Fig. 2. CAD Model of the body frame (rear view) with the elastic spinal column. The joint can be twisted between $-40°$ and $40°$.

Height	44cm
Length	95cm
Width	50cm
Wheelbase	51cm
Weight	9.5kg
Motors	4x Faulhaber 24V DC motors with 46:1 planetary gear
Motor Power	4x 83W
Battery	10Ah/30V, Lithium Polymer Batteries
Battery Weight	1.6kg

TABLE I
PHYSICAL DIMENSIONS OF THE ROBOT

For the low-level-control, a custom-designed FPGA motor control board is used to generate four independent trajectories, one for each legged wheel. The internal control loop on the FPGA board follows the trajectories, using a

runtime-configurable position controller. The locomotion of the robot is performed by pattern generators which describe the trajectory of each leg within the phase of $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$.
The power consumption as well as the position of each leg are measured in real time, providing important proprioceptive information about the system. In contrast to other approaches, we can individually define the trajectory for each leg, allowing us to synchronize the legs with each other, or to shift the phase of each leg trajectory.

B. The Multi-Legged Wheel Design

The compliant legs of the robot are arranged around four hip shafts with an angular distance of $\frac{2\pi}{5}$. Because of the symmetry of the legs, we have only to consider the phase between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$ (cf. Figure 3). With this configuration we can assure that we have a minimum of four legs on the ground, which ensures a stable configuration of the robot. The outer radius of the legged wheel is $22cm$. The inner radius (i.e. the height of the hip joint shaft if two legs have ground contact) of the legged wheel is $18cm$. In order to decrease the physical shock during locomotion, shock-absorbing leg tips were used.
In contrast to that, a wheeled robot would only be able to go

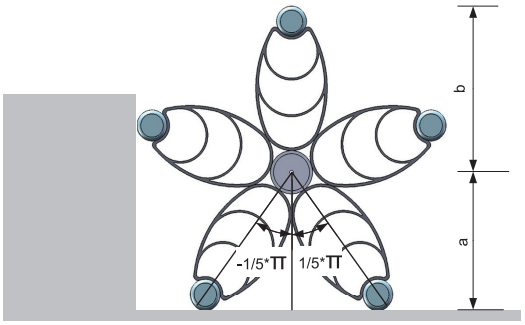


Fig. 3. The ASGUARD wheel. Five compliant legs are mounted around each hip shaft. The dimensions shown are $a = 18cm$ and $b = 22cm$

on a plateau of a height which is much less than the height of the wheel shaft. While driving with high velocities, only the leg tips have direct contact to the ground. In this case, ASGUARD behaves like a wheeled system, with the radius b (cf. Figure 3) reaching velocities of around $2m/s$, which is equivalent to two body lengths per second.

III. CONTROL

A. Using Adaptive Motion Patterns for Hybrid Legged Wheel Control

In order to control the robot ASGUARD, we are facing two requirements. On one hand, we have to control the velocity, i.e. the rotational speed of each of the legs. On the other hand, we have to control the exact position of the robot's legs in order to control the gait and for controlled stair-climbing behaviour. From a variety of motion-pattern based control methods for walking robots [9]–[12] some concepts about timed motion trajectories could be learned. In [13] our approaches for controlling walking robots are described in detail.
From these motion-pattern based control methods, used in a variety of walking machines, we developed an efficient approach to control such systems by using trajectories in the time-angle space. In contrast to many pure legged robots, which have generally more than one degree of freedom for each leg, we have only one angle to control over time. Moreover, as described in Section II-B, we only have to consider the angular space between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$.
For the directional and speed control of ASGUARD, a high

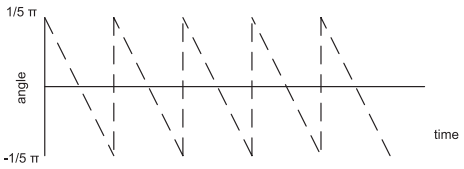


Fig. 4. A typical sawtooth pattern in time-angle space

level controller, which receives its input directly via a joystick, sends the parameters for phase, frequency, and direction to the individual leg trajectories (cf. Figure 5). From our high-level controller we can modify the pattern parameters by changing the pattern frequency, the direction of the generated pattern as well as a phase offset. By this phase offset we can change the synchronization of each of the legs. In our case, a simple sawtooth pattern is used. Figure 4 shows a sample trajectory of one leg in time-angle space.
The patterns are then generated in parallel on a custom-designed FPGA board. A position controller, which is on the next lower level of our control approach, is also implemented in the FPGA. Its task is to follow the generated trajectory in a closed-loop manner. The way how the position controller is working in terms of speed, accuracy, and elasticity, is in our approach directly influenced by the environment of the robot.
During the run, the environment generates a direct feedback on the system, which is in our case the feedback of inclination of the robot as well as the torque feedback of the actuators. An advantage of our design is that we can modify the amplification factor of the proportional control term of the inner control loop at runtime. By changing those parameters on-line, we allow a larger error between the current leg position and the target trajectory. This is an important factor because we are using the proprioceptive information of the applied force to change those parameters. For left/right control of the robot, we use a differential steering approach by individually controlling the speed and the direction of movement for each side of the robot.
Changing the proportional part of the position control parameters on-line has an effect like a mechanical spring. The stiffness at each leg is directly adjustable by the PID parameter of the controller, of which we only use the proportional part. We release the spring for each actuator if the measured torque

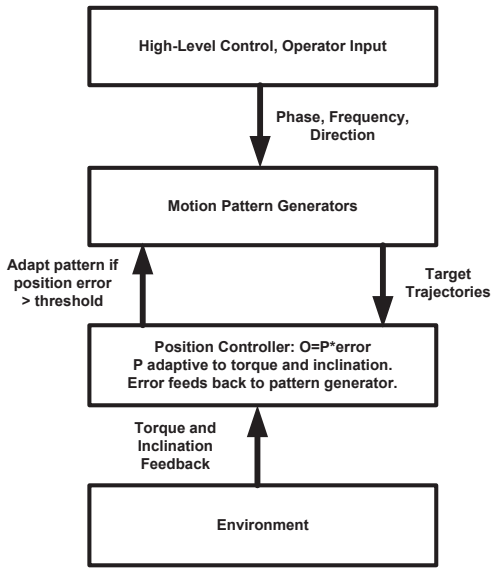


Fig. 5. Our architecture of the behaviour-based control using proprioceptive sensor feedback

is higher with respect to the average torque. This does not directly effect the generation of the motion patterns, only the actual trajectory of each leg. This is comparable to an injury, like a sprained ankle, where humans get a direct negative feedback on the nerves which will result in a more "'elastic"' way of walking without changing the motion pattern directly. The error of the actual trajectory and the target trajectory is then fed back to the motion pattern generator. If the error gets larger than half of one step phase ($\frac{1}{5}\pi$), the pattern of the specific actuator is synchronized with the actual position of the leg. By this we close the loop between the generated pattern and the position controller, which, to our best knowledge, has not been done with other hybrid legged-wheeled robots. This is an important feature because we do not have to synchronize the left and right side of the legs while climbing a stair manually or by a fixed stairs motion pattern. This task is performed directly by the adaptive controller. When ASGUARD climbs a stair, the first front leg which has contact to the first step will have a higher torque on the actuator. The controller will release the leg and therefore allow a larger position error. As stated above, we do not have to ensure that the legs along one cross axis are synchronized, i.e. that they have exactly the same pattern phase. An optimal behaviour in climbing a stair would be to keep the tilt angle of the whole robot minimal while all motors have more and less the same power consumption, depending of course on the inclination and the centre of gravity of the system. The controller keeps implicitly the left and right wheel in phase because it is minimizing the torque discrepancy between all legs. This is achieved, for instance, if the two front legs lift the weight of the robot on the stairs at the same time, given that we have the same motor configuration and mechanical friction within the system, and the weight distribution along the body axis.

In order to distribute the torque of each actuator, which is directly related to the measured motor current, we use an approach to modify the proportional part of the position controller which is responsible for following the trajectory of the generated pattern (cf. Equation 1).

$$P_i = (\kappa_i - (cur_i - \frac{\sum cur}{n}) * \iota_i) \quad (1)$$

P_i refers to the proportional part of the position controller in leg i and cur_i to the measured motor current in A for each motor. The constants κ_i and ι_i are used to map the proportional factor to the controller.

In [2] we showed already that this simple feedback is sufficient for robust stair-climbing. The reader should note that this approach does not directly change the motion pattern in its phase, frequency or direction, but changes the way the internal position controller changes the actual trajectory by allowing a larger error between the target and the actual trajectory. The difference between the torque of a specific motor (which is proportional to the motor current) with respect to the average torque gives a negative feedback on the controller. This results in a higher elasticity, similar to an adaptive spring. In our approach, a high positive discrepancy in the motor torque results in a higher elasticity in the leg.

In contrast to a stair-climbing behaviour, the robot has to bring as much power as possible to the ground, especially while accelerating. This is best achieved by a strict position controller within our control architecture (cf. Figure 5). For this we use a simple proportional controller (cf. Equation 2) with maximal error amplification. Due to the inertia of the system, no differential term within the position control loop is needed.

$$O_i = P_{max} * (error_i) \quad (2)$$

To make our first control approach (cf. Equation 1) versatile for flat as well as for steep terrain and stairs, we take into account the inclination of the robot which is measured by an inertial based tilt sensor. We assume that the robot should adapt its control parameters if the inclination is positive, i.e.

$$0 \leq \frac{pitch}{90^\circ} \leq 1. \quad (3)$$

We extend Equation 1 by applying the measured system inclination, resulting in Equation 4.

$$O_i = (P_{max} - (P_{max} - P_i) * \frac{pitch}{90^\circ}) * (error_i) \quad (4)$$

O_i refers to the motor output of the controller; $error_i$ names the difference between the actual and the target position of leg i, respectively.

IV. EXPERIMENTAL RESULTS

In a first test with the controller, the robot was jacked up on a rack with the actuators running free. Than the front left leg was manually blocked for roughly five seconds and the adaptive part of the internal control loop (i.e. the proportional

factor) was recorded (cf. Figure 6).
Note, that the controller adepts the internal pattern if the error between the target position and the current position of one leg gets larger a specified threshold. This is comparable to a mechanical friction clutch, but with freely configurable parameters. After a synchronization occurred, the error is also reset to zero, resulting implicitly in a change of compliance of the actuator (cf. Equation 4).
The spikes in Figure 6 resemble the numbers of step phases where the front leg was blocked. In can be observed that the compliance in the front left legs goes up (i.e. the proportional factor of the position controller goes down), while the other three legs are made stiffer by increasing the proportional factor of the position controller.
To verify the approach, we did several experiments with a

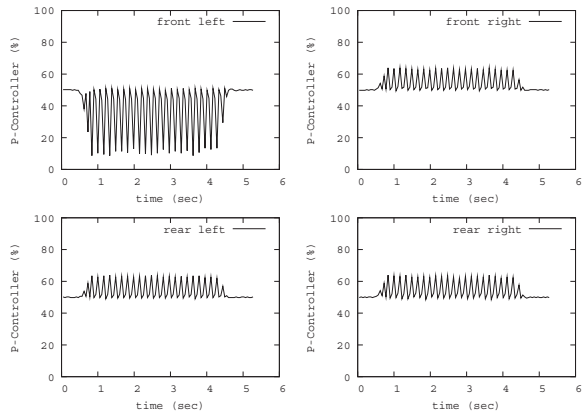


Fig. 6. The compliance of each legs while one leg was manually blocked. The proportional factor can be adapted between stiff (100%) and full compliant (0%). When the front left leg is blocked, the compliance gets close to the maximum. The compliance of the other legs is reduced, i.e. the legs get stiffer.

typical standard stair in our robotic test track with a step hight of 17cm, a step width of 30cm (cf. Figure 7). The flight of stairs of the test run has nine steps until the landing is reached.
To verify the behaviour of the adaptive controller, we analysed the compliance of each leg, based on the proportional factor of the position controller, which adapts in relation to the measured torque of each actuator. Because the legs are automatically left/right synchronized during the stair run, we compared the average front with the average rear compliance of the robot's legs.
The left/right synchronization is done implicitly by the same controller because the leg which hits the first step of the stairs is the most compliant, until the second leg touches the step. If the error between the generated pattern and the actual position of the leg gets larger than a threshold, the pattern is synchronized with the actual position of the leg.
For the experiment we were interested in how the compliance of the front/rear legs is changing in relation to each other. First we observed that due to the adaptivity of the compliance, the climbing behaviour was much more stable on the stairs. Without the torque adaptive controller, the robot tends to rear up on its rear legs, causing a jerky and critical climbing behaviour.
With the adaptivity enabled, the rear legs had not the power

Fig. 7. The stair climbing robot ASGUARD on our outdoor test track. The pattern generator and the proprioceptive feedback take care of the inherent left/right synchronization while climbing a stair. No fixed motion pattern is needed.

any more to rear up the whole corpus because the legs were compliant. At the same time, the front legs were able to pull the system forward instead of the rear legs pushing the robot. Figure 8 shows the results of the nine step stair run. A compliance ratio larger than 0 denotes the percentage of compliance of the rear legs with regards to the front legs. What can be seen in the graph is the nine steps of the stairs (plus one for the landing). It can be seen that during the run, the compliance in the rear leg (i.e. ratio > 0) is significantly larger than the compliance of the front legs (i.e. ratio < 0). The graph can also resemble the alternating pushing and pulling behaviour between the rear and the front legs. With our approach, this is achieved without any fixed or pre-defined walking pattern.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this work we described a proprioceptive control approach for our hybrid legged-wheel robot ASGUARD. The robot is controlled by four individual pattern generators for each of the four actuated leg wheels. We presented our layered architecture which is using a closed loop feedback between the individually generated motion patterns and the internal position controller. In contrast to existing hybrid legged-wheeled robots, we did not use a fixed or predefined motion pattern for stairs or even terrain. The patterns are generated

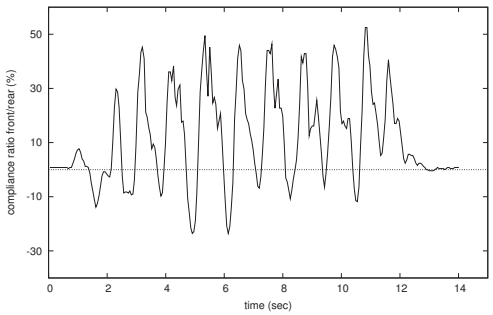


Fig. 8. The average rear leg compliance in respect to the front legs during the stair run. A value > 0 means that the rear legs are more compliant than the front legs at the given percentage and less compliant if the value is < 0. Within the pattern, the nine steps (plus the landing) can also be recognized.

and modified by the direct force feedback applied to each leg. This was achieved by a direct coupling between the applied torques and the stiffness of the position controller. We showed that by only using a proprioceptive torque feedback the robot is able to climb stairs. We found out that a strict controller performs better on flat and even ground. On the other hand, the same strict proportional controller led to several back flips on stairs. We therefore added another proprioceptive tilt feedback in order to perform a weighted merge of the two controllers (maximum stiffness versus maximum adaptation). We showed that this versatile control approach for hybrid legged-wheeled systems was able to perform best possible on a flight of stairs and produced good results on flat ground.

B. Future Works

While the stair-climbing behaviour of ASGUARD is stable, the behaviour on flat ground is still far from being ideal due to a massive leg slippage while accelerating. This could be witnessed by analysing slow-motion videos from several test runs. We could observe that during a fast acceleration the robot's front legs were lifted from the ground due to the system's inertia. We are currently working on an approach in order to detect leg slippage during acceleration.
Another current research is directed to the analysis of different motion patterns with respect to the substrate the robot is moving on. For the time being, we did not yet use any fixed motion pattern but an adaptive pattern depending on the system's proprioceptive data.

To our experience, this is an efficient approach while climbing a stairs, because the feedback which is induced by the physical stair adapts the motion pattern in such ways that the left and right legs are implicitly synchronized on each axis. We suspect that different speeds as well as different types of substrate are an important factor while choosing the type of global motion pattern. This can be observed by many quadruped animals who adapt their walking pattern in regard to locomotion speed and substrate. Ongoing experiments with ASGUARD are focused on the energy consumption in respect of gait, speed, and substrate.

REFERENCES

[1] M. Eich, F. Grimminger, and F. Kirchner, "Adaptive stair–climbing behaviour with a hybrid legged–wheeled robot," in *11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, Coimbra, Portugal, September 8–10 2008, to appear.

[2] M. Eich, F. Grimminger, S. Bosse, D. Spenneberg, and F. Kirchner, "Asguard: A hybrid legged wheel security and sar–robot using bio–inspired locomotion for rough terrain," in *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance*, Benicssim, Spain, January 7–8 2008.

[3] U. Saranli, M. Buehler, and D. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, July 2001.

[4] J. Quinn, R.D.and Offi, D. Kingsley, and R. Ritzmann, "Improved mobility through abstracted biological principles," in *IEEE/RSJ International Conference on Intelligent Robots and System, Volume 3*, 2002, pp. 2652–2657.

[5] E. Moore and M. Buehler, "Stable stair climbing in a simple hexapod robot," 2001. [Online]. Available: "citeseer.ist.psu.edu/moore01stable.html

[6] H. Komsuoglu, D. McMordie, U. Saranli, D. Moore, M. Buehler, and D. E. Koditschek, "Proprioception based behavioral advances in hexapod robot," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, May 21-26, 2001, pp. pp 3650 – 3655.

[7] G. C. Haynes and A. Rizzi, "Gaits and gait transitions for legged robots," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, May 2006.

[8] T. Allen, R. Quinn, R. Bachmann, and R. Ritzmann, "Abstracted biological principles applied with reduced actuation improve mobility of legged vehicles," in *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003), Volume 2*, 2003, pp. 1370–1375.

[9] J. Ayers, "A conservative biomimetic control architecture for autonomous underwater robots," in *Neurotechnology for Biomimetic Robots*, Ayers, Davis, and Rudolph, Eds. MIT Press, 2002, pp. 241–260.

[10] F. Kirchner, D. Spenneberg, and R. Linnemann, "A biologically inspired approach towards robust real world locomotion in an 8-legged robot," in *Neurotechnology for Biomimetic Robots*, J. Ayers, J. Davis, and A. Rudolph, Eds. Cambridge, MA, USA: MIT-Press, 2002.

[11] A. Ijspeert and J. Kodjabachian, "Evolution and development of a central pattern generator for the swimming of a lamprey," 1998. [Online]. Available: citeseer.ist.psu.edu/article/ijspeert98evolution.html

[12] A. H. C. Y. Fukuoka, H. Kimura, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *International Journal of Robotics Research*, vol. 22, pp. 187–202, 2003.

[13] D. Spenneberg and F. Kirchner, *The Bio-Inspired Scorpion Robot: Design, Control & Lessons Learned*. Houxiang Zhang, I-Tech Education and Publishing, 2008, pp. 197–218.

Adaptive Compliance Control of a Multi-legged Stair-Climbing Robot Based on Proprioceptive Data

Markus Eich, Felix Grimminger, Frank Kirchner
DFKI Bremen
(German Research Center For Artificial Intelligence)
Robotics Group
28359 Bremen, Germany
{*first_name.last_name*}*@dfki.de*

Abstract

In this work we describe an innovative compliance control architecture for hybrid multi-legged robots. The approach was verified on the hybrid legged-wheeled robot ASGUARD, which was inspired by quadruped animals. The adaptive compliance controller allows the system to cope with a variety of stairs, very rough terrain, and is also able to move with high velocity on flat ground without changing the control parameters. The control approach takes into account the proprioceptive information of the torques, which are applied on the legs. The controller itself is embedded on a FPGA-based, custom designed motor control board. An additional proprioceptive inclination feedback is used to make the same controller more robust in terms of stair-climbing capabilities. Contrary to existing approaches, we did not use a pre-defined walking pattern for stair-climbing, but an adaptive approach based only on internal sensor information. In this work we show how this adaptivity results in a versatile controller for hybrid legged-wheeled robots. For the locomotion control we use an adaptive model of motion pattern generators. In contrast to many other walking pattern based robots, we use the direct proprioceptive feedback in order to modify the internal control loop, thus adapting the compliance of each leg on-line. The robot is well suited for disaster mitigation as well as for urban search and rescue (USAR) missions, where it is often necessary to place sensors or cameras into dangerous or inaccessible areas to get a better situation awareness for the rescue personnel, before they enter a possibly dangerous area. A rugged, waterproof and dust-proof corpus and the the ability to swim are additional features of the robot.

Index Terms

adaptive compliance control; robot locomotion; legged wheel; pattern generator; stair climbing

I. INTRODUCTION

The robot ASGUARD[1] was designed to be used in harsh outdoor environment with a focus on security and outdoor surveillance as well as on disaster mitigation missions. For those applications, a robot has to transport a variety of mission-depending application sensors inside a difficult terrain. Those missions are commonly named “Three D” missions. “Three D” stands for dull, dirty, and dangerous and implies, e.g., tasks where rescue personnel have to enter a collapse-endangered building in search for injured people, the acquisition of samples in contaminated areas or patrolling every day along the same fence of a security-relevant compound. For all those applications, an in-situ system has to deal with obstacles or uneven and difficult outdoor terrain. Additionally, the robot should be able to move fast where the ground is levelled and easier to cross. In order to cope with these two requirements, the quadruped robot ASGUARD was designed (cf. Figure 1). It makes use of multiple rotating legs attached on one hip shaft. The overall concept and the hardware design is described in [1], [2].

This simple, but very robust, hybrid legged-wheel approach has already been analysed by several research teams. The key idea is to use one rotating actuator for driving one or more simple legs around one axis. Sometimes referred to as compliant legs [3] or spoked wheel [4], this approach makes often use of a very simple and therefore very robust locomotion principle. The concept of using motion pattern is well known and utilized in the area of ambulating robots. An example for generated motion pattern are Central Pattern Generators (CPGs) as the major mechanisms in animals to control and to produce rhythmic motion. CPGs are characterized by the ability to produce rhythmic motion patterns via oscillation of neuronal activity without the need of sensory feedback [5]. However, sensory feedback is normally integrated into the generation of motion pattern. Mostly load and position data of the controlled limb/joint are fed back into the CPG-network which is used to implement a closed-loop control of the rhythmic motion of the system actuators. To modulate the controlled rhythmic patterns, the CPG can change its frequency, phase, and amplitude [6].

For the use in robots it is reasonable to develop an abstract CPG model which inherits only the basic principles of the CPG’s functionality. Many different ways to achieve this have been proposed and tested, e.g. [7]–[10]. In [11] our approaches for controlling walking robots are described in detail.

[1] **A**dvanced **S**ecurity **Guard**

Fig. 1. The hybrid legged-wheeled robot ASGUARD on soft forrest (left) and on rocky terrain (right)

Besides multi-joint walking machines, pattern-based approaches are widely used in hybrid locomotion. In [3], for instance, the hexapod RHEX is described. The robot uses one rotating compliant leg per actuator and is able to ascend and descend stairs. RHEX uses a fixed pattern for the trajectory of each leg. The locomotion is performed by a tripod gait, where the retraction and protraction phases are alternatingly triggered. For the stair-climbing behaviour of the RHEX robot, six phases were defined, based on a fixed transition model [12], [13]. Another proprioceptive information of the robot is used in [14], where the motor current is used to detect the contact with a flight of stairs. In this case, the tripod gait changes to a metachronal wave gait. RHEX uses fixed gait transitions, but in contrast to our approach, the trajectories of the legs are often based on a fixed, hand-adjusted gait configuration. A similar control approach as in RHEX can be found in [15], but uses a tri-lobe wheel to implement a quadruped locomotion. [4] and [16] use a design of a multi-spoked wheel for their hexapod WHEGS, which comes closer to the design of our quadruped ASGUARD because WHEGS uses more than one compliant leg per axis and is able to adapt its walking pattern by pure mechanical compliance of the legs. The bio-inspired mechanical design is derived from an analysis of the cockroach gait. WHEGS uses no sensor information to adjust the tripod gait: it uses only the compliant legs design to adapt to different types of terrain. WHEGS uses only one DC motor for locomotion and one servo for active steering.

To our knowledge, all locomotion concepts for hybrid legged-wheeled approaches are based on fixed motion patterns. Inclination and ground contact are usually used to select from a range of predefined walking patterns. Only WHEGS uses an adaptation based on pure mechanical compliance of the legs, but this compliance cannot be changed on-line during locomotion. Our robust stair-climbing behaviour is based on an adaptive closed-loop approach where the direct torque feedback is used to adapt the overall walking pattern. While running fast on flat ground, e.g. a paved road, the legs of the robot have to be non-compliant in order to bring the required torque on the ground. Therefore the compliance of the controller as to be adaptive to the bodies inclination.

The remainder of the paper is arranged as follows: a short summary of the physical platform ASGUARD is described in Section II. The general control concept and its latest extension is described in Section III. In Section IV we present the experimental results of our approach. In Section V we will discuss those and give some ideas about our future research.

II. ASGUARD PLATFORM

The long-term goal of our research is to develop a robust outdoor platform which is suitable to be included in disaster mitigation as well as in security and surveillance missions. The platform should be able to transport application sensors to areas that are dangerous for humans to access, e.g. a collapse-endangered building or an industrial compound after a chemical accident. In those cases, before they enter, the rescue personnel might need some information about the air contamination or the whereabouts of people inside an area. The robot should be upgradeable with a variety of application sensors, e.g. cameras, thermal vision, or chemical sensors. To be usable in any search and rescue or security application, the robot has to be operational without changing batteries for at least two hours. All these requirements were specified in cooperation with potential end users, like fire fighters and rescue personnel. The robot is also waterproof and can move in the water if a float is attached to the robot's body. For swimming, the legged wheels serve as paddles, allowing a swim speed of around 0.5 body-lengths per second. This defined the minimum size of ASGUARD, as well as the energy budget and the minimum payload. To be usable for a variety of missions, the robot has to be able to carry sensors to areas which are normally not

accessible to wheeled and tracked robots.

A. The Physical Design of ASGUARD

The robot ASGUARD is a hybrid quadruped outdoor robot which was inspired by animal locomotion, as described in [4] and [16]. The first prototype of our system is driven by four directly actuated legs with one rotational degree of freedom. In Figure 2, a CAD model of of the robot frame with the four actuators is shown; in Table I the dimensions and other physical data are given. After testing the ground traction with a rigid corpus, we found out that we could increase ground contact by adding an additional rotational degree of freedom along the body axis, serving as an elastic spinal column. By this we could increase the ground traction significantly.

Fig. 2. CAD Model of the body frame (rear view) with the elastic spinal column. The joint can be twisted between $-40°$ and $40°$.

Height	44cm
Length	95cm
Width	50cm
Wheelbase	51cm
Weight	9.5kg
Motors	4x 24V DC motors with 46:1 planetary gear
Motor Power	4x 83W
Battery	LiPo 100Wh
Battery Weight	1.6 kg

TABLE I
PHYSICAL DIMENSIONS OF THE ROBOT

In order to generate four trajectories in time-angle space, a custom designed FPGA motor control board is used. The internal control loop on the FPGA board follows the trajectories, using a runtime-configurable position controller. The locomotion of the robot is performed by pattern generators which describe the trajectory of each leg within the phase of $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$ (cf. Figure 3). The torques of the legs and the position of each leg are measured in real time, providing important proprioceptive information about the system for the compliance control described later in this work. In contrast to other approaches, we can individually define the trajectory for each leg, allowing the robot to synchronize the legs with each other, or to shift the phase of each leg trajectory.

B. The Multi-Legged Wheel Design

The compliant legs of the robot are arranged around four hip shafts with an angular distance of $\frac{2\pi}{5}$. Because of the symmetry of the legs, we have only to consider the phase between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$. With this configuration we can assure that we have a minimum of four legs on the ground, which ensures a stable configuration of the robot. The outer radius of the legged wheel

is $22cm$. The inner radius (i.e. the height of the hip joint shaft if two legs have ground contact) of the legged wheel is $18cm$. In order to decrease the physical shock during locomotion, shock-absorbing leg tips were used.

In contrast to that, a wheeled robot would only be able to go on a plateau of a height which is much less than the height of

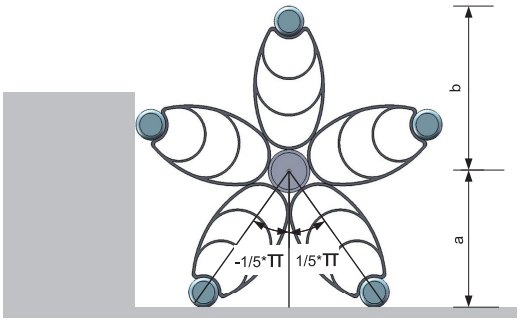


Fig. 3. The ASGUARD wheel. Five compliant legs are mounted around each hip shaft. The dimensions shown are $a = 18cm$ and $b = 22cm$

the wheel shaft. While driving with high velocities, only the leg tips have direct contact to the ground. In this case, ASGUARD behaves like a wheeled system, with the radius b (cf. Figure 3) reaching velocities of around $2m/s$, which is equivalent to two body lengths per second.

III. ADAPTIVE LOCOMOTION CONTROL

A. Adaptive Compliance Control Based on Proprioceptive Data

In order to control the robot ASGUARD, we are facing two requirements. On one hand, we have to control the velocity, i.e. the rotational speed of each of the legs. On the other hand, we have to control the exact position of the robot's legs in order to control the gait and for controlled stair-climbing behaviour. From a variety of motion-pattern based control methods for walking robots [7]–[10] some concepts about timed motion trajectories could be learned. In [11] our approaches for controlling walking robots are described in detail.

From these motion-pattern based control methods, used in a variety of walking machines, we developed an efficient approach to control such systems by using trajectories in the time-angle space. In contrast to many pure legged robots, which have generally more than one degree of freedom for each leg, we have only one angle to control over time. Moreover, as described in Section II-B, we only have to consider the angular space between $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$.

For the directional and speed control of ASGUARD, a high level controller, which receives direct user input, sends the

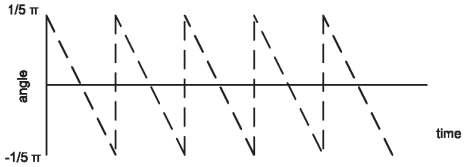


Fig. 4. A typical sawtooth pattern in time-angle space. In this sample, constant speed is assumed.

parameters for phase, frequency, and direction to the individual leg trajectories (cf. Figure 5). From our high-level controller we can modify the pattern parameters by changing the pattern frequency, the direction of the generated pattern as well as a phase offset. By this phase offset we can change the synchronization of each of the legs. In our case, a simple sawtooth pattern is used. Figure 4 shows a sample trajectory of one leg in time-angle space.

The patterns are then generated in parallel on a custom-designed FPGA board. A position controller, which is on the next lower level of our control approach, is also implemented in the FPGA. Its task is to follow the generated trajectory in a closed-loop manner. The way how the position controller is working in terms of speed, accuracy, and elasticity, is in our approach directly influenced by the environment of the robot.

During the run, the environment generates a direct feedback on the system, which is in our case the feedback of inclination of

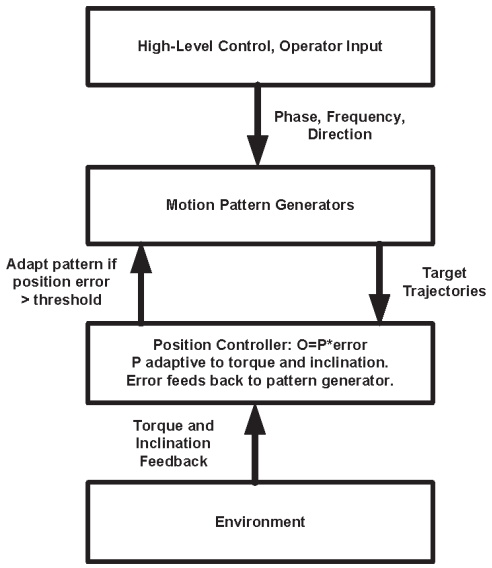


Fig. 5. Our architecture of the behaviour-based control using proprioceptive sensor feedback

the robot as well as the torque feedback of the actuators. An advantage of our design is that we can modify the amplification factor of the proportional control term of the inner control loop at runtime. By changing those parameters on-line, the error between the current leg position and the trajectory is implicitly controlled. This fact is used to control the compliance of each leg individually.

Changing the proportional part of the position control parameters on-line has an effect like a mechanical spring. The stiffness at each leg is directly adjustable by parameter of the controller. We release the spring for each actuator if the measured torque is higher with respect to the average torque. This does not directly effect the generation of the motion patterns, only the actual trajectory of each leg. This is comparable to an injury, like a sprained ankle, where humans get a direct negative feedback on the nerves which will result in a more "'elastic"' way of walking without changing the motion pattern directly.

The position error between current leg position and the generated trajectory is directly fed back to the motion pattern generator. If the error grows larger than half of one step phase ($\frac{1}{5}\pi$), the pattern of the specific actuator is synchronized with the actual position of the leg. By this we close the loop between the generated pattern and the position controller, which, to our best knowledge, has not been done with other hybrid legged-wheeled robots. This is an important feature because we do not have to synchronize the left and right side of the legs while climbing a stair manually or by a fixed stairs motion pattern. This task is performed directly by the adaptive controller. When ASGUARD climbs a stair, the first front leg which has contact to the first step will have a higher torque on the actuator. The controller will release the leg and therefore allow a larger position error.

As stated above, we do not have to ensure that the legs along one cross axis are synchronized, i.e. that they have exactly the same pattern phase. An optimal behaviour in climbing a stair would be to keep the tilt angle of the whole robot minimal while all motors have more and less the same power consumption, depending of course on the inclination and the centre of gravity of the system. The controller keeps implicitly the left and right wheel in phase because it is minimizing the torque discrepancy between all legs. This is achieved, for instance, if the two front legs lift the weight of the robot on the stairs at the same time, given that we have the same motor configuration and mechanical friction within the system, and the weight distribution along the body axis.

In order to distribute the torque of each actuator, which is directly related to the measured motor current, we use an approach to modify the proportional part of the position controller which is responsible for following the trajectory of the generated pattern (cf. Equation 1).

$$P_i = (\kappa_i - (cur_i - \frac{\sum cur}{n}) * \iota_i) \tag{1}$$

P_i refers to the proportional part of the position controller in leg i and cur_i to the measured motor current in A for each motor. The constants κ_i and ι_i are used to map the proportional factor to the controller.
In [2] we showed already that this simple feedback is sufficient for robust stair-climbing. The reader should note that this approach does not directly change the motion pattern in its phase, frequency or direction, but changes the way the internal position controller changes the actual trajectory by allowing a larger discrepancy between the target and the actual trajectory. The difference between the torque of a specific motor (which is proportional to the motor current) with respect to the average torque gives a negative feedback on the controller. This results in a higher elasticity, similar to an adaptive spring. In our approach, a high positive discrepancy in the motor torque results in a higher elasticity in the leg.
In contrast to a stair-climbing behaviour, the robot has to bring as much power as possible to the ground, especially while accelerating. This is best achieved by a strict position controller within our control architecture (cf. Figure 5). For this we use a simple proportional controller (cf. Equation 2) with maximal error amplification. Due to the inertia of the system, no differential term within the position control loop is needed.

$$O_i = P_{max} * (error_i) \tag{2}$$

To make our first control approach (cf. Equation 1) versatile for flat as well as for steep terrain and stairs, we take into account the inclination of the robot which is measured by an inertial based tilt sensor. We assume that the robot should adapt its control parameters if the inclination is positive, i.e.

$$0 \leq \frac{pitch}{90^\circ} \leq 1. \tag{3}$$

We extend Equation 1 by applying the measured system inclination, resulting in Equation 4.

$$O_i = (P_{max} - (P_{max} - P_i) * \frac{pitch}{90^\circ}) * (error_i) \tag{4}$$

O_i refers to the motor output of the controller; $error_i$ names the difference between the actual and the target position of leg i, respectively.

IV. EXPERIMENTAL RESULTS

In order to test the adaptivity of the controller, the robot was jacked up on a rack with the actuators running free. The front left leg was manually blocked for roughly five seconds and the adaptive part of the internal control loop (i.e. the proportional factor) was recorded (cf. Figure 6). Note, that the controller adepts the internal pattern if the error between the target position and the current position of one leg gets larger a specified threshold. This is comparable to a mechanical friction clutch, but with freely configurable parameters. After a synchronization occurred, the error is also reset to zero, resulting implicitly in a change of compliance of the actuator (cf. Equation 4). The spikes in Figure 6 resemble the numbers of step phases where the front leg was blocked. In can be observed that while the compliance of front left legs increases, the compliance of other three actuators are increasing. This is achieved by increasing the proportional factor of the corresponding position controller.
To verify the controller in a real word scenario, we did several experiments with a typical standard stair in our robotic test track with a step hight of 17cm, a step width of 30cm (cf. Figure 7). The flight of stairs of the test run has nine steps until the landing is reached.
We analysed the compliance of each leg, based on the proportional factor of the position controller, which adapts in relation to the measured torque of each actuator. Because the legs are automatically left/right synchronized during the stair run, we compared the average front with the average rear compliance of the robot's legs.
The left/right synchronization is done implicitly by the same controller because the leg which hits the first step of the stairs is the most compliant, until the second leg touches the step.
For the experiment we were interested in how the compliance of the front/rear legs is changing in relation to each other. First we observed that due to the adaptivity of the compliance, the climbing behaviour was much more stable on the stairs. Without the torque adaptive controller, the robot tends to rear up on its rear legs, causing a jerky and critical climbing behaviour.
With the adaptivity enabled, the rear legs had not the power any more to rear up the whole corpus because the legs were compliant. At the same time, the front legs were able to pull the system forward instead of the rear legs pushing the robot.
Figure 8 shows the results of the nine step stair run. A compliance ratio larger than 0 denotes the percentage of compliance of the rear legs with regards to the front legs. What can be seen in the graph is the nine steps of the stairs (plus one for the landing). It can be seen that during the run, the compliance in the rear leg (i.e. ratio > 0) is significantly larger than the compliance of the front legs (i.e. ratio < 0). The graph can also resemble the alternating pushing and pulling behaviour between the rear and the front legs. With our approach, this is achieved without any fixed or pre-defined walking pattern. The

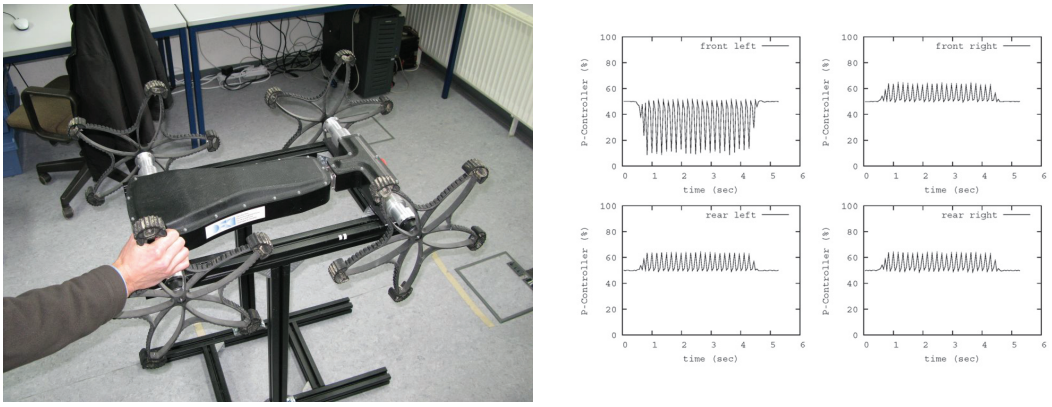


Fig. 6. The compliance of each legs while one leg was manually blocked (left). The proportional factor can be adapted between stiff (100%) and full compliant (0%). When the front left leg is blocked, the compliance gets close to the maximum. The compliance of the other legs is reduced, i.e. the legs get stiffer (right).

Fig. 7. The stair climbing robot ASGUARD on our outdoor test track. The pattern generator and the proprioceptive feedback take care of the inherent left/right synchronization while climbing a stair. No fixed motion pattern is needed.

next results were acquired on a standard stair with nine steps until the first landing was reached. Note, that for the presentation of our results, only the left side of the robots motor are displayed, because on a stair, without active steering, the left and right actuators produce comparable proprioceptive data and can be neglected for further analysis.

For a qualitative analysis of our compliance approach in respect to a non-adaptive, pure position controller, several runs were performed on the same flight of stairs. The first run on the stairs was performed without any compliance in the legs, which means that a pure position controller was used to follow the generated trajectories (Figure 9). This resulted in an unstable stair-climbing behaviour because the rear legs were rearing up the robot's corpus. This again resulted in a critical pitch angle of the robot. According to the data, the current in the rear motors can easily reach 7 ampere. The synchronisation of the generated pattern with the leg position is disabled (no synchronisation signal is triggered).

On a second run on the same stairs we used the described compliance controller which implements Equation 4. The adaptive approach leads to a higher compliance of the legs, especially of the rear legs, reducing their torque significantly while climbing the stairs. This is achieved by reducing the proportional part of the position control, allowing a higher position error while following the trajectory of each leg. This is done for each leg individually and in real-time. Due to the reduction of torque

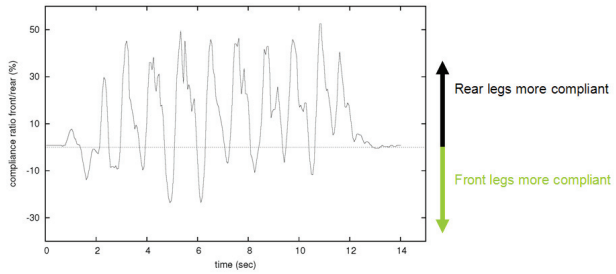


Fig. 8. The average rear leg compliance in respect to the front legs during the stair run. A value > 0 means that the rear legs are more compliant than the front legs at the given percentage and less compliant if the value is < 0. Within the pattern, the nine steps (plus the landing) can also be recognized.

in the rear legs, the front legs have to carry a higher load of the system. This leads to a more balanced torque and power distribution along the body axis (cf. Figure 10). Whenever the error between the trajectory and the current position grows

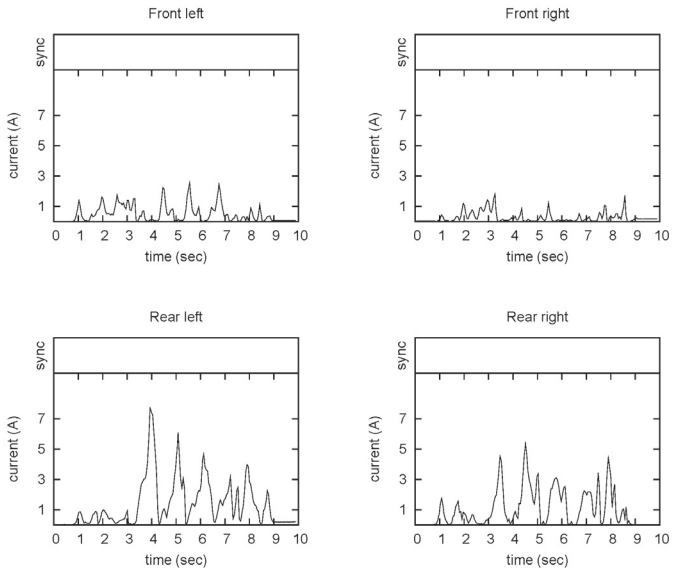


Fig. 9. The proprioceptive dataset of the robot climbing a stair without the adaptive controller. A pure position controller is used to follow the pattern trajectory.

larger that a pre-defined threshold (for our experiments we set the threshold to $|\frac{1}{40}\pi|$, the generated pattern is synchronized with the current leg position. The upper row in Figure 10 shows, when a pattern synchronization signal is raised within the system. This has an effect similar to a mechanical friction clutch, but implemented on an electronic level. The advantage of this implementation is that the force, at which the friction clutch is acting, can be defined on-line, even during a run. The friction clutch-like behaviour can be interpreted as follows: When the position error between the trajectory and the current leg position gets larger, the torque also grows proportionally. When the error (and implicitly the motor torque) crosses the threshold, the internal pattern generator is reset. The pattern generator is not stopped at this point, causing the position error to grow again continuously. While the blocked or overloaded leg is not acting any more, the other legs are used to push the system forward and are eventually reducing the load of the overloaded leg.
An important effect is that by having this clutch like-effect in combination with the leg compliance, the legs are implicitly

left/right synchronized on a stair. Therefore there is no need to select a predefined walking pattern for a stair. The quantitative

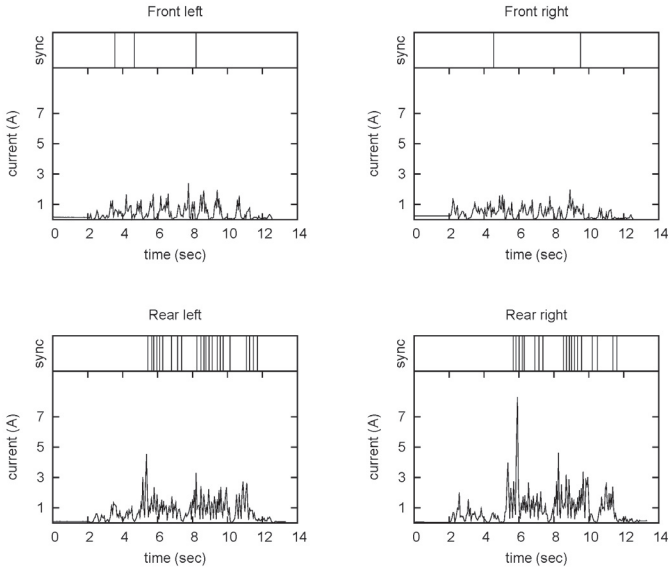


Fig. 10. The proprioceptive dataset of the robot climbing the same stair with the adaptive controller enabled. A proportional part of the internal position controller is adaptive in relation to the torque and inclination.

analysis of the power load distribution is shown in Table II. We found out that by using our adaptive controller, the energy distribution between the front and the rear axis could be reduced from the ratio $1:3.6$ to $1:1.8$ during the experiments. Additionally, the maximum load on the rear axis was reduced significantly from 7 ampere peak to roughly 5 ampere peak.

	Non-adaptive Controller	Our Compliance Controller
Front Legs	2.03mAh	2.77mAh
Rear Legs	7.41mAh	5.02mAh
Power Distribution Front:Rear	1:3.6	1:1.8

TABLE II
POWER CONSUMPTION BETWEEN THE FRONT AND REAR LEGS USING A PURE POSITION CONTROLLER (LEFT) AND THE ADAPTIVE COMPLIANCE CONTROLLER (RIGHT) WHILE CLIMBING THE SAME STAIRS.

V. CONCLUSION AND FUTURE WORK

A. Conclusions

In this work we described a proprioceptive control approach for our hybrid legged-wheel robot ASGUARD. The robot is controlled by four individual pattern generators for each of the four actuated leg wheels. We presented our layered architecture which is using a closed loop feedback between the individually generated motion patterns and the internal position controller. In contrast to existing hybrid legged-wheeled robots, we did not use a fixed or predefined motion pattern for stairs or even terrain. The patterns are generated and modified by the direct force feedback applied to each leg. This was achieved by a direct coupling between the applied torques and the stiffness of the position controller. We showed that by only using a proprioceptive torque feedback the robot is able to climb stairs. We found out that a strict controller performs better on flat and even ground. On the other hand, the same strict proportional controller led to several back flips on stairs. We therefore added another proprioceptive inclination feedback in order to make the approach more versatile for flat ground as well as stair-climbing. We showed that this versatile control approach for hybrid legged-wheeled systems was able to perform best possible on a flight of stairs and produced good results on flat ground.

B. Future Works

While the stair-climbing behaviour of ASGUARD is stable, the behaviour on flat ground is still far from being ideal due to a massive leg slippage while accelerating. This could be witnessed by analysing slow-motion videos from several test runs. We could observe that during a fast acceleration the robot's front legs were lifted from the ground due to the system's inertia. We are currently working on an approach in order to detect leg slippage during acceleration.
Another current research is directed to the analysis of different motion patterns with respect to the substrate the robot is moving on. For the time being, we did not yet use any fixed motion pattern but an adaptive pattern depending on the system's proprioceptive data. To our experience, this is an efficient approach while climbing a stairs, because the feedback which is induced by the physical stair adapts the motion pattern in such ways that the left and right legs are implicitly synchronized on each axis. We suspect that different speeds as well as different types of substrate are an important factor while choosing the type of global motion pattern. This can be observed by many quadruped animals who adapt their walking pattern in regard to locomotion speed and substrate. Additional focus of our ongoing research is the use of the proprioceptive information from the legs in order to augment auto generated maps with this additional data. We belief that proprioceptive information from a robot body can improve state estimated and self-localization significantly.

VI. ACKNOWLEDGMENTS

This work was financed by the German Research Center for Artificial Intelligence (DFKI) in Kaiserslautern, Germany within the SentryBot EP project (project number 9518). Additionally, the authors would like to thank the Electronics Lab of the Robotics Group at the University of Bremen for the electronic support.

REFERENCES

[1] M. Eich, F. Grimminger, and F. Kirchner, "Adaptive stair–climbing behaviour with a hybrid legged–wheeled robot," in *11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, Coimbra, Portugal, September 8–10 2008, to appear.
[2] M. Eich, F. Grimminger, S. Bosse, D. Spenneberg, and F. Kirchner, "Asguard: A hybrid legged wheel security and sar–robot using bio–inspired locomotion for rough terrain," in *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance*, Benicssim, Spain, January 7–8 2008.
[3] U. Saranli, M. Buehler, and D. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, July 2001.
[4] R. D. Quinn, J. Offi, D. Kingsley, and R. Ritzmann, "Improved mobility through abstracted biological principles," in *IEEE/RSJ International Conference on Intelligent Robots and System, Volume 3*, 2002, pp. 2652–2657.
[5] P. Stein, "Motor systems, with specific reference to the control of locomotion," *Ann. Rev. Neuroscience*, vol. 1, pp. 61–81, 1978.
[6] K. T. Sillar, O. Kiehn, and N. Kudo, "Chemical modulation of vertebrate motor circuits," in *Neurons, Networks and Motor Behavior*, P. Stein, S. Grillner, A. I. Selverston, and D. Stuart, Eds. MIT Press, Cambridge, 1997, pp. 183–193.
[7] J. Ayers, "A conservative biomimetic control architecture for autonomous underwater robots," in *Neurotechnology for Biomimetic Robots*, Ayers, Davis, and Rudolph, Eds. MIT Press, 2002, pp. 241–260.
[8] F. Kirchner, D. Spenneberg, and R. Linnemann, "A biologically inspired approach towards robust real world locomotion in an 8-legged robot," in *Neurotechnology for Biomimetic Robots*, J. Ayers, J. Davis, and A. Rudolph, Eds. Cambridge, MA, USA: MIT-Press, 2002.
[9] A. Ijspeert and J. Kodjabachian, "Evolution and development of a central pattern generator for the swimming of a lamprey," 1998. [Online]. Available: citeseer.ist.psu.edu/article/ijspeert98evolution.html
[10] A. H. C. Y. Fukuoka, H. Kimura, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *International Journal of Robotics Research*, vol. 22, pp. 187–202, 2003.
[11] D. Spenneberg and F. Kirchner, *The Bio-Inspired Scorpion Robot: Design, Control & Lessons Learned.* Houxiang Zhang, I-Tech Education and Publishing, 2008, pp. 197–218.
[12] E. Moore and M. Buehler, "Stable stair climbing in a simple hexapod robot," 2001. [Online]. Available: "citeseer.ist.psu.edu/moore01stable.html
[13] H. Komsuoglu, D. McMordie, U. Saranli, D. Moore, M. Buehler, and D. E. Koditschek, "Proprioception based behavioral advances in hexapod robot," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, May 21-26, 2001, pp. pp 3650 – 3655.
[14] G. C. Haynes and A. Rizzi, "Gaits and gait transitions for legged robots," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, May 2006.
[15] S. D. Herbert, A. Drenner, and N. Papanikolopoulos, "Loper: A quadruped-hybrid stair-climbing robot," in *Proceedings of the 2008 IEEE Conference on Robotics and Automation*, Pasadena, USA, May 2008.
[16] T. Allen, R. Quinn, R. Bachmann, and R. Ritzmann, "Abstracted biological principles applied with reduced actuation improve mobility of legged vehicles," in *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003), Volume 2*, 2003, pp. 1370–1375.

A Hybrid Legged Wheel Climbing Robot for Marine Inspection

Thomas Vögele, Markus Eich, Felix Grimminger, Kristin Fondahl

DFKI Robotics Innovation Center (RIC),
28359 Bremen, Germany
E-mail: thomas.voegele@dfki.de
www.dfki.de/robotics

The inspection of marine vessels is currently performed manually. Inspectors use sensors (e.g. cameras, devices for non-destructive testing) to detect damaged areas, cracks, and corrosion in large cargo holds, tanks, and other parts of a ship. Due to the size and complex geometry of most ships, ship inspection is time-consuming and expensive. The EU funded project MINOAS develops concepts for a Marine Inspection Robotic Assistant System to improve and automate ship inspection. A central part of MINOAS is to evaluate the use of a cooperative fleet of robots, including areal drones, magnetic climbing robots, and underwater crawlers, for ship inspection. In this paper we describe a first concept for one component of the MINOAS robot fleet, a magnetic crawler for the inspection of large cargo holds and large tanks. We show how a lightweight system using magnetic wheels (including hybrid leg-wheels) and a passive magnetic tail can successfully climb tall metallic walls and overcome small obstacles.

Keywords: Hybrid Leg-Wheel Locomotion, Magnetic Climber, Marine Inspection

1. Introduction

Seagoing vessels are subject to numerous and regular inspections and maintenance measures. These involve the visual and physical inspection of the ship, including cargo holds and tanks. Many spaces on a ship are difficult and even dangerous to access: Smaller tanks are often narrow, badly ventilated and dirty. Larger spaces (e.g. cargo holds) can easily reach heights of 10 meters and more. Extensive scaffolding, the use of cranes, or other (typically expensive) measures are needed to access the latter. This makes ship inspection a time consuming, expensive, and potentially dangerous task.

Ship inspection is still done mostly "manually", i.e. by surveyors and inspectors. Tools to automate the process, like mobile robots, are not used.

The reason is that ships in general, and confined spaces on ships in particular, are very difficult environments for mobile inspection robots, with the main challenges being in the areas of locomotion, communication, and self-localization/navigation.

The EU-funded R&D project MINOAS (**M**arine **IN**spection r**O**botic **A**ssistant **S**ystem) addresses these challenges in an attempt to develop concepts for the automation of the ship inspection process. Figure 1 displays a typical inspection scenario on a bulk carrier. Extensive scaffolding has to be installed in the cargo hold in order to grant access to the surveyor for close inspection of the ship hull. The surveyor may have to wait several weeks until the scaffolding is installed and the ship can be inspected. Obviously, the costs for such time consuming inspections are high as the ship owner will loose money on every day the ship docked.

Fig. 1. Dock workers have to install scaffolding in order to make the cargo hold accessible for the ship surveyor.

Locomotion of robots in ships is the first challenge to be addressed in MINOAS. Ships are made of solid steel and thus enable the use of magnetic feet, wheels, or tracks for locomotion. However, the movement of magnetic crawlers is hampered by the complex geometries and many obstructions (beams, stiffeners, bolts, welds, pipes etc.) that are typical for ships. Most magnetic crawlers are caused to shear off an otherwise flat metallic wall by even small obstacles, such as bolts, welds or cables. The usability of tracked magnetic crawlers for ship inspection is therefore limited.

Another problem is related to the safety of operation of a magnetic crawler. Most magnetic crawlers used in industrial environments, e.g. for the cleaning and preparation of tanks and ship hulls, weight at least ten kilos. During operation, they have to be secured with cables to prevent damage to both the robot and humans should they shear off and fall. However, securing a robot with cables is not feasible in a scenario where an inspector

wants to inspect, for example, the top section of a tall cargo hold. What is needed here is a small, sturdy, light-weight robot that is able to withstand a fall from several meters and at the same time does not endanger the surveyor and/or other crew members.

In the first phase of the EU-funded MINOAS project, we focus on the locomotion challenge and develop a solution for the inspection of tall cargo holds. A light-weight system design is used to develop an inspection robot that

- can easily be carried by a human (surveyor),
- is able to climb metallic walls fast and efficiently,
- can overcome small obstacle like welds, bolts and thin cables,
- is sturdy enough to survive a fall from several meters and
- is light enough to cause no damage (to man or material) in case of an accident.

This light-weight magnetic crawler will be one member of the MINOAS robot fleet, which also includes flying systems and heavier magnetic crawlers. The task of the light-weight crawler is to climb metallic walls, perform a visual close-up inspection of the wall, and mark spots of interest for further inspection by other team members. The system presented in this paper is remote controlled by the operator (surveyor).

2. Related Work

Many approaches exist already in research and in industry for robots that use magnets or suction pads to climb vertical metallic walls. The robots using magnets can be divided mainly into tracked and wheeled systems, although some legged systems exist as well. Wheeled systems are relatively simple, but most magnetic wheels are heavy because they are made of magnetic material or metal in combination with strong magnets. Some tracked systems use strong magnets located in the mid-section of the robot for traction and (rubber) tracks for locomotion. Others use magnetic tracks for both locomotion and traction. All tracked systems have the disadvantage that they are complex and heavy. Another drawback is that most of them cannot climb on uneven and corrugated walls, e.g. corrugated sheet metal.

The introduction of a mechanism based on dual magnetic wheels, resembling a crawler, not only overcame limitations in negotiating pipe bends but enables robots to climb over sharp obstacles like sleeve and dresser joints.[1] This concept has been further exploited in,[2] where magnetic wheel units

incorporating active lifter-stabilizer mechanisms demonstrated their effectiveness in negotiating complex shaped structures.

In our work we make use of a hybrid legged wheel approach, already found in robotic systems like Asguard,[3,4] the hexapod RHex,[5] or the bio-inspired Whegs.[6]

The main idea of the hybrid legged-wheel approach is to combine the advantages of magnetic wheels, i.e. fast locomotion on metallic surfaces, with that of magnetic feet, i.e. the ability to overcome obstacles. A hybrid legged-wheel crawler will thus be able to climb vertical walls that do not have to be completely flat and obstacle-free. Uneven corrugated surfaces and even small obstacles like welds, bolds and cables, can be overcome with this approach.

3. Climbing Robot Description

3.1. *Application- and System Requirements*

To inspect tall cargo holds and tanks on ships, the surveyor has to be able to reach places high above ground. In many cases, pre-installed means of access, like ladders and gangways, are not available. The standard procedure for the inspection of such spaces is to install temporary scaffolding, or to flood the cargo hold in order to access the upper parts of the hold by raft. In MINOAS, we developed a demonstrator for a robotic tool that will assist a surveyor to get a visual impression of places that cannot be reached otherwise. Using the robot, surveyors will be able to inspect the upper parts of tall structures on a ship while remaining safely and comfortably at the base of the structure. Scaffolding and flooding will not be needed anymore.

To be useful in the application described above, the robot had to be simple and efficient. This implied a number of design requirements the robot had to fulfill:

- Mobility: The robot must be able to climb vertical metallic walls with a reasonable speed.
- Portability: It has to be light enough to be carried by the surveyor.
- Energy autonomy: External power supplies involving cables and tethers should be avoided.
- Safe handling: In the absence of securing cables (which would greatly limit the usability of the robot), the robot must not pose a danger to the surveyor or other personnel.

We addressed these requirements by a system design that combines a

light-weight minimalistic robot corpus with efficient locomotion through magnetic wheels.

3.2. *Light-Weight Robot Design*

The current version of the light-weight inspection robot weights only 650 grams. This includes a Li-Io battery pack supporting energy autonomy for up to one hour, two high-performance Faulhaber electro-motors, electronics for motor control and communication, an LED light source, and a light-weight camera.

The high performance-to-weight ratio was achieved for one by using light-weight materials, like Delrin plastic and fiberglass, for all parts of the robot corpus. In addition, a consequent simplification and reduction of the robot design was implemented. Two wheels in cooperation with a flexible "tail" guarantee the stability of the system (Figure 2). Each wheel is equipped with an individually controlled wheel-hub-engine, enabling high mobility and maneuverability.

3.3. *Locomotion by Magnetic Wheels*

We use magnetic wheels for efficient locomotion of the crawler on metallic surfaces. Compared to tracks, magnetic wheels have the advantage of being less complex. Furthermore, the use of light-weight material (e.g. plastic) makes it possible to build a locomotion system that is much simpler and lighter than a comparable tracked or standard wheeled system. The wheels of our light-weight magnetic crawler are made of Delrin plastic, rubber and strong neodymium magnets.

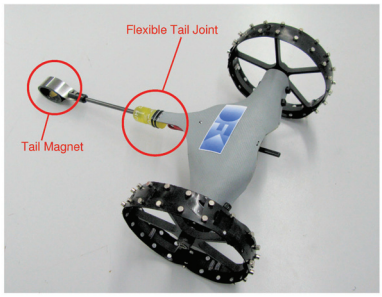


Fig. 2. A flexible tail with an attached neodymium magnet increases stability during rotation on the vertical wall.

For the first demonstrator of the magnetic crawler we tested two different designs for the magnetic wheels (Figure 3) :

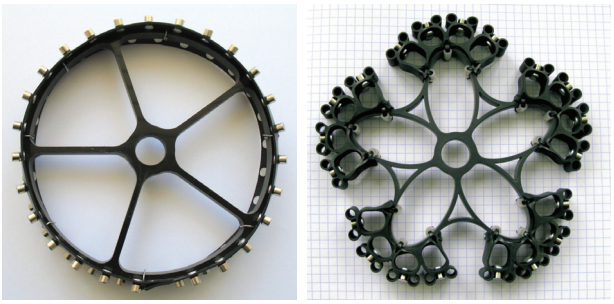


Fig. 3. **Left:** Wheel design A **Right:** Wheel design B

- Wheel Type A follows a standard wheel design. Traction is created by a total of 50 high-power neodymium magnets embedded in a flexible rubber base.
- Wheel Type B applies a hybrid wheel-leg design, where each wheel consists of five legs that end in a foot. Each foot has a flexible rubber base with 8 high-power neodymium magnets.

In addition to the magnetic wheels, a strong neodymium magnet is included in the tail section. The passive but flexible magnetic tail proved to have a strong impact on the stability and maneuverability of the system (Figure 2).

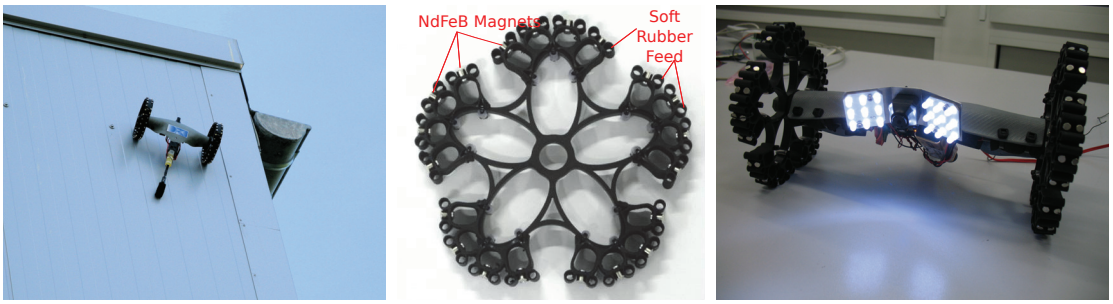


Fig. 4. **Left:** The magnetic crawler climbing a vertical metal wall. For this experiment a round shaped wheel with magnets was used.**Middle:**The hybrid legged wheel design. Each wheel has five legs, including 40 neodymium magnets for each wheel. **Right:** Mounted on the crawler is a camera including an LED-based light source. Image transmission is wireless.

4. First Results of Locomotion Tests

Preliminary tests with the robot on tall metal walls were very encouraging. We could reach climbing velocities of up to 0,5 m/s. The combination of two wheels and a passive magnetic tail proved to enable an extraordinary

flexibility and maneuverability. The robot was able to turn within a radius smaller than 40 cm.

The robot was able to overcome obstacles, such as welds and small metallic bolts, without difficulties. The standard magnetic wheel (Wheel Type A) was able to overcome non-magnetic obstacles such as cables of up to 6 mm diameter (Figure 5). The hybrid wheel (Wheel Type B) could handle cables of up to 13 mm.

However, while Wheel Type A showed a very reliable traction and allowed the robot to climb, turn and decent fast and without problems, the traction of Wheel Type B was not satisfactory so far. Tests indicate that shear forces created through the (by design) uneven roll behavior of the wheel are responsible for a frequent loss of traction. In the next version of the robot we will therefore implement an intelligent motor control to smooth the roll behavior of the wheel and to achieve a better and more reliable traction.

Fig. 5. **Left:** Managing non-metallic obstacles with wheel type A **Right:** Managing non-metallic obstacles with wheel type B

5. Conclusion

In this paper we describe the concept of a light-weight magnetic crawler for the inspection of tall structures on ships, like cargo holds and tanks. This work is embedded in the EU-funded MINOAS project which strives to develop a concept for a heterogeneous robot fleet for ship inspection.

To be light-weight and simple, the crawler was equipped with two magnetic wheels and a passive magnetic tail. The wheels were made of light-weight plastic. Traction on metallic surfaces was was reached through high-performance neodymium magnets embedded in a flexible rubber base attached to the wheels.

With the two wheels, the robot displayed very good climbing and turning properties. The tail, which also includes a high-performance neodymium

magnet, proved to add considerable stability and flexibility to the system. Two different wheel designs were tested. Although the hybrid wheel-leg design showed better performance when obstacles had to be overcome, the classic round wheel design had much better traction on the metallic surfaces. In the next version of the robot, we hope to be able to overcome the traction problem of the hybrid wheel by implementing an intelligent motor controller.

In summary, the two-wheeled light-weight crawler design proved to be a sound basis for a robotic tool that can be used by a surveyor for the visual inspection of regions on a ship that are otherwise difficult to reach. Future work will include a further improvement of the traction of the magnetic wheels. This is necessary to allow for an increase in payload needed to add additional sensors and actuators, for example to mark spots for closer inspection by other robots of the MINOAS robot fleet.

Acknowledgments

This work has been funded by the European Commission under the project MINOAS (SCP8-GA-2009-233715).

References

1. Y. Kawaguchi, I. Yoshida, H. Kurumatani, T. Kikuta, Y. Yamada and O. Ltd, Internal pipe inspection robot, in *1995 IEEE International Conference on Robotics and Automation, 1995. Proceedings.*, (IEEE, 1995).
2. F. Tâche, W. Fischer, R. Moser, F. Mondada and R. Siegwart, Adapted magnetic wheel unit for compact robots inspecting complex shaped pipe structures, in *Proc. of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2007), Zurich, Switzerland*, (Citeseer, 2007).
3. M. Eich, F. Grimminger and F. Kirchner, Adaptive Stair-Climbing Behaviour with a Hybrid Legged-Wheeled Robot, in *Advances in Mobile Robotics. 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, (World Scientific Publishing Co. Pte. Ltd., Coimbra, 2008).
4. M. Eich, F. Grimminger and F. Kirchner, *Adaptive compliance control of a multi-legged stair-climbing robot based on proprioceptive data* (Emerald Group Publishing Limited, 2009).
5. E. Moore and M. Buehler, Stable stair climbing in a simple hexapod robot, in *Int. Conf. Climbing and Walking Rob., Karlsruhe, Germany*, (Citeseer, 2001).
6. R. Quinn, D. Kingsley, J. Offi and R. Ritzmann, Improved mobility through abstracted biological principles, in *IEEE Int. Conf. On Intelligent Robots and Systems (IROS02)*, (Lausanne, 2002).

19th Mediterranean Conference on Control and Automation
Aquis Corfu Holiday Palace, Corfu, Greece
June 20-23, 2011

Design and Control of a Lightweight Magnetic Climbing Robot for Vessel Inspection

Markus Eich* and Thomas Vögele
DFKI GmbH
Robotics Innovation Center
28359 Bremen, Germany
{first_name.last_name}@dfki.de

Abstract— **The inspection of marine vessels is currently performed manually. Inspectors use sensors (e.g. cameras, devices for non-destructive testing) to detect damaged areas, cracks, and corrosion in large cargo holds, tanks, and other parts of a ship. Due to the size and complex geometry of most ships, ship inspection is time-consuming and expensive. The EU-funded project MINOAS develops concepts for a Marine Inspection Robotic Assistant System to improve and to automate ship inspections. As one example of a physical system implementation, we introduce our magnetic wall-climbing robot. This remotely operated lightweight system is able to climb a vessels steel frame and is able to deliver visual inspection data on-line. For any type of surveying process, the raw and meta data are mandatory in order to compare inspection data over time. In this paper, we describe our approach of how the magnetic climbing robot is localized and controlled. Additionally, we describe the design of the robot as well as the localization device which is able to provide a real-time 3D pose of the inspection robot.**

I. INTRODUCTION

Sea-going vessels are subject to numerous and regular inspections and maintenance measures. These involve the visual and physical inspection of the ship, including cargo holds and tanks. Many spaces on a ship are difficult and even dangerous to access: Smaller tanks are often narrow, badly ventilated and dirty. Larger spaces (e.g. cargo holds) can easily reach heights of 10 meters and more. Extensive scaffolding, the use of cranes, or other (typically expensive) measures are needed to access the latter. This makes ship inspection a time consuming, expensive, and potentially dangerous task.

Ship inspection is still done mostly ”manually”, i.e. by surveyors and inspectors. Tools to automate the process, like mobile robots, are not used. The reason is that ships in general, and confined spaces on ships in particular, are very difficult environments for mobile inspection robots, with the main challenges being in the areas of locomotion, communication, and self-localization/navigation. The EU-funded R&D project MINOAS (**M**arine **IN**spection r**O**botic **A**ssistant **S**ystem) addresses these challenges in an attempt to develop concepts for the automation of the ship inspection process. Figure 1 displays a typical inspection scenario on a bulk carrier. Extensive scaffolding has to be installed in the cargo hold in order to grant access to the surveyor for close inspection of the ship hull. The surveyor may have to wait several weeks until the scaffolding is installed and the ship can be inspected. Obviously, the costs for such time consuming inspections are high as the ship owner will loose money on every day the ship is docked.

Fig. 1. Dock workers have to install scaffolding in order to make the cargo hold accessible for the ship surveyor.

Locomotion of robots in ships is the first challenge to be addressed in MINOAS. Ships are made of solid steel and thus enable the use of magnetic feet, wheels, or tracks for locomotion. However, the movement of magnetic crawlers is hampered by the complex geometries and many obstructions (beams, stiffeners, bolts, welds, pipes etc.) that are typical for ships. Most magnetic crawlers are caused to shear off an otherwise flat metallic wall by even small obstacles, such as bolts, welds or cables. The usability of tracked magnetic crawlers for ship inspection is therefore limited.

Another problem is related to the safety of operation of a magnetic crawler. Most magnetic crawlers used in industrial environments, e.g. for the cleaning and preparation of tanks and ship hulls, weight at least ten kilos. During operation, they have to be secured with cables to prevent damage to both the robot and humans should they shear off and fall. However, securing a robot with cables is not feasible in a scenario where an inspector wants to inspect, for example, the top section of a tall cargo hold. What is needed here is a small, sturdy, lightweight robot that is able to withstand a fall from several meters and at the same time does not endanger the surveyor and/or other crew members.

For the vessel surveying process it is mandatory to keep track of the inspection results over time. Inspection data,

*This work is partially supported by FP7 project SCP8-GA-2009- 233715 (MINOAS).

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

e.g. video data of corrosion and cracks or thickness measurements have to be logged together with the time stamp as well as the accurate location where the data was acquired. This will provide the surveyor with the mandatory data to observe the change of the damages over time in a vessel's life cycle.

In this work, we show our robotic inspection tool which is able to provide the needed information to the surveyor. We give a system overview of our mobile lightweight magnetic climber. Additionally we describe our method on how the 3D position of the climber can be tracked in real time. This allows the surveyor to localize the data within a 3D user interface.

The remainder of the paper is organized as follows: In Section II we briefly summarize the state of the art in vessel inspection robotics. In Section III an overview of the system architecture is presented. In Section IV we describe our vessel inspection system in detail, including the inspection robot, the 3D localization mechanism and the user interface. Section V gives an overview of the experimental evaluation of our approach and Section VI concludes our work and gives an outlook on the future work regarding the MINOAS project.

II. RELATED WORK

Many approaches exist already in research and in industry for robots that use magnets or suction pads to climb vertical metallic walls. The robots using magnets can be divided mainly into tracked and wheeled systems, although some legged systems exist as well. Wheeled systems are relatively simple, but most magnetic wheels are heavy because they are made of magnetic material or metal in combination with strong magnets. Some tracked systems use strong magnets located in the mid-section of the robot for traction and (rubber) tracks for locomotion. Others use magnetic tracks for both locomotion and traction. All tracked systems have the disadvantage that they are complex and heavy. Another drawback is that most of them cannot climb on uneven and corrugated walls, e.g. corrugated sheet metal.

Some inspection robots are able to climb on poles and vertical tube like structures [1]. This climbing robot clinches to a pole on two sides and is able to carry a significant payload.

Quite common for vertical climbing robots is the usage of suction pads [2]. This works perfectly on glossy and clean environment like windows or clean and even surfaces. The drawback of using suction pads is the need for an additional vacuum motor, which increases the weight of the robot by several kilograms. Even if the vacuum motor is externally, the robot still has to carry the pressure hose. Because of the heavy weight these systems have to be secured by a safety rope. This again results in a higher installation effort.

In our approach we use a lightweight system which can be transported easily by a surveyor, even to less accessible parts of the ship, e.g. parts which are only accessible by man holes. Dust and dirt also prevents a vacuum based climbing robot from working in unclean environment e.g. inside a bulk carrier.

The introduction of a mechanism based on dual magnetic wheels, resembling a crawler, not only overcame limitations in negotiating pipe bends, but enables robots to climb over sharp obstacles like sleeve and dresser joints [3]. This concept has been further exploited in [4], where magnetic wheel units incorporating active lifter-stabilizer mechanisms demonstrated their effectiveness in negotiating complex shaped structures.

Obstacle avoidance and robot localization is also an issue. For the wall-climbing robot CROMSCI [5] an approach is developed which uses a 2D laser range finder to detect holes in the wall. Holes are a critical issue for vacuum based climbers, because these systems lose the mandatory vacuum between wall and robot. The system is also able to avoid obstacles and is currently investigated as a system for concrete dam monitoring.

A system which is able to climb complex ferrous structures is described in [6]. This quadruped legged system has compliant magnets as feet and is able to walk and climb on complex structures. Unfortunately, the author didn't mention the system speed performance.

For our developed system the locomotion speed was one of the project criterions. During a vessel inspection process, the robot has to cover an area of several tenth of square meters, therefore a wheeled system approach was favored over a legged robot approach.

III. SYSTEM ARCHITECTURE

The MINOAS crawler inspection system consists of several hardware and software components (cf. Figure 2). The crawler itself is tele-operated by a human operator using a 35MHz remote control. The crawler can be equipped with a variety of application sensors for the vessel inspection. Currently, the crawler has an integrated wireless analog camera providing a real-time video stream. Due to the light weight of the crawler, video is currently not processed on the robot itself, but is transmitted to a control station using 2.4GHz transmission.

For the surveying process, it is mandatory that not only the time of the video is recorded, but also the position where the data was acquired. The crawler has a total weight of 700 gram, including batteries and camera. This weight does not allow to carry sensors for localization like laser range finders or sonar which are commonly used in robotics.

To provide the mandatory information about the crawlers position, an external tracking device was designed. The custom designed 3D tracker consists of a pan-tilt unit, a high resolution camera and a single point laser range finder. This tracker provides the real-time position and is directly connected to the control station. The control station combines the application data from the inspection crawler with the position information and time information. The tracker itself communicates with the control station using the communication framework from ROS [7].

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

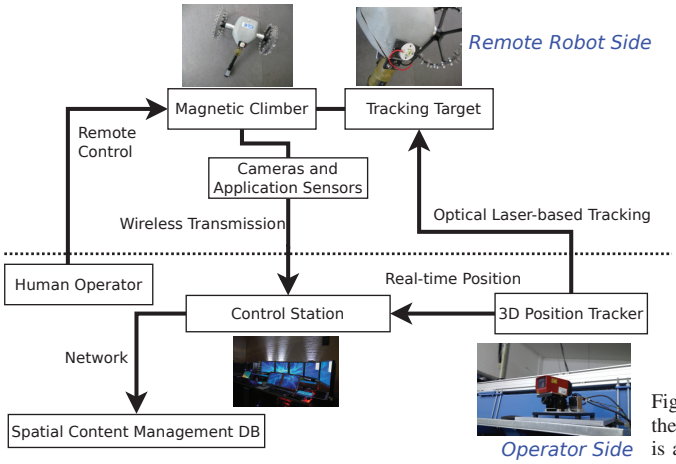


Fig. 2. An overview of the architecture of the vessel inspection systems. The main components are the magnetic climber robot, the 3D tracking unit and the spatial content management database, allowing to store acquired inspection data with the information of place and time.

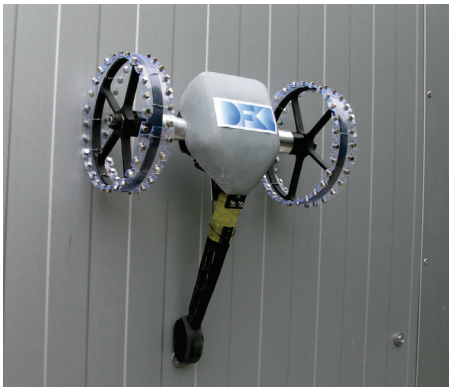


Fig. 3. The MINOAS magnetic crawler. The flexible tail section improves the maneuverability significantly on a vertical wall. A neodymium magnet is attached to the tail section and stabilizes the system while climbing.

The 3D tracker uses a camera to track a bright LED target, which is mounted on the inspection robot. Together with the laser range finder on the tracking device, the position in 3D space can be assessed in real-time. The wireless transmitted inspection data from the robot is matched with the inspection meta data. The position of the magnetic crawler calculated in the trackers coordinate system. In order to give the surveyor a better impression where the robot is inside the vessel, a 3D model of the ship is used. The reference frame of the tracker has to be manually aligned once with that of the ship's 3D model. In the following section, the different components of the inspection crawler system are described in detail.

IV. SYSTEM DESIGN AND CONTROL

A. The Magnetic Crawler

In this section we describe the hardware design of the MINOAS magnetic climbing robot. The system was designed to be lightweight in order to be safely usable during the inspection process. There are a lot of heavy weight systems magnetic inspection systems on the market and within the research community. Most of the inspection systems have the weight of several kilograms and need to be secured by safety ropes. One goal of our system design was to make the crawler an easy-to-use tool for the vessel surveying process. It should be usable without long training and become as usable as a hammer or a wrench.

For the locomotion on the ferrous frame of a sea-going vessel the system has two wheels which are equipped with 50 neodymium magnets each. In order to increase the adhesive power, every pair of the magnets is oriented in opposite magnetic field orientation. The system is controlled using a differential steering approach which allows the system to turn on the spot while climbing a vertical wall. On the tail is a third neodymium permanent magnet attached which stabilizes the system on the wall (cf. Figure 3).

The current version of the magnetic crawler has an elastic tail section which increased the maneuverability significantly compared to a static tail approach. The crawler can directly be controlled by the operator using a 35 MHz remote control (cf. Figure 4).

A wireless camera is attached to the crawler for on-line video transmission to the control station or a handheld video device. If no control station is used for on-line video storage, the operator can also use the handheld device to get an impression of what the robot is looking at (cf. Figure 4). During a normal vessel surveying process the area is usually poorly lit, because most of the vessel inspection takes place inside the vessel. For that reason the magnetic crawler carries its own light source which is attached on both sides of the on-board camera.

If no control station or position tracking is needed, the equipment of the base system is lightweight and man-portable. This is especially the case if a surveyor only wants to get a quick visual impression about some hard accessible parts of the vessel, e.g. at high altitude inside a vessel's cargo hold. Figure 5 shows the design of the lightweight crawler. Table I summarizes the technical data of the inspection system.

Dimensions (l x w x h) :	38 cm x 28 cm x 15 cm
Weight:	670 g
Velocity:	50 cm/sec on vertical ferrous surfaces
Actuation:	2 x 12 V DC gear motors
Battery:	11,1 V - 800 mAh - lithium polymer battery
Wheels:	2x50 neodymium magnets
Sensors:	Wireless video camera with internal light source
Control:	35 MHz remote control

TABLE I

TECHNICAL SPECIFICATION OF THE MINOAS LIGHTWEIGHT CRAWLER.

During a vessel surveying process, the visual information about the vessel's condition has to be recorded with the accurate time *when* the measurement was taken together with the spatial information *where* the data was acquired.

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

Fig. 4. Top: The magnetic crawler with the back cover removed. During the inspection a mobile display shows what the robot is looking at to the surveyor. Bottom: The magnetic crawler with the remote control and the portable video receiver.

Fig. 5. The concept of the magnetic crawler from a a 3D CAD perspective.

Because of the lightweight requirement, our mobile system is not able to carry laser range finders which usually find their application in mobile robotics.

SLAM (self-localization and mapping) approaches allow the mapping of a 3D area in combination with probability-based localization approaches. For the application inside a large sea vessel the application of such techniques is likely to fail because of the amount of data which has to be collected and because of the weight of the needed sensors. For the MINOAS approach we decided to use an external 3D tracking approach which makes use of a pan-tilt camera in combination with a single point laser range finder.

The tracking of the lightweight crawler is based on vision, thus the crawler has to be in line of sight of the tracking unit. The tracking itself is realized using a bright power LED, which is mounted on the back of the crawler (cf. Figure 6). The 3D tracker is able detect the light source on the back of the system an is therefore able to estimate the position of the crawler.

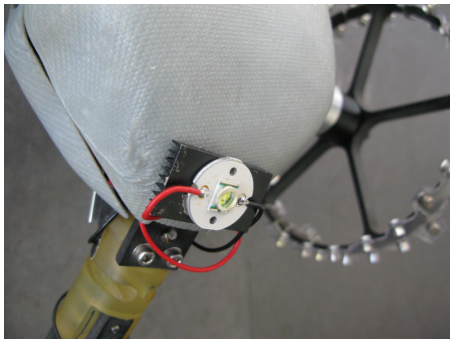


Fig. 6. The tracking target is attached to the back of the robot. It consists of a very bright power LED. This target is used for the visual tracker to aim the laser at the moving crawler.

B. 3D Localization

As mentioned above, the surveyor needs beside the visual data of the robot also the accurate position of *where* the data was acquired. Because self-localization is not feasible with on-board sensors, we have chosen the approach of an external portable tracking device. The main component is a custom designed pan-tilt unit with a high resolution firewire camera and two Dynamixel DX-117 servos. These are connected to a control station which does the image processing in order to track the crawler robot. The servos are controlled using a RS485 serial bus communication (cf. Figure 7).

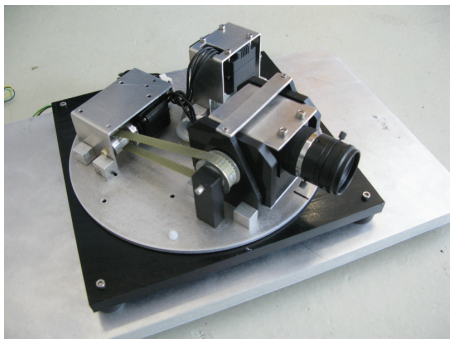


Fig. 7. The MINOAS 3D optical tracker without the attached laser range finder. A high resolution camera is used to track the magnetic crawler. The servo provides the system with pitch and yaw angle with respect to the trackers coordinate frame.

The camera detects the bright light source which is attached to the back of the mobile robot. The controller of the 3D tracking unit uses a blob tracking approach to track the LED on the moving crawler. The image is first filtered using two Gaussian filters which are subtracted from each other. The image is searched for the blob containing the

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

highest brightness level and the center of gravity of the bob is calculated. The tracking module keeps the center of gravity of the target blob in the middle of the tracking image, calculating the before mentioned tilt and yaw angle of the tracker. In order to get the accurate position of the crawler, i.e. the position of the magnetic crawler, the distance to the target has to be measured.

Attached to the visual tracking system is a one axis laser range finder (Leuze ODSL30, cf. Figure 8). The combination of two angles with the distance provides the 3D position of the crawler. The laser of the tracking device is calibrated to point on the back of the crawler, i.e. it measures the distance of the back cover to the 3D tracker.

The offset from the laser device to the rotation axis of the pan-tilt unit have to be calibrated. The calibration is done measuring several points along a straight wall along the two individual axes. The laser point is a visible spot, so the ground truth can be measured easily by hand.

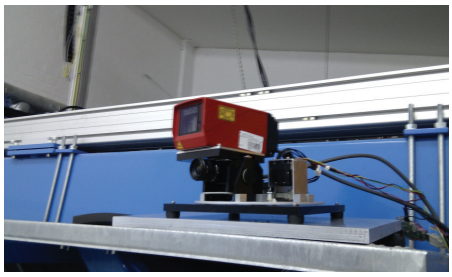


Fig. 8. The MINOAS 3D optical tracker with the Leuze ODSL30 attached. The device is directly connected to the control station using firewire for the image transmission and RS-485 for servo and laser data acquisition.

The driver of the tracking unit was developed using parts of the Robotic Operation System ROS [7]. ROS assures that all data transmitted have a time stamp which is important to combine all important data coming from different sensor sources.

C. User Interface and Data Storage

The user interface and the data storage are the back-end of the MINOAS inspection system. The control software module collects all information from the different sensor sources and combines them to time stamped data packages. More specifically, the information of the 3D tracker (giving the current position of the climbing robot) and the video images are combined using the time stamp of each individual sensor information.

Prior to the vessel inspection process,the position of the 3D tracker has to be aligned manually to 3D environment of the user interface. This assures that the 3D environment and the real environment have the same reference frame. In order to display the real time information about the crawler, a 3D ship model is added to the graphical user interface (cf. Figure 9). The 3D model of the inspection crawler is also added to the user interface to visualize the ongoing inspection process. The 3D model of the robot moves in the virtual 3D environment of the vessel and gives the surveyor an intuitive overview of the process.

The reason behind this approach is that the operator, e.g. the surveyor of the vessel can see on-line where the trajectory of the robot was recorded. Whenever the robot acquires new visual information, like damages of the coating or decay of the ship's frame, the information is allocated to the 3D ship model and the defect is annotated in the 3D environment as 3D objects. This allows an easy and intuitive access to the data. The video and image data itself is stored on a local file system. With all data we also associate where the data was acquired. The data together with the meta data are stored in a spatial content management database system. The key idea behind this is that the data is searchable by the location. The user can specify in the 3D ship model environment which data has to be accessed. This also includes a nearness search, i.e. the interface provides all relevant inspection data near a certain, user-specifiable position.

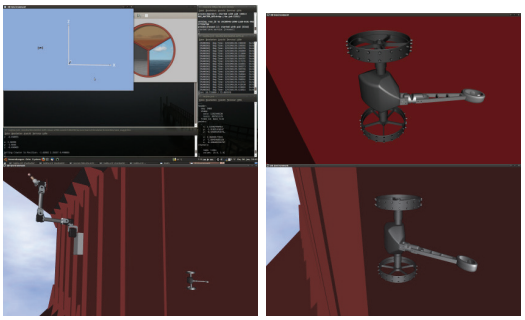


Fig. 9. Some screen shots of the MINOAS 3D user interface: The 3D model of the inspected vessel is loaded into the interface. The model of the inspection robot is projected into the 3D model based on the real-time position data. This gives the surveyor a visual 3D impression of the inspection data.

The user interface is developed independently of the type of inspection robot. In this work we presented a magnetic climbing robot as an example for a robotic vessel inspection system. Within the MINOAS project a variety of different sensor-equipped plattforms is envisaged [8].

The data within the spatial content management system is arranged in a hybrid metric/topological manner. The reason for this is that the tracker in the real world needs always a direct line of sight to the climbing robot. Intuitively, inside a large vessel with the size of a PanMax class ship (which has a length of around 300 Meter) is not quite possible to realize. Therefore each inspection spot has its own 3D reference frame. The reference frame has to be attached manually to the ship model as well as to the real world. The reference system is a metrical one, providing a 3D position in space with respect to the 3D trackers reference frame. Each reference frame itself is linked to only one part of the ship and the ship's 3D model. Every single part is than ordered topologically within a tree structure.

V. EXPERIMENTAL EVALUATION

At the current state of the MINOAS project, the tracker, the climbing robot and the user and control interface, which is combining the video data with the spatial 3D information,

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

are functional. The crawler is tele-operated while the real time position data is projected into the reference frame of the tracker. Figure 10 shows the experimental setup. The experiments are done on a steel wall inside the lab. The wall has a size of 1500 m x 3000 mm x 3 mm and is suitable to get the first tracking data. On the right side of Figure 10 the 3D Pose is visualized. The first experiments showed

Fig. 10. The experimental set up with the magnetic crawler and the control unit. The crawler is remotely operated while tracker measures the 3D pose information.

that the tracker is able to follow the crawler accurately. A problem of the time delay of 1-2 second showed up, which means that the tracker is always 1 second behind the crawler. It could also be seen that the accuracy is very dependent on the tracker to crawler distance. The angular accuracy of the tracker is depending on several system inherent issues, e.g. mechanical play, servo accuracy, camera resolution and camera objective. The laser accuracy is not significant in contrast to the angular accuracy, because the accuracy is below 1 cm. More critical is the angular accuracy. On the first experiments we had a combined position error of maximum 20 cm, given a distance of 7 meter between the tracking unit and the crawler. Because the laser measurement spot is visible on the wall, it is straight forward to measure the position error along the wall.

In the MINOAS project specification an accuracy requirement of 30 cm was defined. This means that at a tracking distance of 7 meter the accuracy is still within the specification. A systematic evaluation of the key parameters of the tracking unit is left for future work.

VI. CONCLUSION AND FUTURE WORK

In this work we gave a system overview of a magnetic climbing robot for marine vessel inspection. The system is part of a fleet of robot-based inspection systems which are currently developed within the EU-founded MINOAS project. The robot is supported by a custom designed 3D tracker which is based on vision and laser. The tracker provides the accurate 3D position of the robot during the inspection process.

The application data from the robot, i.e. the video stream and single images are combined in order to produce spatially and time tagged data. This is a mandatory requirement for the comparison of inspection data over the lifetime of a vessels. Future works of the project will include the extension of the localization approach to other systems which are currently developed by the project partners. Additional systems include a heavy magnetic crawler with an attached manipulator arm as well as an unmanned areal vehicle. All those system have to be tracked, localized and integrated into our framework.

Improvement will also be done in the tracking system because it may fail if the crawler is moving very close to another bright light spot. To prevent this, a defined frequency can be applied to the LED which than again can be synchronized with the image processing module on the tracker.

Additional work will be done towards system autonomy. Currently, the system is purely tele-operated, only the 3D position in the reference frame and within the 3D ship model is provided. In a next step the navigation of the climbing robot is going to be closed loop in combination with the tracker. During the project it is envisaged to provide an inspection pattern on a flat steel wall and to let the system gather visual data autonomously using a predefined motion pattern. The motion pattern or, more specifically, the points of interest, have to be defined by the surveyor using the virtual 3D vessel model.

REFERENCES

[1] W. Fischer and R. Siegwart, “Wheeled Pole-Climbing-Robot with High Payload Capability, Using a Clamping Mechanism which is Inspired by the Rope-Clamps in Human Climbing,” in *Emerging Trends in Mobile Robotics*, H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, Eds., Nagoya, 2010.

[2] S. Raut, A. Patwardhan, and R. Henry, “Design and Construction of Glossy Surface Climbing Biped Robot,” in *Emerging Trends in Mobile Robotics*, H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, Eds. Nagoya,Japan: World Scientific Publishing Co. Pte. Ltd., 2010, pp. 423–430.

[3] Y. Kawaguchi, I. Yoshida, H. Kurumatani, T. Kikuta, and Y. Yamada, “Internal pipe inspection robot,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1. IEEE, 1995, pp. 857–862.

[4] F. Tâche, W. Fischer, R. Siegwart, R. Moser, and F. Mondada, “Compact magnetic wheeled robot with high mobility for inspecting complex shaped pipe structures,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, Oct. 2007, pp. 261–266.

[5] M. Jung, D. Schmidt, and K. Berns, “Behavior-Based Obstacle Detection and Avoidance System for the Omnidirectional Wall-Climbing Robot CROMSCI,” in *Emerging Trends in Mobile Robotics*, H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, Eds. Nagoya, Japan: World Scientific Publishing Co. Pte. Ltd., 2010, pp. 73–80.

[6] G. Peters, D. Pagano, D. K. Liu, and K. Waldron, “A Prototype Climbing Robot for Inspection of Complex Ferrous Structures,” in *Emerging Trends in Mobile Robotics*, H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, Eds. Nagoya, Japan: World Scientific Publishing Co. Pte. Ltd., 2010, pp. 150–156.

[7] ROS, “Robot Operating System.” [Online]. Available: www.ros.org

[8] A. Ortiz, F. Bonnin, A. Gibbins, P. Apostolopoulou, W. Bateman, M. Eich, F. Spadoni, M. Caccia, and L. Drikos, “First steps towards a roboticized visual inspection system for vessels,” in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*. IEEE, 2010, pp. 1–6.

978-1-4577-0123-8/11/$26.00 ©2011 IEEE

A Magnetic Climbing Robot for Marine Inspection Services

Kristin Fondahl, DFKI GmbH, Bremen/Germany, kristin.fondahl@dfki.de
Markus Eich, DFKI GmbH, Bremen/Germany, markus.eich@dfki.de
Johannes Wollenberg, DFKI GmbH, Bremen/Germany, johannes.wollenberg@dfki.de
Frank Kirchner, DFKI GmbH, Bremen/Germany, frank.kirchner@dfki.de

Abstract

Currently, the inspection of sea-going vessels is performed manually. Ship surveyors do a visual inspection; in some cases they also use cameras and non-destructive testing methods. Prior to a ship surveying process a lot of scaffolding has to be provided in order to make every spot accessible for the surveyor. In this work a robotic system is presented, which is able to access many areas of a cargo hold of a ship and perform visual inspection without any scaffolding. The paper also describes how the position of the acquired data is estimated with an optical 3D tracking unit and how critical points on the hull can be marked via a remote controlled marker device. Furthermore first results of onboard tests with the system are provided.

1. Introduction

Marine vessels are subject to numerous and regular inspections and maintenance measures. Ship surveyors inspect the vessels on a regular basis. In most of cases, the surveyor performs only a visual inspection. In order to reach each spot on the ship, scaffolding has to be erected in the cargo holds. Typical heights of cargo holds are 15-20 m. Fig.1 shows two parts of a cargo hold of a bulk carrier with different wall structures. The installation of the scaffolding usually takes several days, before the surveyor can start the inspection process. Every day the ship stays in the dock and out of service results in a significant loss of money for the ship owner, making this (currently necessary) preparation time is very expensive. The EU-funded R&D project MINOAS (**M**arine **IN**spection r**O**botic **A**ssistant **S**ystem) addresses this challenge in an attempt to develop concepts for the automation of the ship inspection process.

Fig. 1: One of the four cargo holds of a 10,000 DWT bulk carrier.

The key idea of the project is to develop and test a fleet of semi-autonomous robots which can provide visual data as well as thickness measurement data to the surveyor without the need for setting up scaffolding prior to the inspection process. While the idea to employ robotic agents for the inspection of hazardous environments is not new, see *Sawada et al. (1991), Kawaguchi et al. (1995), Pack et al. (1997*), a fully autonomous inspection of a cargo ship still is a long-term goal. The idea of the project is not to develop an autonomous inspection but rather focus on robotic tools that can enhance and simplify the current inspection process.

One of the fleet's robots is a lightweight magnetic crawler which is able to climb along the vertical walls of a vessel. The crawler was introduced in *Vögele et al. (2010)* and the first design concept was described in *Eich and Vögele (2011)*. The robot provides a live video stream as well as offline images of the ship for later inspection. Apart from the locomotion capability of the inspection system, it is mandatory for the inspection process that the data is localized, i.e. the position of images and video streams are known within the vessel. For this purpose a 3D tracking unit was developed which acquires the position of the magnetic crawler in real-time. This allows a meaningful comparison of inspection data over a vessel's lifetime, because the exact position of the data can be stored and therefore replicated. A 3D user interface provides the necessary information to the surveyor and allows access to all acquired data within a 3D view.

2. The Lightweight Crawler: Design and Control

The basic design of the lightweight magnetic crawler was described in *Eich and Vögele (2011)*. In the meantime further adaptations to the system have been made to increase the robustness of the locomotion as well as the maneuverability.

As the silicon rubber used as tread in the previous version showed high temperature dependence and hindered successful test runs during the cold season, a less temperature dependent rubber material was tested and integrated. The galvanized rubber is water-jet cut to form distinct anchor points for the magnet mountings thus providing a better adaptation to the wall and an overall higher adhesive force. Neoprene foam mounted in between two strips of rubber provides additional friction. The three strips are glued onto a polycarbonate sheet that connects the tread to the rim, Fig.2 (right). This new assembly increases the traction especially in cold weather, but it also increases the weight of the wheel assembly. To compensate for this additional weight stronger neodymium magnets are used.

Adjustments were also made to the tail. The form was altered into a bow-like structure to enable the transition from vertical to horizontal planes without detaching the tail. Furthermore two magnet rings replaced the previous sliding magnet and serve as passive adaptive rear-wheels. These provide strong adhesion in any orientation yet little to no friction. This adaptation increased the maneuverability and robustness of the crawler drastically.

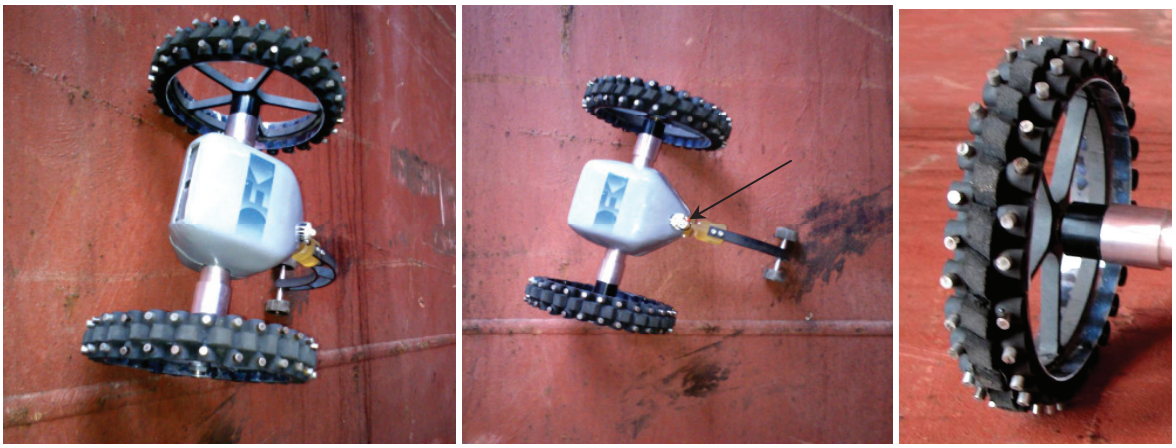


Fig. 2: Lightweight magnetic crawler after the onboard tests. The arrow indicates the tracking LED used for localization. The image on the right is a close-up of the new wheel structure.

3. Position Estimation Using Optical Tracking

While the components of the tracking unit did not change, compared to the previously described version, *Eich and Vögele (2011*), the current design is more compact and mountable to a tripod which allows an easy deployment inside the vessel. The servos and cabling are integrated inside the unit to provide a higher robustness against external influences such as dirt. To achieve a level stance on the tripod a circular level is integrated into the unit. See Fig.3 for an overview of the design.

Fig. 3: System overview of the tracking unit that localizes the inspection robot within the vessel. The camera is mounted above the laser range finder and both are actuated with the servo motors situated beneath the covers.

In order to localize the crawler inside the vessel information from the two Dynamixel RX-28 servo motors, one Hokuyo laser scanner and one monocular Guppy F-036C camera are fused. The LED mounted on the crawler, Fig.2, is tracked by the monocular camera using Difference of Gaussian, *Lowe (2004)*. The discrepancy between the current and desired position is mapped to servo motor commands to hold the crawler in focus. The 3D position is calculated based on distance measurements provided by the laser and current angles of the servo motors resulting in a 3-dimensional point which is sent to the user interface described in Section 4.

Using a laser range finder instead of a stereo camera rig for depth measurements saves computation time and is more accurate on larger distances. It is also more practical in our case, since vessels provide relatively homogeneous image content which is generally not beneficial for stereo vision, *Barnard (1987)*.

The external position mapping depends on the position of the tracking unit. To generate replicable data over time the exact location of the tracking unit inside the vessel must be known.

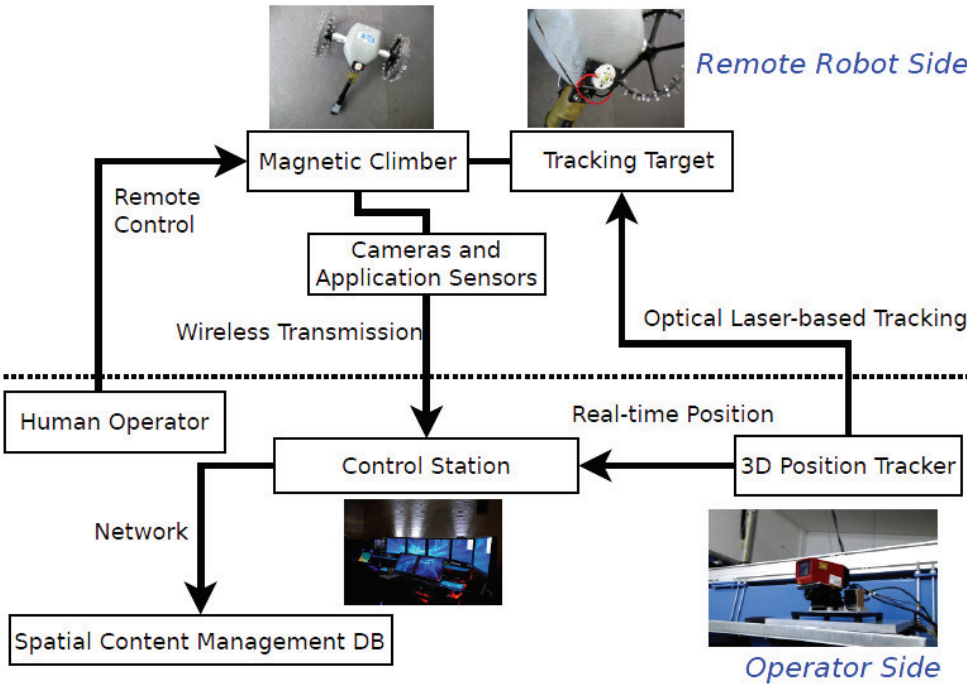


Fig. 4: The 3D tracking concept

The tracking system works together with the magnetic crawler as one inspection unit, Fig.4. The robot platform is operated via a remote control. The onboard video device transmits the data using a wireless 2.4 GHz connection. The position and the video data are stored together in a spatial content management system.

3.1. Laboratory experiments

The first test of the tracking unit was to estimate the positioning accuracy of the tracking system under laboratory conditions. The magnet crawler was remotely controlled using the 35 MHz control unit. The crawler had an ultra bright LED (910 Lm, 12 W) attached on the back which was detectable by the 3D tracker. The tracker was programmed to follow the LED with the tracking unit. The laser, which is attached to the camera pan-tilt, gives the distance to the target. Based on the two pan tilt angles and the distance the pose can be calculated via the law of sine. The point measured is always in the reference frame of the tracker. The distance of the magnet crawler to the tracker was 3 m and 4.5 m. It was tested if the tracker was able to follow the target. The error between the laser spot emitted by the laser scanner as reference point and the bright LED was measured in cm. The crawler was moved and the measurement was taken as the laser reached the closest spot near the crawler. The objective used for the tracker has a focal length of 8 mm. The nominal accuracy of the laser range finder is 1 cm. The tracker was able to follow the magnet crawler on the test wall.

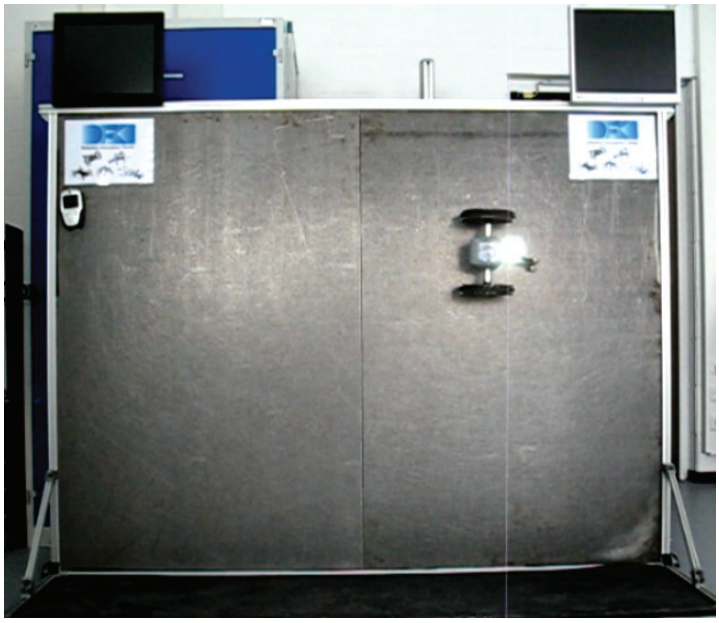


Fig. 5: The DFKI lightweight crawler on the test wall with the tracking LED switched on.

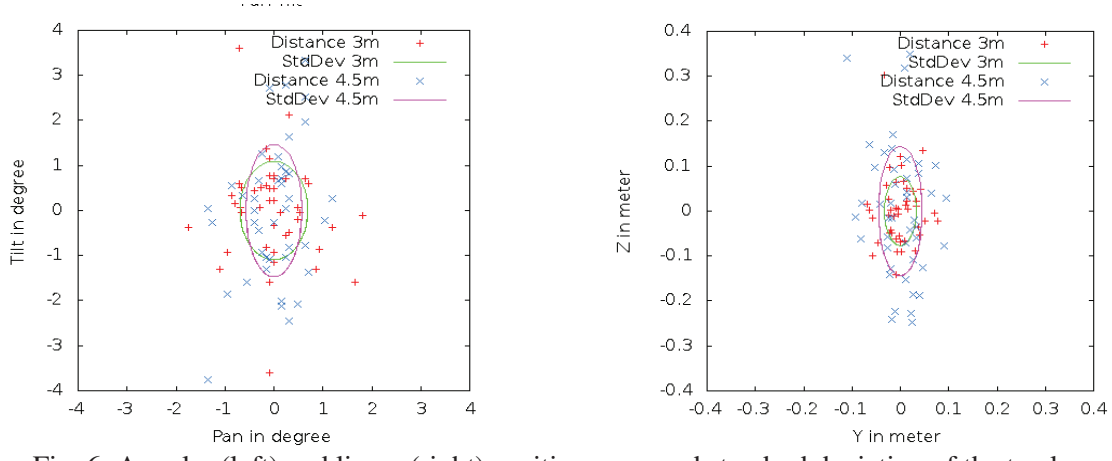


Fig. 6: Angular (left) and linear (right) position error and standard deviation of the tracker.

The error in the angles and distance is depicted in Fig.6. The experiment showed that the distance error was within the range of 10 cm at a distance of 3 m and within 20 cm range at a distance of 4.5 m. This corresponds to an error of ± 2° at the 3 m distance and ± 3° at the 4.5 m distance.

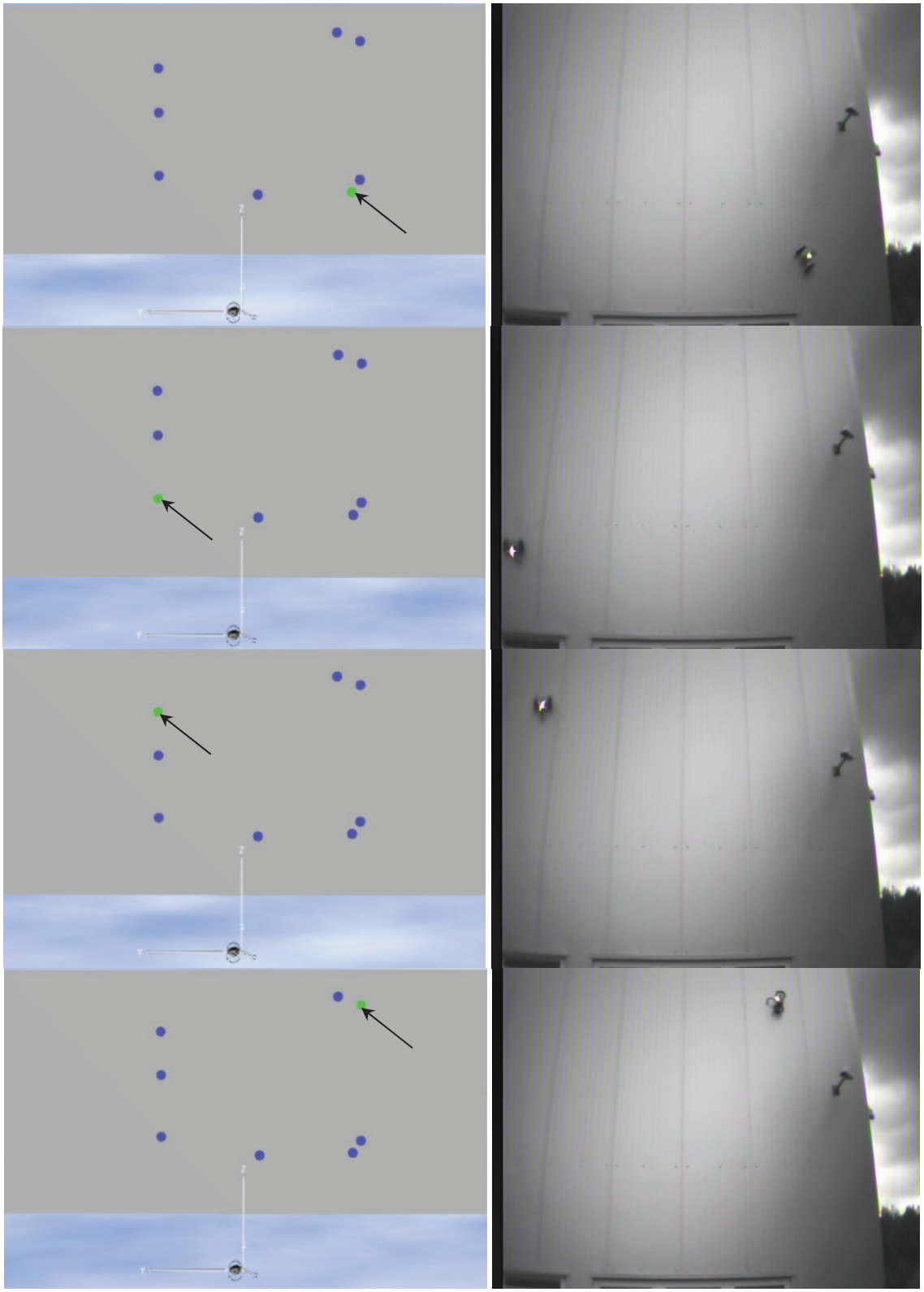


Fig. 7: Tracking results. Left: The tracked position in the model is indicated by the program with a green dot and marked with an arrow in the images. Right: The real position of the system.

Further testing was carried out on an outer metallic wall with a distance of 6 m to the tracking unit and a larger width and height to test a wider range of the tracker. The position error of the tracking unit in these experiments could not be measured exactly due to the height, but lay in the range of 30 cm for all angles. Fig.7 compares the tracked and real positions of the lightweight crawler in these trials.

3.2. User Interface and 3D Representation

For a user friendly inspection process, a graphical user interface (GUI) was added to the system where all data is easily accessible. Since positional information sent by the tracking unit and visual data sent by the camera are transmitted separately the user interface synchronizes all incoming data based on time stamps and constructs data items containing aligned positional and visual data. For an overview of the process, see Fig.8. ROS (Robot Operating System) is a communication middleware for robotic applications and takes care of the sensor data processing, *Quigley et al. (2009)*. In the left part of the interface, Fig.9, data items collected while the crawler moves along the wall are displayed in a list. In the right part, a 3D view of the vessel including a blob-like visualization of data items is given. It is possible to inspect data by selection either inside the list or directly inside the 3D environment, which in the latter case opens the corresponding item on the left. Data items are organized into "runs" whereby a run represents an entire acquisition process while the crawler moves along the wall. This assists inspection of vessels over their life time in making a comparison of data from different time periods possible. Since the availability and interchangeability of data is a common problem in the inspection of a vessel, *Bijwaard et al. (2009)*, standard XML-Files are used to save all information concerning one run. As the crawler provides offline visual data as well, an import for videos is available which automatically synchronizes the input video with the temporally closest item in the currently considered run. The timestamp of the video is extracted from its meta-data.

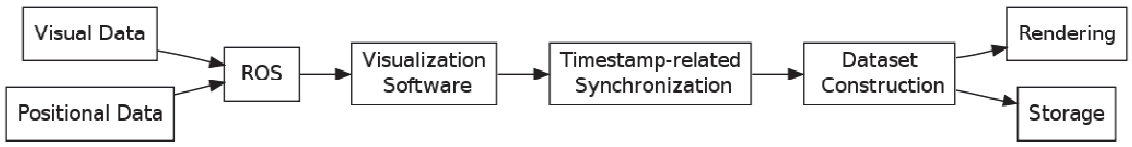


Fig. 8: Workflow of the visualization

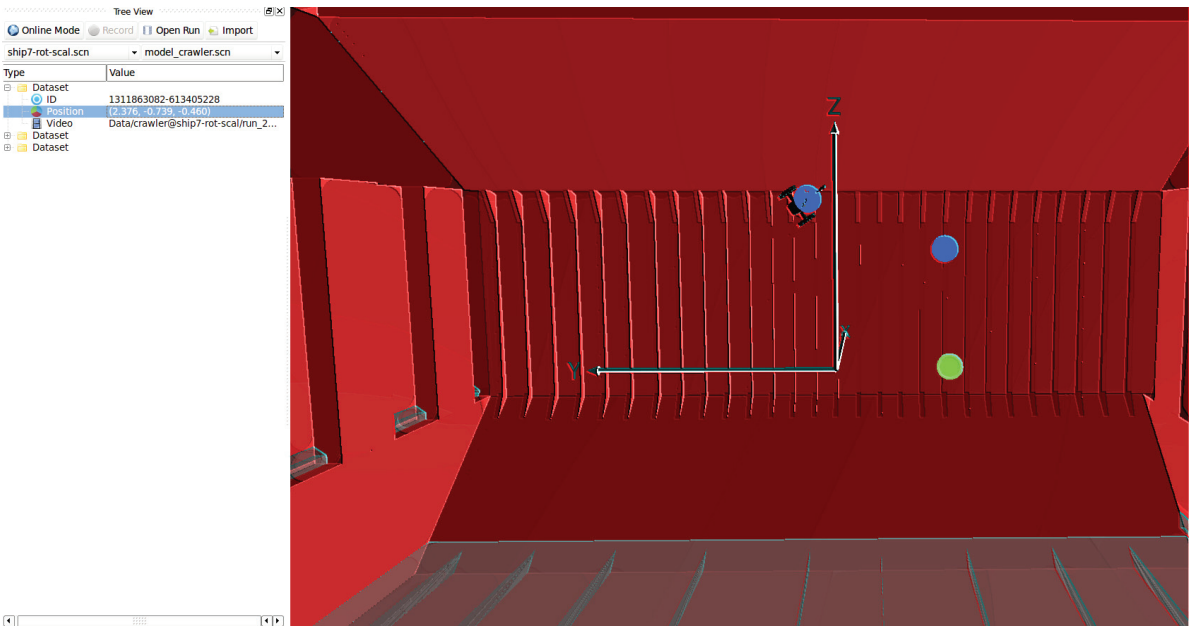


Fig. 9: GUI displaying three data items. One item is selected for further inspection.

4. Marking Defects on a Vessels Hull

During the first two stages of the inspection process, the surveyor will make a preliminary evaluation of the corrosion and defects of the ship hull. In some cases the surveyor wants to mark the defects found during this process for later inspection. This is currently done with a pen or paint, Fig.10.

Fig. 10: Defects on a vessel are marked directly by the surveyor

A mobile inspection robot should also be able to mark defects. As proof of concept, the magnetic climbing robot was equipped with an actuated micro pump, which contains varnish to be sprayed on the surface. The pump spray container can be equipped with a variety of inks, to account for different surface colors and structures in the ship. It is actuated via a small electrical motor and can be triggered remotely by the operator. The device is stand alone and provides an interface for triggering the spraying process and for a 12 V-DC power input. In a first step, this device was integrated on the MINOAS lightweight crawler. Integration in the heavy-weight crawler of the MINOAS project is possible, as the marker unit is self-contained. The unit was attached at the bottom of the lightweight crawler and is pointing towards the ground/wall, Fig.11. The motor was linked via a control unit to the RC-receiver of the robot, so that it could be triggered with the same remote that controls the lightweight crawler.

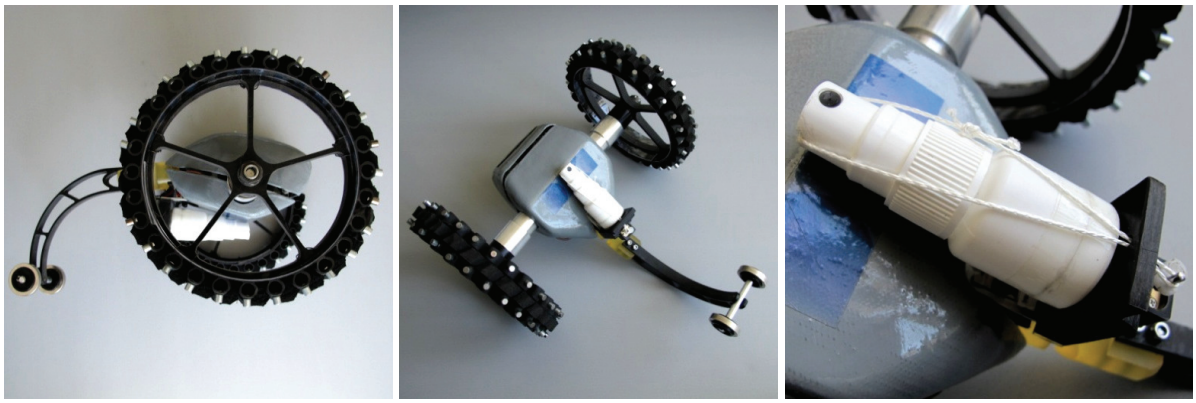


Fig. 11: The lightweight crawler with the marking unit attached

Test trials were carried out with the pump-spray unit attached under the lightweight crawler robot to provide an upright position of the paint container during the tests. For the test runs an acrylic water-diluted varnish was used. In one of the runs the paint was applied directly to a metal surface to test the adherence of the ink to the surface. In the following test runs, a sheet of paper was used to cover the metal surface and avoid intensive cleaning.

4.1. Experimental Results

The acrylic varnish used for the test runs was diluted to a very low viscosity, so as not to clog the outlet of the spray unit. This led to running of the paint on the vertical wall, if too much paint was

used in one position. Otherwise the spray unit showed good results for continuous marking as well as for single point marking on both surfaces (metal and paper), Fig.12.

Fig. 12: Experimental results for lightweight crawler and spray unit on a metallic (left) and paper surface (right)

5. Ship Trials and Lessons Learned

To evaluate the performance in a realistic environment, the lightweight crawler and the tracking unit were tested in the cargo hull of the 10,000 DWT bulk carrier “Alaya”, Fig.13. The tracking test was performed inside one cargo hold of this vessel. The equipment was placed on the bottom of the cargo hold and the tracker was mounted on a tripod. The magnet crawler was equipped with the tracking LED in order to be followed by the tracker. Fig.14 shows the testing environment.

The magnetic crawler was tested on various walls with different properties. The adhesion forces on the vertical walls did generally not suffice for robust maneuvering, while a simple ascent was possible on most walls. On a sloped wall with an inclination of approximately 30° to the vertical plane, Fig.14, the crawler could navigate easily even over small obstacles such as welding seams or dents. This shows that the current design of the magnet crawler is too heavy for the magnetic force provided by its magnets or – vice versa – the magnetic adhesion force is too small for the robot’s weight. It also indicates that the testing surfaces for which the crawler was optimized in the lab do not match the real

world accurately enough. Furthermore the magnetic crawler had problems with transitions between different planes. This was expected as difficulties with these maneuvers already arose in the lab. A test of the marking unit inside the ship was not possible to avoid a contamination of the cargo hold.

Fig. 13: The bulk carrier “Alaya”

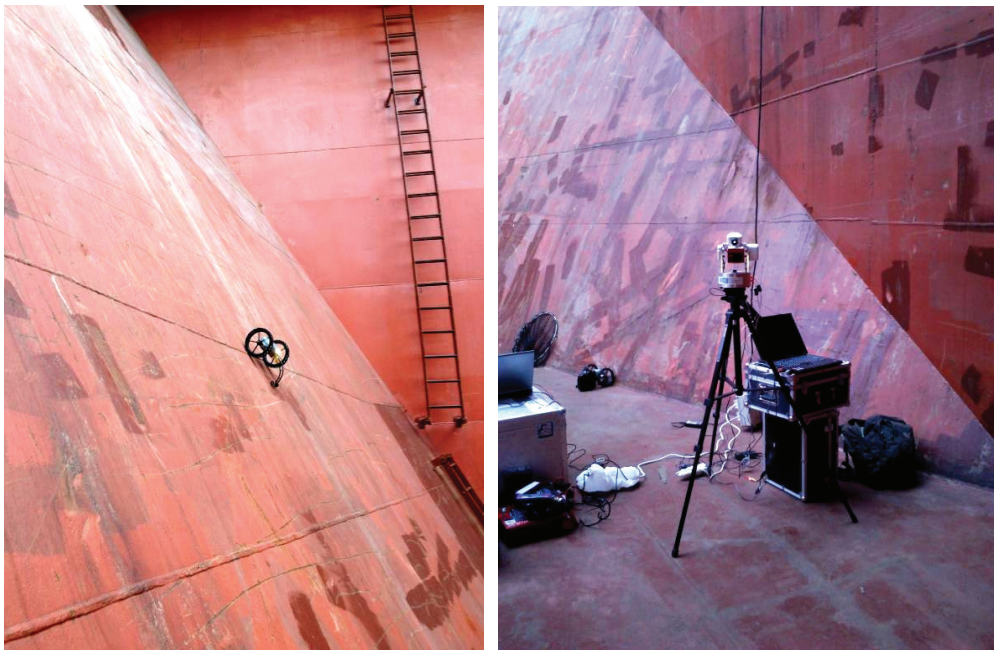


Fig. 14: The field test environment inside the cargo hold showing the lightweight crawler on a side wall of the hold and the setup of the tracking unit on its bottom

While the transmission of the 35 MHz RC signal to control the robot worked reliably even in a distance of about 20 m, the transmission of the live video stream in the 2.4 GHz range was very susceptible to noise. In regular conditions at a distance of a few meters the video signal was normally sufficient, though already noisy. But once other sources of electromagnetic noise were in use nearby, such as a hydraulic pump used during the ongoing repair process, the video signal was lost completely.

The tracking unit could follow the crawler reliably, as long as a certain speed limit was not exceeded. Nevertheless the tracking error remained quite high, detecting the position of the crawler in the 0.5 m range at a maximum distance of 15 m. As the tracking unit only adapts its orientation as soon as the crawler is almost out of focus and then only in one direction (no two servos acting at one time) the

tracking curve provided by the unit is a step curve, while the crawler itself can follow smooth lines. The data matching of the acquired images to the detected location worked well, although only one measurement could be completed due to the problems with the video transmission.

6. Conclusion and Future Work

The new developments in the inspection system presented include an optimized mechanical design of both the crawler robot itself and the tracking unit. Furthermore the software to localize the data taken by the lightweight crawler and its graphical representation for the user is introduced. The presented marking unit can provide a way to label defects inside the hull for later inspection. The lab-trials and the real-world tests aboard the "Alaya" showed some issues that need to be corrected, before an application of this robotic system becomes a useful addition to conventional inspections.

On the mechanical side the robot needs stronger magnets to provide a robust attachment to the steel walls in any orientation. Simply replacing the current magnets with stronger ones though, might lead to different issues that have to be considered, such as a decreasing motor speed due to higher loads or a possible failure of the current mounting system resulting in a loss of the magnets during the runs. To test possible outcomes the testing surface in the lab is to be adapted to resemble an actual ship wall more closely and allow for meaningful experiments without the need to board a ship. Stronger magnets may also help to transit from a horizontal to a vertical surface or between two vertical surfaces. Otherwise a new wheel design, e.g. *Tâche et al. (2009),* might help overcome these problems.

To make a use of the marking unit in a real world environment, it needs to be equipped with a more suitable varnish for metallic surfaces. This varnish must not clog the outlet of the spray container and has to be suitable for multiple if not all surfaces in a ship. Another option for a marking system may be the use of a servo actuated marker pen.

The transmission problems of the video images have to be prevented, as other repairs cannot be put to a hold during the inspections. Therefore a new video transmission was later integrated into the robot with a 5.8 GHz submission rate and 2.5 times stronger signal. The transmission remains to be tested onboard a ship but the noise ratio inside the lab decreased drastically with this setup.

On the software side the tracking algorithms need to be optimized and sped up to enable the tracking unit to follow the robot robustly at all times. This could be achieved by using for instance a particle filter which estimates the motion before the motion is executed, *Fox et al. (1999*).

The synchronization and matching of the sensor data with the localization data works well and reliably. Yet some problems remain: To actually use the data, an accurate computer model of the inspected ship is needed, which, oftentimes, is not available for the inspection process. Another robot of the MINOAS fleet might provide this model in later adaptations, namely the flying inspection unit, *Ortiz et al. (2011)*. Furthermore the tracking unit needs reproducible anchor points inside the ship. This might be put into practice by welding markers (i.e. screw nuts) to the anchor points or otherwise marking them inside the ship. Or it might be possible to map the anchor points virtually and retrieve them during the inspection process by measurements from certain landmarks in the hull.

Nevertheless after the optimization of these factors the lightweight crawler can serve as a useful tool during the inspection of large cargo holds or even (with a watertight cover) for the outer hull. Before that can happen not only the technical problems need to be solved but also a wider acceptance of the reliability of such robotic tools for the commercial use has to be sought.

Acknowledgments

This work has been funded by the European Commission under the project MINOAS (SCP8-GA-2009-233715).

References

BARNARD, S.T. (1987), *A stochastic approach to stereo vision*, In Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, Morgan Kaufmann Publ., pp.21-25.

BIJWAARD, G. E.; KNAPP, S. (2009), *Analysis of ship life cycles − The impact of economic cycles and ship inspections*, Marine Policy 33/2, pp.350-369

EICH, M.; VÖGELE, T. (2011), *Design and control of a lightweight magnetic climbing robot for vessel inspection*, 19th Mediterranean Conf. Control & Automation (MED), Corfu, pp.1200-1205

FOX, D.; BURGARD, W.; DELLAERT, F.; THRUN, S. (1999), *Monte Carlo localization: Efficient position estimation for mobile robots*, 16th Nat. Conf. Artificial Intelligence, Orlando

KAWAGUCHI, Y.; YOSHIDA, I.; KURUMATANI, H.; KIKUTA, T.; YAMADA, Y. (1995), *Internal pipe inspection robot*, IEEE Int. Conf. Robotics and Automation, Nagoya, Vol.1, pp.857-862

LOWE, D.G. (2004), *Distinctive image features from scale-invariant keypoints*, Int. J. Computer Vision 60/2, pp.91-110.

ORTIZ, A.; BONNIN-PASCUAL, F.; GARCIA-FIDALGO, E.; BELTRAN, J.P. (2011), *A control software architecture for autonomous unmanned vehicles inspired in generic components*, 19th Mediterranean Conf. Control & Automation (MED), Corfu, pp.1217-1222

PACK, R.T.; CHRISTOPHER, J.L.; KAWAMURA, K. (1997), A Rubbertuator-based structure-climbing inspection robot, IEEE Int. Conf. Robotics and Automation, Albuquerque, Vol.3, pp.1869-1874

QUIGLEY, M.; GERKEY, B.; CONLEY, K.; FAUST, J.; FOOTE, T.; LEIBS, J.; BERGER, E.; WHEELER, R.; NG, A. (2009), *ROS: An open-source robot operating system*, IEEE Int. Conf. Robotics and Automation, Kobe, pp.151-170

SAWADA, J.; KUSUMOTO, K.; MAIKAWA, Y.; MUNAKATA, T.; ISHIKAWA, Y. (1991), *A mobile robot for inspection of power transmission lines*, IEEE Trans. Power Delivery 6/1, pp.309-315

TÂCHE, F.; FISCHER, W.; CAPRARI, G.; SIEGWART, R.; MOSER, R.; MONDADA, F. (2009), *Magnebike: A magnetic wheeled robot with high mobility for inspecting complex shaped structures*, J. Field Robotics 26(5), pp.453-476

VÖGELE, T.; EICH, M.; GRIMMINGER, F.; FONDAHL, K. (2010), A hybrid legged wheel climbing robot for marine inspection, 13th Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR), Nagoya, pp.1-8

A Robot Application for Marine Vessel Inspection

Markus Eich
DFKI - Robotics Innovation Center
28359 Bremen, Germany
markus.eich@dfki.de

Francisco Bonnin, **Emilio Garcia** and **Alberto Ortiz**
University of the Balearic Islands
07122 Palma de Mallorca, Spain
alberto.ortiz@uib.es

Gabriele Bruzzone
ISSIA - CNR (Via De Marini 6)
16149 Genoa, Italy
gabriele.bruzzone@ge.issia.cnr.it

Yannis Koveos
Glafcos Marine Ltd.
18537 Piraeus, Greece
ykovaios@glafcos-marine.com

Frank Kirchner
DFKI - Robotics Innovation Center
28359 Bremen, Germany
frank.kirchner@dfki.de

Abstract

Seagoing vessels have to undergo regular inspections, which are currently performed manually by ship surveyors. The main cost factor in a ship inspection is to provide access to the different areas of the ship, since the surveyor has to be close to the inspected parts, usually within hand's reach, either to perform a visual analysis or to take thickness measurements. The access to the structural elements in cargo holds, e.g., bulkheads, is normally provided by staging or by 'cherry-picking' cranes. To make ship inspections safer and more cost-efficient, we have introduced new inspection methods, tools, and systems, which have been evaluated in field trials, particularly focusing on cargo holds. More precisely, two *magnetic climbing robots* and a *micro-aerial vehicle*, which are able to assist the surveyor during the inspection, are introduced. Since localization of inspection data is mandatory for the surveyor, we also introduce an external *localization system* which has been verified in field trials, using a climbing inspection robot. Furthermore, the inspection data collected by the robotic systems is organized and handled by a *spatial content management system* which permits comparing the inspection data of one survey with that from another, as well as documenting the ship inspection when the robot team is used. *Image-based defect detection* is addressed by proposing an integrated solution for detecting corrosion and cracks. The systems' performance is reported, as well as conclusions on their usability, all in accordance with the output of field trials performed onboard two different vessels under real inspection conditions.

1 Introduction

For obvious reasons, large tonnage vessels, such as bulk carriers, dry cargo ships, or tankers (see Fig. 1), undergo regular inspections to prevent structural damage that can compromise the vessel's integrity. These inspections are usually performed in accordance with an inspection programme that depends on the requirements of the so-called *classification societies* (in short, *the classes*), and comprise visual close-up surveys as well as thickness measurements obtained by means of non-destructive testing methods (NDT) (Tanneberger and Grasso, 2011). For a close-up survey, the surveyor has to get within hand's reach of the part under observation for adequate visual inspection. Structural damage, pitting, and corrosion are visually estimated based on the experience of the surveyor, and the inspection process is usually documented, using cameras to take images, chalk and pen for defect marking, and a clipboard for note taking. Some solutions based on unmanned underwater vehicles (UUV) have been proposed lately for the inspection of underwater areas, e.g., the hybrid ROV solution by ECA Robotics (http://www.eca-robotics.com), the

Figure 1: Illustration of traditional inspection methods: (left) general cargo ship, (center) staging previous to inspection, and (right) cherry-picking. [Source: (left,center) Lloyd's Register, (right) http://www.standard-club.com]

HAUV by Bluefin (Kaess et al., 2010), and the VideoRay ROVs (http://www.videoray.com/). The first two systems are primarily intended for underwater hull survey while the latter is for water tank inspection.

Regarding the inspection of dry areas, providing access to the relevant parts of the ship, e.g. inside the cargo hold of a bulk carrier, is the most time-consuming part of the inspection process. As can be seen in Fig. 1(center), traditional ship surveying methods comprise, prior to the survey, the installation of scaffolding to allow the surveyor to inspect structures such as bulkheads, beams, stiffeners and brackets, which are usually several meters above the bottom of the hold. Besides the scaffolding, "cherry-picking" methods are also employed: in this case, the surveyor reaches the point of interest inside a basket, transported by a tower crane or by a hydraulic arm (Fig. 1(right)). Clearly, these procedures can be dangerous for the surveyor. For this reason, and because of the high costs of gaining access to a ship for inspection, the EU-funded research project MINOAS (Caccia et al., 2010b) set up a consortium to introduce robots into the ship surveying process. The basis of the consortium's expertise comprised two classification societies, different marine service providers, and partners involved in robotics research. The key idea was to introduce a set of novel tools to enhance the ship surveying process. For the interested reader, a more detailed discussion of the application scenario can be found in (Ortiz et al., 2010).

This paper reports results of the introduction of heterogeneous robots to the area of close-up surveys of the structural elements of large-tonnage vessels, where most of the work is still performed manually. The effort is a mixture of novelty and integration, and the proportion of each is different for every platform. Nevertheless, the main contribution is the fully integrated inspection system, covering all the stages of an inspection procedure based on the use of robots and supporting software, something that did not exist before the project MINOAS. We introduce a heterogeneous robot team with different locomotion abilities, namely a micro-aerial vehicle (MAV), a lightweight magnetic crawler supported by an external positioning unit, and a heavyweight magnetic crawler equipped with a manipulator arm for thickness measurement. Two additional systems were developed to assist the surveyor: a spatial content management system (SCMS) to host and present in a centralized way the inspection data collected by all the platforms, and a visual defect detection solution for automatic defect detection. They are all described in the following sections.

2 Re-Engineered Inspection Procedure

On the basis of the expertise of the classes involved and the maritime industry, and as part of the MINOAS project working plan, a total of three stages were defined in order to implement an inspection procedure based on robots and compatible with the requirements of the classes (Tanneberger and Grasso, 2011).

Stage 1: *Fast visual inspection overview*. The goal of this stage is to cover large areas of selected parts of the inner hull, supplying visual inspection data to get a rough overview about the state of the vessel, and searching for defects, such as coating breakdown, corrosion and cracks. The images collected must be tagged with pose information since the areas of interest for Stages 2 and 3 are visually selected in this phase. Due to the aforementioned, an aerial platform turns out to be the best option. In more detail, the vehicle must be able to perform vertical, stationary and low speed flight in indoor environments.

Stage 2: *Visual close-up survey*. The procedure in Stage 2 is to get a better impression of the coating and the possible damages, and mark the defective areas in order to repair the damage or perform thickness measurements. The vehicle must be capable of moving on horizontal, sloped, and vertical ferromagnetic surfaces. At this stage, the camera of the robot has to be steady and very close to the structural parts in order to provide high resolution, high quality images. Proper lighting of the area under inspection should be provided. Marking the defects directly on the structure, once confirmed by the surveyor, would be an optional feature of the robot. The acquired images have to be tagged with positioning information. Finally, ease of system setup is also required. Under these constraints, a lightweight, magnetic crawler was considered to be a suitable platform.

Stage 3: *Thickness measurement collection*. At selected parts identified during Stage 2, the thickness of the material is determined by taking measurements at selected points of the hull. The locomotion speed is not important, and the setup requirements are permitted to be more onerous than for Stages 1 and 2. The main goal of Stage 3 is to collect thickness measurements at selected points of the hull structures. Motion over horizontal, sloped and vertical ferromagnetic surfaces is still required. Compared to Stage 2, a higher payload capability is required in order to transport thickness measurement devices and a suitable manipulator arm. The climbing robot has to be able to move inside shell frames of the type shown in Fig. 2. Constraints on the vehicle size are imposed by the need of maneuvering between the T-shaped shell frames of the size indicated in Fig. 2(right). A heavyweight inspection system was selected to fulfill the requirements for this stage. Table 1 specifies the relation between the different systems and the mission stage where they are used.

Table 1: Relation between systems and inspection stages

System	Stage 1	Stage 2	Stage 3
Aerial platform	×		
Lighweight inspection system (crawler & localization device)		×	
Heavyweight inspection system (crawler & NDT thickness measurement device)			×
Defect detection system	×	×	
Spatial contents management system	×	×	×

The MINOAS inspection procedure places suitable robots into the inspection process, but is generic enough to cover all the requirements of a traditional close-up survey, and becomes therefore the basis for a fully robot-based inspection procedure for marine vessels. Nevertheless, since a very significant fraction of the inspection effort at the structural level (as specified by the rules set for the classes) is spent on cargo holds and on the different elements they involve, e.g., bulkheads, stiffeners, and cross-decks, the developments and tests in this paper refer mostly to cargo holds.

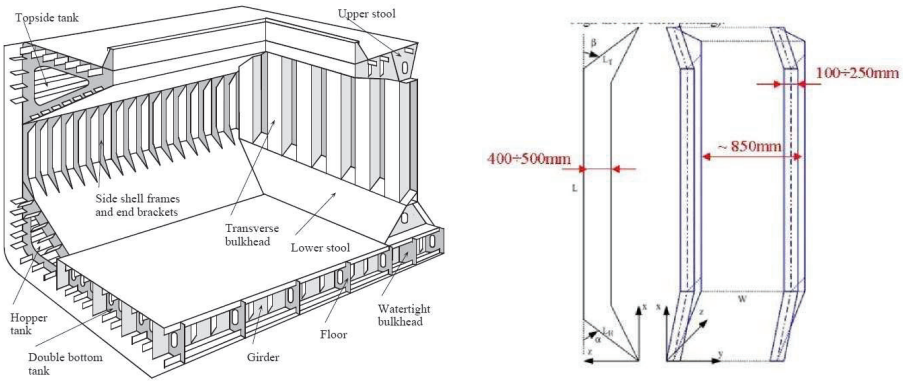


Figure 2: (left) Bulk carrier hold. (right) Shape and size of a shell frame.

3 MINOAS Inspection Platforms

3.1 Related Work on Inspection Platforms

This section reviews previous work that could match the requirements for the MINOAS platforms. Due to their different natures, aerial platforms and crawling systems are considered separately, the latter referring jointly to both the lightweight and the heavyweight inspection systems.

3.1.1 Aerial platforms

MAVs have increased their popularity as robotic platforms in recent years. Their development has been driven by commercial, research, government, and military purposes. This kind of vehicle allows accessing hazardous environments, usually difficult to reach by humans or ground vehicles. These robots are an adequate solution for inspection tasks at, e.g., remote or safety-compromised areas. In order for these platforms to achieve autonomy, a full navigation solution is required. Lately, a number of navigation solutions have been proposed for multi-rotors, including platform stabilization, self-localization, mapping, and obstacle avoidance. They mainly differ in the sensors used to solve these tasks, the amount of processing that is performed onboard/off-board, and the assumptions made about the environment. The laser scanner has been extensively used due to its accuracy and speed. For instance, (Dryanovski et al., 2013; Grzonka et al., 2012) propose full navigation systems using laser scan matching and IMU fusion for motion estimation embedded within SLAM frameworks that enable such MAVs to operate indoors. In (Bachrach et al., 2011; Dryanovski et al., 2013) a multi-level approach is described for 3D mapping tasks. Infrared or ultrasound sensors are other possibilities for implementing navigation solutions. Although they typically have less accuracy and require higher noise tolerance, several researchers (Bouabdallah et al., 2005; Matsue et al., 2005; Roberts et al., 2007) have used them to perform navigation tasks in indoor environments, since they are a cheaper option than laser scanners. Vision based navigation has become quite popular for MAVs lately. Cameras' success in general robotics comes mainly from the richness of the sensor data supplied, combined with their low weight, low power designs, and relatively low prices. Nevertheless, for the particular case of MAVs, the associated higher computational cost has made researchers find optimized solutions that can run over low-power processors. Among the most recent papers published in this regard, some propose visual SLAM solutions based on feature tracking, either adopting a frontal mono or stereo camera configuration, e.g. (Fraundorfer et al., 2012), or choosing a ground-looking orientation, e.g. (Chowdhary et al., 2013). Others focus on efficient implementations of optical flow calculations, either dense or sparse, and mostly from ground-looking cameras, e.g. (Zingg et al., 2010), or develop methods for landing, tracking and taking off using passive, e.g. (Meier et al., 2012), or active markers, e.g. (Wenzel et al., 2011), also using a ground-looking camera.

3.1.2 Climbing robot platforms

The field of wall climbing robots, which naturally turns out to be relevant for this application, has received a certain amount of attention since the late 1990s (Silva and Tenreiro, 2010). Referring specifically to marine applications, a robotic system which was developed to inspect hot welding seams was introduced by (Shang et al., 2008). This small-sized system has a weight of 30 kg, requires a safety-rope during operation, and uses an infrared sensor to check the temperature of the hot seam after welding. Heavyweight hull cleaning robots have also been used for ship surveying and repair. These robots weigh more than 100 kg and are used to remove large areas of coating on the ship's hull using water jetting techniques or brushes (Ortiz et al., 2007). Some robots are already available for marine inspection services, such as the robot Steel-Climber from Miko Marine (Miko, 2013) or the Magnet Crawler M250 from Jetstream Europe (Jetstream, 2013). Both are heavyweight magnetic crawlers for blasting, ship cleaning, and inspection. The robot CROMSKI (Jung et al., 2010) is used for dam inspection and is able to climb vertical walls, independently of the material of the wall. Another robot using a tracked system with suction pads is described in (Kim et al., 2010). The robot is a self-contained system which also integrates the motor that produces the vacuum. Another suction-pad based climbing approach is described in (Raut et al., 2010). It enables the robot to walk on glossy and flat surfaces for window cleaning in buildings, and also relies on clean surfaces. An example of a robot using magnetic wheels for inspection purpose is described in (Tâche et al., 2009). The adaptability of the system is provided by different joints which allow the adjustment of the robot's kinematics. A tracked robot using permanent magnets is described in (Kalra

et al., 2006).

3.2 Aerial Inspection Robot for Stage 1

3.2.1 General overview and design of the robot

The MAV prototype is based on the well-known Pelican quadrotor from Ascending Technologies (see Fig. 3(a)). This is a 50-cm diameter platform with 25.4 cm propellers, able to carry a payload of 650 g, and equipped with a standard navigation sensor suite: a barometric pressure sensor for height estimation, a GPS receiver, and a full 3-axis inertial measuring unit (IMU). Furthermore, the MAV has a Hokuyo lightweight laser scanner with a range of up to 30 m, which is not only used for obstacle detection, but, by deflection of lateral beams using mirrors, to estimate the distance to the floor as well as to the ceiling. Visual information is collected by means of a flexible vision system with an appropriate structure for supporting one ground-looking camera and two additional units, which can be tailored for the particular inspection mission to be performed, such as: two forward-facing cameras forming a stereo vision system, one camera facing forward and the other facing upwards, or, to save weight, a single camera facing forward. Apart from the onboard controllers, the vehicle carries an additional high level processor (HLP) which obviates the need to send sensor data to a base station, since it processes it onboard, thus avoiding any communications latency inside the critical control loops.

The configuration shown in Fig. 3(a) includes a CoreExpress board fitted with an Intel Atom 1.6 GHz processor and 1 GB RAM. The different sensors are attached to the HLP through USB. Finally, communications are implemented through a WiFi link. The wireless device attached to the vehicle is connected to the HLP using a dedicated PCI Express port, avoiding the need for wireless communications to share USB bandwidth.

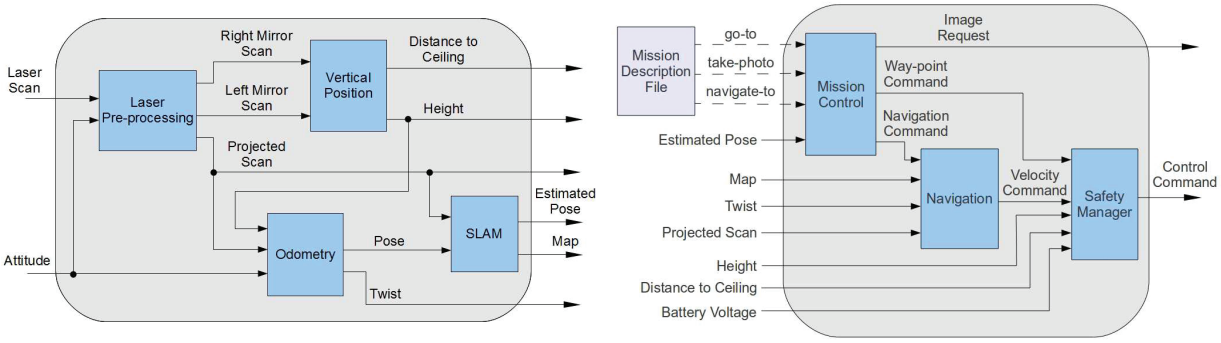
3.2.2 Control architecture

As on similar platforms, the control software architecture comprises at least two physically separated agents: the MAV itself and a ground station. More specifically, the different computational resources of the MAV run the control algorithms as follows (either as firmware or as software): (1) as is well known in the Pelican, the main ARM7 low level controller (LLC) runs the low-level software taking care of attitude stabilization and direct motor control (Gurdan et al., 2007); (2) the secondary ARM7 high level controller (HLC) runs the position controller described in (Achtelik et al., 2011); and (3) the HLP executes, on top of the Robot Operating System (ROS) running over Linux Ubuntu, ROS nodes providing platform motion estimates as well as platform safety, interaction with the onboard platform controllers, and WiFi communication with the ground station. Finally, in our configuration, the ground station comprises a cluster of laptops running ROS/Linux Ubuntu to perform off-board operations.

Figures 3(b–c) depict the control software running on the HLP and on the ground station. It features self-localization, mapping, obstacle avoidance, and path planning modules as well as mission control and supervision modules. Self-localization, as a central capability for this platform, is implemented following a 2D laser-based motion estimation approach (in accordance with the requirements of Stage 1). In more detail, the control software permits operation in both semi-autonomous and autonomous modes. In the first mode of operation, an operator is expected to send velocity commands in x, y and z using the sticks of a remote control (RC) unit, while the vehicle provides hovering and height control functionalities using the onboard sensors and the low-level attitude/position controllers. In the second mode of operation, the vehicle performs autonomously missions described by means of mission specification files (MSF, see Fig. 3(d) for a very simple example). In short, MSFs are parsed in order to identify and perform the actions requested (*go-to*, *navigate-to* and *take-photo*) making use of the sensor data processing components, laser pre-processing, vertical position, and odometry, whose results feed the SLAM and the navigation components, together with the low-level controllers. Finally, a safety manager implements a number of safety behaviors, such as monitoring *go-to* and *navigate-to* actions before sending the corresponding motion commands to the HLC, preventing the robot from flying too high or too close to the ceiling, and monitoring the battery voltage.

Among the different components enumerated above, the ground station runs those control modules that can tolerate

(a)



(b) (c)

(d)

```
1  <mission>
2  <goto x="0.0" y="0.0" z="1.0" spx="0.2" spy="0.2" spz="0.2" yaw="0.79" accpos="0.1" accori="0.0" timeout="30" stay_time="5.0" />
3  <takephoto camera="1" path="picture1.jpg" stay_time="5.0"/>
4  <goto x="2.0" y="0.0" z="1.0" spx="0.2" spy="0.2" spz="0.2" yaw="0.79" accpos="0.1" accori="0.0" timeout="30" stay_time="5.0" />
5  <takephoto camera="1" path="picture2.jpg" stay_time="5.0" />
6  <goto x="0.0" y="0.0" z="1.0" spx="0.2" spy="0.2" spz="0.2" yaw="0.79" accpos="0.1" accori="0.0" timeout="30" stay_time="5.0" />
7  <goto x="0.0" y="0.0" z="0.0" spx="0.2" spy="0.2" spz="0.2" yaw="0.79" accpos="0.3" accori="0.0" timeout="30" />
8  </mission>
```

Figure 3: The aerial vehicle: (a) platform overview, (b) self-localization and mapping modules, (c) mission execution modules, (d) example of a mission specification file.

latency in the communications, namely simultaneous localization and mapping, path planning, and mission execution/supervision, while critical control loops run onboard the vehicle in order to ensure minimum delay, a requirement also reported by other authors (Achtelik et al., 2009) to permit autonomous flying. Furthermore, information exchange between the laptops is performed by wire and the wireless datalink is left only for communication with the MAV. That is to say, only one laptop talks directly with the MAV, which reduces multiple point-to-point communications for the same data, but they are republished by this laptop to provide the full cluster with all the information. This configuration permits adding new computers to the cluster as needed, ensuring there is no extra wireless communication with the vehicle. For more details about the MAV control software, refer to (Bonnin-Pascual et al., 2012).

3.3 Lightweight Inspection Robot for Stage 2

3.3.1 General overview and design of the robot

The lightweight climbing robot is depicted in Fig. 4. It is actuated by two 12 V DC motors that drive the two front wheels, on which a total of 112 neodymium magnets are attached. The adhesion force on an iron surface is 12.16 N per magnet. The polarities of the magnets are oriented alternately to increase the adhesion force. Each wheel consists of two rows of magnets with a foam material applied in between, to increase the traction during climbing. The magnets are integrated into flexible rubber holdings which provide adaptability to the surface, and allow the system to traverse between surfaces. Furthermore, a tail is attached to the system using a flexible tail joint. During wall climbing, the tail

bends around the z-axis, allowing motion in circles with a small radius. At the tail tip, a rocker has been incorporated with two additional magnetic rings. This provides an additional degree of freedom that allows the tail wheels to adapt to uneven parts and climb on corrugated metal parts. Finally, since the system weight is well below 1 kg, no safety precautions need to be taken, except for a small catching net to be used by a second person to catch the robot in case it drops from, e.g., a bulkhead, in contrast to many existing magnetic climbing robots which need to be secured by a crane or a safety rope. As a result, the system can be quickly deployed to obtain visual close-up data from areas that are hard to reach. To provide visual data, a camera is attached to the front of the robot, including an LED light

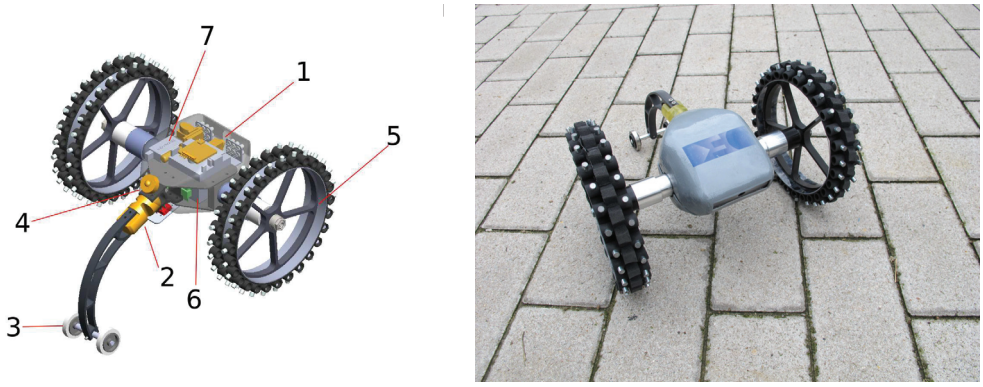


Figure 4: The lightweight crawler: (a) isometric view, with 1) video camera and LED light source, 2) elastic tail joint, 3) magnetic rear wheel rocker, 4) high-power LED for position tracking, 5) magnetic front wheel, 6) 12V LiPo battery pack, and 7) 2.4GHz M-PCM receiver; (b) vehicle realization.

source to enhance the image quality. The robot also stores high resolution images and video streams directly on a local SD-card for post-processing. The lightweight crawler is also capable of marking defects directly on the bulkhead of a ship, using a micro-pump to apply acrylic varnish on the surface. See (Eich and Vögele, 2011) for more details on the design of the lightweight crawler.

3.3.2 Control architecture

This section describes the control software of the lightweight inspection system, a graphical overview of which is shown in Fig. 5. As can be observed, the diagram also includes the SCMS, although the corresponding details will not be given until Section 5, since it is a system which is orthogonal to the robot team. We include the SCMS in this discussion to illustrate how, in particular, the lightweight crawler interacts with it. Any other platform interacts in approximately the same way, thanks to the modularity principles that have been adopted for the design of the full inspection system[1]. With regard to the lightweight inspection system in particular, the operator controls the robot using a 2.4 GHz M-PCM RC unit which provides the speed signal to the motors, and gets direct feedback from the robot via a first-person-view (FPV) interface. The remote control also provides a PCM interface to the control computer, allowing different control modalities, e.g., direct control by an operator, where the robot receiver sends the commands to the DC drivers; or indirect control by a navigation algorithm. The pump spray unit is actuated using a micro-actuator, which is also directly triggered via the RC. The FPV interface is implemented through an analog wireless 5.8 GHz link. During field trials, two different options were evaluated (see Fig. 6): hand-held devices and video goggles, the latter providing the operator with an immersive view in the scenery through the camera of the robot. Because maneuvering the robot only from the FPV can cause nausea, especially if the robot is at a significant height, it has been observed that it is desirable that a direct line of sight exist between the operator and the robot.

3.3.3 Associated vision-based localization unit

Due to the size and weight of the lightweight crawler, which were directly imposed by the requirements, the system cannot carry additional sensors that could be used to derive its position. However, in Stage 2, metric localization is

[1] The same would apply to the optical tracking unit and the heavyweight crawler.

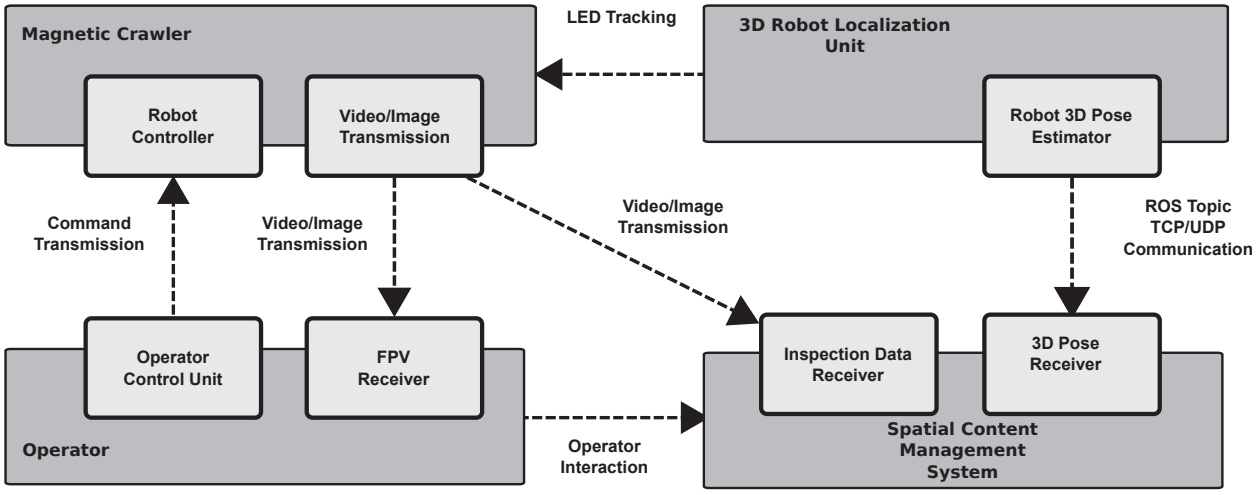


Figure 5: Overview of the control software of the lightweight inspection system. Besides the operator as man-in-the-loop, the system architecture comprises the mobile robot and the 3D localization unit. The drawing also gives the relation between the lightweight crawler and the SCMS.

Figure 6: User interfaces for the FPV of the lightweight inspection system: (left) small and portable handheld 5-cm screen device; (center) custom developed screen box comprising a 5.8 GHz video receiver and an 18-cm TFT screen; (right) full immersive video goggles used by the surveyor during field trials.

mandatory in order to localize the acquired data. The 3D localization system and the control architecture are shown in Fig. 7. In this diagram, U corresponds to the camera's horizontal image resolution and V to its vertical one. The camera is in the fixed reference frame of the laser range finder, and both are actuated, in accordance with the estimated pitch and yaw angles (θ, φ) of the target, by the two servo motors of the pan-tilt unit (PTU). The image provided by the camera is pre-processed using a Laplacian of Gaussian filter (LoG) (Haralick and Shapiro, 1992). The resulting image $g(u, v)$ has steep intensity gradients and is well suited to detect the bright light source of the LED in the images. The output of the detector is the position (u, v) of the center of gravity (CoG) of the image. Common problems when tracking a light source are that the position might jump to other bright sources during the tracking process, or the target itself can get temporarily occluded.

Due to its well-known robustness to multi-modal distributions and occlusions (Isard and Blake, 1998), we have adopted a Monte Carlo particle filter which is used in combination with sequential importance re-sampling (SIR). The method we use for the blob estimation in the image plane is a variant of the sample-based Monte Carlo localization (MCL) method, which is well-known for robot pose tracking within a 2D map (Thrun et al., 2000). In general, the following recursive equation is solved using the particle filter in order to estimate the position of a system (μ is a normalization

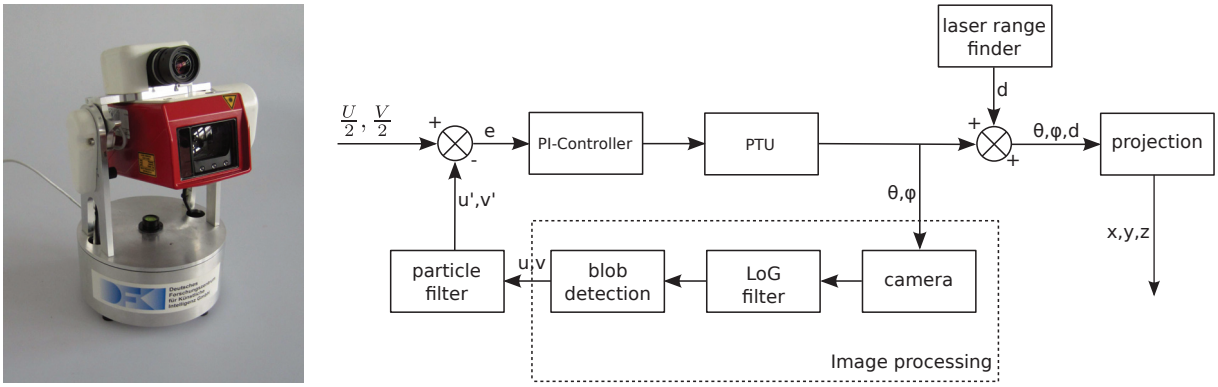


Figure 7: 3D localization unit: (left) integrated system; (right) control architecture.

factor):

$$Bel(x_t) = \mu \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\|o_t - x_t\|}{\sigma}\right)^2} \int p(x_t|x_{t-1}, \mathcal{T}(x_{t-1}, x_{t-2}), g(x_{t-1}, \sigma)) Bel(x_{t-1}) \mathrm{d}x_{t-1} \quad (1)$$

In contrast to the application of the particle filter in robot localization, no direct action update a is available to the system, therefore we estimated the last action by the translation between the two positions. To account for the change of speed and direction between two pose estimation steps, we add Gaussian noise into the motion model. The last motion at time x_{t-1} is thus based on the translation between the two prior estimated positions, $\mathcal{T}(x_{t-1}, x_{t-2})$. The Gaussian noise is incorporated as the function $g(x_{t-1}, \sigma)$. In our case, we take the distance between the estimated position and the new blob measurement given by the blob detection algorithm.

In the sequential importance re-sampling step, each sample is redrawn from the sample set according to its weight $\omega^i = \mu \frac{1}{\sigma\sqrt{2\pi}} \exp(-\|o_t - x_t^i\|^2/(2\sigma^2))$, which arranges that good samples are reproduced while samples with a low likelihood are removed from the set[2]. The output of the system is, thus, the 3D position of the robot in polar coordinates, which is projected onto the reference frame of the tracker using $x = d\sin\theta\cos\varphi$, $y = d\sin\theta\sin\varphi$ and $z = d\cos\theta$. This estimate is regularly provided to the operator and to the SCMS.

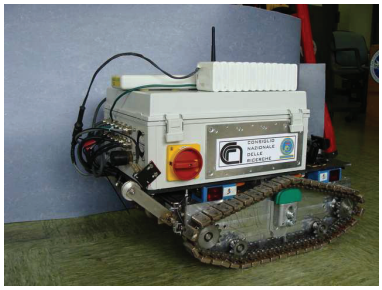
3.4 Heavyweight Inspection Robot for Stage 3

3.4.1 General overview and design of the robot

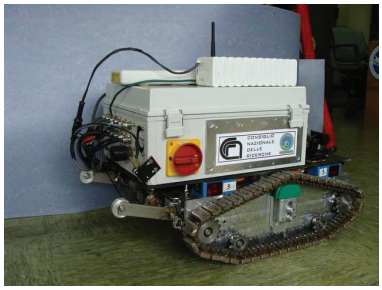
As mentioned before, unlike the lightweight climber, this crawler requires a larger payload capacity in order to carry an electrical robotic arm and an ultrasound probe for non-destructive thickness measurement, apart from other sensors and devices. As a consequence, the mechanical design of this heavier vehicle has been essentially determined by the choice of the climbing system. In order to minimize the risk of detaching and crashing while using a passive adherence system, magnetic tracks were preferred to wheels in virtue of their ability to maximize the contact area with the metal surface. The result was the Magnetic Autonomous Robotic Crawler (MARC) shown in Fig. 8.

As can be seen, a cart, actuated by a DC brushless servomotor and rigidly connected to a couple of wheels, has been mounted at the rear of the vehicle. The wheeled cart, positioned in *home*, *up*, and *down* configurations, as shown in Fig. 8, helps the vehicle to cross slope discontinuities and to not overturn when working on vertical slopes. As far as the magnetic tracks are concerned, custom-made platelet-shaped magnets have been designed, built, and directly fixed to the holes of the track chain brackets. Each magnet, which measures 40 × 22 mm, exerts a maximum magnetic force of about 100 N. Since, under nominal operating conditions, 8–10 magnets per track are in contact with the metallic surface, the corresponding attractive force is of 800–1000 N per track (1600–2000 N in total). Each track is actuated

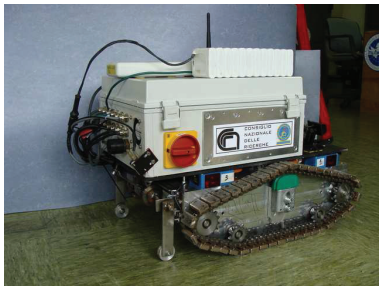
[2] μ ensures the weights ω^i sum up to 1



(a) Wheels in *home* position: used for operations over non-vertical slopes



(b) Wheels in *up* position: used for handling slope discontinuities between 45 and 90 degrees.



(c) Wheels in *down* position: used when working on vertical slopes.

Figure 8: MARC rear wheel configurations.

by a 140 W brushless motor coupled to a gearhead with reduction ratio 85:1 for a resulting maximum speed of 0.12 m/s. Motors and gear-boxes are mounted at the bottom part of the vehicle, together with Li-Ion battery packs placed along the longitudinal axis and at the rear part of the platform.

3.4.2 Control architecture

The MARC sensor system consists of: a high performance, miniature attitude heading reference system (AHRS) with GPS, combining MEMS sensor technology and a highly sensitive embedded GPS receiver, providing orientation, inertial, and, when possible, GPS measurements; and four laser range sensors, able to measure the distance to, e.g., lateral bulkheads, and allowing the computation of the platform's position and orientation inside an operating lane. The vehicle-embedded real time computing platform is based on commercial-off-the-shelf (COTS) hardware and free software: in particular, a single-board computer hardware architecture, possessing PC-compatible CPUs, has been adopted to support a Linux-operated system configured for real time performance as discussed in (Bruzzone et al., 2009).

The MARC control system is able to follow a linear surface on the basis of lateral range measurements. The measurements provided by the pair of laser range sensors mounted perpendicularly to the vehicle side allow estimating the distance d and orientation φ of the vehicle with respect to a linear vertical shell frame. The proposed control law for following a linear surface, using a simple Lyapunov-based design, is

$$\begin{cases} u = u^* \\ \dot{\varphi} = -k_d u^* (d - d^*) - k_\varphi \sin \varphi \end{cases} \tag{2}$$

where u^* is the linear reference speed of the vehicle, $\dot{\varphi}$ is its yaw rate, d^* is the reference range from the tracked linear surface, d is the current distance to the surface, and k_d and k_φ are positive constants. This allows MARC to follow a linear surface at a reference range, remaining parallel to the tracked surface. See (Bibuli et al., 2012) for more details about MARC.

3.4.3 Associated robotic arm for NDT measurement

Given the complexity of ship structures and their common environmental conditions, e.g., highly corroded, rusted or even bent structural members, ultrasonic thickness measurement (UTM) becomes a tedious task even for skilled UT operators, involving a number of phases: surface preparation, which can become the longest task since it is not uncommon to have significant scale and rust on the metallic surface; application of couplant, usually a high viscosity liquid, such as glycerin, whose purpose is to increase the transmission coefficient of the medium traversed by the ultrasonic pulse that goes from the probe to the uneven surface of the specimen; and UT probe application and measurement, which can require additional surface preparation if the UTM is not steady or if unreliable readings are obtained. Due to these conditions, and in order to require only a minimum of skills from the user, the robotic arm

carried by MARC exhibits a certain degree of adaptability and intelligence in order to allow the measurement of a wide range of significant ship structural components, such as frame stiffeners, brackets, and face plates, as required by the classes. Five degrees of freedom (4 angular and 1 linear), a holding torque of 1 kg/m (@18.5V) and an angular resolution of 0.3° are the key resulting specifications of the arm at the mechanical level. Regarding the arm controller, it features a USB interface with MARC and an ARM Cortex M3 (@72 MHz) processor, running a 400 Hz control loop which in particular implements spline-based trajectory following. Finally, about the end-effector, it comprises a grinder for surface preparation (using current monitoring for torque control), a peristaltic pump to inject the couplant that improves the transmission of the ultrasonic pulse, and the corresponding ultrasound sensor. Referring particularly to the task of thickness measurement, the robot is equipped with an FPGA-based board responsible for the high voltage pulse generation (< 390 V), the return signal amplification (0–100 dB), and the digitization at a maximum sampling rate of 100 MHz and 10-bit resolution. For a more detailed description, refer to (Koveos et al., 2012). UTM relies on measuring the time for an ultrasonic pulse to propagate along the material under test and extract its thickness on the basis of the speed of sound c. It is typical from NDT to use A-Scan waveforms, i.e. ultrasonic amplitude *versus* time of arrival, for visual inspection. However, since the raw waveform is usually noisy and may contain signals from multiple reflections, time is measured from a filtered signal, y_n, representing the energy of the ultrasound. This is obtained by low-pass filtering a rectified input signal x_n.

Since the waveform is buffered, a zero-phase formulation is used to preserve phase information (in the following, P and Q are, respectively, the feedforward and the feedback filter orders, b_i and a_i are the corresponding filter coefficients and N is the number of samples):

$$\begin{aligned}
&\text{1) Forward filtering:} && y_n^* = \tfrac{1}{a_0}\left(\textstyle\sum_{i=0}^{P} b_i\,|x_{n-i}| - \sum_{j=1}^{Q} a_j y_{n-j}^1\right),\ n = 0\,..\,N \\
&\text{2) Reverse-time filtering:} && y_n = \tfrac{1}{a_0}\left(\textstyle\sum_{i=0}^{P} b_i y_{N-n-i}^* - \sum_{j=1}^{Q} a_j y_{n-j}\right),\ n = 0\,..\,N \\
&\text{3) Time reversing:} && y_n = y_{N-n,}\ n = 0\,..\,N
\end{aligned} \tag{3}$$

Regarding thickness calculation, it can be implemented through a peak detection algorithm: leveraging the fact that ultrasonic echoes are monotonically decreasing due to absorption and reflection losses, time of flight can be estimated as the time between those peaks. Despite this method being simple, it relies on single points of the waveform and is subject to variations of the user selected parameters, e.g. filter bandwidth, amplification, etc. A second algorithm, based on the auto-correlation of the unfiltered input signal, is preferred instead. Although more computationally intensive, it provides robust estimations due to the use of the entire waveform, minimizing any dependency on the user-provided parameters, and augmenting the level of autonomy. The thickness is then estimated as follows:

$$\text{thickness} = \frac{c}{2}\,dt\left(\arg\max_{j}\sum_{n} x_n x_{n-j}\right),\ j \in n_{min}\,..\,n_{max}\,, \tag{4}$$

with dt the sampling period, and n_{min} , n_{max} defined by the minimum and maximum measurable thicknesses, which, in this work, come from a wide range 3–30 mm, satisfying the survey's requirements, typically 5–25 mm.

4 A vision-based solution for defect detection

This section proposes a vision-based integrated solution for detecting two kind of defects of metallic surfaces that are generally considered relevant to determining the condition of a vessel, namely, coating breakdown/corrosion and cracks. (Bonnin-Pascual, 2010) performs a thorough review of the different techniques which have been used so far for defect detection in general. Some of those proposals are specifically related to vessel inspection, although they mostly refer to the inspection of the external part of the hull by means of (mainly tethered) unmanned underwater vehicles. Their main goal is to assist with the detection of the loss of the external coating, aquatic life adhering to the hull (to prevent future corrosion), artificial objects attached to the hull (to avoid sabotage), weld inspection, etc.

The approach described in this section aims at detecting the aforementioned defects from the digital images captured by the MINOAS robotic platforms. It must be regarded as an assistance tool in the hands of the surveyor, which essentially means there will always be a human making the last decision about the state of the corresponding structural

component. In more detail, the defect detector is an integrated solution that considers both kinds of defect through an algorithm comprising two stages. The first stage, corrosion detection, is in charge of labeling all areas of the input image suspected of being affected by corrosion. The second stage, crack detection, makes use of the pixels classified as *corroded* as clues for detecting cracks. The rationale behind this approach comes from the observation that, typically, in metallic surfaces, cracks are surrounded by corroded areas, so that guiding the crack detector in accordance with the output of the corrosion detector becomes a highly effective strategy for enhancing its performance. The two stages are outlined in the following sections. For more details, see (Bonnin-Pascual, 2010).

4.1 Corrosion detection

The corrosion detector is based on a supervised classification scheme that comprises two steps that can be considered as two weak classifiers. The idea is to chain different fast classifiers with poor performance in order to obtain a global classifier attaining a higher global performance. The first classifier is based on the premise that a corroded area exhibits a rough texture. Roughness is measured as the energy of the symmetric gray-level co-occurrence matrix (GLCM), calculated for downsampled intensity values between 0 and 31, for a given direction α and distance d. The *texture energy* E is defined as $E = \sum_{i=0}^{31} \sum_{j=0}^{31} p(i,j)^2$, where $p(i,j)$ is the probability of the occurrence of gray levels i and j at distance d and orientations α or $\alpha + \pi$. Patches with an energy lower than a given threshold τ_E, i.e., those which exhibit a rough texture, are finally candidates to be more deeply inspected.

The second classifier operates over the pixels of the patches that have survived the roughness step. This classifier makes use of the colour information that can be observed from the corroded areas, unlike the first classifier. It works over the hue-saturation-value (HSV) space after the realization that HSV-values that can be found in corroded areas are confined to a bounded subspace of the HS plane. Although the V component has been observed to be neither significant nor necessary to describe the color of the corrosion, it is used to prevent the well-known instabilities in the computation of hue and saturation when the color is close to white or black. This second step requires a prior training step to learn the color of the corroded areas, which consists in building a bidimensional histogram of HS values for image pixels labelled as 'corroded'. The resulting histogram is subsequently filtered by zeroing entries whose value is below 10% of the highest peak, and, next , applying a bilateral filter, considering the bins' heights as the intensity values of a digital image. This approach filters the histogram using a kernel consisting of two Gaussians, one for the spatial domain and another for the range domain. The colour classifier proceeds as follows for every 3-tuple (h, s, v): given thresholds mV, MV and mS, pixels close to black, $v < mV$, or white, $v > MV \wedge s < mS$, are labelled as non-corroded, while the remaining pixels are classified as corroded if the HS histogram contains a non-zero value at the (h, s) entry.

4.2 Crack detection

The crack detector proceeds in accordance with a percolation model, similarly to the approach described in (Yamaguchi and Hashimoto, 2010), although their detector was constructed for searching cracks in concrete; this fact makes the authors assume a geometrical structure that does not match exactly the shape of the cracks that are formed in steel. The percolation model used is developed from a region-growing procedure which starts from a seed pixel and propagates in accordance to a set of rules. In our case, the rules are defined to identify dark, narrow and elongated sets of connected pixels, which are then labelled as cracks. Once a seed pixel has been defined, the percolation process starts as a two-step procedure: during the first step, the percolation is applied inside a window of $N \times N$ pixels until the window boundary is reached; in the second step, if the elongation of the grown region is above ϵ_N, a second percolation is performed until either the boundary of a window of $M \times M$ pixels ($M > N$) is reached or the propagation cannot progress because the gray levels of all the pixels next to the current boundary are above a threshold T. Finally, all the pixels within the region grown are classified as crack pixels if the region elongation is larger than ϵ_M.

Within the $N \times N$- or the $M \times M$-pixel window, the percolation proceeds in accordance to the next propagation rules: (1) all 8 neighbours of the percolated area are defined as candidates, and (2) each candidate p is visited and included in the percolated area only if its gray level value $I(p)$ is lower than a threshold T, which has been initialized to the seed pixel gray level. At the end of the percolation process, the mean gray level of the set of pixels is checked to determine

if the region is dark enough to be considered a crack; otherwise, the region is discarded, and another percolation starts at a different seed pixel. Seed pixels are defined over a regular grid with a step of Δ pixels, and are required to coincide with an edge whose gray level is below γ_s. To ensure that the relevant edges are always considered, a dilation step follows the edge detection, where the dilation thickness is in accordance with Δ. Furthermore, since the crack detector operates under the guidance of the corrosion detector, as mentioned above, seed pixels are required to lie within corroded areas. The different conditions for being a seed pixel, taken together, speed up the crack detection process and reduce the false positive rate thanks to the corrosion-based guidance.

5 Spatial Content Management System for Robot-acquired Inspection Data

Classification societies use software packages such as the Germanischer LLoyd HullManager [3], which provides the means to assess the ship's condition and store the relevant data for later use. Mobile tools for damage assessment also exist from Sertica[4]. With this tool, the surveyor can incorporate inspection data using a portable device. Publications regarding inspection data management tools can be found in (Fletcher and Kattan, 2009). All these tools, however, require an operator who manually inputs the inspection data.

The SCMS that is introduced in this section integrates the surveyors and the inspection robots in a sort of man-in-the-loop strategy. The inspection data acquired by the robots, are collected with the SCMS and can be evaluated during or after the inspection. Currently, no system of this kind is available. The SCMS is a central element of the inspection system which is proposed in this work. In more detail, it permits collecting, sharing and displaying in a centralized manner all the information gathered using a 3D representation of the vessel under inspection. All robot systems introduced in this paper can provide images, videos or thickness measurements during the inspection to the SCMS using common interfaces. As mentioned in (Caccia et al., 2010a), a surveyor does not use a metric representation for localizing inspection data, but rather uses the spatial topology of the ship, i.e., "cargo hold 1, port side, bulkhead, left side": inspection data are topologically assigned to the named parts of the ship within the SCMS. The ship as a whole is thus internally represented as a tree structure, each node corresponding to a certain part of the ship. The 3D interface of the SCMS is shown in Figure 5. The data is internally assigned to the topological parts of the vessel (cf.

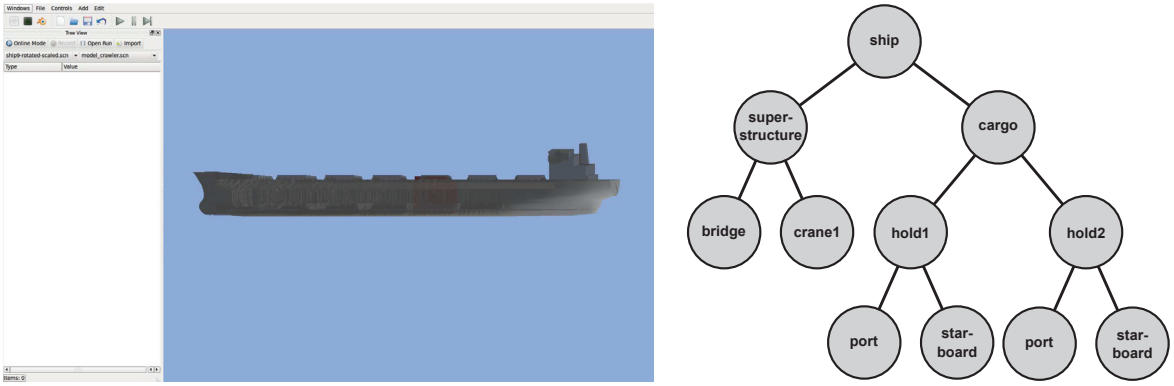


Figure 9: The Spatial Content Management System: (left) The 3D user interface. (right) The data is internally assigned to topological parts of the vessel. Within the nodes, the localization of the data is performed metrically.

Figure 5). The inspection data are anchored to the nodes of the tree structure. This representation as a graph, apart from being logical to the users, allows a quick access to the data, since all the inspection data for a certain topological unit, e.g., a certain cargo hold, are located under a certain branch of the tree structure. Nevertheless, the location of the inspection data is also represented in a metrical way within the data node leafs, each leaf node having its own local, metric reference frame aligned to the 3D representation of the ship. The origin of the reference frame corresponds to

[3] http://www.gl-maritime-software.com/gl-hullmanager.php
[4] http://www.sertica.dk/

a fixed position, e.g., where the localization unit was placed during the inspection, with the only constraint being that, across surveys, approximately the same position must be used. Within the SCMS, the surveyor navigates through a 3D model of the ship to get access to the different areas, while the inspection data items are projected metrically into the 3D model as data blobs which can also be selected by the surveyor. Furthermore, all the data associated to a certain node, irrespective of their modality and recording date, are available once the item is selected. Finally, 3D sub-models can be associated to each tree node in order to obtain a higher level of detail of the corresponding structural element when selected.

6 Systems performance evaluation

The marine inspection system described in this work was tested both in the laboratory and in field trials. This section reports the results collected for every platform during the different experiments that were performed. The first trial was performed on a bulk carrier ship which was under repair in a shipyard in Varna, Bulgaria. Several ship surveyors attended the trials and provided comments on the usability of the various platforms and on possible improvements. The second trial was also performed on a container cargo ship in the same shipyard which served also as a final validation point for all systems. The two test cargo holds are shown in Fig. 10.

Figure 10: Vessel holds involved in the first and the second field trials: (left) cargo hold of the bulk carrier after it was cleaned, and (right) cargo hold of the container ship.

6.1 Performance of the aerial platform

This section reports on the results of a number of laboratory and field experiments to demonstrate the navigation capabilities of the MAV. These experiments were planned to cover the kind of capabilities required to perform Stage 1 of a MINOAS inspection mission. To begin with, Fig. 11 shows results for the MAV operating in semi-autonomous mode, with the operator trying to reach, with the assistance of the platform, a set of waypoints within a laboratory of the University of the Balearic Islands (see Fig. 11(a) and (c)) and during the first field trial onboard the bulk carrier ship (see Fig. 11(b) and (d)). As can be noticed, the platform effectively provides the operator with height control and hovering functionalities. The first experiment in autonomous mode that is reported, whose results are provided in Fig. 12(a–d), shows the ability of the platform to reaching, without the intervention of an operator, a number of waypoints at different heights (within a building of the University of the Balearic Islands). More precisely, the mission describes a wall sweeping task, as would be done onboard a ship, although at the scale of a laboratory. The mission consists in attaining, with an accuracy of 0.1 m, a total of ten waypoints along a zig-zag path, as indicated in Fig. 12(b). A different kind of experiment (also performed within a building of the University of the Balearic Islands) is shown in Fig. 12(e–h). In this case, apart from reaching a set of waypoints autonomously, the goal is to perform two short sweeping tasks in adjacent rooms, so that a safe path through the door communicating booth rooms has to be planned and followed during the flight. That is to say, in this experiment, the MAV faces all the difficulties of a real scenario. As well as in the previous experiment, the different waypoints are correctly attained—within the accuracy requirements

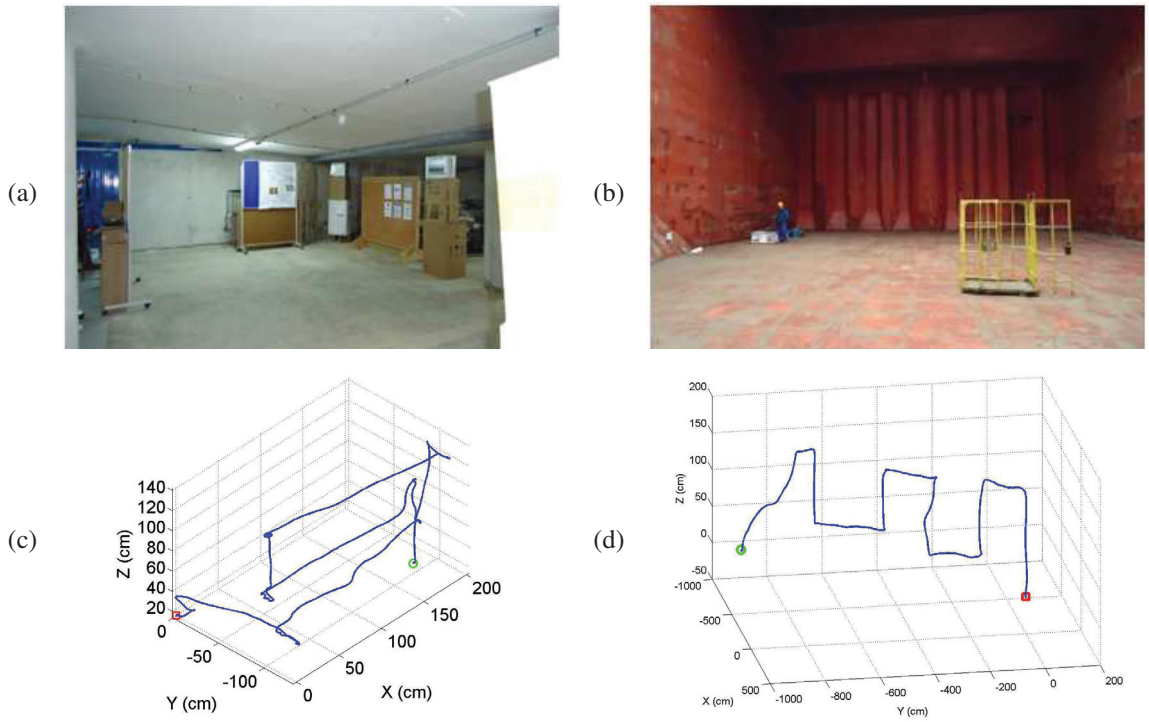


Figure 11: The MAV operating in semi-autonomous mode: (a,b) snapshots of the environments; (c) path followed by the MAV during the laboratory experiment; (d) path followed by the MAV during the field trial. (The red square indicates the take-off position and the green circle, the landing position.)

determined by the mission specification file (0.1 m)—while the local and global path planners produce safe paths towards the targets.

A third set of experiments in autonomous mode taking place onboard the container ship, during the second field trial, is here reported (see Fig. 13(a)). This time the mission description file specified a sweeping task, consisting in reaching a collection of sixteen waypoints along a zig-zag like path (see Fig. 13(b)), and the experiment was performed several times to compare the results of the consecutive executions. The results, which can be found in Fig. 13(c–d), show that (1) the paths estimated by the vehicle are consistent with the environment, as well as the map built by the laser-based SLAM component, and (2) the behaviour of the MAV is repeatable (up to the accuracy requested in the mission specification file, 0.1 m), both of which suggest the validity of the navigation approach. Pictures taken by the onboard cameras at some of the waypoints can be found in Fig. 13(e).

6.2 Performance of the Lightweight Inspection System

6.2.1 Lightweight crawler

As for the climbing capability of the robot, several tests were performed over mainly vertical bulkheads, and also along 70° slopes. Fig. 14(a) shows the climbing robot on a slope and Fig. 14(b) on a transverse bulkhead. In this regard, Table 2 shows the maximum tear-off force against the maximum vertical drag force exerted by the robot while climbing vertically on a bulkhead along a straight line. The reference measurement for the robot was taken while climbing a clean, bare steel frame without any coating. The system was able to create a lift of 19.8 N while climbing vertically. The wheels of the robot blocked because the motors were not able to create more torque. Therefore, this number corresponds to the maximum lift the robot could produce under optimal conditions. The maximum tear-off force was measured as 45.1 N. In all the experiments, except the reference measurement, the wheels started to slip

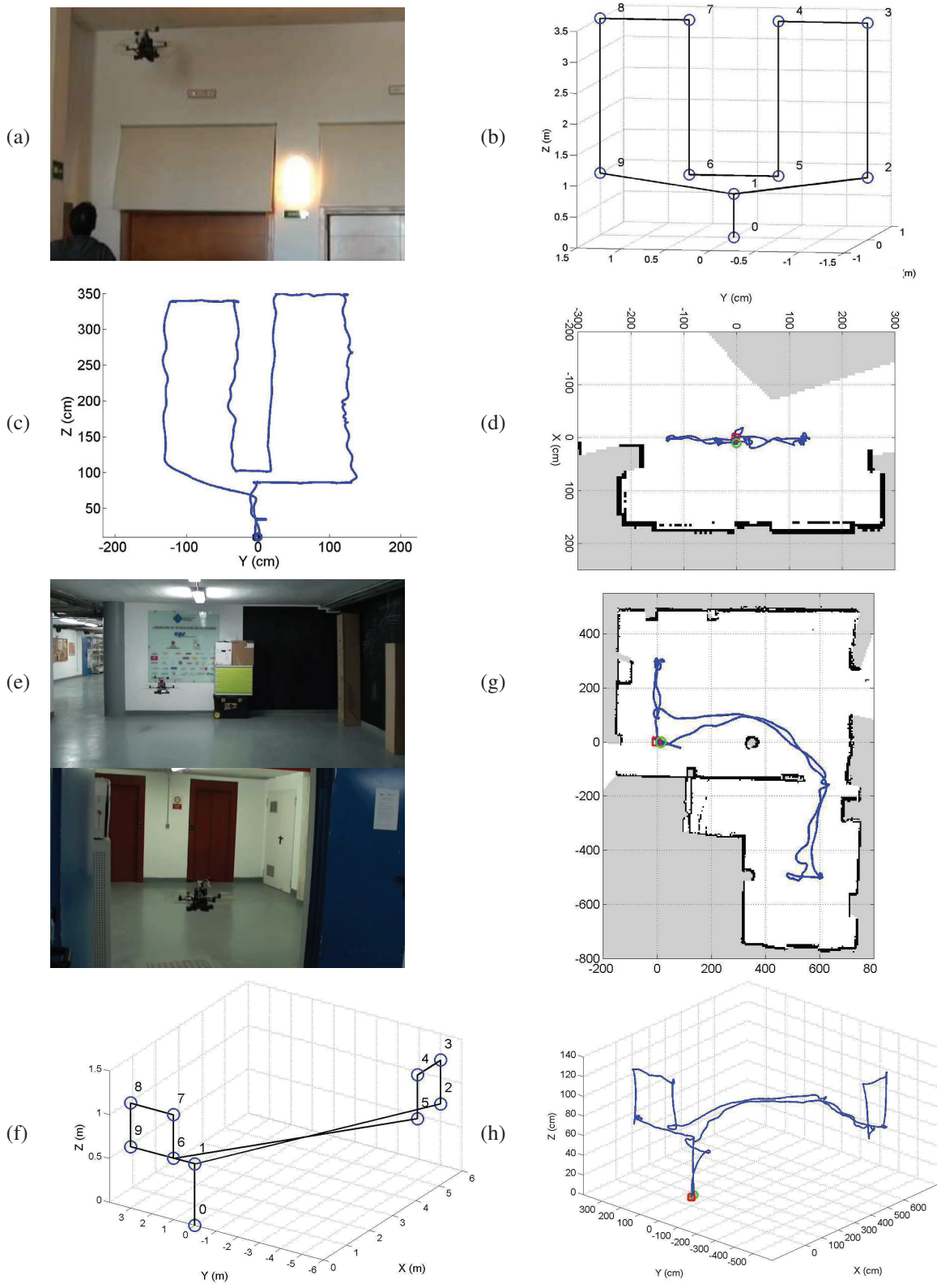


Figure 12: MAV results in autonomous mode under laboratory conditions: [1st experiment] (a) snapshot of the environment, (b) sequence of waypoints to be reached: 0–9/1/0, (c) path followed by the MAV, and (d) 2D path followed by the MAV superimposed over part of the map; [2nd experiment] (e) snapshots of the two rooms where the experiment takes place—the MAV departs from *above*, goes *below*, and finally returns *above*—, (f) sequence of waypoints to be reached: 0–9/1/0, and (g–h) map and 2D/3D path followed by the MAV.

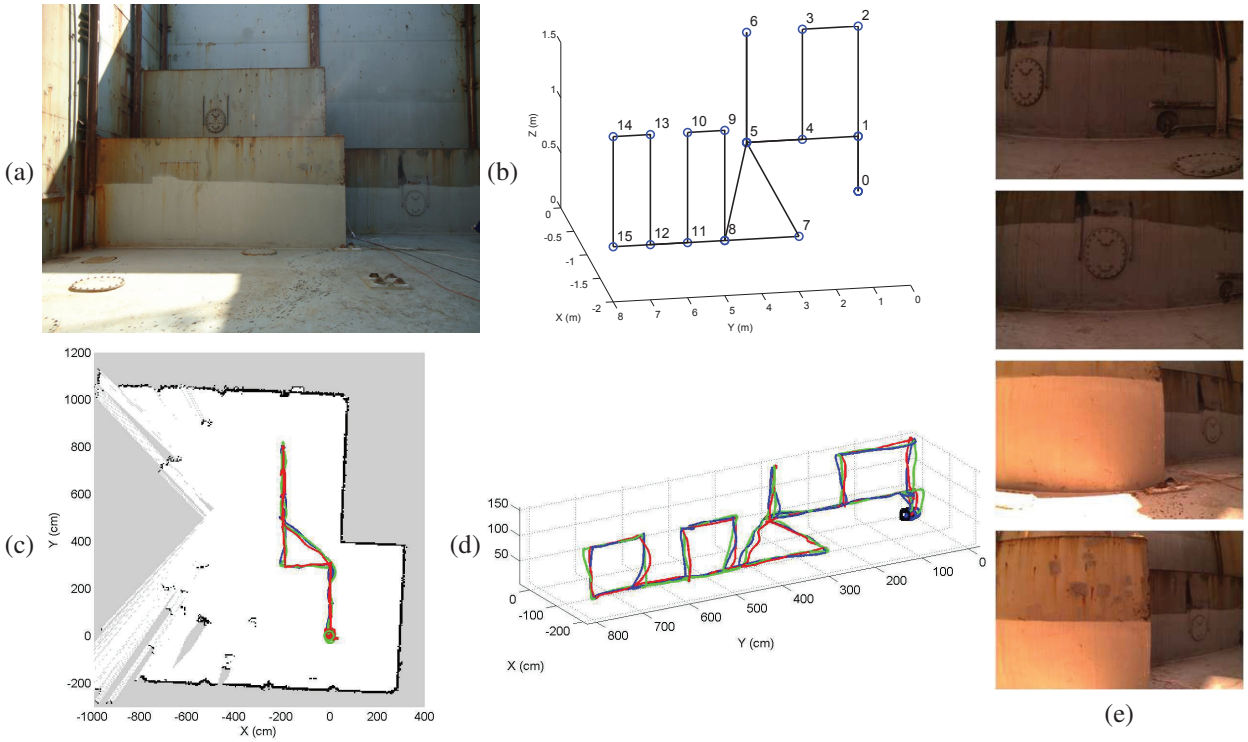


Figure 13: MAV results in autonomous mode from the second field trial: (a) hold area where the experiment takes place, (b) sequence of waypoints to be reached by the MAV: 0–6/5/7–15/8/5/1/0, (c-d) map and 2D/3D paths followed by the MAV, and (e) pictures taken at waypoints 1, 4, 8 and 9 [from top to bottom].

and the magnets could not produce enough adhesive force for the robot to move upward. A live image of the crawler

Table 2: Performance of the climbing robot on different vertical surfaces

Surface	Condition	Max. Vertical Drag Force (N)	Max. Tear-off Force (N)	Robot Behavior
5 mm bare steel plate	clean	45.1	19.8	block
1 mm sheet metal	clean	25.5	12.4	slip
coated steel	clean	30.3	15.7	slip
coated steel	corrosion	28.6	12.2	slip
bare steel	corrosion	25.5	17.1	slip

was received by the small hand-held device depicted in Fig. 6(left). In contrast to the laboratory experiments, the 2.4 GHz analog video signal suffered from several problems, being significantly distorted at a distance of more than three meters. Because of the video transmission issue, only the videos and images stored directly on the robot could be used by the surveyor. Of particular relevance in this regard is the fact that the robot was able to take high-quality snapshots of some damage at some welding seams which were of interest to the surveyors because the damage was not visible to the surveyor without using the robot (cf. Fig. 14(c–e)). The quality of the images was sufficient to estimate the level of corrosion. The experimental setup during the trial within a container hold, which involved a surveyor, an operator, the 3D tracking unit, and the lightweight crawler, is shown in Fig. 15(a). The robot was constantly sending images to the SCMS, which was running on a laptop. Some snapshots of the transmitted images are shown in Fig. 15(c) and 15(d). To check the performance of the crawler, an operator within the line of sight controlled the robot, while the surveyor would indicate where to move. The robot was able to climb on the vertical walls of the cargo hold. Besides, it was useful for inspecting the spaces behind the guidance rails for the containers (see Fig. 15(b)) that were not visible to the surveyor without using the robot. However, rust and dirt would stick to the magnets, causing problems for the crawler (see Fig. 15(e)).

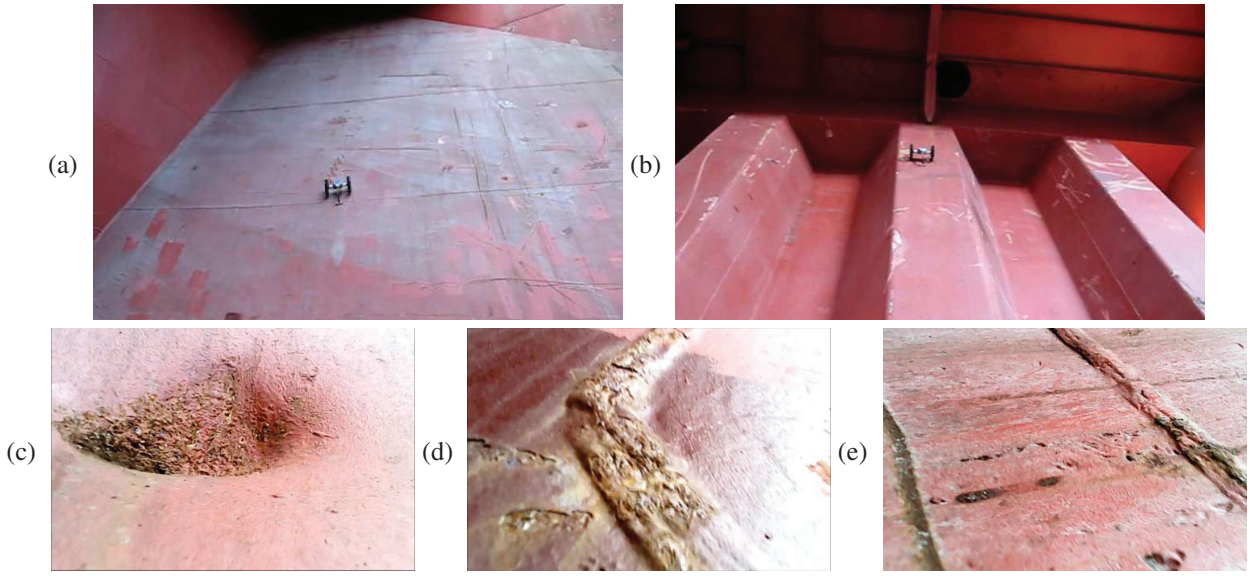


Figure 14: First trial for the lightweight crawler: the climbing performance was tested on a 70° slope (a), as well as on a vertical transverse bulkhead (b); (c-e) sample images taken by the robot.

Summarizing the trials, we can say that on the one hand, the main strength of the lighweight crawler is its ease of use, which was proved by the surveyors themselves' being able to control the robot without prior training. Even without the tracking system and the SCMS, the robot alone proved able to serve as an extension of the surveyor's eyes, supplying him with a quick look at parts of the ship which otherwise are accessible only using staging or cherry-picking methods. On the other hand, the second trial also showed clearly the limits of using a magnetic climbing system in a real environment: rust and dirt did not allow the crawler to climb freely on the bulkhead, although it was possible for the operator to navigate the robot around such critical patches. Since rusty patches provide less adhesion force for magnets, a system with stronger magnets would solve the problem, such as the heavy weight crawler MARC described in this paper. The heavy weight crawler had no problems in heavily corroded bulkheads, but on the other hand, needed more safety precautions and more setup time.

6.2.2 Localization experiments

In this section, we report the performance of the 3D tracking unit while tracking the lightweight crawler. To this end, the absolute error was measured and compared with a reference value of 0.3 m. This value comes from the structure of cargo holds in different sorts of vessels (e.g., bulk carriers and containerships), and corresponds to the average distance between the sections in which bulkheads and walls are typically divided. In this way, the surveyor can know the section from which the inspection data were taken. During the experiments, the crawler would move along a vertical bulkhead of the cargo hold along a constant height of 1.5 m. Results from a first experiment can be found in Fig. 16(left). As can be observed, the resulting accuracy is around 15 cm up to a distance of 11 m, while, when the target is more distant from the tracker, the error of the system is significantly larger. This is due to the resolution of the camera and the focal length of the optics. In order to increase the overall accuracy, a different camera with higher resolution and a different focal length could be used. In this particular case, however, the height of the cargo hold during the second field trial was around 12 m, which indicates that the tracking performance was good enough to localize the robot everywhere inside the hold, i.e., in accordance with the requirements.

A second experiment was carried out to evaluate the repeatability of the tracker. Eight different positions were selected on a bulkhead, and the tracker was placed at a distance of around 5 m. The target was placed randomly at the selected positions and a total of 70 measurements were taken by means of the tracker. As expected, the tracker produced eight point clusters, one for each distinct position. The center of mass and the standard deviation for each cluster is shown in Fig. 16(right). Taking into account those results, it becomes obvious that the tracking is highly repeatable. Fig. 17

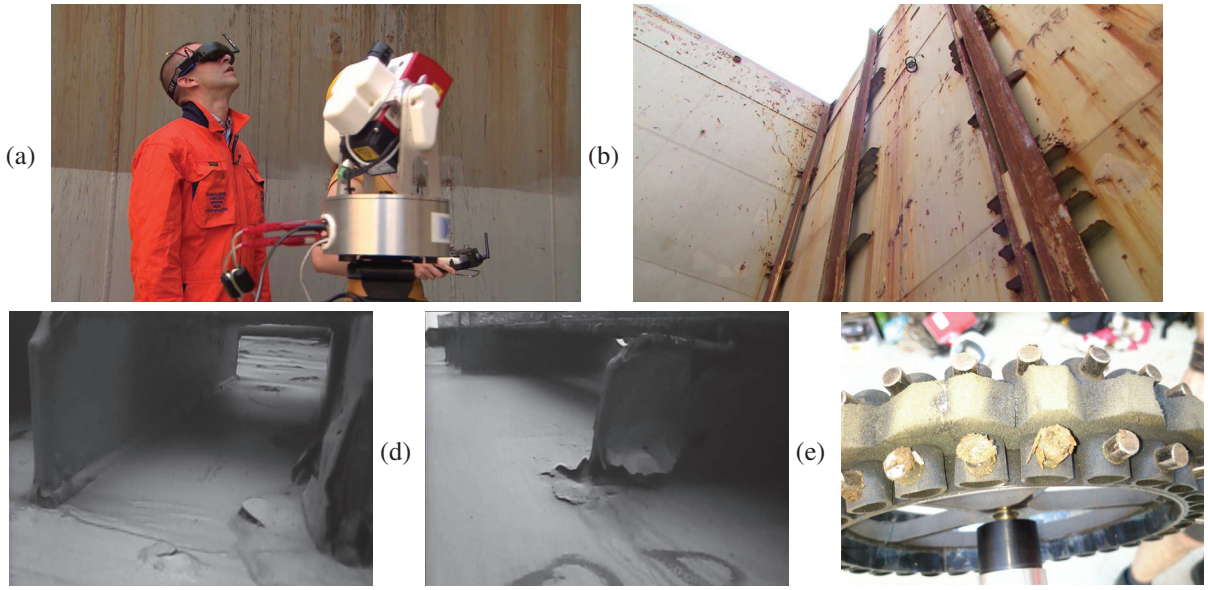


Figure 15: Second trial of the lightweight crawler: (a) experimental setup, (b) view of the wall climbed by the robot, (c–d) sample of transmitted video images from the inspection run, and (e) state of the magnets, covered in dirt and rust, after one of the runs.

shows the trajectories estimated by the tracker along two of the many runs that were performed during the trials. The area covered by the runs was around 20 m^2. The tracking unit was located at a distance of around 6 m from the bulkhead.

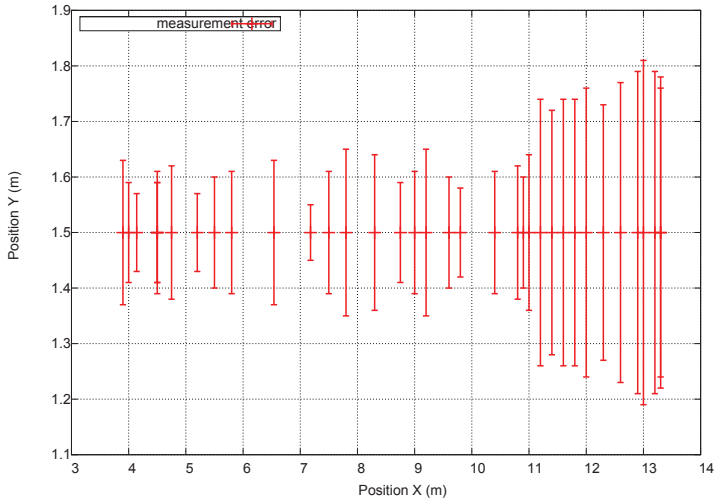
6.3 Performance of the Heavyweight Inspection System

6.3.1 Heavyweight crawler

The main goal during the ship trials was to demonstrate the ability, inside a ship hold, of the MARC (1) to climb vertical walls, in general, and bulkheads, in particular, and (2) to follow vertical shell frames in autonomous mode. The ability of MARC to climb a vertical painted metallic wall inside a ship hold in a dirty environment in the presence of rust were verified in a first set of experiments. During these preliminary tests, the vehicle was positioned on the vertical walls with the support of human operators. Fig. 18(left) shows MARC climbing a bulkhead. The rope visible in the pictures is just a safety measure, i.e., it would not support the vehicle under normal working conditions. The experiments performed consisted in MARC's following a vertical stiffener frame tracked with two of the lateral laser range finders with which MARC is fitted, while climbing a bulkhead. By way of illustration, Fig. 18(right) shows the vehicle telemetry while executing one of these experiments. The different plots show, from top to bottom, the measured ranges and estimated distances, and the range and orientation errors. As expected, the vehicle manages to reduce the tracking error to zero after deliberately forcing the robot to move closer to one of the sides.

6.3.2 Thickness measurements

In order to test MARC's ability to measure thicknesses when fitted with the arm and the NDT end-effector, the crawler was made to climb a vertical bulkhead (Fig. 19(left)) while aligning itself to the stiffener frames using its lateral laser distance sensors. Once in place, the operator measuring the thickness with the ultrasound can use the robotic arm to perform the measurement. The measuring procedure starts, then, by setting a rough estimate of the desired point (x, y, z, θ) where the measure has to be taken.



$\bar{x}$	$\bar{y}$	$\bar{z}$	dist.	σ
3.866	-3.267	-0.771	5.120	0.026
4.637	-2.039	-0.743	5.120	0.018
5.013	-0.687	-0.774	5.119	0.029
5.003	$+0.732$	-0.797	5.118	0.035
4.974	$+0.710$	$+1.303$	5.190	0.032
4.996	-0.690	$+1.214$	5.187	0.029
4.619	-2.034	$+1.196$	5.187	0.035
3.844	-3.251	$+1.258$	5.189	0.023

Figure 16: 3D tracker performance: (left) absolute positioning error as a function of distance, (right) center of mass $(\bar{x}, \bar{y}, \bar{z})$ and standard deviation (σ) for each cluster of the repeatability test [units are in meters].

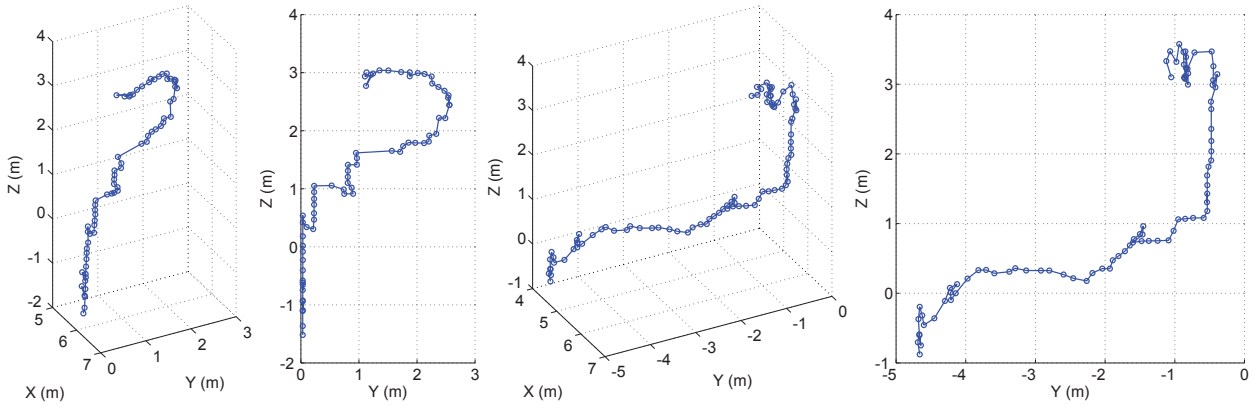


Figure 17: Two trajectories estimated by the 3D tracker: (1st,3rd) 3D perspective; (2nd,4th) YZ projection.

Next, the arm starts to move along the previously defined vector, until contact with the surface is detected by means of reed switches or the motor's loading measure provided by the servos, which makes unnecessary any previous knowledge of the exact relative location of the surface point. Since the angle θ is provided as a rough estimate of the orientation of the surface to be measured, the actual value must then be identified. This is essential for a reliable measurement, since grinding and the UT probe placement are highly sensitive to orientation misalignment; reed switches and the servo's angular sensors are again used to this end.

The next step starts grinding at a pre-defined load level and time duration which must be selected by the operator depending on the ship's state and amount of rust. The UT probe is next placed on the ground spot and a couplant is applied through the peristaltic pump, suitable for highly viscous fluids. Finally, the measurement is performed and a filtered A-Scan along the real UT waveform is presented to the operator. The measured waveform and its location can be stored prior to the start of a new measurement. Fig. 19(right) shows the results for one of the measurements, where the UT waveform is in red and the A-Scan is in blue. Numerous settings familiar to the UT operator, such as gain, filter bandwidth, time window, etc., can be altered at any point in time, although these settings are typically adjusted during calibration at the beginning of the survey, as required by the surveyor. The figure shows, for comparison, thickness estimates from both the peak detection algorithms and algorithms based on the auto-correlation function. The measurement, 9.1 mm, was verified by the shipyard's representative. Given the distant operation (typically above 10 m height), the number of steps to perform, and the accuracy of the operations required by the UTM, the arm's

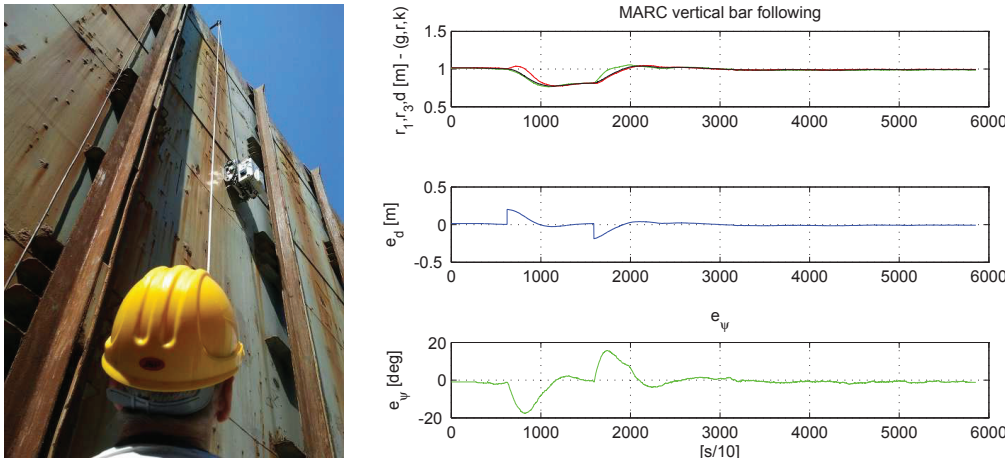


Figure 18: (left) MARC climbing a bulkhead inside the ship cargo hold. (right) MARC telemetry while following a side frame: measured ranges and estimated distance, distance error and orientation error (from top to bottom), where, in the upper plot, r1 (green) refers to the front right sensor, r3 (red) refers to the rear right sensor, and d (black) is the estimated distance to the side frame.

micro-controller software was designed and developed so that the measuring process was as automatic as possible, requiring minimal user intervention or knowledge of the environment.

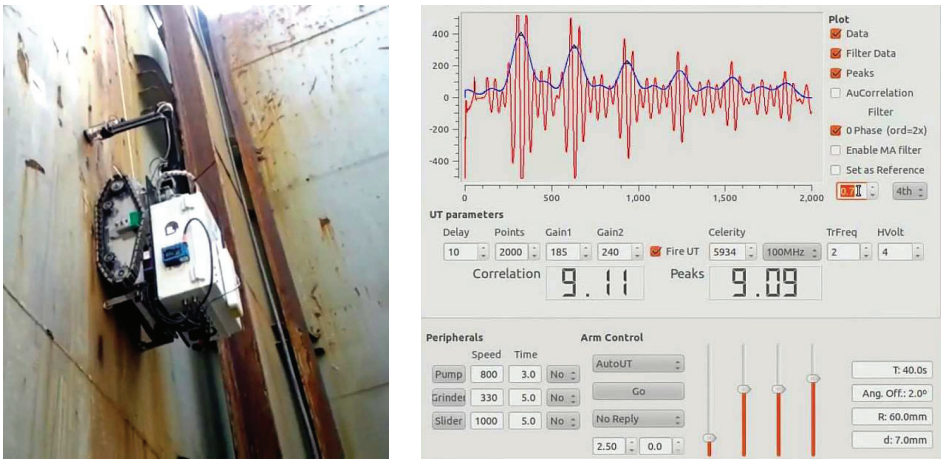


Figure 19: (left) MARC on a vertical bulkhead, taking a thickness measurement during a field trial. (right) Screenshot of the UT software.

6.4 Performance of the defect-detection solution

The performance of the defect detector has been evaluated using digital images provided by the MINOAS partners or else collected from real vessels during field trials. In general, the different images were captured without specific settings for the imaging process, particularly as to the illumination. Performance figures have been obtained for each stage of the classifier, using manually labelled images as the ground truth. Illustrative results of the defect detector are provided in Fig. 20. On the one hand, Fig. 20(a) provides results for the same input image using different energy thresholds. As can be observed, τ_E can be tuned to decrease false positives and just allow the detection of the most significant corroded areas. In the images, pixels labelled as corroded are colour-coded as red, orange, green and blue in accordance to the likelihood provided by the histogram, with red for high likelihood and blue for low likelihood.

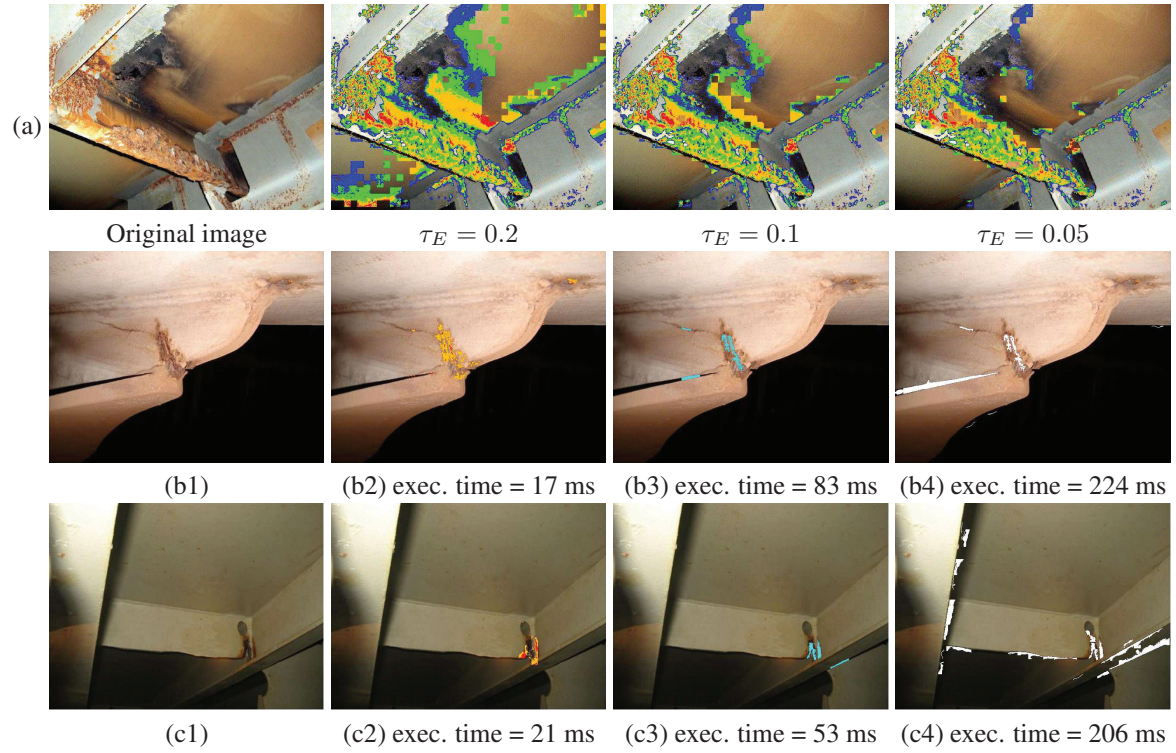


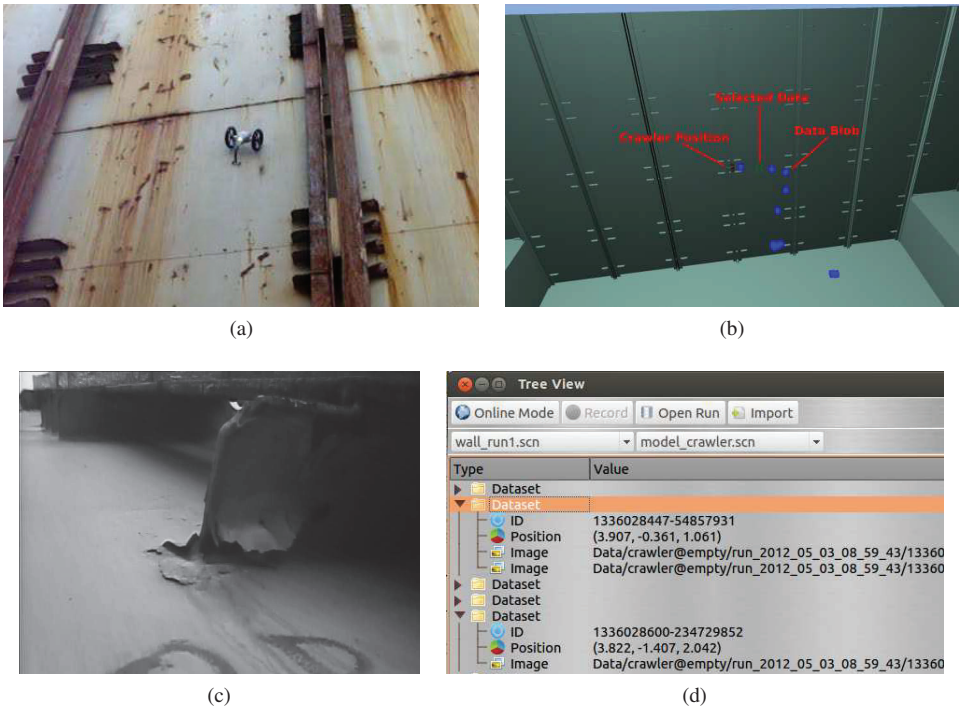
Figure 20: (a) Corroded areas detected using different energy threshold values τ_E. (b–c) Image samples and corresponding defect detections: (1) original image, (2) corrosion detector output, (3) crack detector output, and (4) crack detector output if the corrosion stage is skipped.

On the other hand, Fig. 20(b–c) shows the results for the full defect detector. The figure also compares the output of the crack detection stage with the output of an unguided configuration. As can be seen, the number of false positive detections is conspicuously reduced when the crack inspection makes use of the output of the corrosion detection.

Quantitative performance results for the full set of images have been obtained by calculating the percentages of false positives, *FP/no. pixels*, and of false negatives, *FN/no. pixels*. On the one hand, for the corrosion detector, the results are 9.80 (FP%) and 5.86 (FN%), while, for the crack detector, the results are 0.72 (FP%) and 0.57 (FN%). On the other hand, when the corrosion stage output is used to guide the crack detector, the result is a reduction in the FP% from 2.29% to 0.72%, and a speed-up in the execution time of 3.1x. This performance is in accordance with our assumption that most of the cracks in a metallic surface appear in areas which are also affected by corrosion, and prove that using corrosion to guide the crack detection results in a faster and more robust detector. Referring to the execution times, the defect detector provides corrosion-labelled images in 7–25 ms, and corrosion-guided crack-labelled images in 30–180 ms. These execution times correspond to images ranging from 120,000 to 172,000 pixels, and for a runtime environment including a laptop fitted with an Intel Core2 Duo processor (@2.20GHz, 4GB RAM).

6.5 Evaluation of the spatial content management system

The content management system was introduced during the second trial on the container ship. The tracking system was placed in a fixed position within the cargo hold. The 3D model was assigned to the topological node of the vessel as described in Section 5 and the position of the 3D tracker was calibrated by measuring several of the distances from the bulkheads. The lightweight crawler was used during the experiments. Transmitted inspection data were synchronized with the position provided by the tracking unit, thus providing information which was tagged with time and position. Live inspection views of the SCMS can be found in Fig. 21. For a start, Fig. 21(a) shows a 3D view of the hold, the position of the inspection robot, and the data blobs which had been recorded by the system. Within this view, the



(a) (b)

(c) (d)

Figure 21: (a) Cargo hold where the SCMS evaluation took place. Illustration of SCMS operation: (b) 3D view, (c) image at the data node, and (d) tree view.

operator can directly select inspection data just by clicking on the corresponding blob (Fig. 21(b)). The SCMS can also show a tree view (Fig. 21(c)), where the operator can select inspection data based on the recording time, the type of data, or search for specific data contents.

7 Conclusion and future research

This paper has introduced a vessel inspection system comprising a number of robotic systems fitted with different locomotion capabilities (aerial and magnetic crawling) and sensing modalities. Auxiliary systems, such as an integrated defect detection solution and a spatial content management system (SCMS) have been verified in field trials. The abilities of each platform have been derived directly from the requirements imposed by a re-engineered inspection procedure using robots, consisting of three distinct inspection stages. An integrated defect detection solution has been applied in order to select the critical spots for the use of non-destructive testing methods during the third stage. The areas of improvement include, apart from the industrialization of the platforms for a more robust operation, the enhancement of the locomotion capabilities together with the augmentation of the platforms' intelligence to increase its level of autonomy. The SCMS can also be enhanced by means of the semantic annotation of the structural parts, in order to lead to a more understandable representation of the inspection data, e.g., 'blistering on the upper stool between the third and fourth shell frame', as well as with the incorporation of 3D reconstruction functionality, using, e.g., laser-generated point clouds, in order to avoid any dependence on the availability of CAD models of the vessel under inspection. Finally, the collaboration between the platforms is a topic for further research.

Acknowledgments

This work was funded by the European Commission under project MINOAS (SCP8-GA-2009-233715). Francisco Bonnin-Pascual and Emilio Garcia-Fidalgo have been partially supported by the European Social Fund through, respectively, grants FPI10-43175042V and FPI11-4312 3621R (Conselleria d'Educació, Cultura i Universitats, Govern

de les Illes Balears). The authors want to thank Felix Grimminger and Kristin Fondahl for their work on the hardware design of the lightweight crawler and the 3D tracking unit. Finally, the authors thank Giorgio Bruzzone, Edoardo Spirandelli, and Mauro Giacopelli for their extraordinary contribution to the design, development, and testing of the MARC platform.

References

Achtelik, M., Achtelik, M., Weiss, S., and Siegwart, R. (2011). Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments. In *Intl. Conf. Robotics and Automation*, pages 3056–3063.

Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N. (2009). Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments. In *Intl. Aerial Robotics Competition*, pages 582–586.

Bachrach, A., Prentice, S., He, R., and Roy, N. (2011). RANGE-Robust autonomous navigation in GPS-denied environments. *J. Field Robotics*, 28(5):644–666.

Bibuli, M., Bruzzone, G., Bruzzone, G., Caccia, M., Giacopelli, M., Petitti, A., and Spirandelli, E. (2012). MARC: Magnetic Autonomous Robotic Crawler Development and Exploitation in the MINOAS Project. In *Intl. Conf. Comp. and IT Appl. in the Maritime Industries*, pages 62–75.

Bonnin-Pascual, F. (2010). Detection of cracks and corrosion for automated vessels visual inspection. Master's thesis, University of Balearic Islands. Available at http://srv.uib.es/ref/1233.html.

Bonnin-Pascual, F., Garcia-Fidalgo, E., and Ortiz, A. (2012). Semi-Autonomous Visual Inspection of Vessels Assisted by an Unmanned Micro Aerial Vehicle. In *Intl. Conf. Intell. Robots and Systems*, pages 3955–3961.

Bouabdallah, S., Murrieri, P., and Siegwart, R. (2005). Towards Autonomous Indoor Micro VTOL. *Autonomous Robots*, 18:171–183.

Bruzzone, G., Caccia, M., Bertone, A., and Ravera, G. (2009). Standard Linux for embedded real-time robotics and manufacturing control systems. *Rob. and Comp.-Integr. Manufacturing*, 25(1):178–190.

Caccia, M., Bibuli, M., and Bruzzone, G. (2010a). Definition of navigational patterns. Technical report, CNR ISSIA, Genoa, Italy.

Caccia, M., Robino, R., Bateman, W., Eich, M., Ortiz, A., Drikos, L., Todorova, A., Gaviotis, I., Spadoni, F., and Apostolopoulou, V. (2010b). MINOAS a marine inspection robotic assistant: System requirements and design. In *Symp. Intell. Autonomous Vehicles*, pages 479–484.

Chowdhary, G., Johnson, E., Magree, D., Wu, A., and Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *J. Field Robotics*, 30(3):415–438.

Dryanovski, I., Valenti, R. G., and Xiao, J. (2013). An open-source navigation system for micro aerial vehicles. *Autonomous Robots*, 34(3):177–188.

Eich, M. and Vögele, T. (2011). Design and control of a lightweight magnetic climbing robot for vessel inspection. In *Mediterranean Conf. Control and Automation*, pages 1200–1205.

Fletcher, J. and Kattan, R. (2009). A ballast tank coating inspection data management system. http://www-.elcoship.com/papers.htm.

Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., and Pollefeys, M. (2012). Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV. In *Intl. Conf. Intell. Robots and Systems*, pages 4557–4564.

Grzonka, S., Grisetti, G., and Burgard, W. (2012). A Fully Autonomous Indoor Quadrotor. *Trans. Robotics*, 28(1):90–100.

Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.-M., Hirzinger, G., and Rus, D. (2007). Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. In *Intl. Conf. Robotics and Automation*, pages 361–366.

Haralick, R. and Shapiro, L. (1992). *Computer and Robot Vision*. Addison-Wesley.

Isard, M. and Blake, A. (1998). CONDENSATION—Conditional density propagation for visual tracking. *Intl. J. Computer Vision*, 29:5–28.

Jetstream (2013). Jetstream Magnetic Crawler M250. http://www.jetstreameurope.co.uk/products/magnetic-crawler.html.

Jung, M., Schmidt, D., and Berns, K. (2010). Behavior-based obstacle detection and avoidance system for the omni-directional wall-climbing robot CROMSCI. In Fujimoto, H. et al., editor, *Emerging Trends in Mobile Robotics*, pages 73–80. World Scientific.

Kaess, M., Johannsson, H., Englot, B., Hover, F., and Leonard, J. (2010). Towards autonomous ship hull inspection using the Bluefin HAUV. In *Intl. Symp. on Techn. and the Mine Problem*.

Kalra, L., Guf, J., and Meng, M. (2006). A Wall Climbing Robot for Oil Tank Inspection. In *Intl. Conf. Robotics and Biomimetics*, pages 1523–1528.

Kim, H., Seo, K., Lee, K., and Kim, J. (2010). Development of a Multi-Body Wall Climbing Robot With Tracked Wheel Mechanism. In Fujimoto, H. et al., editor, *Emerging Trends in Mobile Robotics*, pages 439–446. World Scientific.

Koveos, Y., Kolyvas, T., and Drikos, L. (2012). Robotic Arm Development for Ultrasonic Inspection. In *Intl. Conf. Comp. and IT Appl. in the Maritime Industries*, pages 258–268.

Matsue, A., Hirosue, W., Tokutake, H., Sunada, S., and Ohkura, A. (2005). Navigation of Small and Lightweight Helicopter. *Trans. Japan Soc. for Aeronautical and Space Sciences*, 48(161):177–179.

Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1–2):21–39.

Miko (2013). Steel-Climber. http://www.miko.no/steel-climber/.

Ortiz, A., Bonnin, F., Gibbins, A., Apostolopoulou, P., Bateman, W., Eich, M., Spadoni, F., Caccia, M., and Drikos, L. (2010). First steps towards a roboticized visual inspection system for vessels. In *Intl. Conf. Emerging Technologies and Factory Automation*.

Ortiz, F., Pastor, J. A., Alvarez, B., Iborra, A., Ortega, N., Rodriguez, D., and Conesa, C. (2007). Robots for hull ship cleaning. In *Intl. Symp. on Ind. Electronics*, pages 2077–2082.

Raut, S., Patwardhan, A., and Henry, R. (2010). Design and Construction of Glossy Surface Climbing Biped Robot. In Fujimoto, H. et al., editor, *Emerging Trends in Mobile Robotics*, pages 423–430. World Scientific.

Roberts, J. F., Stirling, T., Zufferey, J.-C., and Floreano, D. (2007). Quadrotor Using Minimal Sensing For Autonomous Indoor Flight. In *European Conf. Computer Vision*.

Shang, J., Bridge, B., Sattar, T., Mondal, S., and Brenner, A. (2008). Development of a climbing robot for inspection of long weld lines. *Ind. Robot: An Intl. Journal*, 35(3):217–223.

Silva, M. F. and Tenreiro, J. (2010). *A Survey of Technologies and Applications for Climbing Robots Locomotion and Adhesion*, Climbing and Walking Robots, Behnam M. (Ed.), chapter 1, pages 1–22. InTech.

Tâche, F., Fischer, W., Caprari, G., Siegwart, R., Moser, R., and Mondada, F. (2009). Magnebike: A magnetic wheeled robot with high mobility for inspecting complex-shaped structures. *J. Field Robotics*, 26(5):453–476.

Tanneberger, K. and Grasso, A. (2011). MINOAS Deliverable (D1): Definition of the Inspection Plan / Definition of Acceptance Criteria. http://minoasproject.eu/excibition/Publications/PDF/D1.pdf.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2000). Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1–2):99–141.

Wenzel, K. E., Masselli, A., and Zell, A. (2011). Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle. *J. Intel. Robotic Systems*, 61(1–4):221–238.

Yamaguchi, T. and Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, 21(5):797–809.

Zingg, S., Scaramuzza, D., Weiss, S., and Siegwart, R. (2010). MAV Navigation through Indoor Corridors Using Optical Flow. In *Intl. Conf. Robotics and Automation*.

Semantic Labeling: Classification of 3D Entities Based on Spatial Feature Descriptors

Markus Eich, Malgorzata Dabrowska, and Frank Kirchner
DFKI Bremen
(German Research Center For Artificial Intelligence)
Robotics Innovation Center
28359 Bremen, Germany
{*first_name.last_name*}*@dfki.de*

***Abstract*—Understanding the three-dimensional working environment is one of the most challenging tasks in robotics. Only by labeling perceived objects with semantics, a robot can reason about its environment, execute high-level plans and interact autonomously with it. A robot can perceive its enviroment by using 3D LIDAR systems, which generate 3D point cloud images of the environment. This data is perceived in a spatial domain, i.e. the raw data gives only positions of the measured points. The transfer from the spatial domain to the semantic domain is known as the gap problem in AI and one of the hardest to solve. In this paper we present an approach on how to extract spatial entities from unorganized point cloud data generated by a tilting laser scanner. Additionally, we describe how the extracted spatial entities can be mapped to entities in the semantic domain using feature descriptors. We also discuss, how a-priori knowledge about typical indoor environments can be used for semantic labeling.**

I. INTRODUCTION

Mobile robots are usually moving in and interacting with a 3D environment, so 3D perception is mandatory for such systems. Besides path planning and map building approaches, which have been thoroughly investigated over the last decade, robots need an understanding about their environment. Most navigation approaches in robotics are dealing with building a coherent metric world model, localize the robot within this model, and perform a trajectory planning within this generated model. Those algorithms are described as concurrent localization and map building or simply called SLAM (Simultaneous Localization and Mapping) [1].

As more processing power and more sophisticated 3D range sensors become available, an increasing number of approaches are dealing with the generation of coherent, metric 3D models. Nevertheless, it becomes clear that simple metric information about the environment is not sufficient to establish real autonomy in terms of interacting with and reasoning about the robot's environment. For intelligent behavior, it is preferable to send the robot high-level commands, like "Move to the table in the office!" or "Take the box from the table in the office!" instead of sending the robot to pure metric coordinates.

For high-level behavior, semantic reasoning and planning is mandatory but this cannot be achieved by looking only at metric information. The transfer between the semantic domain and the spatial domain is often referred to as the "semantic gap problem" in AI literature [2].

Figure 1 gives a motivation for our work. It shows a point cloud taken in our lab as well as the mobile robot used in our experiments. As a human it is quite simple to identify at least some of the spatial entities. A helpful a-prior knowledge in identifying the entities is that the point cloud was taken in our lab and that its shows a typical office environment. The point cloud is pre-processed using our segmentation algorithm described in this paper. What we do as humans automatically is, we look for structure. When we look at indoor environments usually everything has a rectangular shape (desks, tables, monitors, doors, shelves). A 3D shape of a door will not be easily identifiable by a human unless a handle is detected or the model of the door is embedded into a spatial entity, which is already identified as a wall. In this paper, we describe how spatial features can be extracted from 3D point cloud data and how extractable features can be used for entity classification. Additionally, we discuss how spatial reasoning can be applied on those features in order to increase the classification success rate of objects in an indoor environment.

The reminder of the paper is organized as follows: First, we give a brief overview of the related work in Section II. In Section III, we describe our approach on how to extract spatial features from unorganized point clouds. The concept on how spatial entities can be used for semantic scene recovery is given in Section IV. The results of our approach are given in Section V. We conclude our work and give some ideas about our research direction in Section VI.

II. RELATED WORK

The process of analyzing the environment using 3D laser range finders is basically done in two consecutive steps. In a first step, laser data is acquired using a tilting laser setup or 3D LIDAR (light detection and ranging) system and matched to an existing point cloud model. In a second step of the scene recovery process, geometric information is extracted from the merged point cloud data. This can be achieved by using 2D plane extraction [3] or the direct extraction of 3D primitives [4]. Some common surface reconstruction methods include the ball pivoting algorithm [5] and the Delaunay triangulation [6].

Several scans are usually taken and merged into a common reference frame of a global world model. If scans are taken from different positions, the problem is often referred to as scan registration or as SLAM problem [1]. A widely used method for matching two consecutive scans is the iterative

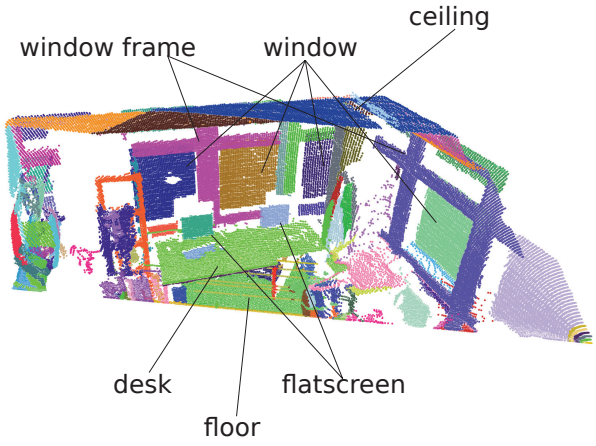


Fig. 1. A point cloud taken by the robot in our lab. It shows a typical office environment with a table, two flat screen monitors and several windows in the wall, which are easily recognizable by humans. The points are already clustered by our algorithm and displayed in different colors.

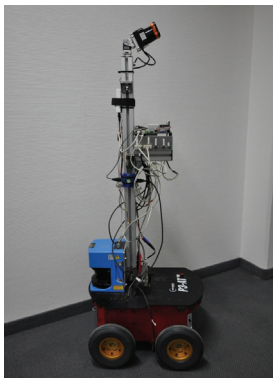


Fig. 2. The Pioneer 3 AT platform used in our experiments. The system is equipped with a tilting Hokuyo UTM-30LX laser range finder for point cloud generation. The system contains an embedded PC for navigation and data processing running ROS as communication framework.

closest point matching algorithm (ICP) [7]. The algorithm uses the closest point in two point sets and reduces the overall distance error by applying affine transformations until the two point sets match. The drawback of the ordinary ICP method is that a good pre-alignment of the two point sets is mandatory. ICP is otherwise likely to fail. Recent approaches use heuristics and a reduced information to pre-align the scan [8] or a high dimensional features descriptor in order to pre-align two consecutive scans [9].

Most of the described algorithms are aimed at reconstruction accurate surfaces and finding their application in reverse engineering of 3D structures. The accuracy of these algorithms results in high computational costs. They are thus not suited for robotic applications because the surface reconstruction can take up to several minutes.

The plane extraction algorithm described in [3] works well on noisy sensor data and uses an iterative region growing algorithm. The drawback of this approach is that it needs organized point clouds (i.e. the neighbor of each point is known). This is common for 3D LIDAR systems but not true for merged or registered scans. The approach we present in this paper will allow scene recovery from unorganized point sets and will also extract features from the spatial domain. The approach described in [10] provides a fuzzy clustering algorithm in order to segment a 3D scenery into cluster subsets without model learning. An approach of iterative triangulation of unorganized point clouds is described in [11]. All the described algorithms above are dealing with the spatial domain and are usable for identifying regions in LIDAR generated 3D point cloud data.

We want to bridge the gap between semantic and spatial representation of environment representation. Coming from the semantic side, [12] and [13] describe how semantic maps are used for high level planning and spatial reasoning. The authors describ in their work the bridging between the spatial domain and the semantic domain which they call S-Box (spatial box) and T-Box (taxonomy box). The semantic interpretation of physical objects is done by optical marker identification but not directly on spatial interpretation of point cloud data.

In approach described in [14], a constraint network is chosen in order to identify spatial entities such as wall, floor and doors. That work shows how an environment can be described efficiently by using only the two constraints “parallel to” and “orthogonal to”. We will extend this idea by adding additional spatial features which can be directly extracted using shape recovery on segmented regions.

Research in the RoboCup @Home [15] league among other things focusses on human-machine interaction in terms of abstract (verbal) goal descriptions and navigation and mapping in obstructed environments. Robots have to understand commands given by humans and their environment, i.e. they need to know where to find requested items and where to deliver them to. The approach we present in this paper fits perfectly well into this scenario, facilitating the adaptation to new and previously unknown (office) environments.

Two main contribution to the state of the art are given in this paper:

- An efficient method for scene recovery of unorganized point clouds based on iterative region growing and alpha shape recovery.
- The classification of spatial entities by extracted feature descriptors.

III. SHAPE RECONSTRUCTION

3D data of the scenery is generated using a tilting laser scanner. From this, we generate a data set Ψ with vector elements $\psi_i = (x_i, y_i, z_i)^T$, representing points in space. We process the raw data in two consecutive steps. First, we apply a fast region growing algorithm to segment the point cloud data into regions, which belong to a coherent plane. In a second step, the geometric data of the detected planes are extracted. The shape of the planes are of major interest. Therefore, the segmented point regions are polygonized using alpha shapes.

A. Region Growing

Our region growing approach follows partly the algorithm described in [3], with the extension that our approach is able to process unorganized point clouds by efficiently integrating k-nearest neighbor (KNN) search into the growing process. Computationally, the most expensive function in the algorithm is the KNN search which can be approximated with a runtime of $\mathcal{O}(n\log(n))$ [16]. Due to the requirement of being able to process unorganized point clouds which occur in registered scans, composed by using ICP [7], we have to optimize the KNN search during the region growing process.
Our key idea is to perform the cost-intensive KNN search at the beginning and store each point separately with a pointer to its KNNs. The growing plane keeps track of its growing frontier, i.e. each plane has its assigned nearest neighbors which are the next candidate for the growing process. During the region growing process, the KNNs of each point which is grown into the plane, are merged with the region frontier RC_{front}. In our approach (cf. Algorithm 1, R denotes the set

Algorithm 1 Region Growing Algorithm

1: $R \leftarrow \emptyset, RC \leftarrow \emptyset, RC_{front} \leftarrow \emptyset$
2: **for all** $\psi \in \Psi$ **do**
3: calculate $knn_i \in KNN$ with $\|\psi_i - knn_i\| < \delta$
4: **end for**
5: **while** $\Psi \neq \emptyset$ **do**
6: select random $\psi \in \Psi$
7: $RC \leftarrow \psi$
8: remove ψ from Ψ
9: $RC_{front} \leftarrow knn_\psi$
10: **while** $RC_{front} \neq \emptyset$ **do**
11: **if** $MSE(RC \cup \Omega(RC_{front},1)) < \varepsilon$ and $\|BFP(RC) - \Omega(RC_{front},1)\|_\perp < \gamma$ **then**
12: $RC \leftarrow RC \cup \Omega(RC_{front},1)$
13: remove $\Omega(RC_{front},1)$ from Ψ
14: $RC_{front} \leftarrow RC_{front} \cup knn_\psi$
15: **end if**
16: pop RC_{front}
17: **end while**
18: **if** $size(RC) < \theta$ **then** drop RC
19: **else**
20: $R \leftarrow R \cup RC$
21: **end if**
22: **end while**

of regions which are identified during the process, while RC denotes the region candidate. Ψ denotes the set of unprocessed points. The priority queue $knn_i \in KNN$ denotes the list of the k-nearest neighbors of point $\psi_i \in \Psi$. All $knn \in KNN$ as well as RC_{front} are implemented as priority queues and therefore arranged according to their distance. In order to increase memory efficiency, knn_i contains only the pointer to $\psi \in \Psi$. We introduce the function $\Omega(RC_{front}, i)$, which selects $\psi \in \Psi$ being the i-th neighbor of the growing region front.
The set of KNNs for each point ψ is computed once and stored in a priority queue, arranged according to their distance of the corresponding seed points (Line 1-4). The first point ψ is selected from the remaining points of set Ψ and added to the region candidate RC (Line 6-8). The set knn_ψ of ψ becomes the first region growing frontier RC_{front} (Line 9).
The algorithm processes each point in RC_{front} and checks if the merging criteria are met. This is done by checking the mean square error (MSE) of the plane including the new point as well as the orthogonal distance ($\|\|_\perp$) of the new point to the best-fitting plane (BFP) of RC (Line 11). In our implementation, we use the least square-fitting algorithm implemented in CGAL [17]. Eventually, the point selected by $\Omega(RC_{front},1)$ in Ψ is added to RC and removed from Ψ (Line 12-13). The corresponding k-nearest neighbors of ψ are merged into the priority queue RC_{front} (Line 14). If the queue RC_{front} contains no unprocessed points, the region candidate is added to the set of classified regions if it contains sufficiently many points (θ) or dropped otherwise (Line 18-21). The algorithm terminates if no unprocessed points are left in Ψ.

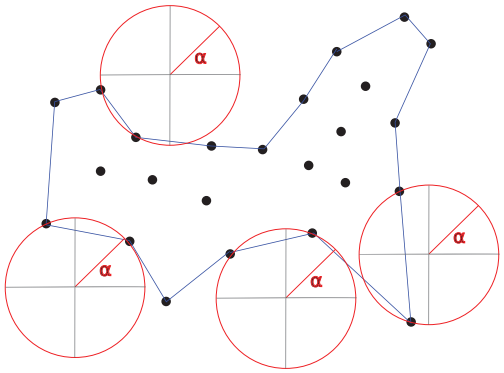


Fig. 3. Alpha shape example in 2D. Alpha shapes can be described as the shape which is created if a disk (or a sphere in 3D) with the radius α is rolled along the point cloud, starting with a border point. Every time a point is hit by the disk, it becomes the new pivoting point and the previous and the new pivoting point are connected.

B. Shape Recovery

After the region growing process described earlier, the shape of each segmented point region is extracted. Because the appearance of each shape is important for later scene analysis, we extract the concave hull, i.e. the polygon which approximates the shape of the original object. We choose the alpha shape approach, which is described more detailed in [18] and [19] in order to extract the concave polygon of the detected regions. Figure 3 gives an idea of how alpha shapes are computed. For the analytic description on shapes the reader is pointed to [18]. The alpha shapes are a special case of a Delaunay triangulation and are mathematically well defined. For each point in a segmented region R, a vertex is created. Two vertices p_i and p_j are connected if there exists a 3D sphere with radius α which has p_i and p_j on the boundary and does contain any other point in R. In our implementation,

we use the alpha shape reconstruction provided by [17]. Every alpha shape is then projected to a 2D plane for later shape analysis, e.g. eccentricity, size, shape, center of gravity. Note the dependency of the recovered shape on the radius α. If $\alpha \to 0$, each point in R form a separate shape, if $\alpha \to \infty$ the convex hull is calculated.

IV. SHAPE CLASSIFICATION

After the shapes are recovered from the unorganized point cloud, the goal is to classify the structure the robot perceives and to label the structure with semantics. To make semantic labeling possible in indoor environments, we make use of some basic assumptions. If we look around in a typical indoor environment like a household environment or an office environment, it is clear that most structures are of rectangular shape and mostly parallel or orthogonal to each other.

We will explain our semantic labeling approach using a simple example: Think of two persons who are able to communicate with each other. One person is able to perceive his environment, the other is not but has a perfect knowledge about the environment. One can think of the first person being in the spatial domain (without any understanding) and the other person being in the semantic domain.

Now think of how the person in the spatial domain would describe a *table* without knowing what a table is. A few spatial features would be sufficient until the person in the semantic domain can guess what is meant by the description (e.g. rectangular shape, parallel to the floor (or ceiling), height less than one meter from the floor, etc.).What happens is that the person in the semantic domain matches the available information to its internal model. Similar to the processing in a decission tree, every additional information given will increase the likelihood for a certain entity in the semantic model space.

Similar to the example above, the robot has to extract a vector of feature descriptors of the spatial entities in order to compare them with the semantic knowledge database. In a first approach, we define a set of vectors which are able to describe spatial entities of an environment. The feature vector Φ is defined as

$$\Phi \quad = \quad (A, E, R, \Theta)^T,$$

where $A \in \mathbb{R}^2$ is the area covered by the shape, $E \in \mathbb{R}$ describes the maximum extension of the detected entity, and $R \in [0,1]$ describes the "rectangularness" of the entity, with $R = 1$ describing a perfect rectangle. In our current approach, we only consider perfect rectangles as a binary decision. In later implementations, we want to also consider similarities to rectangles in order to increase the robustness of our approach. The reason to look for a rectangular structure is given by the observation that most of artificial objects have a rectangular plane in their structure (e.g. doors, shelves, closets, walls, monitors, fridges).

$\Theta \in [0,1]$ describes the relationship to other detected entities and is given by

$$\Theta \quad = \quad P(\omega, \alpha)$$

where $\omega \in \Omega$ is the target of the relationship and $\alpha \in A$ is the definition of spatial relationship. Ω is a set of targets, i.e. labels in the semantic domain; A holds the attribute space, which maps semantic attributes to spatial attributes. The labels in the semantic domain are defined in a model data base and include entities like *desk, table, door*, etc.

The attributes describe the spatial relationship between the detected entities (i.e. the parent of the relationship) and the target entity. An example for an attribute is

$$\begin{array}{lcl} \alpha & \to & [0,1]: \\ above & \to & (Pos\_Z_{parent} - Pos\_Z_{target}) < 0, \end{array}$$

which means that the target is below the parent entity. In our current implementation, we again consider a likelihood function in order to deal with uncertainties. For instance, two shapes can be parallel with the certainty of 0.9 due to noise and rounding differences in the extraction process.

$P \in [0,1]$ maps the relationship between the parent entities and the target entity, where 1 is the maximum likelihood. Another advantage of our approach is that it can be chained forward. For instance, an object on the table can be identified as a monitor with likelihood 0.8. Because the likelihood of the table is 0.5, the likelihood of being a monitor for the object is reduced to 0.4

Mapping semantic attributes to spatial relationship is the main contribution in our approach. A mapping between attribute *A above B* and a geometric expression would include that the z-value of the spatial entity A is higher than the z-value of B. Simply speaking, we solve the semantic classification by recursively calling the relationship function Θ until a **spatial axiom** is reached. The spatial axioms are defined by entities which do not depend on a relationship with other entities. They are defined as shapes having the spatial feature vector

$$\Phi \quad = \quad (A, E, R, 1)^T,$$

implying that there is no relationship needed in order to put semantic labels on spatial axioms, so the likelihood is set to 1.

An example of a spatial axiom is, for instance, a floor which is the lowest horizontal entity in reference to the robot coordinate system. So the floor is not identified by a relationship to other entities.

From the spatial axioms, the relationships Θ are resolved until the root of the classification tree is reached. The function Θ is a recursive call to the semantic domain space. In order to define a match between a model entity and an extracted spatial entity, we define the following similarity equation. The spatial feature descriptor of a model Φ_{model} and an extracted spatial entity Φ_{entity} are similar if

$$\Phi_{model} \quad \odot \quad \Phi_{entity} < \delta,$$

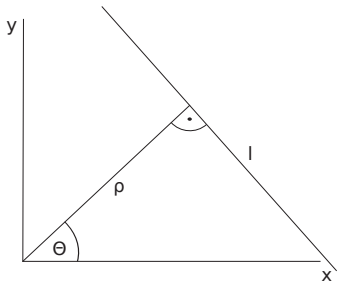


Fig. 4. A representation of a straight line using Hough parameters θ and ρ.

where δ is the similarity threshold. The feature disparity function $\odot$ between Φ_{model} and Φ_{entity} is defined as:

$$\begin{aligned}\Phi_{model} \odot \Phi_{entity} \quad := \quad & w_1\|A_{model} - A_{entity}\| \\ & + w_2\|E_{model} - E_{entity}\| \\ & + w_3\|R_{model} - R_{entity}\| \\ & + w_4\|\Theta_{model} - \Theta_{entity}\|\end{aligned}$$

The weights $w_1,..,w_4 \in \mathbb{R}$ are used to calibrate the disparity function.

When classifying structures, we are not dealing with full 3D perception but with a projection of 2D shapes in 3D space, which is typical for a LIDAR recovered structure. Considering shape analysis, all detected shapes are projected onto a 2D plane. In order to analyze the recovered shapes (cf. Section III), the planes are quantized by a projection in a 2D occupancy grid with a fixed grid of 1cm per pixel.

In our first approach, we focus on rectangular shapes, which will cover most of the objects found in indoor environment. In order to detect rectangular shapes in 2D, a Hough transformation is used [20]. The Hough transform is used to detect linear structure in images. It makes use of the fact that each line can be represented by $\rho = xcos\theta + ysin\theta$. This is shown in Figure 4 where ρ is the normal distance of the line and θ the normal angle. The 2D image is then searched with parameters x_i, y_i and the Hough transformation is evaluated if x_i, y_i belong to the edge. The local maxima in the Hough space $C(\rho, \theta)$ determine the parameters for the detected lines.

First, all line segments in the 2D image are extracted using the Hough transform. A pre-processing step is not necessary (with Canny or Sobel operators), because we work directly on the extracted alpha shapes as described in the previous chapter. In order to detect a rectangular shape, all lines are analyzed with regard to their intersection in four points and if the angle between the lines are $\sim 90^\circ$. Another approach which directly extracts rectangles in the parameter space is given in [21]. This approach is currently under investigation.

We now summarize our approach for the semantic labeling of spatial entities.

- A spatial database with labels and spatial description is set up. Each entity is represented by a spatial feature descriptor SFD_{model}. Each element is described by a spatial feature descriptor $\Phi_{model} = (A, E, R, \Theta)^T$. Some entities must be spatial axioms and not depending on the relationship to any other entity (i.e. $\Theta = 1$).
- The parameters A,E,R of Φ are extracted from the detected shapes in the spatial domain using the rectangle detection. Θ is evaluated until a spatial axiom is found.
- The disparity function $\Phi_{model} \odot \Phi_{entity}$ is evaluated. If the spatial feature descriptors are similar, the detected entity is matched with the model.

TABLE I
SEGMENTATION RESULTS OF OUR ALGORITHM. THE *-PARAMETERS WERE USED FOR FIGURE 5

δ (m)	γ (m)	ε (m)	θ (points)	Processing time (s)	Regions
*0.25	*0.02	*0.01	*100	*15.4	*109
0.25	0.02	0.01	300	13.4	52
0.25	0.10	0.01	100	18.4	63
1.00	0.02	0.01	100	28.5	105
0.25	0.02	0.10	100	16.7	96

V. RESULTS

We tested our algorithm in a typical indoor environment using the robot setup pictured in Figure 1. For the scan, we chose 100.000 points at an opening angle of 180° horizontally and 90° vertically. For the segmentation process, we selected the parameters given in Table I. The experiments were conducted on an Intel® Core2 Duo T9600 running on one core. The segmentation results are shown in Figure 5.

It is obvious that important spatial entities are segmented correctly and the shape is recovered correctly. Besides the correct outer shape, important spatial information can be derived from the shapes, e.g. their appearance, the plane normal, and the metric extension. By this means, the 3D scene can be interpreted using constraints based on pre-knowledge about the environment. In a typical office environment, a table can be described as a rectangular shape parallel to the floor. The floor itself may be identified as being the largest horizontal plane with the lowest z-value, while the perpendicular, rectangular shape above the desk may be classified as flat screen monitors. Figure 6 shows an example in order to extract the features A,E,R from the shape polygons. In oder to classify the shapes recursively, we define three semantic entities, i.e. *desk, screen, floor* with their spatial relationships.

$$\begin{aligned}\Phi_{desk} &= (1.28, 1.60, 1, \Theta(floor, parallel))^T \\ \Phi_{screen} &= (0.24, 0.57, 1, \Theta(desk, orthogonal) \\ &\quad \wedge \Theta(desk, above))^T \\ \Phi_{floor} &= (2.0, 2, 0, 1)^T\end{aligned} \tag{1}$$

The relationships *parallel*, *orthogonal*, and *above* map the corresponding entities to the spatial relationships

$$\begin{aligned}\alpha &\rightarrow [0,1]: \\ parallel &\rightarrow (N_{parent} \cdot N_{target}) - (|N_{parent}| \cdot |N_{target}|) < \varepsilon \\ orthogonal &\rightarrow N_{parent} \cdot N_{target} < \varepsilon \\ above &\rightarrow (Pos\_Z_{parent} - Pos\_Z_{target}) > 0\end{aligned} \tag{2}$$

In the relations above, N denotes the normal vector of the extracted regions belonging to the regarded shape (cf. Section

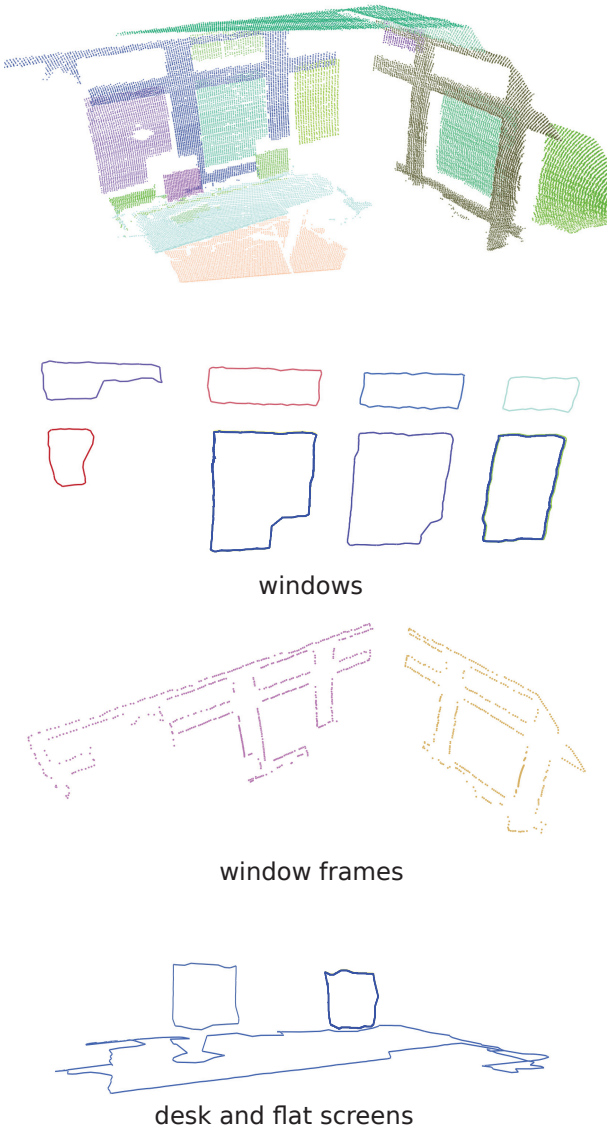


Fig. 5. The results of our segmentation algorithm. The planes in the point cloud are segmented correctly and assigned to different colors. Some selected alpha shapes are presented. The window shapes, the shape of the window frame, and the shape of the flat screens on the desk are clearly recognizable after the polygonization.

III). *Pos_Z* denotes the position of the shape (represented by the center of mass of the corresponding shape) in the vertical direction. Note that alpha shape or Hough transformed structures do not contain any relationship to each other. The extracted relationship features are processed during the region growing process, e.g. center of gravity, or normal vector.

Currently, we are able to extract all the spatial features mentioned in this paper and chain the extraction of the relationship function Θ. What is still missing to finally proof our concept is the implementation of the search algorithm that is able to match the spatial feature descriptors of detected entities to existing model feature descriptors. First experiments are promising considering only three simple entities, such as floor,

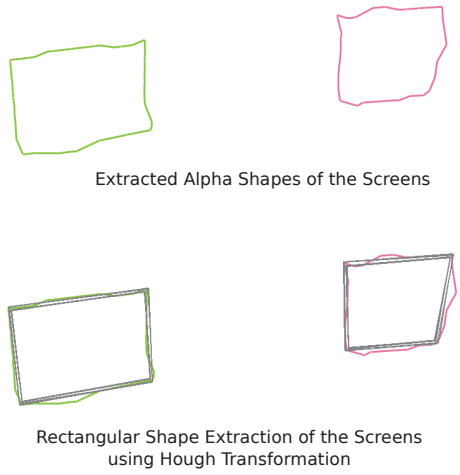


Fig. 6. The two flat screen monitors from our scenery in Figure 5. First, the alpha shapes are extracted. Using our rectangle detection algorithm, the shape is clearly defined as rectangular. From the shape, the parameters A (area) and E (extension) can be easily extracted. The rectangularness (R) is set to 1 for each of the shapes. The relation function Θ is omitted in this example because no other entities are related to the shapes.

table and monitor standing on the table. The next step is to match the features described in this paper to a semantic database of objects.

VI. CONCLUSION AND FUTURE WORK

In this paper we combined a method for recovering structure from unorganized 3D point clouds in the robotics domain. We presented two algorithms: The first transforms the point cloud into independent plane segments. The planes are then processed by computing the concave hull using an alpha shape algorithm. By this means, the shapes of objects can be recovered efficiently. We showed how rectangular structure can be detected from the extracted shapes after the polygonization and which features can be extracted in order to apply semantic labeling to spatial entities. We introduced a spatial feature description together with a spatial relationship mapping, allowing to find labels for detected entities. We finally provided first results of the geometrical extraction process.

Future work focusses on the implicit mapping between semantic and spatial entities. For the time being, we are able to fill the feature vectors for known entities and define a similarity function. The last step, i.e. the automatic classification of detected objects is still to be realized. Another research question is to build a descriptive onthology of the semantic space, allowing spatial reasoning in the semantic space and using our approach for bridging the gap between the semantic and the spatial domain. Or main research goal is to have a decriptive language for spatial entities which can be searched for object classification based on similarities in

the feature space.

VII. ACKNOWLEDGMENTS

The work is sponsored by the German Federal Ministry for Education and Research (BMBF) within the "SEMPROM" project under contract number 01IA08002.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[2] S. J. Russell and Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2003.

[3] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga, "Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors," in *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*. IEEE Press, 2007, p. 16.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, Juni 2007.

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The Ball-Pivoting Algorithm for Surface Reconstruction," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4. Los Alamitos, CA, USA: IEEE Computer Society, 1999, pp. 349–359.

[6] C. Kuo and H. Yau, "A Delaunay-based region-growing approach to surface reconstruction from unorganized points," *Computer-Aided Design*, vol. 37, no. 8, pp. 825–835, 2005.

[7] P. J. Besl and H. D. Mckay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[8] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "Heuristic-based laser scan matching for outdoor 6D SLAM," in *KI*, vol. 3698. Springer, 2005, pp. 304–309.

[9] R. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France*. Ieee, September 2008, pp. 3384–3391.

[10] N. Lomenie, "A generic methodology for partitioning unorganised 3D point clouds for robotic vision," in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings.* Ieee, 2004, pp. 64–71.

[11] Z. Marton, R. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics*, 2009.

[12] C. Galindo, J. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," in *Robotics and Autonomous Systems*, vol. 56, no. 11. Elsevier, 2008, p. 955966.

[13] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," in *Proc. IROS*. Citeseer, 2005, p. 34923497.

[14] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, p. 915926, 2008.

[15] "Robocup@home website," http://www.robocupathome.org.

[16] P. M. Vaidya, "An optimal algorithm for the all-nearest-neighbors problem," in *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 1986, pp. 117–122.

[17] CGAL Editorial Board, *CGAL User and Reference Manual*, 3rd ed., 2009.

[18] W. Shen, "Building boundary extraction based on lidar point clouds data," in *ISPRS08*. ISPRS, 2008, p. 157.

[19] F. Bernardini and C. L. Bajaj, "Sampling and reconstructing manifolds using alpha-shapes," in *CCCG*. IEEE Computer Society, 1997.

[20] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communication of the ACM*, vol. 15, January 1972.

[21] C. Jung and R. Schramm, "Rectangle detection based on a windowed Hough transform," in *Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium*. Citeseer, 2004, p. 113120.

An Application of Fuzzy DL-Based Semantic Perception to Soil Container Classification

Markus Eich
Robotics Innovation Center
German Research Center for Artificial Intelligence
28359 Bremen, Germany
Email: markus.eich@dfki.de

***Abstract*—Semantic perception and object labeling are key requirements for robots interacting with objects on a higher level. Symbolic annotation of objects allows the usage of planning algorithms for object interaction, for instance in a typical fetch-and-carry scenario. In current research, perception is usually based on 3D scene reconstruction and geometric model matching, where trained features are matched with a 3D sample point cloud. In this work we propose a semantic perception method which is based on spatio-semantic features. These features are defined in a natural, symbolic way, such as geometry and spatial relation. In contrast to point-based model matching methods, a spatial ontology is used where objects are rather described how they "look like", similar to how a human would described unknown objects to another person. A fuzzy based reasoning approach matches perceivable features with a spatial ontology of the objects. The approach provides a method which is able to deal with senor noise and occlusions. Another advantage is that no training phase is needed in order to learn object features. The use-case of the proposed method is the detection of soil sample containers in an outdoor environment which have to be collected by a mobile robot. The approach is verified using real world experiments.**

I. Introduction

In this work we propose a novel approach for object detection in cluttered outdoor environment using semantic perception. The application of this work is the detection of soil sample containers which have to be located and transported by an autonomous robot. The approach was developed primarily for a lunar demonstration scenario where a team of robots has to explore and to locate soil sample containers or drill cores and transport them to a collection point. The same application is also relevant for field robotics in outdoor and agricultural scenarios where samples of containers (e.g. soil or field sample equipment) have to be collected by a robot. An example is given in Figure 1. Other applications for our semantic perception approach include also e.g. humanitarian demining or SAR application (e.g. define mines as round, flat shapes, lying on the ground) if combined with a suitable platform (e.g. [1]). Furthermore, the approach can be applied to structural robot inspection on marine vessels [2] (e.g. define conduits as cylindrical shapes of certain size running vertically along walls or horizontally along ceilings). These are examples of spatio-semantic description which can be used to support perception and object detection in areas where structural knowledge and spatial relation are available to describe objects. In order to detect 3D objects, they usually have to be trained first. In contrast to this approach, we uses spatio-semantic descriptions of the objects defined in an ontology.

(a) (b)

Fig. 1. (a) Cylindrical core drill samples. (b) Soil sample containers used during the experiments.

The key idea behind the approach is that objects are semantically described how they look like in contrast to accurate 3D model matching. The features are extracted from 3D point clouds and are matched to a spatial ontology which describes how objects look like in a symbolic way. Beside the individual features which are extracted directly from the point cloud cluster (e.g. shape and geometry), spatial relations between objects are used to distinguish soil sample containers from other objects. This includes the pose of objects and the spatial relation to other spatial entities, such as ground, walls or other objects. In contrast to prior geometric model learning, our approach needs a spatio-semantic description of the objects. This allows an easy migration of the perception module to other domains. Another feature of our approach is that the spatial ontology is easily extendable, because spatio-semantic object description is intuitive to humans. The challenge with sensor based perception in general is that sensor input is usually error-prone, even if 3D LIDAR systems are used. Other problems in perception are occlusions and partial visibility. Because of the sensor inaccuracy, spatial reasoning has to deal with vagueness and probabilities. Therefore we have chosen an approach using Fuzzy DL which allows modeling of inaccuracies. The experiments within this work have been carried out with the mobile manipulation robot AMPARO (cf. Figure 2). The system was developed on the basis of the Pioneer 3-AT robot. The robot is equipped with a 3D LIDAR system, consisting of a Hokuyo UTM-30 laser range finder mounted on a pan-tilt unit. The robot has also a 6-DOF robot arm which is used for grasping and transport of soil containers.

II. Related Work

The process of object classification using 3D laser range finders has been in focus of robotic perception research in the

Fig. 2. The mobile robot AMPARO. The robot is equipped with 2D and 3D sensors for navigation and perception.

last few years. Perception is a mandatory field of robotics, because only by using perception, a robot is able to interact with the environment. Any high-level task usually includes interaction with objects (e.g. localization, transportation or manipulation). The perceived objects have to be semantically annotated in order to interact with them on a task level (e.g. "Grab a cup from the kitchen table" or "Put a stone sample into the container").

In order to extract objects from a 3D point cloud, the general structure of the point cloud has to be analyzed. This can be achieved by using 2D plane extraction [3] or by extraction of 3D primitives [4]. The approach described in [5] provides a fuzzy clustering algorithm for the segmentation of a 3D scenery into cluster subsets without prior model learning. An approach of iterative triangulation of unorganized point clouds is described in [6]. In another approach described in [7], Normal Aligned Radial Features are extracted from the point cloud. These features can be used to globally locate pre-learned objects within an unorganized point cloud. This approach is comparable to the extraction of a Persistent Feature Histogram (cf. [8]). The aforementioned algorithms are dealing with point-based features which are used to identify trained objects and structures in 3D point cloud data. No semantic information and object relationships are used in the aforementioned cases. The object models have to be trained, i.e. the objects have to be scanned before they can be matched with the sample point cloud.

On a higher semantic level, [9] and [10] describe how semantic maps can be used for robot navigation. The spatial domain and the semantic domain which is denoted as S-Box (spatial box) and T-Box (taxonomy box) are used to localize a mobile robot within a domestic environment. The semantic interpretation of physical objects is based on the detection of optical markers and is not directly based on the spatial interpretation of point cloud data. In [11] a constraint network is used for the identification of spatial entities, such as walls, floor, and doors. The paper shows how an indoor environment can be described efficiently by using only the two spatio-semantic constraints "parallel to" and "orthogonal to". The work does not include spatial features for object classification. An approach using semantic information based on the ontology of an indoor environment is used in [12]. The authors use plane detection and spatial relation between planes in order to classify objects of a typical indoor environment, such as shelves and tables.

An approach using spatial relation in 2D images by applying Fuzzy Ontology is given in [13]. The authors describe how spatial relationships and reasoning in a fuzzy domain can be used to annotate 2D images. Using description logic and relationships between objects and locations is presented in [14]. In this paper, a probabilistic DL language CR$\mathcal{ALC}$ (Credal $\mathcal{ALC}$) is used to model the ontology of an office environment. Objects are detected using SIFT features using a camera. The location of the robot can than be inferred based on the perceived objects. In contrast to the work mentioned above, we use spatial features which are extracted from segmented clusters of a 3D point cloud. We model the ontology of the sample containers based on the perceivable features. We used a fuzzy description logic approach which was first introduced by [15] to classify the sample container. In this paper we significantly extent the approach presented in [16] and [17] where we used planar features for object classification in an indoor environment. In this work, we extend the basic principles of spatial ontology by applying imprecise reasoning and knowledge about shape and structure.

III. Semantic Perception

A. Spatial Feature Description and Extraction

In order do detect soil sample containers, which are basically cylindrical shapes of different sizes, we first introduce a perception pipeline, depicted in Figure 3. The point cloud is generated using a tilting 2D laser scanner which is mounted on the robot AMPARO (cf. Figure 2). The use-case for our approach is the classification and the localization of soil sample containers as a prequisite for a fetch-and carry-application. For the spatial features extraction from a 3D point cloud we

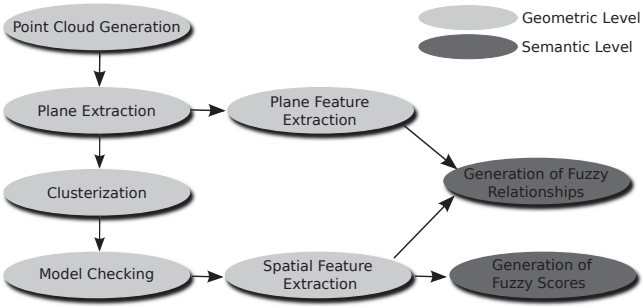


Fig. 3. Perception Pipeline

use the RanSaC-based segmentation approach described in [4]. Because the spatial relations between objects and the ground is needed, the ground plane is segmented and extracted first. This is done using the region growing algorithm described in [16]. The remaining points, not belonging to the ground plane, are clusterized and the features are extracted in the next step. The spatio-semantic features which are used for the classification are in detail:

$$\Phi_1 = ModelTypeFit \qquad |\{\Phi_1 \in \mathbb{R} | 0 \leq \Phi_1 \leq 1\}$$
$$\Phi_2 = Radius \qquad |\Phi_2 \in \mathbb{R}^1$$
$$\Phi_3 = Height \qquad |\Phi_3 \in \mathbb{R}^1$$

$$\Phi_4 = Orientation \quad |\Phi_4 \in \mathbb{R}^3$$
$$\Phi_5 = Position \quad |\Phi_5 \in \mathbb{R}^3$$
$$\Phi_6 = HeighToGround \quad |\Phi_6 \in \mathbb{R}^1$$
$$\Phi_7 = OrthoToGround \quad |\{\Phi_7 \in \mathbb{R} | 0 \leq \Phi_6 \leq 1\}$$
$$\Phi_8 = ParallelToGround \quad |\{\Phi_8 \in \mathbb{R} | 0 \leq \Phi_6 \leq 1\}$$

Φ_1 is extracted directly from the RanSaC classifier and given by the percentage of how many points of the cluster are accepted by the model. Φ_4 is calculated by the two points with normals (n_1, n_2) defining the cylindrical model by calculating $n_1 \times n_2$. The diameter of the infinite cylinder Φ_2 is created by projecting the lines going through the normals n_1 and n_2 onto the plane being orthogonal to the main axis (i.e. Φ_4). The intersection point of the two lines in the orthogonal plane defines the center of the cylinder shape model. The distance between the center point and the point n_1, projected onto the same plane, defines the radius of the model Φ_2. The height of the detected object Φ_3 is not directly extracted by the RanSaC model, because the RanSaC model describes a cylinder of infinite length. To get the height of the object, we project each point p_i from the cluster onto the infinite axis, defined by Φ_4. The height feature Φ_3 is then defined by

$$\Phi_3 = \|\arg\min_{p_i \in \Psi}(p_i \cdot \Phi_4), \arg\max_{p_i \in \Psi}(p_i \cdot \Phi_4)\| | p_i \in \mathbb{R}^3$$

The position feature Φ_5 of the object is defined by the point on the main axis which is closest to the ground. Hence

$$\Phi_5 = \arg\min_{p_i \in \Psi}(p_i \cdot \Phi_4, z) | p_i \in \mathbb{R}^3.$$

All spatial features described above are extracted from every cluster Ψ from the original point cloud. The features Φ_6 and Φ_7 are used to relate the extracted object to the "GroundPlane" which was detected using the region growing method described in [16].

Φ_6 is generated by projecting Φ_5 onto the ground plane. Because sensor inaccuracies lead to the fact that objects are rarely strict orthogonal or parallel to each other in a mathematical sense, a fuzzy representation is used for these concepts. We define

$$\Phi_7 = 1.0 - max((a,0,0) \cdot \Phi_4, (0,b,0) \cdot \Phi_4) | P(a,b,c) \in \mathbb{R}^3$$
$$\Phi_8 = max((a,0,0) \cdot \Phi_4, (0,b,0) \cdot \Phi_4) | P(a,b,c) \in \mathbb{R}^3$$

where the ground plane is described by ax+by+cz+d=0. This allows an estimation of the likelihood of two objects being orthogonal or parallel to the ground plane.

In order to classify and to locate the soil sample containers, the containers are described in a spatio-semantic way using an structural ontology. Using spatial features instead of an algorithm which only detects the shape of an object has several advantages. The main advantage is that these features mentioned above are easily extendable to other domains. For instance, the model feature can be replaced by other types of shape models, such as cuboids, planes and spheres. Applying a combination of these features, as described in the following sections, makes the approach more robust in contrast to other perception approaches. In the next section, an ontology is defined which hosts the knowledge base about how the soil sample containers look like. Based on a fuzzy reasoning approach, the extracted features are matched against the fuzzy logic concepts.

B. Building an Ontology for Soil Sample Container Classification

In this work, we combine perceivable features of objects and the environment with a knowledge base system. Semantic object annotation is this case accomplished by a query to an ontology. The knowledge about the sample containers and the spatial relation are defined in the T-Box (describing the terminology of the ontology). The A-Box describes the assertional part of the ontology, i.e. the individuals. The ontology of the domain is manually generated using the spatial knowledge about the sample containers in scope. The geometric features are described as concepts of the T-Box. The individuals of the knowledge base are automatically generated by the perception layer (cf. Figure 4). The A-Box is updated after each 3D scan and the reasoner is triggered to classify the soil sample containers within the point cloud, based on the spatio-semantic description of the T-Box. We use a

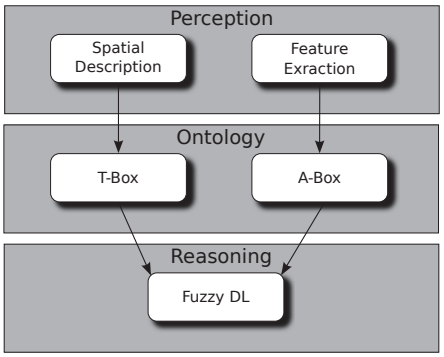


Fig. 4. The reasoning architecture using Description Logic. The A-Box is provided by the feature extractor from the perception side. The T-Box is modeled using Fuzzy DL syntax.

FuzzyDL reasoner which operates on geometric features and the spatial relationships between the extracted point clusters. By using DL we are able to describe features and spatial relationships of the soil container in terms of roles, concepts and individuals. An general overview of DL can be found in [18, Chapt.1]. Because of measurement inaccuracies and occlusions, extracted features cannot be processed by a discrete DL reasoner. We use therefore a fuzzy $\mathcal{SHIF}(D)$ language which was firstly introduced by [15]. Fuzzy $\mathcal{SHIF}(D)$ is a description logic language which allows concept negation, concept union and intersection, concept negation as well as existential and universal restrictions on roles. The syntax of the DL language $\mathcal{SHIF}(\mathcal{D})$ is as follows (see also [15] for a more detailed description on FuzzyDL):

$$C, D := \top | \bot | A | C \sqcup D | C \sqcap D | \neg C | \forall R.C | \exists R.C$$

where C and D are concepts, A defines the atomic concept, and R.C the role of a concept. For our spatial reasoning approach we make also use of the Gödel t-norm and Gödel t-conorm to express Fuzzy DL union and intersection (i.e. $C \sqcap_G D := min(C, D)$ and $C \sqcup_G D := max(C, D)$ respectively. For modeling of the perceptual domain, we make use of the weighted sum concept. The weighted sum of a FuzzyDL

concept is defined by $\omega_1 C_1 + ... + \omega_k C_k$. In order to model the ontology of the soil sample containers, the following concrete names are assigned to the different DL concepts (denoted as C_i) and roles (denoted as R_i):

$$
\begin{aligned}
C_1 &:= CylinderType & R_1 &:= fitModel \\
C_2 &:= Height & R_2 &:= hasHeight \\
C_3 &:= Radius & R_3 &:= hasRadius \\
C_4 &:= DistanceToGround \\
C_5 &:= OrthogonalToGround \\
C_6 &:= ParallelToGround \\
R_4 &:= hasDistanceToGround \\
R_5 &:= isOrthogonalToGround \\
R_6 &:= isParallelToGround
\end{aligned}
$$

In the case of the soil sample containers, the objects in scope are cylinders of different heights and diameters. Each object has also a defined pose (orientation and position) and a relation to the ground plane. We are only interested in objects which are standing on the ground and are not located on top of a rock formation or have fallen down. The pose in relation to the ground plane, the orientation and the object features like height and diameter have to be taken into account. The problem we are mainly facing during the perception process is that perception data is error-prone. In order to cope with inaccuracy and occlusion, fuzzy concepts are used to described the spatio-semantic features (cf. Figure 5). For the container

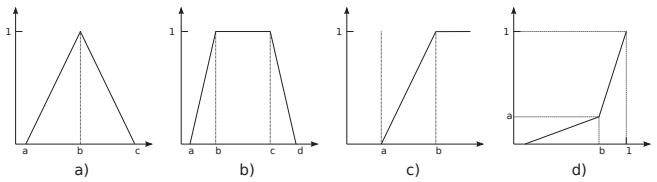


Fig. 5. The used fuzzy concepts for the spatial reasoning approach. The features are represented using the fuzzy concepts of triangular_set (a), crisp_set (b), right_shoulder (c) and the linear_modifier (d).

detection, we use the following concrete name assignment for the fuzzy sets which is used to define the ontology of the sample container:

$$
\begin{aligned}
CylinderType &:= right\_shoulder(a,b) \\
Height &:= triangular\_set(a,b,c) \\
Radius &:= triangular\_set(a,b,c) \\
DistanceToGround &:= triangular\_set(a,b,c) \\
OrthogonalToGround &:= linear\_modifier(a,b) \\
ParallelToGround &:= linear\_modifier(a,b)
\end{aligned}
$$

After all needed concepts are defined for the sample container extraction, the ontology can be generated. For the experiments, two different types of sample containers have to be classified: Containers which are standing upright and can be transported and containers which have fallen down and might be damaged. We use the two concepts of "UsableContainer" and "DamagedContainer". In terms of the T-Box of the sample container ontology we define

$$
\begin{aligned}
UsableContainer &\sqsubseteq SampleContainer \\
DamagedContainer &\sqsubseteq SampleContainer \\
DamagedContainer &\sqcup UsableContainer = \bot \\
UsableContainer_{sum} &\equiv \\
&\omega_1(\exists fitModel.CylinderType) + \\
&\omega_2(\exists hasHeight.Height) + \\
&\omega_3(\exists hasRadius.Radius) + \\
&\omega_4(\exists hasDistanceToGround.DistanceToGround) + \\
&\omega_5(\exists isOrthogonalToGround.OrthogonalToGround) \\
UsableContainer_{intersect} &\equiv \\
&(\exists fitModel.CylinderType) \sqcap_G \\
&(\exists hasHeight.Height) \sqcap_G \\
&(\exists hasRadius.Radius) \sqcap_G \\
&(\exists hasDistanceToGround.DistanceToGround) \sqcap_G \\
&(\exists isOrthogonalToGround.OrthogonalToGround) \\
DamagedContainer_{sum} &\equiv \\
&\omega_1(\exists fitModel.CylinderType) + \\
&\omega_2(\exists hasHeight.Height) + \\
&\omega_3(\exists hasRadius.Radius) + \\
&\omega_4(\exists hasDistanceToGround.DistanceToGround) + \\
&\omega_5(\exists isParallelToGround.ParallelToGround) \\
DamagedContainer_{intersect} &\equiv \\
&(\exists fitModel.CylinderType) \sqcap_G \\
&(\exists hasHeight.Height) \sqcap_G \\
&(\exists hasRadius.Radius) \sqcap_G \\
&(\exists hasDistanceToGround.DistanceToGround) \sqcap_G \\
&(\exists isParallelToGround.ParallelToGround)
\end{aligned}
$$

For the experiments, the concepts of $UsableContainers$ and $DamagedContainers$ are modeled in two ways for later comparison. The weighted sum concepts makes use of a more soft definition of the role implication. This means that if features which are satisfied with a lower likelihood can be compensated by other features, having a higher likelihood. This approach needs a sensitive setting of the weights $\omega_1 ... \omega_5$, as we discuss in the result chapter. The use of the intersection implication, which uses the Gödel T-norm, is a more strict model of the concepts $DamagedContainer$ and $UsableContainer$. This implies that the concept is unsatisfied if at least one existential restriction is unsatisfied. The comparison between the different rule implications is discussed in the results section. The A-Box of the ontology is populated with the roles (properties) of the individual instances, consisting of the features described in Section III-A. The individuals are generated by the perception layer and provided to the ontology. For each extracted point cloud cluster, denoted as $object_i$, the extracted spatial features are added to the ontology.

$$
\begin{aligned}
fitModel(object_i, \Phi_1) \\
hasRadius(object_i, \Phi_2) \\
hasHeight(object_i, \Phi_3) \\
hasDistanceToGround(object_i, \Phi_6)
\end{aligned}
$$

$$isOrthogonalToGround(object_i, \Phi_7)$$
$$isParallelToGround(object_i, \Phi_8)$$

To query the knowledge base, after the individuals are updated in the A-Box of the ontology, a min-instance query is executed in order to check the satisfiability of the models DamagedContainer and UsableContainer. This is achieved by inf$\{n|\mathcal{K} \models$ (instance $object_i$ DamagedContainer n)$\}$ and inf$\{n|\mathcal{K} \models$ (instance $object_i$ UsableContainer n)$\}$, respectively.

IV. Experimental Results

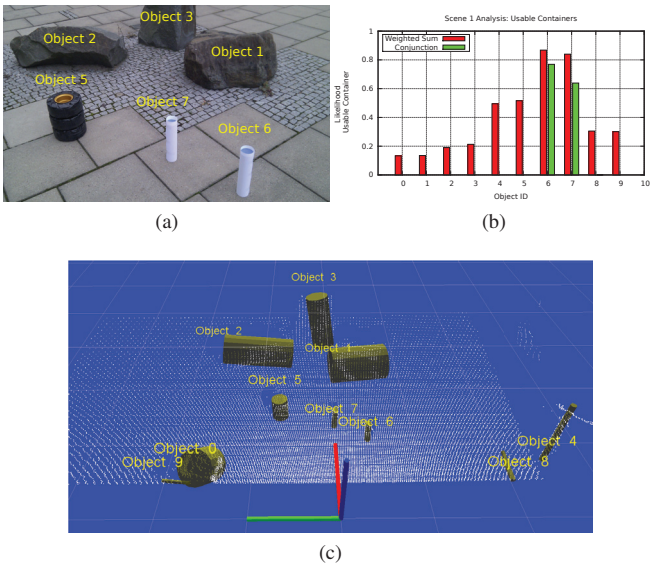
For the experiments the mobile robot Amparo is used which is equipped with 3D LIDAR System. We want to locate the sample containers which have a height of 30cm and a radius of 3.5cm. The containers have a cylindrical shape so each point cluster is checked if it fits the cylindrical model. The sample containers are modeled using the following fuzzy concepts:

$$\begin{aligned} CylinderType &:= right\_shoulder(0.2, 0.8) \\ Height &:= triangular\_set(0.20, 0.30, 0.40) \\ Radius &:= triangular\_set(0.01, 0.035, 0.07) \\ DistanceToGround &:= triangular\_set(0.0, 0.20, 0.40) \\ OrthogonalToGround &:= linear\_modifier(0.7, 0.1) \\ ParallelToGround &:= linear\_modifier(0.7, 0.1) \end{aligned}$$

Thereby it is differentiated between damaged containers (i.e. they are lying on the ground) and graspable containers. In the following experiments, the object descriptions given above are used to extract and classify the soil sample containers from a 3D point cloud scene. This is the prequisite for the fetch-and-carry use-case where the robot has to grasp and transport usable soil sample containers.

A. Scene 1 Analysis

In the first experiment, two soil sample containers and similar objects have been placed within in the detection range. The correct sample containers are segmented as Objects 6 and 7, respectively. A similar object (Object 5) is also visible in the scene (cf. Figure 6a). The point cloud is analyzed and all point clouds of objects are clusterized, resulting in nine sample containers candidates (cf. Figure 6c), resulting in seven false positives. After applying the FuzzyDL classification method, the two soil containers are identified correctly. We test two different fuzzy implications on the same scene, i.e. the weighted sum of the features as well as the more strict conjunction rule (cf. Figure 6b). It becomes obvious that objects, which have similar features, have a higher overall score, even if some features are not matched at all if the weighted sum implication is used. This is true for Object 5 where all features get a high overall score from the reasoner, even if the diameter has a lower score. The same holds for the Object 4 which is detected outside the scene. In order to use the weighted sum implication, a threshold has to be carefully chosen in order to classify the soil containers. Better results are provided by the use of the conjunction implication of the knowledge base. A comparison between the different rule implications are shown in Figure 6b. Only the correct Objects 6 and 7 are identified as correct soil containers if the conjunction is used and no false positives are classified.



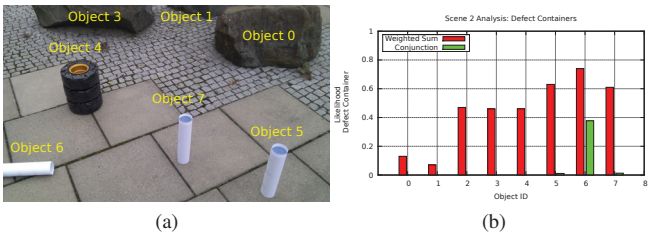
(a) (b)

(c)

Fig. 6. Experiment using scenario setup 1. (a) The image taken of scene 1. The reasoner should provide all sample containers which are standing upright on the ground plane. (b) Comparison between weighted sum vs. the conjunctive implication of the reasoner. (c) The raw point cloud generated from the scene, with all extracted objects as seen by the laser.

B. Scene 2 Analysis

Scene 2 has a similar setup as scene 1. In this scenario, the damaged container has to be detected. The container is identified by the rule implying that a damaged container is horizontally aligned with the ground plane. Other features used are the same as for scene 1 analysis. The results are shown in Figure 7. In this experiment, the difference between the use of the weighted sum implication and the conjunction implication becomes more obvious. The weighed sum provides an almost equal match for the Objects 5 and 7 which match the model of a damaged container, except the relation-to-ground feature. The query to the knowledge base using the conjunctive



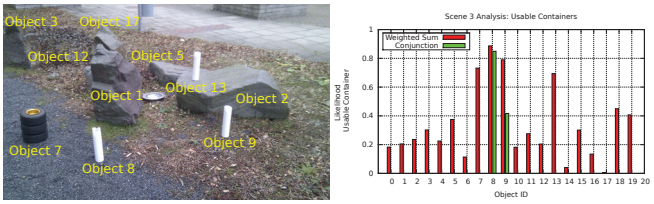
(a) (b)

Fig. 7. Experiment using scenario setup 2. (a) Image taken of scene. The reasoner should provide all soil containers which are lying on the ground. Seven candidates are detected and the reasoner produces the right match (i.e. Object 6). (b) The perception results using our approach showing the difference between weighted sum vs. the conjunctive implication of the reasoner.

implication results only in one significant match (Object 6) and no false positive. This shows that the more strict conjunctive implication is better in differentiating between similar objects.

C. Scene 3 Analysis

In this experiment, a more cluttered environment is chosen to verify the robustness of our approach (cf. Figure 8). Again, a similar cylindrical shape is placed in the vicinity of the sensor (Object 7). Other objects, such as bushes, rocks and obstructions can be detected by the laser. Object 8 and 9 are the correct soil sample containers which should be classified. The segmentation of the point cloud extracts 19 container candidates. Using the weighted sum implication results in a high score for Objects 7,8,9 and 13. This is due to the fact that in the weighted sum implication low scores in one features are compensated by good matches in other features. This can be seen in the scores of Object 7 (which matches all criterions except the diameter) and Object 13 (which matches all criterions except the distance-to-ground relation). Using the restrictive



(a) (b)

Fig. 8. Experiment using scene setup 3. (a) The image taken of scene 3. It shows three sample containers (8,9,13), of which one is not on the ground (13). Object 7 does not match the required features, even if the shape matches. (b) Scene 3 is analyzed using our semantic perception approach.

conjunctive implication results in a match only for Objects 8 and 9. The last experiment suggests the robustness of our approach, even if the environment is cluttered. If the restrictive conjunction implication is used, no false positives are detected. All experiments show also, that the fuzzy detection approach is able to cope with sensor inaccuracies.

V. Conclusion and Future Work

In this work we presented a novel approach for the detection of soil sample containers using semantic perception. We combined spatial features, which are extracted from a laser generated point cloud, and a Description Logic based reasoning approach. This allows the definition of an spatial ontology about the objects, their perceivable features, and their spatial relation. In order to cope with sensor noise, occlusions and imprecise knowledge about the environment, we made use of a fuzzy based reasoning approach which deals with vagueness about the perceived features. We showed that our approach can be used to detect objects in a cluttered environment based only on the semantic description of the features. Future work will include the extension of the perceivable features to other shapes. We only used the cylindrical form as a proof of concept in this work. We are going to implement other basic shapes, such as cuboids and spheres. We are going to apply our approach to other domains as well, e.g. to the domain of indoor environments and to structural analysis of parts of cargo holds on seagoing vessels.

Acknowledgment

The project IMPERA is funded by the German Space Agency (DLR, Grant number: 50RA1111) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament.

References

[1] M. Eich, F. Grimminger, and F. Kirchner, "A Versatile Stair-Climbing Robot for Search and Rescue Applications," in *IEEE International Workshop on Safety, Security and Rescue Robotics, 2008. SSRR 2008*. Ieee, Oct. 2008, pp. 35–40.

[2] M. Caccia and R. Robino, "MINOAS a marine inspection robotic assistant: System requirements and design," in *Proc. of the IFAC Symposium on Intelligent Autonomous Vehicles. (IAV-10)*, no. 1, Lecce, Italy, 2010.

[3] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga, "Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors," in *IEEE Int. Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*. IEEE Press, 2007, pp. 1–6.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.

[5] N. Lomenie, "A generic methodology for partitioning unorganised 3D point clouds for robotic vision," in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings.* Ieee, 2004, pp. 64–71.

[6] Z. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics*, 2009.

[7] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "NARF: 3D Range Image Features for Object Recognition," in *Proc. of IROS2010, Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, Taipei, Taiwan, Oct. 2010.

[8] R. B. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *In Proc. of the 21st IROS2008, Nice, France*. Ieee, Sep. 2008, pp. 3384–3391.

[9] C. Galindo, J. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.

[10] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *IEEE/RSJ IROS 2005, In Proc. of.* IEEE, 2005, pp. 2278–2283.

[11] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[12] S. Albrecht, T. Wiemann, M. Günther, and J. Hertzberg, "Matching CAD object models in semantic mapping," in *ICRA 2011 Workshop Semantic Perception, Mapping and Exploration*, Shanghai, China, 2011.

[13] C. Hudelot, J. Atif, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation," *Fuzzy Sets and Systems*, vol. 159, no. 15, pp. 1929–1951, Aug. 2008.

[14] R. B. Polastro, F. Corrêa, F. G. Cozman, and J. Okamoto, "Semantic mapping with a probabilistic description logic," in *SBIA 2010, In Proc. of.*, A. C. da Rocha Costa, R. M. Vicari, and F. Tonidandel, Eds., vol. 6404. Springer, 2010, pp. 62–71.

[15] F. Bobillo and U. Straccia, "fuzzyDL: An expressive fuzzy description logic reasoner," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008.* Ieee, Jun. 2008, pp. 923–930.

[16] M. Eich, M. Dabrowska, and F. Kirchner, "Semantic Labeling: Classification of 3D Entities Based on Spatial Feature Descriptors," in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, Alaska, USA, 2010.

[17] M. Eich and F. Kirchner, "Reasoning About Geometry: An Approach Using Spatial-Descriptive Ontologies," in *In Workshop Artificial Intelligence and Logistics (AILog), (ECAI-10)*, K. Schill, B. Scholz-Reiter, and L. Frommberger, Eds., Lisbon, Portugal, 2010.

[18] D. Nardi, R. Brachman, and Others, "An introduction to description logics," *The description logic handbook: theory, implementation, and applications*, pp. 1–40, 2003.