A Self-Contained Localization Unit for Terrestrial Applications and Its Use in Space Environments

Markus Emde, Bjoern Sondermann, Juergen Rossmann

Institute for Man-Machine Interaction, RWTH Aachen University, Germany
e-mail: {emde, sondermann, rossmann}@mmi.rwth-aachen.de

Abstract

Global navigation satellite systems (GNSS) have been introduced on earth as standard localization services in many domains allowing a wide range of applications. Unfortunately, extraterrestrial exploration missions do not benefit from these developments, nor do indoor environments on earth. We developed a general concept for solving the localization problem in unstructured environments. Firstly, this concept was applied to the forestry domain on earth and general purpose software modules were implemented. Secondly, these modules were reused on a mobile robot in a planetary exploration scenario. Based on the application-independent localization concept we built a mobile self-contained localization unit equipped with several sensors and a processing unit allowing precise estimation of its pose in different domains.

1 Introduction

Mobile robots are equipped with a multitude of different sensors to percept their environment. Thus, they are able to detect obstacles and to fulfill certain tasks. With an increasing complexity of tasks the requirements to the sensors increase as well. Simple distance sensors suffice for a reliable collision avoidance [1], but for object recognition or localization more sophisticated environment information is needed. Therefore, optical sensors like laser scanners or camera systems are often used. With techniques like stereo image processing, polarization mode dispersion (PMD) or light projection sensors (Kinect 2010) camera systems are available that provide color or intensity as well as depth information. On the one hand, this data provides much more environment information, on the other hand, the processing of the detailed data and the extraction of the required information is more complex. Thus, with regard to localization, there are many different approaches for environment perception and modeling [2, 3, 4, 5]. A simple way to represent the environment in a model is to find and describe distinctive features [2] in the sensor data. These features can be stored and managed in databases resulting in a simple environment model, as in [3]. The representation of the environment by a set of features in the sensor data is suitable for small operational areas. When the area of the application scenario is bigger, other or additional information is used [4, 5].

In the automotive sector stereo cameras and radar sensors are commonly used for observing the area in front of the car [6, 7, 8]. The depth information can be used to evaluate the current situation on the street and possible hazard sources. Occupancy grids are commonly used [9]. They divide the visible space in discrete grid cells and check them for occupancy by using the sensor data [10]. Often, additional information like velocity and moving direction of the occupied cells are extracted and stored for predicting future occupancy. Occupancy grids taken from multiple data-gathering positions can be composed into a global map of the environment. For position estimation of mobile robots a registration between a new grid and the global map has to be found [11]. This is possible in highly structured, but not self-similar environments.

A semantic environment model as presented in this paper is used in very few applications, where it is also limited to specific scenarios [12, 13].

In the following, we will explicate the localization concept in chapter 2. Particularly, we will focus on the semantic environment model and the essential idea of semantic landmarks. By combining all components, that are relevant for localization into one localization unit, we built a self-contained prototype allowing an application-independent pose estimation on the basis of semantic landmarks. We present the resulting device and its use in forestry on earth, as well as in a planetary exploration scenario in chapter 3. The evaluation of the pose estimation results is presented in chapter 4, and in chapter 5 we discuss further application areas and some possible adaptations to the landmark detection modules of the localization method. At last we will summarize and give a short outlook on future work and expectations.

2 Application-independent localization

Starting point of our approach is an application-independent localization framework which is integrated in an existing simulation software. The idea was firstly introduced in [15]. The localization module itself is the central element of our application-independent localiza-

tion framework. On the one hand, this module uses local sensor data of its environment as input. On the other hand, semantic landmarks which are stored in databases, are used by matching them against the perceptions gathered from the local sensor data. In this section, we will present the main modules or sub-frameworks (see Figure 1) in detail.

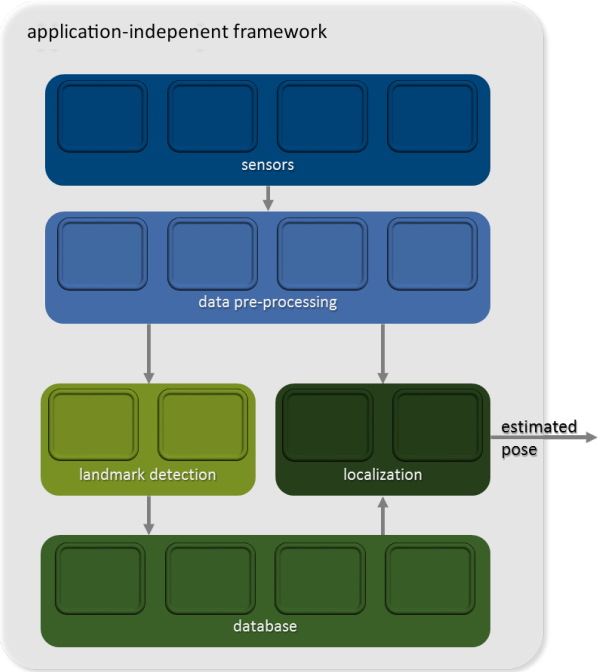

Figure 1. Flow-chart of the presented application independent localization concept

2.1 Sensor Integration and Simulation

As the application-independent localization approach needs sensor data of its environment as input data, the underlying simulation software has been extended by a sensor framework (introduced in [16]). It provides methods for the modeling, simulation and visualization of a wide range of sensing components, while logging and playback mechanisms allow for an efficient offline development based on sensor data. Furthermore, the introduction of various error models enables the detailed analysis of sensor data processing algorithms under different boundary conditions. Within the simulation environment, the sensor framework itself uses a consistent data interchange concept for interoperability between all components. It is based on an IO-board metaphor and allows for a standardized input and output of sensor data. As all data types representing sensor data output or input inherit from this generic data type, this data interchange concept represents the first layer of the sensor framework. The data interchange concept has been applied to the whole application-independent localization approach in a second step.

A special feature of the sensor frameworks is the distinction between simulated and virtual sensors. Due to hight cost and other limiting factors real hardware sensors may not be available at all times to carry out necessary test series. Using simulated sensors in an appropriate testbed provides an alternative. By combining simulated sensors, which yield ideal data, with error models, the behavior of real components are emulated, providing realistic sensor data for subsequent algorithms. Virtual sensors are used to feed algorithmic results or recorded data into the network of connected sensor framework components. Virtual sensors have the same features as the envisioned type of sensors. Data ports and recorded data or algorithmic results are treated as conventional sensor data.

The result of the localization algorithm represents an absolute position estimation. This information is provided by the use of a virtual sensor setting the absolute position information on the output of an adequate sensor component, which provides absolute position information in the same output format as used by a GPS sensor, but its position estimation is based on different sensor data inputs.

2.2 Semantic landmarks

Our localization concept is based on local environment perception as well as on semantic navigation maps as presented in [18]. The localization uses an abstract landmark definition as central datatype. There are two specifications: a *Landmark* and a *Perception*. Figure 2 shows a short excerpt of the used data structure. A navigation map consists of landmarks that have been generated from existing data like preliminary maps or remote sensing data as described in detail in [18]. Landmarks contain a position and a reliability value indicating the certainty of the detection algorithm when the landmark was observed. Perceptions are semantic objects that are detected by the sensors of the localization unit. They contain a direction and a distance which is equivalent to a position in the local reference system of the sensor at the moment of acquisition. The specific inheritances of landmarks and perceptions contain additional attributes describing the particular semantic characteristics of the objects.

Firstly, this method was applied to the forestry domain by implementing general purpose modules for sensor data processing, semantic world modelling and a general, semantic landmark-based localization algorithm. After verifying the accuracy of the implemented algorithms

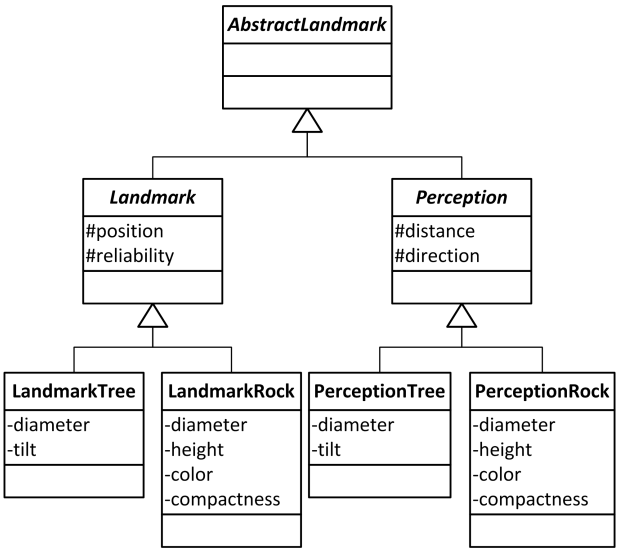

Figure 2. Example for the landmark class hierarchy in UML

in virtual testbeds using 3d-simulation techniques, the approach is in daily use on a prototype harvester in the forest enterprise of North Rhine-Westphalia, Germany. Secondly, the general purpose software modules were reused in an exploration scenario of a mobile robot. Only scenario specific modules like the semantic landmark detector for planetary surfaces had to be adopted. As all improvements on the general modules effect both scenario adaptations similarly, the development process has become economically efficient. Furthermore, this concept allows an adaptation to other terrestrial applications with little effort.

The adaptation to forest environments for example is straight forward, as trees constitute easily detectable landmarks. A distinct position can be assigned to them, they are easily to detect with laser scanners and stereo cameras [19] and they appear often in the forest. In planetary exploration missions rocks can be used as landmarks. By setting limits in parameters like height, diameter, compactness and color they can be distinguished from hills and cliffs. An arbitrary amount of landmark types can be used simultaneously for localization, which is beneficial in scenarios where one class of landmark is not present all the time.

For the detection of individual landmark types, an appropriate detector has to be implemented for each sensor class, for example for laser scanners and stereo cameras. When using one sensor in two different ways, for example a camera on a driving platform and one on a flying drone, the implementation of two detectors is also reasonable, as the appearance of objects often heavily change with a changing perspective. In [18] the detection of appropriate landmarks during the landing phase of a space probe equipped with a camera is introduced. For the detection of landmarks during a subsequent exploration mission on the surface we present a rock detector for stereo cameras at this point (see figure 3).

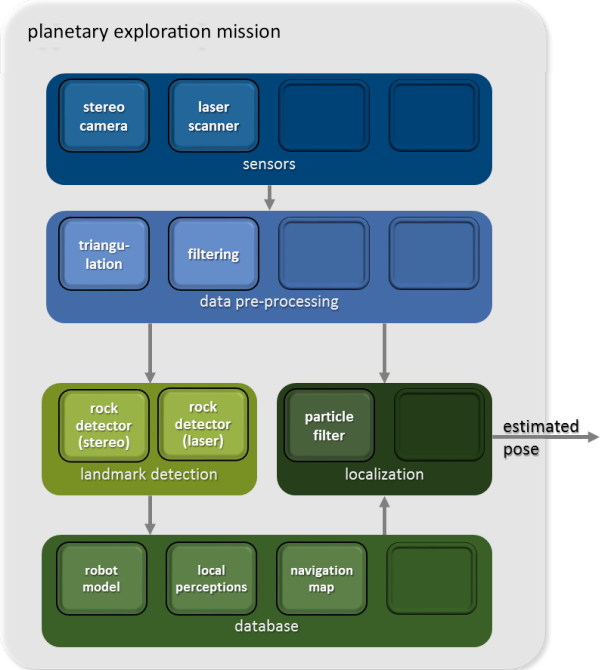

Figure 3. Flow-chart of the exemplary adaptation to a planetary exploration scenario.

2.3 Rock detection on stereo images

The presented approach is based on the idea of [20]. Instead of searching for rocks in the incoming stereo depth map or point cloud directly, some preprocessing is done to transform the distance measures of the stereo data into a height map according to a floor plane. With this transformation we can use fast image processing techniques for finding connected structures. In a first implementation we used a method like [21] to find rock-like structures in the 3d point cloud directly, but it was not suitable for real-time applications. The actual procedure of the rock detector can be divided into three steps:

1. Detect floor plane and transform depth map into height map

2. Segment the height map into a fixed number of height classes

3. Cluster adjacent height regions to rocks and determine attributes

For floor plane extraction we use the iterative least-squares-approach proposed in [20]. The floor plane is estimated by a principal component analysis (PCA) of the stereo depth information. By iteratively removing outliers according to their standard deviation and reestimating the plane a best fitting floor plane is computed. Afterwards the distance coded 3d information can be transformed into a height map according to the detected floor plane.

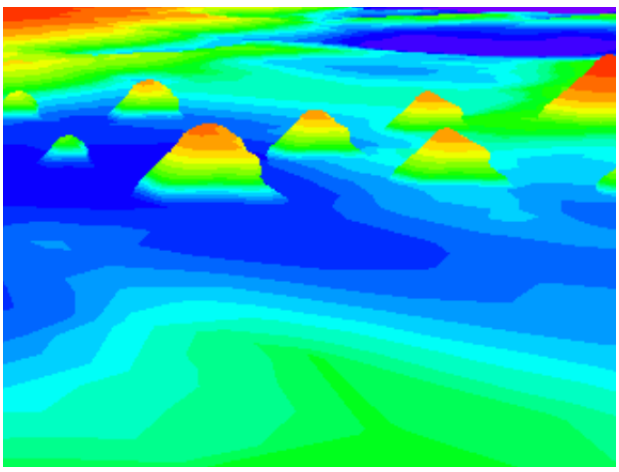

Figure 4. Result of the kMeans-segmentation on a height map after sorting the classes.

For segmenting the resulting map into a fixed number of height classes we use kMeans clustering. Gor et al. propose a number of 21 classes. From this point we will go on differently to [20] and firstly sort the classes according to their corresponding key values resulting in a map as in Figure 4. Beginning on the highest height class, we are using an asymmetric floodfill algorithm for descending the classes from the top of the rock to the floor following this idea: The top of a rock has no neighbor with a higher class and exactly one neighbor of one height class below. Inside a rock each region has one direct neighbor above and below differing in one class step. Based on this idea, we can traverse the height classes from the top down to the point reaching a class which has more than one neighbor above. We use an absolute-difference-based floodfill algorithm allowing to increasingly add only one new class to the current cluster respectively.

After reaching a class with more than one upper neighbor, all regions traversed so far are united to one cluster and the classes are again traversed from top to bottom, but this time we use a gradient-based floodfill algorithm collecting all regions from the top of the rock down to the lowest area of the map reachable by traversing, allowing only directly neighboring class traversals. These regions are marked as already visited and are no longer regarded in the following analysis of the height map. This procedure is repeated until all regions above a given border are examined. The border can be be parametrized freely according to the reliability of the estimated floor plane approximation.

After combining the height classes to clusters of possible rocks, the rock candidates have to be examined for their applicability for localization. Therefor, the 3d information given by the stereo data for each selected cluster is analyzed. The height and the diameter of the region can easily be determined from this data. The main attribute for the distinction between a rock and no rock is the compactness c given by the quotient between the circumference of a circle with the same area A as the candidate region in image space and the length of the contour of the region C:

$$c = \frac{2\sqrt{\pi A}}{C}.$$

According to this compactness measure possible compact rock landmarks can be distinguished from longish elevations of the ground. An appropriate border can be given by mission control, as well as a minimal and maximal height and diameter to be considered for detection during runtime.

2.4 Particle-filter-based Localization

After the surrounding landmarks have been detected and attributed they are given to the localization algorithm for pose estimation according to a preliminary given navigation map, for example generated by the techniques proposed in [18]. The used localization algorithm is basically a particle filter as introduced in [22], adjusting the positions of the locally observed perceptions with the positions of the landmarks in the navigation map. An adaptation of this algorithm to a scenario-independent localization concept was firstly proposed in [15]. The algorithm evaluates randomly distributed pose hypotheses by comparing the position, type and characteristics between the perceptions and the landmarks of the navigation map. If the initial position of the exploring robot is known via GNSS in the forest or a landing site on a planetary exploration mission, the radius of the particles that are distributed from the algorithm can be reduced for quicker and more reliable initialization.

The estimated pose is given in the internally used data format for further processing or can be emitted in a navigational standard format like NMEA 0183 defined by the National Marine Electronics Association and used in GNSS-devices for giving the current position to the processing software. This allows to use the pose information of the proposed approach in existent navigational software without further adaptations. In addition and much more important, our approach allows the usage of existing GNSS software solutions in areas where no or not

sufficient GNSS-reception is available like in indoor environments, in mining facilities under ground and in exploration missions on planets other than earth.

3 Localization Unit

Based on the application-independent localization approach previously introduced, a prototype of a self-contained localization unit was designed and manufactured. It is intended for use in outdoor and harsh environments and can, among other carrier systems, be mounted on mobile robots, cars or work machines. In order to obtain a housing as sturdy as possible, but to simultaneously ensure the heat transfer of the individual components and to keep the weight as low as possible, the entire housing is made of aluminum. The unit itself in its current configuration weighs 8.7 kg and has a size of 358 x 314 x 121mm (HWD). On carrier systems without power supply, an additional external battery is in use which has a weight of 3.6 kg. It currently includes an industrial laser scanner and a stereo camera as the primary sensors. In addition, an inertial measurement unit (IMU) with three orthogonally disposed acceleration sensors and three gyroscopes is installed in the housing of localization unit. The sensor data processing and localization are performed on a passively cooled industrial PC, which is mounted in the same enclosure and passes its heat to the housing via heat pipes.

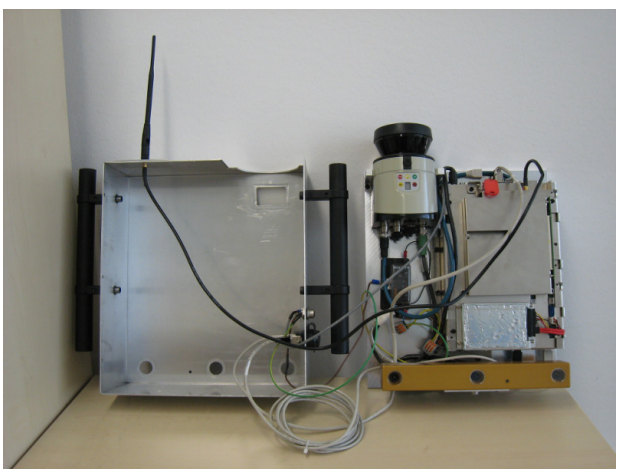

Figure 5. Opened self-contained localization unit with laser scanner, stereo camera and industrial PC.

The self-contained localization unit with passively cooled industrial PC, laser scanner, stereo camera, IMU and additional wiring for power supply and connection of external components is shown in figure 5.
For mounting on vehicles, a system was chosen, which simultaneously can serve as handles for a human carrying the localization unit (see figure 6). Furthermore, the location unit has external interfaces to connect additional sensors which can be used directly by the localization framework.
The self-contained localization unit can be operated via a protocol provided by the simulation system VEROSIM – "Virtual Environments and RObot SIMulation" [14]. The sensor data as well as the results of the data processing algorithms can be recorded at any point in the processing chain in order to perform a later analysis. For data transmission, a wired or wireless communication can be used. Existing applications can easily connect to the localization unit and make use of the position information provided in standard protocols such as the well-known NMEA 0183 format. Thus, the localization unit can provide position data to applications such as navigation software, which originally can only be used in conjunction with a GPS receiver.
The presented prototype is the first completely application-independent implementation of the developed localization concept. Additional external sensors can be connected to the localization unit and are addressed directly by the localization framework. Furthermore, the landmark detectors can be reconfigured and replaced during operation as well. This reconfigurability allows the use of the self-contained localization unit in other areas of application.
Moreover, by using semantic landmarks as introduced in 2.2 instead of features in the sensor data, environment models can be generated which include additional attributes of the detected objects. These may be the basis for many other applications based on semantic information.

Figure 6. Self-contained localization unit mounted on mobile robot.

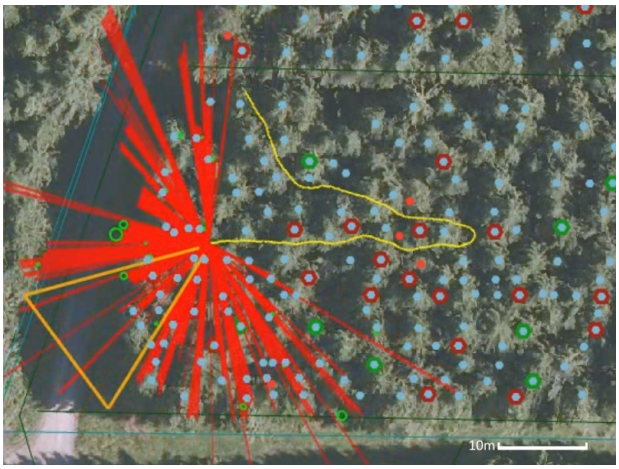

Figure 7. Result of the localization algorithm during one test run in the forest scenario. The laser scanner data is visualized as red rays and the stereo cameras field-of-view as a yellow triangle.

Figure 8. Result of the localization algorithm during one test run in a planetary exploration scenario in a stone quarry.

4 Testing and Results

The results of the localization components have been tested in simulation during development and implementation using virtual testbeds as described in [23]. The localization results in the virtual testbed were compared to ground truth known from the simulation system. Thus, it is possible to determine the localization accuracy accurately.

We tested our algorithms in both scenarios, forest and planetary exploration. In the forest the localization accuracy lies within a mean translational error of 0.1 meters. In the scenario planetary exploration the mean translational error lies within the dimension of 0.2 meters. The implemented algorithms were also tested in physical testbeds. Here, the ground-truth is more difficult to achieve. In the forest, a surveyor team was hired to measure the position of the robot from calibrated measuring points. Therefor, the robot was equipped with two well visible markers. The tree positions were also manually measured by the surveyors and recorded in the navigation map. During the test runs the robot was stopped in frequent intervals and the surveyors measured the accurate position as a reference for our localization analysis. The mean accuracy in translation has been measured under 0.1 meters. Figure 7 shows the localization result underlaid with remote sensing data and the landmark map of the area. In comparison, GNSS receivers provide accuracies in the range of 1 to 15 meters under the canopy of trees in the forest. The real world planetary exploration tests have been carried out in a stone quarry. To verify the results of the localization as exact as possible, the rock landmarks were stored in the navigation map manually. The terrain was pictured and geo-tagged by a remote controlled aircraft and in addition 3d laser scans were made to use the data inside the simulation system for further analyses. The start and end position of the robot was determined by measuring the distance of a marker on the robot from two measuring points. The robots movement was also recorded with a camera for identifying approximate deviations between localization and actual movement. Figure 8 shows the localization result of one of the test runs through the test area. The start position is marked by a green and the end position by a red circle. The localization error at start and end position averages out at 0.25 meters.

5 Conclusion and Outlook

In this paper, a universal scenario-independent localization concept and the implementation in a self-contained localization unit has been introduced. The localization unit itself can be mounted on various mobile carrier systems as well as carried by human.

All in all, we were able to estimate the position of our localization unit within a precision of 0.5 meters. The device was attached to a mobile robot during most of the test runs. There was no recognizable difference in precision or reliability between test runs, where the localization unit was carried by foot (see figure 9) or attached to the robot. All data has been recorded, so that further improvements in the algorithms can be compared directly to the current results.

Currently, we are working on adaptations to domains, where GNSS-services is not available like indoor applications or underground mining facilities. Our next goal is

Figure 9. Self-contained localization unit carried by humans in a forest.

to use the presented localization concept in the domain of automated guided vehicle systems (AGV) for warehouse logistics. This adaptation can be achieved with little effort. Here, artificial markers like qr-tags are used as landmarks. These tags can also carry additional information, so that the reliability and accuracy of the localization can be increased. We are planning to use an omnidirectional camera as primary sensor, because its field-of-view covers the whole surrounding area and allows a maximum number of landmarks. A first prototype in a virtual testbed shows promising results (see figure 10)

Furthermore, a lighter and cheaper localization unit is planned to be built. By simply using components for mobile devices instead of industrial ones, a weight reduction of 60-80% is realistic. As these components are assembled in great numbers the price is also significantly lower than the components used in the current prototype. It is planned to introduce the lightweight localization unit in the forestry domain for computer aided stand inventory.

Acknowledgments

Parts of this work were developed in the context of the research projects Virtual Forest and SELOK.
Virtual Forest: This project is co-financed by the European Union and the federal state of North Rhine-Westphalia, European Regional Development Fund (ERDF). Europe - Investing in our future.
SELOK: Funded by the German Aerospace Center (DLR) with funds provided by the Federal Ministry of Economics and Technology (BMWi) under grant number 50 RA 0911.

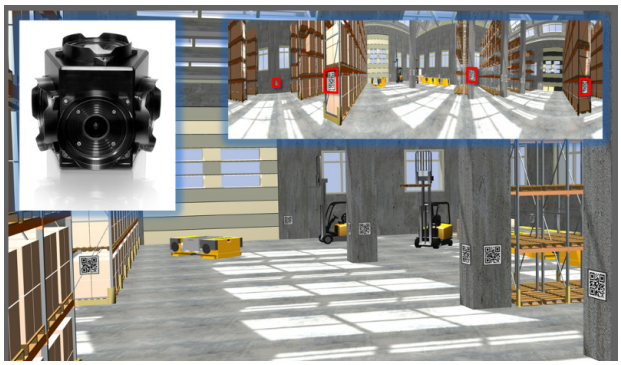

Figure 10. Virtual testbed for automated guided vehicle systems (AGV) in a warehouse logistics scenario.

References

[1] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[2] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[3] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, Aug. 2002.

[4] M. Lingenauber, "Shape Context Matching for the Self-Localization of Planetary Rover," *ASTRA*, 2011.

[5] S. Frintrop and P. Jensfelt, "Attentional Landmarks and Active Gaze Control for Visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1054–1065, Oct. 2008.

[6] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, and R. Schmidt, "Driving environment perception using stereovision," *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pp. 331–336, 2005.

[7] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6d-vision: Fusion of stereo and motion for robust environment perception," *Pattern Recognition*, pp. 216–223, 2005.

[8] J. Wenger, "Automotive radar - status and perspectives," in *IEEE Compound Semiconductor Integrated*

Circuit Symposium, 2005. CSIC '05. IEEE, 2005, p. 4 pp.

[9] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic," in *Programming, Proc. International Conf. Computer Vision, Workshop Dynamical Vision*, 2007.

[10] R. Danescu, F. Oniga, and S. Nedevschi, "Particle grid tracking system for stereovision based environment perception," *2010 IEEE Intelligent Vehicles Symposium*, pp. 987–992, Jun. 2010.

[11] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," *ArXiv e-prints*, 2013.

[12] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Simultaneous localization and map building: a global topological model with local metric maps," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1. IEEE, 2001, pp. 421–426.

[13] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[14] J. Rossmann, M. Schluse, C. Schlette, and R. Waspe, "Control by 3D Simulation - A New eRobotics Approach to Control Design in Automation," in *Proceedings of the 5th International Conference on Intelligent Robotics and Applications (ICIRA 2012)", October 3-5, 2012, Montreal, Quebec, Canada*, ser. LNAI 7507, C.-Y. Su, S. Rakheja, and H. Liu, Eds., vol. Part II, 2012, pp. 186–197.

[15] J. Rossmann, C. Schlette, M. Emde, and B. Sondermann, "Discussion of a Self-Localization and Navigation Unit for Mobile Robots in Extraterrestrial Environments," in *The 10th International Symposium on Artificial Intelligence,Robotics and Automation in Space (i-SAIRAS 2010)*, 2010, pp. 46–53.

[16] M. Emde, J. Rossmann, B. Sondermann, and N. Hempe, "Advanced Sensor Simulation In Virtual Testbeds : A Cost-Efficient Way To Develop And Verify Space Applications," in *AIAA Space 2011 American Institute of Aeronautics and Astronautics (AIAA) Confrence and Exposition*, 2011, pp. 1–11.

[17] M. Emde and J. Rossmann, "Validating a simulation of a single ray based laser scanner used in mobile robot applications," in *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE, Oct. 2013, pp. 55–60.

[18] J. Rossmann, G. Jochmann, and F. Bluemel, "Semantic Navigation Maps for Mobile Robot Localization on Planetary Surfaces," in *12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2013), Session 9 Navigation & Localisation II, 15 - 17 May*, ESA / ESTEC. ESA, 2013, pp. 1–8.

[19] J. Rossmann, B. Sondermann, and M. Emde, "Virtual Testbeds for Planetary Exploration: The Self-Localization Aspect," in *11th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2011)*, 2011.

[20] V. Gor, R. Castano, and R. Manduchi, "Autonomous rock detection for Mars terrain," *Space*, pp. 1–14, 2001.

[21] R. Manduchi, a. Castano, a. Talukder, and L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," *Autonomous Robots*, vol. 18, no. 1, pp. 81–102, Jan. 2005.

[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics. 2005*. MIT Press, 2005.

[23] B. Sondermann, M. Emde, M. Rast, and J. Rossmann, "Simulation-based Engineering with Hybrid Testbeds," in *David Al-Dabass, Alessandra Orsoni, Zheng Xie (Eds.): Proceedings of the UKSim-AMSS 7th European Modelling Symposium on Computer Modelling and Simulation (EMS 2013)*, 2013, pp. 442–447.