# VIRTUAL TESTBEDS FOR PLANETARY EXPLORATION: THE SELF-LOCALIZATION ASPECT

**Juergen Rossmann, Bjoern Sondermann and Markus Emde**

*Institute for Man-Machine Interaction, RWTH Aachen University*
*Ahornstr. 55, D-50274 Aachen, Germany*
*{rossmann, sondermann, emde}@mmi.rwth-aachen.de*

## ABSTRACT

Mobile robots are recognized as being essential for the exploration of planets in our solar system. In this paper we introduce a new approach for a self-localization and navigation unit for mobile robots in extraterrestrial environments supported by virtual testbeds. In addition to an absolute self-localization strategy within a global navigation map a relative localization component is attached based on a modular sensor framework. This testbed framework allows for the integration of real and simulated sensors in virtual and real testbeds for a smooth transition between simulation and real world tests. Introducing virtual sensors, algorithmic results can be treated as ordinary sensor information and therefore seamlessly addressed by the sensor framework.

Key words: self-localization; virtual testbeds; sensor simulation; simulation frameworks.

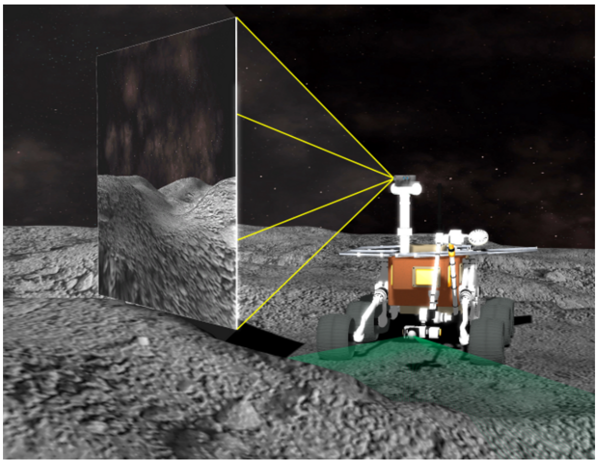

*Figure 1. Simulated mobile robot with attached laser scanner and stereo camera in virtual testbed for planetary exploration. The sensor data is visualized using comprehensive metaphors.*

## 1. INTRODUCTION

Our previous work focused on the development of new methods for highly accurate self-localization and navigation of work machines in the forest. Fusing various sensors and implementing localization algorithms using Sequential Monte Carlo methods (SMC) led to a robust localization unit appropriate for unstructured environments. Mounted on a forest machine, this system allows for the detection and localization of objects in the surrounding area, as well as the self-localization of the vehicle and its well-directed navigation.

We generalized and extended the underlying concepts from forestry to other e.g. extraterrestrial environments. The resulting localization unit is able to handle different classes of landmarks and sensors. In addition to the terrestrial approach we extend our laser scanner based localization unit with stereo cameras and a visual odometry component. The results of this relative localization approach are treated as ordinary sensor data to initialize and enhance the estimation of the absolute positioning algorithms.

The modular sensor framework is used as a generic sensor interface. Besides the connection of real and simulated sensors and their corresponding error models it is capable of providing algorithmic results as ordinary sensor data, called virtual sensors.

The integration of the developed components in our testbed concept allows for evaluating algorithms and sensors in virtual, real and hybrid testbeds. Virtual testbeds are a cost-efficient alternative e.g. for planetary exploration missions, especially if setting up real mockups for testing and verifying is too expensive. Our virtual testbed implements a 3d geometric and functional simulation model incorporating the mobile robot, sensors and the environment e.g. of the planet (figure 1). In addition the simulation system allows for testing and verifying the localization, navigation and control algorithms. Furthermore it provides the possibility to rerun test series with slightly different parameters to find the best parameter sets.

The results lead directly into the project ”SELOK” (self-localization on planetary surfaces - ”Selbstlokalisation auf planetaren Oberflächen”) funded by the German Aerospace Agency (DLR).

*Figure 2. Virtual and real testbeds; left: Smulated forest machine with laser scanner visualization; right: photo of the real harvester with mounted laser scanners.*

## 2. LOCALIZATION STRATEGY

Exploration of unknown planetary surfaces with the help of mobile robots will be essential for the success of future space missions, as it is possible at lower cost and less dangerous. Due to operation in remote areas and gaps in the communication, these mobile robots need robust self-localization strategies. Starting with a self-localization approach optimized and validated for unstructured environment on earth, we introduce a new generalized modular concept applicable in different environments.

### 2.1. Implemented localization strategy for terrestrial unstructured environments

Our previous work focused on the development of new methods for precise self-localization of work machines in unstructured forestry environments. In this domain the Global Positioning System (GPS) has been introduced and is utilized as the standard location service but it suffers from low position accuracy or even signal loss.
Therefore a new approach to determine the position of a work machine has been implemented. As described in [1] a forest machine has been equipped with several sensors to measure aspects of its environment and is able to determine its position with higher accuracy than with a GPS receiver as its only sensor.
The foundation of this approach is the fusioning of the sensor data with information of landmarks calculated beforehand from aerial survey data. Figure 3 illustrates the concept of this approach. In a first step, a local landmark map is generated using the point cloud data of the mounted laser scanners. Implemented object extraction algorithms determine relevant features in this sensor data. According to the position and orientation of the laser scanners and the parametrization of the object extraction algorithms detected features are treated as trees. Using the collected information a local landmark map of trees can be generated.
Next the matching algorithm is started. This calculation is based on a particle filter algorithm as described in [2].

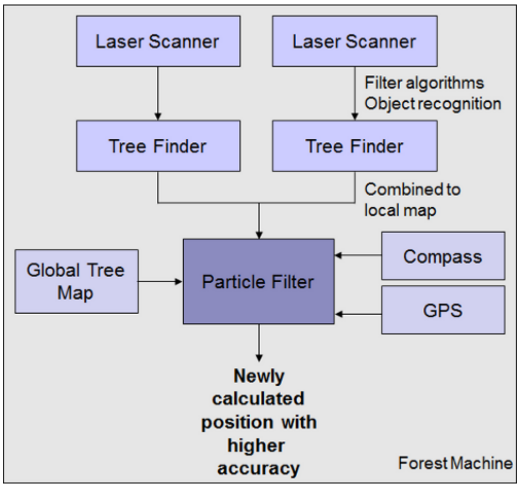


*Figure 3. Implemented localization approach for forestry environments.*

Each hypothesis of a possibly true world state is represented as a single particle with an importance weight. Generated particles are distributed uniformly on an area of interest which is assumed to be the target area.
Landmarks are extracted from the global tree map and are used to recalculate the importance weight of the particles in each sampling step. In the following step the evaluation threshold for the particles is increased. Particles with low importance weight are eliminated and new ones are generated positioned around the remaining ones. This leads to an accumulation of particles at the best position estimations. The resampling steps are repeated until the particles accumulate in one point. In [3] the calculation steps are described in detail as well as information on parametrization of the algorithm.
Using additional information sources like the mounted low-cost GPS receiver and an electronic compass an imprecise position estimation as a starting point as well as an estimated orientation are given and the algorithm can be sped up.

### 2.2. Generalization

The aforementioned system has been tested, evaluated and optimized in forest stands over the last two and a half years. The accuracy of the results has been verified by a surveyor's office. In chapter 4 the results are presented.
Evaluating and optimizing the system in these iteration steps and being faced with different problems we came to the conclusion to re-engineer the approach. At first glance the system is highly optimized to the needs of the forestry environment. It is fast and reliable but at the same time inflexible and non-modular.
Abstracting the components of the localization approach and generalizing the communication among them led to the new modular and flexible concept as illustrated in figure 4. The foundation of this implementation is described

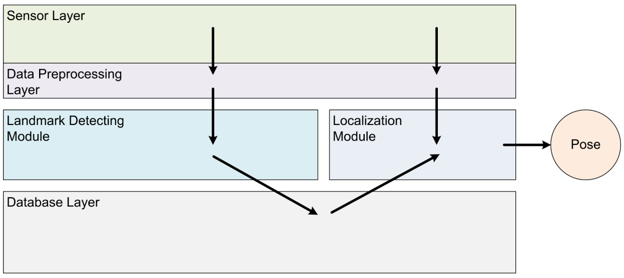


*Figure 4. Abstact concept generalizing the introduced forestry based localization approach.*

in detail in chapter 3. The self-localization approach for forest environments has been broken down into modules according to the new concept. The modules have been implemented context-independent, so that they can be continued to use in different application areas. Furthermore, modules of the same kind, e.g. landmark detectors, can be substituted without effecting the other modules of the processing chain.

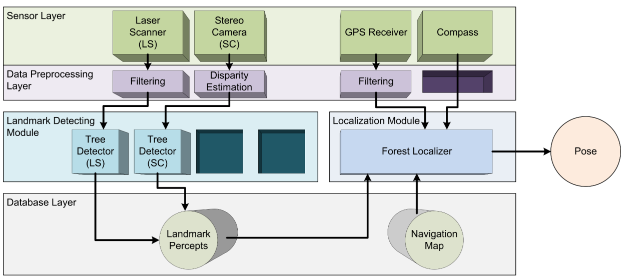


*Figure 5. New modular VisualGPS approach with a configuration for forest environments.*

The adaption of the self-localization approach for forest environments to the new general concept is shown in figure 5. The general concept can now be used for further localization tasks, such as extraterrestrial exploration missions, or localization in urban environments.

### 2.3. Adaption to extraterrestrial environment

By the abstraction and modular reassembly of the localization unit, attaching new sensors to it is now a straight forward task. Furthermore the modular landmark detector unit now allows for many different detection instances as tree detectors in forest environments or for example rock detectors in extraterrestrial environments. Each detector instance defines the required sensors. For example, the tree detector used so far based on 2d laser scanner data, deliver sufficient data and are rugged enough for the use on forest machines. An alternative implementation for tree detection uses stereo cameras, as they are more cost-saving and lighter compared to laser scanners. Each of the detectors extracts a set of tree landmarks from the sensor data and provide the localizer with it.
The localization module itself defines the set of landmark types it is using, so is the forest environment localizer just using tree landmarks independently in which detection instance they were generated. In extraterrestrial environments many different types of landmarks are usable, for example rocks, craters, hills and so on. Therefore the localization module has to evaluate the landmarks semantic information as well as their position to match them correctly to the navigation map, wherein the landmarks are divided in the same semantic classes.
The navigation map for both forest environments and planetary surfaces are generated from remote sensing data. In forest scenarios the data comes from aerial surveys made for this purpose and earth observing satellites. On extraterrestrial exploration missions the data is acquired during the landing process and from observing satellites as well. The automated surface reconstruction and mapping issue using landing data from planetary exploration missions is subject of the DLR funded research project *FastMap*, intruduced in section 1. Figure 6 demonstrates a configuration of modules which could be used for a planetary exploration mission.

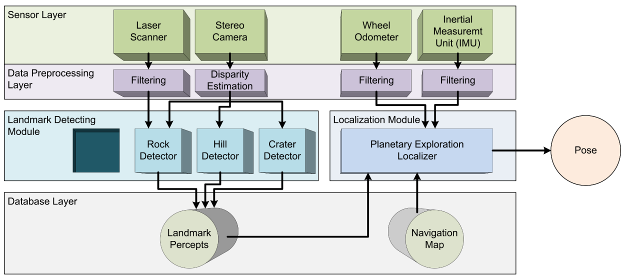


*Figure 6. VisualGPS approach with a module configuration for extraterrestrial exploration missions.*

### 2.4. Landmark detection on stereo image data

The detection of landmarks using stereo cameras has been subject of publications like [4] and [5]. In most of them, the object detection is carried out on the input images directly. In indoor application this is a promis-

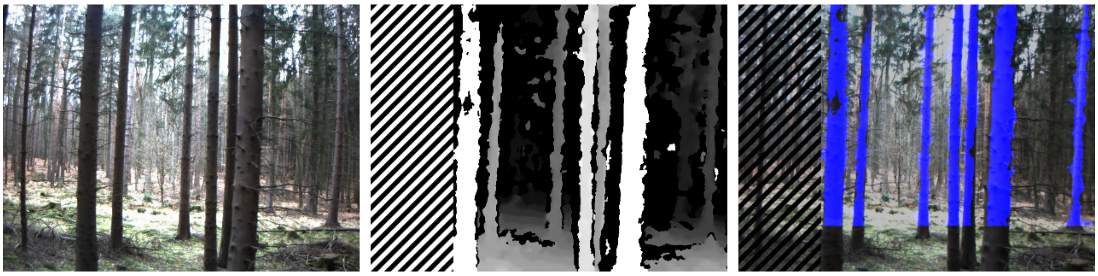

*Figure 7. Calculation steps for tree detection in disparity images. Left: Input image (left view of stereo image); Middle: Disparity image (shaded area is not visible in both stereo images); Right: Detected trees as blue overlay on left input image.*

ing approach, as edges and corners can be used to identify object borders. Tree edges also can be detected by analyzing the horizontal image gradients, but branches or consecutive standing trees in a dense forest make the problem much more challenging. A better approach to solve the problem of landmark detection is to first compute a dense disparity map using one of the methods analyzed in [6]. We use an implementation of the semi global block matching method as introduced in [7]. The resulting disparity map is a discrete depth map color coded in a grayscale image. Object faces pointing directly to the camera result in single-colored areas in the disparity map and can easily be extracted from the background.
Figure 7 shows a sequence of three images representing the three steps image aquisition, disparity map estimation and tree detection. The tree detection algorithm selects all single-colored regions in the disparity map and evaluates their minimal bounding boxes. If the aspect ratio of the bounding box is under a given threshold, the region is marked as a tree.
A more challenging landmark type are rocks, e.g. on extraterrestrial planetary surfaces. A rock detector has to consider more aspects than the depth values from laser scanner or stereo camera as the geometry of rocks is arbitrary. A combination of obstacle detection based on disparity maps as well as analyses of color and texture information on raw image data is more promising, than relying on just one aspect.
In our approach for a rock detecting algorithm, the disparity map of stereo input images is estimated first. Obstacles as rocks or steep slopes can be distinguished from the surface plane in the disparity map, as on the one hand the depth gradient of the objects differs from the one of the plane they are arranged on. On the other hand obstacle edges as they are visible in one of the stereo cameras are partially occluded in the other camera and vice versa, which results in uncertain regions in the disparity map around the obstacle boarders. Regions that cannot be assigned to a disparity class are denoted black in the disparity map. Using the sobel edge detector on the disparity image finally results in clear identifiable lines along distance jumps belonging to obstacle boarders, cf. figure 8. In a last step the dominant edges are extracted and ordered according to their depth in the disparity map. If there is no disparity value known, which will occur quiet often as the dominant edges develop from gaps in the

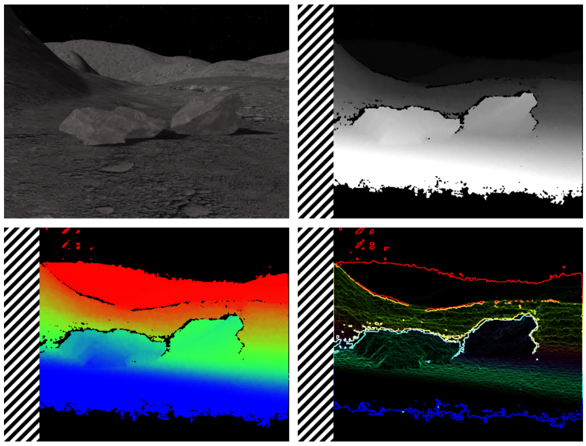

*Figure 8. Obstacle boarder detection using stereo vision. Top-left: input image from virtual testbed (left view of stereo pair); top-right: disparity map from stereo input; bottom-left: disparity map stretched to RGB-color space for a better visualization; bottom-right: obstacle boarders identifiable after applying the sobel operator on the colored disparity map.*

disparity map, the maximum disparity value in a block of surrounding pixels will be taken. This is because the edges separate a foreground object with higher disparity values from the background, and we assign the edges to the object. In a last step the objects can be generated by fitting ellipsoids into the depth-sorted contours. The median depth of the pixels inside the ellipsoid will be taken as distance to the landmark object.

### 2.5. Relative localization approach based on visual odometry

The localization unit for forest environments uses compass and GPS data for initializing the localization algorithm and to reduce the search space. Furthermore, the additional data can be used to detect incoherent pose jumps between GPS/compass data and the estimated pose from Visual GPS. In extraterrestrial environments compasses and GPS receivers are worthless, thus an alternative initialization and correcting method is

needed. Stereo cameras are already used to observe the environment, so the image data can also be used for visual odometry, as it has been already used to assist in resolving wheel odometry problems on mars exploration rover missions [8] and other terrestrial in- and outdoor localization tasks [9], [10], [11].
Visual Odometry uses optical flow between consecutive images to calculate object movement relative to the camera. In our case the camera will move in its environment, that is why the optical flow specifies the inverse camera movement in projected image space. By using rectified stereo images of a calibrated stereo camera system with known baseline distance of left and right camera, the estimated optical flow yields a metric three dimensional camera movement. The accuracy depends on the horizontal resolution and the focal length of the stereo cameras used.
With the additional visual odometry technique the localization unit works in two modes. In the absolute localization mode, the Visual GPS algorithm tries to find enough landmarks to estimate an absolute pose in the localization map. When the number of detected landmarks is not sufficient, the unit switches into the relative localization mode. In this mode the localization algorithm estimates the relative movement of the robot from the last known absolute position using visual odometry. During the relative localization the landmark detector still works in a background process in order to detect further useful landmark configurations.

## 3. THE VIRTUAL TESTBED

As mentioned in chapter 2.2 the implemented approach had to be re-engineered to meet the requirements of new domains. In addition, connections among the implemented components should be flexible and modular. To meet all these requirements the underlying concepts to connect components and to communicate among them have been revised. The design allows easy setups of virtual testbeds using a generic communication concept for the interaction of all components. The newly developed sensor framework allows the parallel integration of real and simulated sensors into the system and provides a smooth transition between simulation and real world tests. The virtual testbed allows a focused view on every component of the system to analyze and optimize its behavior.

### 3.1. Communication concept

In a first step the communication has been standardized to an input-output handler concept which employs an IO-board metaphor and manages all related inputs and outputs. Using an abstract base class for the envisioned types of data, it is possible to flexibly connect all kinds of components to another component in use. The input-output hander additionally informs affected IOs of changes of values. Furthermore it is possible to configure a component to actively post a modification of a value.
Based on this new standardized communication subordinated components have been implemented. To illustrate the new flexible design of the components of the localization algorithm, the sensor framework is described in detail. Other components like the landmark detection module or the localization module are realized in the same way implementing communication and data ports as needed.
Figure 9 shows an UML diagram as the conceptual view

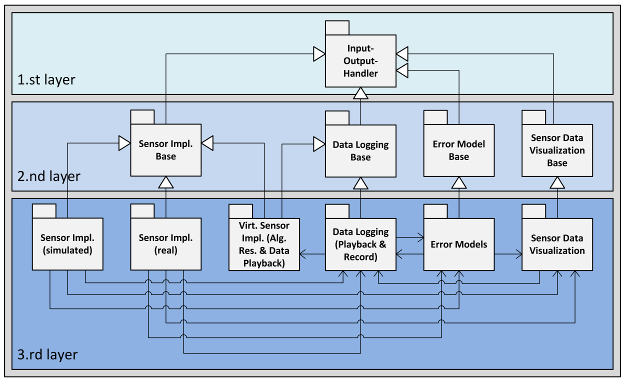


*Figure 9. UML-diagram of the structure of the sensor framework*

of the sensor framework, emphasizing the three major levels of inheritance.
The first layer represents the aforementioned communication components. Abstract implementations of sensors, error models, data logging components or the visualization inherit from the first layer and constitute the second layer of the sensor framework. Basic inputs and outputs which are indispensable to all inherited components are positioned in this layer. The "switch" to enable a component of the sensor framework is located in the abstract base class layer.
The third layer inherits the abstract realization and efficiently realizes the real, virtual and simulated sensors. Furthermore, it specifies error models especially designed to represent an error scenario of a sensor or specialized sensor data visualization for different use cases.

### 3.2. Differentiation of sensor types

The third layer of the framework differentiates among three types of sensors. Sensors are classified as implementation of real hardware API, simulated sensors and virtual sensors. This is necessary to consider the characteristics of each type with its unique features.
Implementation of APIs of real sensors are in use to connect to real hardware components. Implementing interfaces to real hardware allows to control the components and to alter parameters at runtime. Using this feature is indispensible to algorithms or applications e.g. if they need different resolutions of sensor data to calculte correct results.

The second class represents the simulated sensors. As real hardware components are not available at all times, for instance due to high costs, it is not possible to use them and for example carry out necessary test series. Using simulated sensors in an appropriate testbed avoids this effect. By combining simulated sensors, which yield ideal data, with error models (cf. 3.3) the behavior of real components are emulated, providing realistic sensor data for subsequent algorithms.

Virtual sensors represent the last class of sensors in our framework. They are used to induct algorithmic results or recorded data into the network of connected components. As the data ports use an abstract base class for the envisioned types of data as mentioned beforehand recorded data or algorithmic results are treated as conventional sensor data. Regarding the introduced localization approach in this paper the result of the particle filter 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 this component which we call VisualGPS-sensor as it provides absolute position information like a GPS-receiver but its position estimation is based on optical sensor data.

### 3.3. Error Modelling

Besides the ability to simulate sensors it is indispensible to add error models to a virtual testbed. As described above simulated sensors yield ideal data. In reality the output of a sensor is based upon several criteria. Focusing on the sensors of the VisualGPS algorithm, e.g. the implementation of a laser scanner, this means the sensor is able to determine the distance to objects in the virtual testbed without considering maximum scanning distances, reflection properties of surfaces or typical error characteristics like random noise. Comparing the output of this simulated sensor to a real hardware component shows obvious deviations.

The discrepancy between the output of the real hardware component and the simulated one is flattened by adapting characteristics of real hardware components like biased and statistical error. The simulation system in use provides an graphical user interface to compose a sensor network base on input and output data ports. In addition standard error models for depth noise, dirt, reflection characteristics context-dependent errors (e.g. based on detected colors) have been implemented to optimize the behavior of the simulated laser scanner and to emulate real hardware components in a realistic way. Additionally filters have been implemented to limit the maximum as well as minimum depth and to smooth features. Figure 10 shows a network with a simulated laser scanner sensor, error models for depth limitations, depth noise and standard image processing filters. Furthermore visualizations for optimal sensor values as well as sensor values with included error models are added to the network. The results are visualized in figure 11.

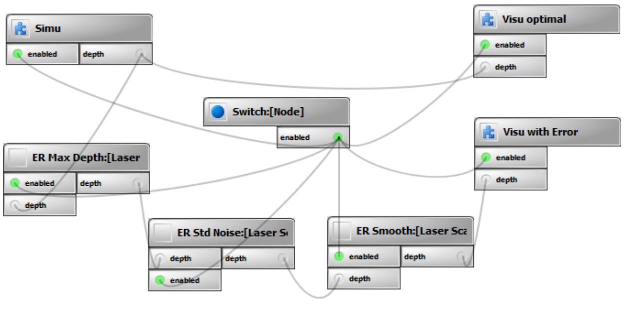


*Figure 10. Graphical user interface to model sensor networks.*

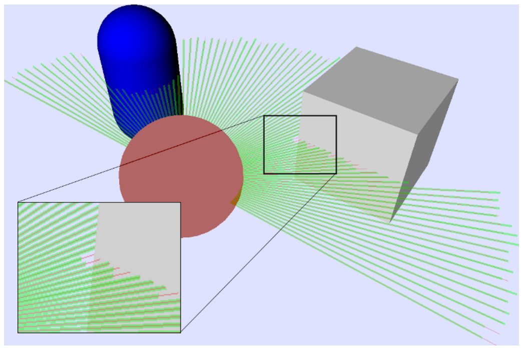

*Figure 11. Abstracted visualization of laser scanner data. Red sphere represents the laser scanner; red lines visualize optimal scanner results; green lines represent results with error models.*

### 3.4. Data logging and playback

Reproducibility is an essential feature to analyze and understand complex processes and to verify results. Simulation systems take advantage of their ability to record and playback tasks and to analyze information in detail independent from temporal restrictions.

In combination with the aforementioned virtual sensors the data logging and playback module is used to reproduce test series and to analyze test data in detail. Logging sensor data outputs, parameters of error models like random seeds or results of algorithms allows verification and optimization of implemented algorithms like the VisualGPS approach.

### 3.5. Sensor fusion

Whenever there are different sensors working in one system, the question of timing is essential, because every sensor has its own test frequency and algorithms using more then one sensor at a time need to be informed, when the relevant sensor data has changed. It would be inefficient if the sensors would be synchronized to the slowest data rate on the one hand, as data packages of faster sensors would be ignored, and on the other hand it would be a waste of processing time to synchronize

the sensors to the fastest sensor clock, as the algorithms using slow sensors had to determine whether the data had changed from one cycle to another.
In our sensor framework all sensors are attached in a non-polling (*pushing*) manner, i.e. whenever there is a new frame of sensor data it is written into a buffer from which the consuming instances can grab it when in need. Every connected instance, this can be a preprocessing mechanism or error model as well as any algorithm depending on the sensor data, will be informed via its connection on the IO-Board. Not until the data is actually needed it will be passed through to the requesting instance. Thereby unnecessary data transfer can be avoided.

### 3.6. Testing

The ability of parallel testing in real and simulated environments and the smooth transition between simulation and real world measurements is the main aspect of the virtual testbed presented in chapter 3. Modules can be developed in a pure virtual environment and after an iterative calibration process with real components, they are validated and can be used in real environments as well. Therefore the implemented modules have to be developed and tested context-independent, resulting in all-purpose modules.
The first implementation of the localization algorithm introduced in detail in chapter 2 has already been developed and optimized using the virtual testbed approach. In general the development process is divided in the following iterative steps:

1. Using a first conceptual implementation in the testbed allows estimating the basic plausibility. All necessary sensor data is provided as ideal simulated sensor output for the developed algorithms.
2. After promising tests of the first stage, error models are added to the simulated sensor data to measure the robustness of the developed algorithms, resulting in first requirements for the applicable hardware in the target system.
3. The algorithms can now be optimized with respect to the constraints defined in the step before. The input data of the simulated components is continuously transformed from ideal to realistic data by adding further error models, according to the stated requirements. The identification of errors in the algorithms is always possible by moving individual components independently of the rest in the degree of their realism.
4. Finally, all components of the simulation are replaced by their real counterparts for testing the algorithms on the target system.

Similar to the design of high-level modules simulated sensors are developed in an iterative process as well. In a first step a basic concept of the sensor according to its sensor class (optical sensor, absolute or relative position sensor, etc.) is implemented. In a second step it will be augmented by specific properties gathered from real components. Standard error models are available as templates and can be adjusted corresponding to real hardware sensor output. Comparing the data of simulated and real components in parallel differences can be identified and minimized to a close to reality implementation.

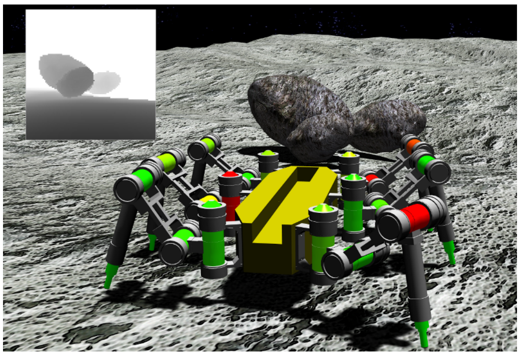

*Figure 12. Visualization of PMD-sensor data and forces at joints and engines of a mobile robot (Scarabaeus robot design by DFKI[1])*

## 4. RESULTS AND FUTURE WORK

The self-localization algorithm discussed in 2 has been tested, evaluated and optimized in forest stands over the last two and a half years using real and virtual testbeds. Generalizing the underlying concept allows for adapting the approach to new domains as e.g. unstructured environment on planetary surfaces. A Virtual Testbed for planetary exploration missions has already been set up wherein the interaction of exploration robots with their environment can be simulated as well as relevant internal processes of the mobile system [12]. All methods in the field of landmark detection in extra-terrestrial environment can already be simulated in the virtual testbed. As the scene model is well known by the system the output data of the algorithms can be verified using the scene as ground truth. Currently the landmark detection is being expanded onto new types. In the future rock-landmarks are treated as primary landmarks on planetary surfaces as introduced in chapter 2.4. The validated modules developed in the forest domain are used to produce planetary mission data in the virtual testbed, as real planetary mission data is not available.
Simulated sensors with corresponding error models and real ones are already integrated into our system. The sensor data can be visualized as comprehensive metaphors allowing easy understanding of complex correlations during development processes. Furthermore the generated sensor data can be logged and played back using the mechanisms described in chapter 3.4 allowing to rerun

[1] Deutsches Forschungsinstitut für Künstliche Intelligenz, Bremen, Germany

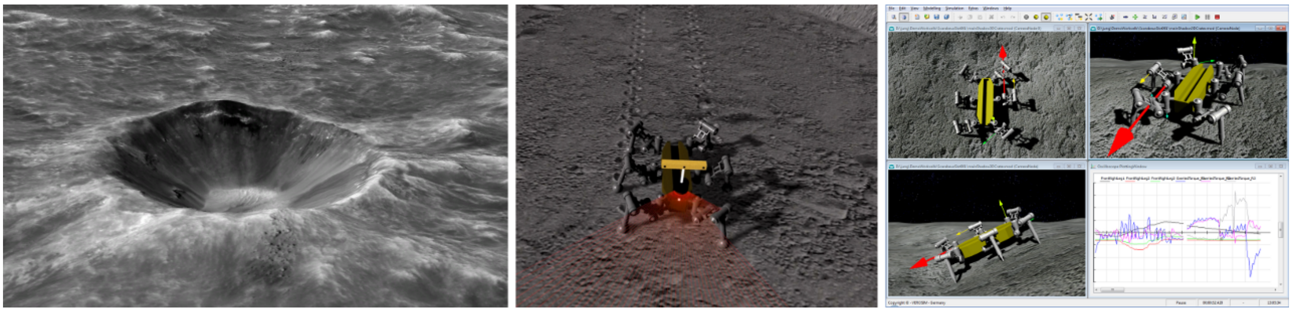

*Figure 13. Images from a virtual testbed for mobile robots in space; left: 3d planetary surface visualization based on HiRISE remote sensing data; middle: simulation of mobile robot with laser scanner interacting with its environment leaving footprints on surface; right: simulation and visualization of internal forces of mobile robot components (Scarabaeus robot design by DFKI)*

test series with slightly different parameters for optimization purposes.
Main objective of future work is a virtual mobile robotics testbed containing relevant sensors used in mobile robotics. The modular concept of the underlying system provides the possibility of developing new modules independently from specific domains or other modules of the system.

## 5. ACKNOWLEDGMENTS

Gefördert von der Raumfahrt-Agentur des Deutschen Zentrums für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter den Förderkennzeichen [50 RA 0911] für SELOK und [50 RA 1034] für FastMap.

## REFERENCES

[1] Juergen Rossmann, Michael Schluse, Christian Schlette, Arno Buecken, Petra Krahwinkler, and Markus Emde. Realization of a highly accurate mobile robot system for multi purpose precision forestry applications. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1 –6, 2009.

[2] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Cambridge: MIT Press, 2005.

[3] Juergen Rossmann, Petra Krahwinkler, and Christian Schlette. *Navigation of Mobile Robots in Natural Environments : Using Sensor Fusion in Forestry*. 2009.

[4] Matthew N. Dailey and Manukid Parnichkun. Landmark-based simultaneous localization and mapping with stereo vision. In *Proc. Asian Conference on Industrial Automation and Robotics, Bangkok, Thailand*. Citeseer, 2005.

[5] Stephen Hirtle. Landmarks for Navigation in Human and Robots. In Margaret Jefferies and Wai-Kiang Yeap, editors, *Robotics and Cognitive Approaches to Spatial Mapping*, volume 38 of *Springer Tracts in Advanced Robotics*, pages 203–214. Springer Berlin / Heidelberg, 2008.

[6] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, 47(1):131–140, 2002.

[7] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):328–41, February 2008.

[8] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 24(3):169–186, March 2007.

[9] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3946–3952, September 2008.

[10] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, January 2006.

[11] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43:215–229, February 2003.

[12] Juergen Rossmann, Thomas J. Jung, and Malte Rast. Developing virtual testbeds for mobile robotic applications in the woods and on the moon. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010*, pages 4952 – 4957, 2010.