

3D Perception and Modeling

Motion-level Teleoperation and Intelligent Autonomous Functions

Andreas Birk, Narunas Vaskevicius, Kaustubh Pathak, Soeren Schwertfeger, Jann Poppinga, and Heiko Buelow.

published in:
IEEE Robotics and Automation Magazine
(RAM), December, 2009

Autonomous robots are increasingly suited for operations in unstructured application domains. One impressive example is the result of the Defense Advanced Research Projects Agency (DARPA) Grand Challenge 2005, where five teams managed to get autonomous vehicles to complete a 213 km track in the Mojave desert. Some conditions of the Grand Challenge do not hold for Space Robotics, especially the availability of a drivable track defined by GPS positions. In general, it can be argued that Space Robotics and especially Planetary Exploration have to deal with one of the most challenging unstructured environments. There is absolutely no infrastructure, the cost of mission failure is very high, and many core sensors that form the foundations of intelligent autonomous behaviors are not space proof, yet. But autonomy is not a binary all or nothing property. Intelligent autonomous functions can supplement different levels of teleoperation. In the context of the 2008 Lunar Robotics Challenge (LRC) of the European Space Agency (ESA), the Jacobs Robotics team investigated 3D Perception and Modeling as an important basis of autonomy in unstructured domains. Concretely, the efficient modeling of the terrain via a 3D Laser Range Finder is addressed. The underlying fast extraction of planar surface patches can for example be used to improve situational awareness of an operator or for path planning.

I. INTELLIGENT AUTONOMOUS FUNCTIONS FOR PLANETARY EXPLORATION

As already pointed out in [1] in the context of Safety, Security, and Rescue Robotics (SSRR), several different stages of teleoperation can be distinguished that feature different levels of intelligent autonomous functions. For almost all possible application scenarios of mobile robots in unstructured domains including Planetary Exploration, there is a human operator in the loop issuing control commands. What potentially differs are the levels of abstraction with which the operator interacts with the robots (figure 1).

On *stage 1* or *motion level* teleoperation, there is a direct mapping between operator inputs - typically via a joystick -

and motor activations. This level already involves many non-trivial challenges that motivate the use of intelligent capabilities onboard of the robot. One obvious example for planetary exploration is advanced locomotion control [2][3][4][5][6]. But good situational awareness of the operator is also of high importance for motion level teleoperation in planetary exploration [7]. In doing so, 3D perception and representation of the environment is of high interest.

Teleoperation for space robotics suffers from many communication related restrictions. There are long time delays and links may even completely break down at particular locations or for some time period. Furthermore, there is the risk that the environment representation for the operator misses certain details that pose a risk to the robot. These are the reasons for *stage 2* teleoperation by devising systems that can follow telecommands on a *behavior level*. An example is short-term autonomous driving through a narrow passage between two rocks. But the challenges for telerobotics do not stop at level 2. Especially if multi-robots are to engage in complex cooperative tasks. Imagine for example the preparation or even set-up of a lunar base. There is the need for *stage 3* or mission level teleoperation where the operator can specify high level goals that the robots pursue in a largely autonomous fashion.

We consider 3D perception and modeling as an important basis for autonomous intelligent functionalities in unstructured environments including especially planetary exploration. This holds for all three levels of teleoperation starting with the low level control. For example, the Mars Exploration Rover (MER) missions [8][9][10][11] are an impressive demonstration of the possibilities of using mobile robots for planetary exploration. Their very advanced capabilities can mainly be categorized somewhere between stage 1 and stage 2 teleoperation with a few level 3 elements. Enabling excellent situational awareness for the operator via 3D perception and modeling is hence of high importance.

In the context of space robotics, stereo vision is the dominant approach for 3D perception and modeling [12] [13] [14] [15] [16], and also some other vision based techniques can be found [17][18]. But stereo vision has its limitations for planetary exploration, especially in planetary settings with few features [19]. Also, computer vision in general has difficulties in coping with extreme lighting conditions like the ones found in the scenario of the ESA Lunar Robotics Challenge. Furthermore, stereo has its limitations in terms of maximum range and precision. There is hence an interest in exploring alternative technologies for robotic planetary exploration like 3D Laser Range Finders (3D-LRF) [20], which as further benefits provide a large field of view and quite precise, high

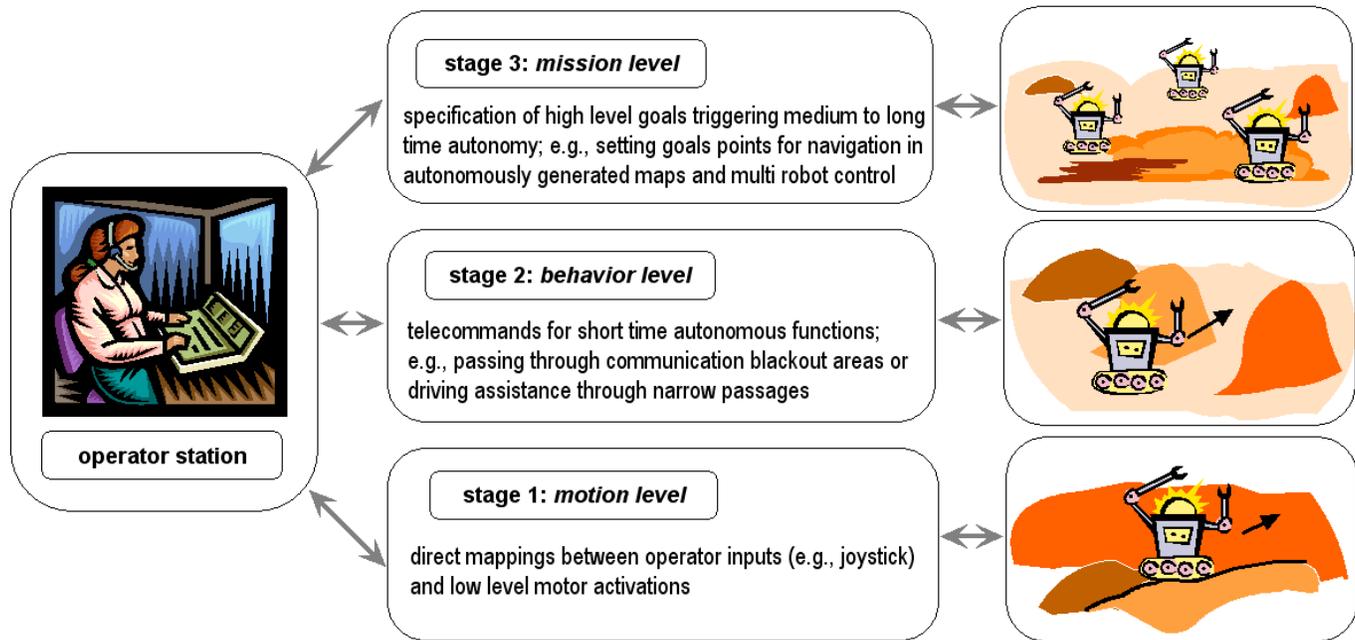


Fig. 1. Teleoperation can involve different degrees of intelligent autonomous functions onboard of the robots.

resolution data.

But what in principle is a significant advantage, namely the provision of large amounts of high resolution 3D data, is also a drawback when it comes to the processing of the data as well as the transmission of the data to the operator's station on earth. A way out of this dilemma is the representation of 3D data through efficient surface models. In the following, some of our work is presented that was also tested on data collected as part of the 2008 Lunar Robotics Challenge (LRC) of the European Space Agency (ESA).

II. THE ESA LUNAR ROBOTICS CHALLENGE



Fig. 3. The two robots of the Jacobs team during the ESA Lunar Robotics Challenge. On the left, the relay robot with a 3D Laser Range Finder to generate models of the environment. On the right, the probe robot to take soil samples in the crater. The photos are taken during the day with good visibility conditions.

The European Space Agency (ESA) introduced the Lunar Robotics Challenge (LRC) [21] in 2008 as a research oriented competition for university student teams. The motivation for the ESA LRC is the scientific interest to search for the presence of water on moon, especially at the bottom of craters in the polar regions. For this purpose, a probe robot has to get from a landing site to the crater, to descend into it, to take probes of interesting soil spots at the bottom, to climb out

of the crater, and to return back to the lander where the soil probe has to be delivered for an automated analysis. This leads to several particular challenges (figure 2). First, the crater is steep and loose soil makes locomotion challenging. Second, there is no RF-communication from the lander - that provides the communication link to the operator station on earth - into the crater, hence direct teleoperation is not possible. Third, there is very bright, horizontal illumination by sunlight on the top and the rim of the crater and absolute darkness inside of it, making teleoperation via video-streams extremely hard and vision based environment modeling impossible.

Eight teams qualified for the 2008 ESA LRC and received funding from the ESA General Studies Programme (GSP):

- Universität Bremen, Germany,
- Jacobs University Bremen, Germany,
- Universidad Politecnica de Madrid, Spain,
- Oulun Yliopisto (University of Oulu), Finland
- Universit di Pisa, Italy,
- Scuola Superiore Sant'Anna Pisa, Italy,
- Surrey Space Centre, University of Surrey, United
- Swiss Federal Institute of Technology (ETH) Zurich, Switzerland

After a few months of design and implementation phases, the actual LRC took place in October 2008 in the volcanic landscape of the Teide National Park on the island of Tenerife. The team of Jacobs University built two robots (figure 3) for this event. One major goal of the team was to use the unique opportunity of this field test to investigate intelligent onboard features. One particularly interesting aspect in this scenario is that vision based approaches to environment modeling are severely hindered by the light conditions (figure 4).

The Jacobs team hence used a second robot in addition to a probe robot. This second robot serves as a communication

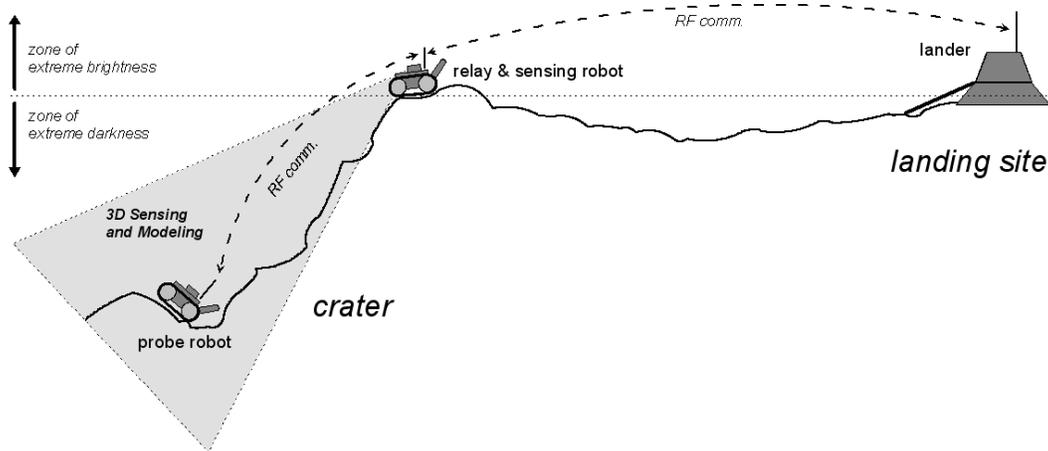


Fig. 2. The setup of the ESA LRC features a typical scenario where 3D sensing and modeling is highly desirable, but vision based approaches would have to cope with very challenging light conditions.



Fig. 4. The robots of the Jacobs team during the actual ESA Lunar Robotics Challenge taking place in the night with simulated light conditions as found in the polar regions of the moon. On the left top, the robots heading for the crater rim. On the right top, the probe robot has already passed the rim of the crater. The second robot serves as communication relay station and assists in the search operation. On the bottom, the light conditions within the crater can be seen.

relay between the lander and the probe robot and furthermore as a 3D environment modeler. For this purpose, the robot is equipped with an actuated laser range finder (LRF). The sensor is based on a commercial 2D LRF of the type SICK S300 combined with a simple servo for a pitching motion for the additional degree of freedom. The SICK S300 has a horizontal field of view of 270° of 541 beams. The servo allows a maximum motion from -90° to $+90^\circ$ at a maximum spacing of 0.5° . This gives a 3D point-cloud of a total size of $541 \times 361 = 195,301$ per sample in the highest resolution. The maximum range of the sensor is about 20 meters. The time to take one full scan in maximum resolution is about $T_{\text{scan}} \approx 32$ seconds. This scan time is the main bottleneck for the envi-

ronment modeling. But there are - more costly - off-the-shelf 3D-LRF alternatives that provide even higher resolution 3D data at significantly faster rates; an example is the Velodyne HDL-64E that provides an update rate with > 1.333 million points per second. The preparation of the team for the LRC, as well as the related research are supplemented by additional experiments in the Unified System for Automation and Robot Simulation (USARSim) [22] where a model of a Velodyne HDL-64E is available for testing purposes. Furthermore, we have imported ground truth data from Mars into USARSim (figure 5).



Fig. 5. Planetary exploration in the Unified System for Automation and Robotics Simulator - USARsim.

III. ESTIMATION OF PLANE PARAMETERS AND THEIR UNCERTAINTY

A general idea of an analytical solution to the *Approximate Least Squares Problem* (ALSP) formulated for the optimal plane fitting under the assumption of the radial Gaussian noise is presented here. It is based on our previous work [23][24].

The equation of a plane is $\hat{\mathbf{n}} \cdot \mathbf{r} = d$, where $\hat{\mathbf{n}}$ is the plane's unit normal and d the distance to the origin. Assume that the sensor returned a point-cloud $\mathbf{r}_j = \rho_j \hat{\mathbf{m}}_j$, $j = 1 \dots N$, where, $\hat{\mathbf{m}}_j$ are the measurement directions for the sensor, usually accurately known, and ρ_j are the respective ranges which are noisy. We make an assumption that $\rho_j \sim N(\bar{\rho}_j, \sigma^2(\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j))$, where, $\bar{\rho}_j = \frac{d}{\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j}$ is the true range of j -th measurement.

For commonly available 3D sensors like actuated Laser-Range-Finders (aLRF) the standard deviation is modeled as ([25], [26], [27])

$$\sigma(\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j) = \frac{\hat{\sigma}(\bar{\rho}_j)}{|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|}, \quad \hat{\sigma}(\bar{\rho}_j) \triangleq \kappa \bar{\rho}_j^2 \quad (1)$$

where $\hat{\mathbf{n}}$ is the local normal to the surface the point \mathbf{r}_j lies on. The coefficient $\kappa > 0$ can be estimated by doing initial calibration experiments with the sensor. Using this we get the following log-likelihood function

$$\mathcal{L} = K - \sum_{j=1}^N \log \frac{\hat{\sigma}(\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)}{|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|} - \frac{1}{2} \sum_{j=1}^N \frac{[(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j) \rho_j - d]^2}{\hat{\sigma}^2(\bar{\rho}_j)} \quad (2)$$

\mathcal{L} has to be maximized w.r.t. $\hat{\mathbf{n}}$, d . This can not be handled analytically, especially as σ is a function of $\frac{d}{\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j}$, therefore we make the assumption $\hat{\sigma}(\bar{\rho}_j) \approx \hat{\sigma}(\rho_j)$. We define $\hat{\sigma}_j \triangleq \hat{\sigma}(\rho_j)$ and note that $\hat{\sigma}_j$ is now no longer a function of $\bar{\rho}_j$ and hence of $\hat{\mathbf{n}}$ and d . Using Eq. (2) and ignoring constant terms together with sum of logarithms we get the ALSP formulation

$$\max_{\hat{\mathbf{n}}, d} \mathcal{L}_{\text{ALSP}} = -\frac{1}{2} \sum_{j=1}^N \frac{(\hat{\mathbf{n}} \cdot \mathbf{r}_j - d)^2}{\hat{\sigma}_j^2} \quad (3)$$

After solving the constrained optimization problem using Lagrange multipliers we get the estimates of the plane parameters

$$d^* = \hat{\mathbf{n}}^{*\text{T}} \left(\frac{\sum_{j=1}^N \frac{1}{\hat{\sigma}_j^2} \mathbf{r}_j}{\sum_{j=1}^N \frac{1}{\hat{\sigma}_j^2}} \right) \triangleq \hat{\mathbf{n}}^{*\text{T}} \mathbf{r}_G \quad (4)$$

where $\hat{\mathbf{n}}^*$ is the eigenvector corresponding to the smallest eigenvalue of the positive semi-definite weighted scatter matrix and \mathbf{r}_G can be seen as its center of gravity when thinking of the weights as masses

$$M = \sum_{j=1}^N \frac{1}{\hat{\sigma}_j^2} (\mathbf{r}_j - \mathbf{r}_G)(\mathbf{r}_j - \mathbf{r}_G)^{\text{T}}. \quad (5)$$

The Hessian of log-likelihood evaluated at the optimum is

$$H^* = \begin{bmatrix} H_{nn}^* & H_{nd}^* \\ H_{nd}^{*\text{T}} & H_{dd}^* \end{bmatrix}, \quad (6)$$

where

$$H_{nn}^* = -\sum_{j=1}^N \frac{\mathbf{r}_j \mathbf{r}_j^{\text{T}}}{\hat{\sigma}_j^2} + \left[\sum_{j=1}^N \frac{\hat{\mathbf{n}}^{*\text{T}} (\mathbf{r}_j - \mathbf{r}_G) (\hat{\mathbf{n}}^{*\text{T}} \mathbf{r}_j)}{\hat{\sigma}_j^2} \right] I_3 \quad (7)$$

$$H_{nd}^* = \sum_{j=1}^N \frac{\mathbf{r}_j}{\hat{\sigma}_j^2}, \quad H_{dd}^* = -\sum_{j=1}^N \frac{1}{\hat{\sigma}_j^2}$$

Then the covariance matrix is [28]

$$C(\hat{\mathbf{n}}^*, d^*) = -(H^*)^{-1} \quad (8)$$

H^* has a zero eigenvalue in the direction of $(\hat{\mathbf{n}}^*, d^*)^{\text{T}}$ therefore the Moore-Penrose generalized inverse has to be used. This property of the Hessian is discussed in [23], [29].

IV. RANGE IMAGE SEGMENTATION

In this section, the principle of our algorithm for identifying regions of points that lie on one plane is presented. This extends the work presented in [30] by modifying the algorithm by reformulating the underlying mathematics to an incremental version, which allows a highly efficient implementation. The detailed description of the algorithm can be found in [24]. Here only an overview and the most important optimizations are provided.

The algorithm proceeds as follows. The best seed for a new region is selected from points yet unassigned to any region. This is done by using local planarity criteria. Concretely, the local planarity of a point is evaluated by fitting a plane in a small 10×10 window around the point and calculating the mean square error of the optimal fit. A preference list of the seeds is created in a preprocessing step by sorting all the points according to this error. Then the region is expanded using breadth-first-search. The neighbors of each boundary point are taken from its Moore neighborhood of range 1 in the range image. The new point is assigned to the region only if it satisfies the mean square error and the distance to the optimal plane constraints. Once the region cannot be grown further, a new region is started. The algorithm terminates when no new seeds are available.

The most crucial operation of the algorithm is the calculation of the optimal fit and its mean square error. It has to be performed whenever a new point is investigated. Therefore, we introduce a minimization of the computational complexity for this.

We know that calculating the optimal fit for a set of 3D points $\mathbf{r}_j = (x_j, y_j, z_j)^{\text{T}}$, $j = 1 \dots k$ requires to find the eigenvector $\hat{\mathbf{n}}^*$ corresponding to the smallest eigenvalue of

the weighted scatter matrix M_k (Eq. (5)). Now suppose the matrix M_k would be calculated from the start every time a new point is added to the region. This would mean that one would need to traverse every point in the current region leading to a huge overhead. Here a way to an incremental update of the matrix M_k and its mean square error is presented, which takes previous calculations into account.

Three variables can be defined that describe a state of k points \mathbf{r}_j

- The sum of all weights - $w_k = \sum_j^k \frac{1}{\sigma_j}$
- The weighted sum of all points - $\mathbf{s}_k = \sum_j^k \frac{\mathbf{r}_j}{\sigma_j^2}$
- The weighted sum of all product matrices -

$$P_k = \sum_j^k \frac{\mathbf{r}_j \mathbf{r}_j^T}{\sigma_j^2}$$

We can express the matrix M_k and the mean square error MSE_k using these state variables in the following way

$$\begin{aligned} M_k &= P_k - \frac{\mathbf{s}_k \mathbf{s}_k^T}{w_k}, \\ MSE_k &= \frac{1}{w_k} \hat{\mathbf{n}}_k^{*T} P_k \hat{\mathbf{n}}_k^* - \frac{2d_k^*}{w_k} \hat{\mathbf{n}}_k^* \cdot \mathbf{s}_k + d_k^{*2}. \end{aligned} \quad (9)$$

Now suppose we want to add a new point \mathbf{r}_{k+1} . We can easily update the state variables

$$\begin{aligned} w_{k+1} &= w_k + \frac{1}{\sigma_{k+1}^2}, \quad \mathbf{s}_{k+1} = \mathbf{s}_k + \frac{\mathbf{r}_{k+1}}{\sigma_{k+1}^2}, \\ P_{k+1} &= P_k + \frac{\mathbf{r}_{k+1} \mathbf{r}_{k+1}^T}{\sigma_{k+1}^2} \end{aligned} \quad (10)$$

From equations (9) and (10) we get M_{k+1} and MSE_{k+1} . In other words we have updated M_{k+1} and MSE_{k+1} indirectly by updating the variables \mathbf{s}_k , w_k and P_k . Please note that for MSE_{k+1} the new normal vector $\hat{\mathbf{n}}_{k+1}^*$ and d_{k+1}^* have to be calculated based on the M_{k+1} matrix.

Note that the Hessian matrix in Eq. (7) can be efficiently evaluated during the optimal fit calculation in the following way. Suppose we have a set of k points $\{\mathbf{r}_j\}$ with the state $\{P_k, \mathbf{s}_k, w_k\}$ and the estimate $(\hat{\mathbf{n}}^{*T}, d^*)^T$, where $\hat{\mathbf{n}}^*$ is the eigenvector corresponding to the smallest eigenvalue λ of matrix M_k , then it can be shown that

$$H^* = \begin{bmatrix} -P_k + \lambda I_3 & \mathbf{s}_k \\ \mathbf{s}_k^T & -w_k \end{bmatrix} \quad (11)$$

As we always keep track of the state $\{P_k, \mathbf{s}_k, w_k\}$ and the eigenvalue λ is calculated during the optimal fitting, we need only a few extra operations to find the Hessian matrix, which can be used to obtain the covariance matrix in Eq. (8).

The incremental formulation leads to a linear $O(n)$ computational complexity, where n is the number of points in a point cloud.

V. FINDING POLYGONAL REPRESENTATION OF PLANAR SURFACES

This is the part where most of the compression of the original point cloud can be made. Once the range image is segmented, the points are projected to their optimal planes, thus reducing the representation dimension to 2, i.e., each segment can be represented by the parameters $(\hat{\mathbf{n}}^*, d^*)^T$,

H^* (actually, it is enough to store only the Hessian H^* , as $(\hat{\mathbf{n}}^*, d^*)^T$ is in its nullspace) of the optimal plane and a set of 2D points defined in the planes' frame.

We note that the most important part per segment is the boundary of its 2D set of points. As the neighborhood information of points is sustained from the pixel space of the range image to the Cartesian space, the boundary of the segment is the same in both spaces. Finding border pixels of a segment in a range image is straightforward. One just need to check the Moore neighborhood of range 1 for each pixel in the segment. If all neighbors of the pixel are from the same segment then it is an inner pixel otherwise it is a boundary pixel. Taking corresponding Cartesian coordinates of border pixels gives us the outline of the planar patch. The inner points are redundant and can be removed. Note that the outline usually consists of several components - one outer boundary and several boundaries of inner holes.

The boundary representation already reduces the initial size quite significantly, however it is usually very dense and can be approximated, thus allowing us to increase memory efficiency even more. One option is to use a Minimum Description Length (MDL) encoding of the chain code, which is computationally expensive. A simple, but sufficient alternative is a convex hull algorithm in form of Graham's scan, which achieves very high compression rates in a short runtime of $O(n \cdot \log(n))$. Convex hull has certain limitations such as holes closing and loosing concave features, which sometimes ends in very rough approximations. Therefore one should always consider which detail level is needed before applying it. Usually in datasets with large openings (which is true in our case), the large scale features are more interesting, therefore the convex hull is suitable candidate.

VI. EXPERIMENTS AND RESULTS

The performance of the presented approach was among others evaluated in the context of the ESA LRC. As mentioned before, one of the Jacobs LRC robots is equipped with a self-made actuated Laser Range Finder (LRF). This is a rather low-end solution for acquiring 3D LRF data. The 3D measurements were made in the volcanic landscape of the Teide National Park on Tenerife where the ESA Lunar Robotics Challenge (LRC) took place. To test scalability with high amounts of data, we also conducted experiments in a virtual environment, where the Eagle crater on Mars is modeled in USARsim based on ground truth data from the Mars Exploration Rover (MER) mission data archives. There, a high end 3D LRF sensor in form of a Velodyne HDL-64E is used. Examples of the point clouds used in the experiments are shown in figures 6 and 7.

Examples of the results of the plane fitting and polygonalization are shown in figures 8 and 9. Multiple views of the 3D models are shown in these two figures; nevertheless, a full 3D visualization - like, for example, the one provided in the operator GUI of the Jacobs robots - provides a much better overview. For the convenience of the reader, several movies with 3D visualizations of the raw point clouds as well as of the results of the plane fitting can be downloaded from <http://robotics.jacobs-university.de/projects/3Dmap>.

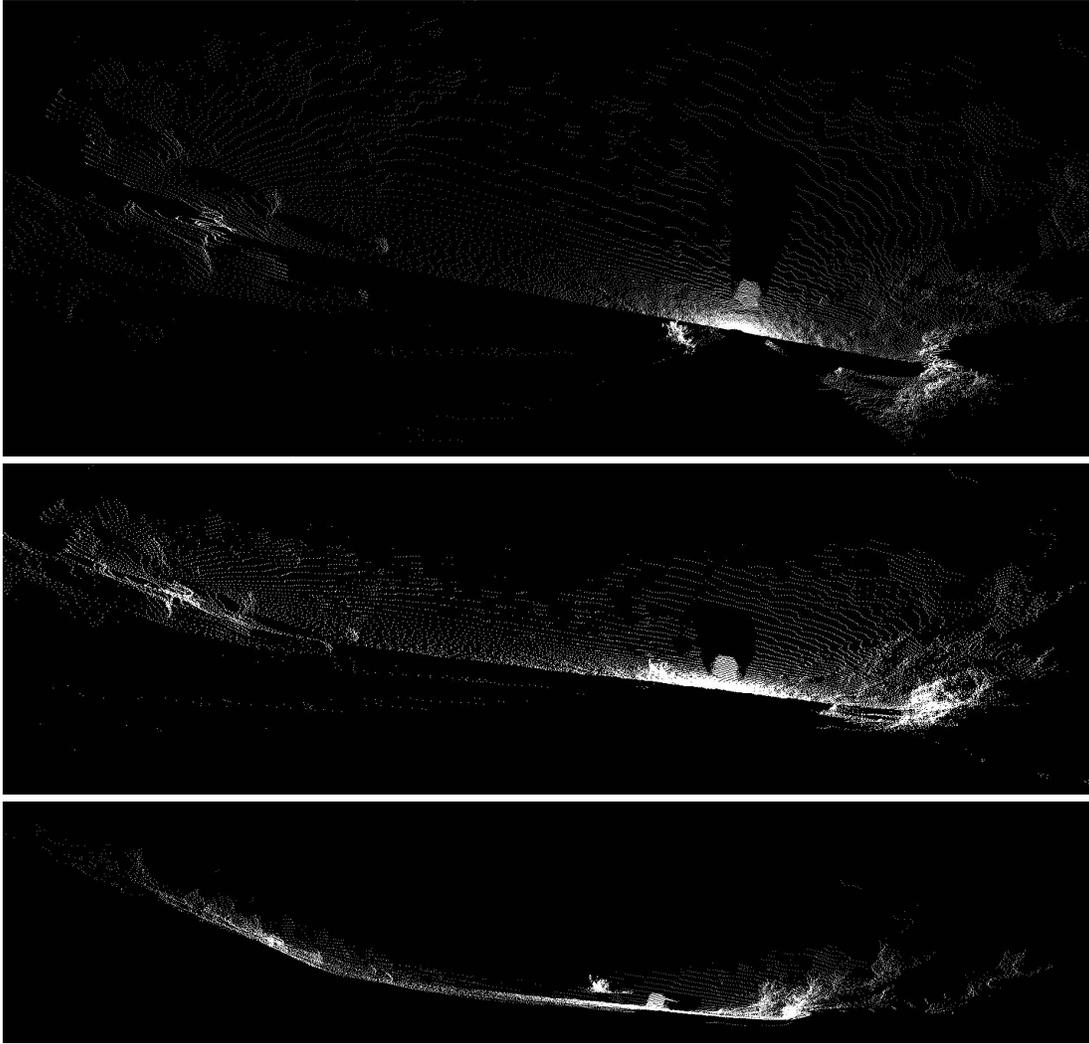


Fig. 6. Several views of a 3D-LRF point cloud acquired at the ESA Lunar Robotics Challenge.

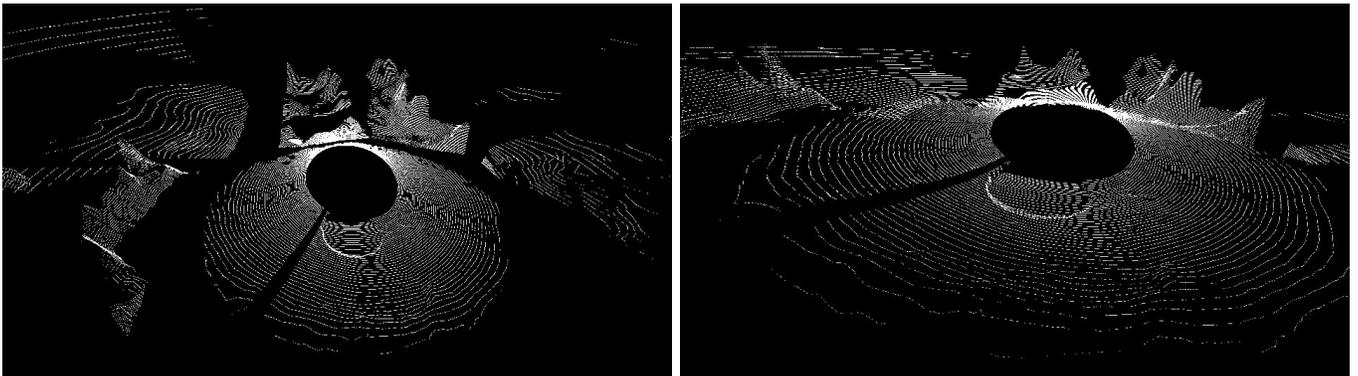


Fig. 7. A point cloud from a simulated robot with Velodyne 3D-LRF in a scenario with ground truth data from Mars.

TABLE I
AVERAGE PERFORMANCE STATISTICS ON THE DATASET FROM THE ESA LRC (11 SCANS, $\sim 4 \cdot 10^4$ POINTS PER SCAN)

Polygonal Representation	Raw Size, KB, (1)	Compressed Size, KB, (2)	Compression Ratio, (1)/(2)	Space Savings $1 - (2)/(1)$
Boundary	576.93	39.24	14.7	0.932
Convex Hull	576.93	4.02	143.37	0.993

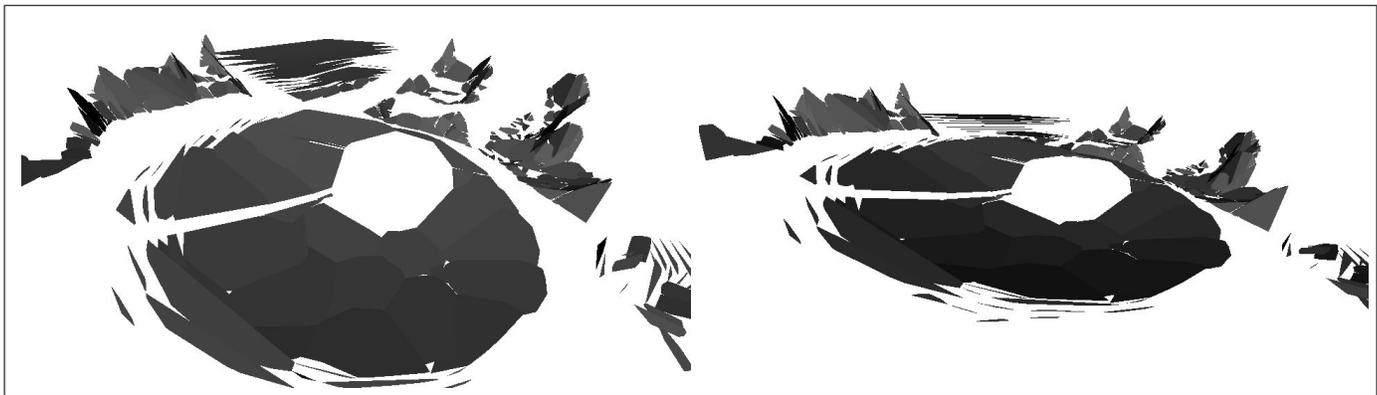


Fig. 9. Results of the plane patch fitting for the three point clouds from the Velodyne 3D-LRF on Mars shown in figure 7. Again, two views are shown for each case. Despite the compression by three orders of magnitude, the general overview is still very useful; easy as well as hard to negotiable areas can be well recognized.

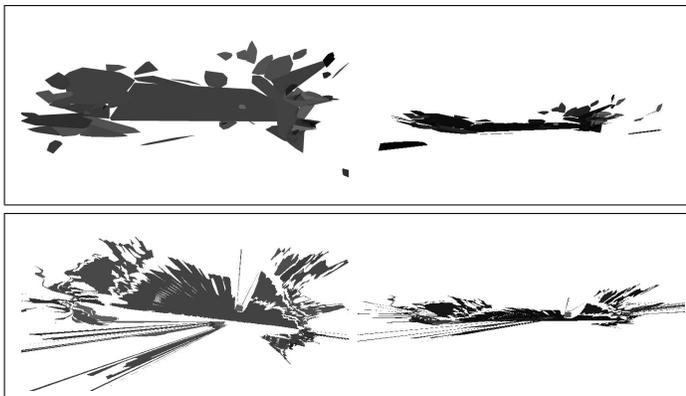


Fig. 8. Results of the plane fitting for the point cloud shown in figure 6. On top, two views of a representation where the boundaries of the plane patches are computed using the convex hull; on the bottom, the boundaries are computed by MDL polygonalization.

TABLE II

AVERAGE RUNTIMES ON THE DATASET FROM THE ESA LRC (11 SCANS,
 $\sim 4 \cdot 10^4$ POINTS PER SCAN)

Polygonal Representation	Runtimes (sec)		
	Segmentation	Polygonalization	Total
Boundary	0.66	0.05	0.71
Convex Hull	0.66	0.14	0.80

Tables I and II show the compression rates and run-times for the ESA LRC data. All experiments were carried out on a 64-bit mobile platform with Intel Core 2 Duo T7100 (1.8GHz) CPU. As can be seen, the plane fitting with convex hull polygonalization leads to compressions of two orders of magnitude. The computations are very fast, the total runtimes are in the order of a few hundred milli-seconds; the approach is hence well suited for online computations on the robot.

TABLE III

THE MINIMUM, MAXIMUM, AND AVERAGE RUNTIMES (SEC) FOR COMPUTING A VORONOI DIAGRAM FOR PATH PLANNING ON THE POINT CLOUDS, RESPECTIVELY ON THE OPTIMIZED POLYGONS

	min	max	average
point clouds	72.81	97.91	85.48
fitted polygons	0.92	2.44	1.49

The planar surface models are helpful in increasing the situational awareness of the operator. They are in addition a useful basis for more complex autonomous operations, especially path planning. Based on the point cloud data, respectively the boundary points of the optimized polygon patches, Voronoi diagrams can be computed as the basis for roadmaps of the environment. Concretely, the 3D points are projected down to the plane, a 2D Voronoi diagram is computed, and edge segments can be assigned a cost that corresponds to the inclination of the related surface in the original 3D data; steep surfaces exceeding the robots locomotion capabilities are hence associated with infinite cost, the corresponding edges in the roadmap are marked as non-traversable. As shown in table III, the computation of the Voronoi diagrams benefits significantly from the generation of large surface patches. The more compact representation of the environment has a very positive influence on the runtime.

VII. CONCLUSIONS

3D perception and modeling is an important basis for mobile robot operations in planetary exploration scenarios as it supports good situation awareness for motion level teleoperation as well as higher level intelligent autonomous functions. It is hence desirable to get long range 3D data with high resolution, large field of view, and very fast update rates. 3D Laser Range Finders (3D-LRF) have a high potential in this respect. In addition, 3D-LRF can operate under conditions where standard vision based methods fail, e.g., under extreme light conditions. But it is non-trivial to transmit the huge amounts of data delivered by a 3D-LRF to an operator station or to use this point cloud data as basis for higher level intelligent functions. Based on our participation in the Lunar Robotics Challenge (LRC) of the European Space Agency (ESA), it is shown how the huge amounts of 3D point cloud data from 3D-LRF can be tremendously reduced. Concretely, large sets of points are replaced by planar surface patches that are fitted into the data in an optimal way. The underlying computations are very efficient and hence suited for online computations onboard of the robot.

ACKNOWLEDGMENTS

The authors would like to thank the European Space Agency (ESA) for their support of the Jacobs Team as part of the 2008 Lunar Robotics Challenge (LRC). The presented research on 3D modeling received the financial support of *Deutsche Forschungsgemeinschaft* (DFG).

KEYWORDS

space robotics, planetary exploration, telerobotics, autonomy, 3D mapping, surface representation, plane fitting

REFERENCES

- [1] A. Birk, S. Schwertfeger, and K. Pathak, "A Networking Framework for Teleoperation in Safety, Security, and Rescue Robotics (SSRR)," *IEEE Wireless Communications, Special Issue on Wireless Communications in Networked Robotics*, vol. 6, no. 13, pp. 6–13, 2009.
- [2] K. Iagnemma, C. Brooks, and S. Dubowsky, "Visual, tactile, and vibration-based terrain analysis for planetary rovers," in *IEEE Aerospace Conference*, vol. 2, 2004, pp. 841–848 Vol.2.
- [3] K. Iagnemma, H. Shibly, and S. Dubowsky, "On-line terrain parameter estimation for planetary rovers," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 2002, pp. 3142–3147.
- [4] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains: Functions and integration," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 917–942, 2002.
- [5] —, "Autonomous rover navigation on unknown terrains functions and integration," in *Experimental Robotics VII*, ser. Lecture Notes in Control and Information Sciences, 2001, vol. 271, pp. 501–510.
- [6] D. B. Gennery, "Traversability analysis and path planning for a planetary rover," *Autonomous Robots*, vol. 6, no. 2, pp. 131–146, 1999.
- [7] J. Norris, M. Powell, M. Vona, P. Backes, and J. Wick, "Mars Exploration Rover Operations with the Science Activity Planner," in *IEEE International Conference on Robotics and Automation, ICRA*, 2005, pp. 4618–4623.
- [8] J. Erickson, "Living the dream - an overview of the Mars exploration project," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 2, pp. 12–18, 2006.
- [9] J. Biesiadecki, E. Baumgartner, R. Bonitz, B. Cooper, B. A4 Cooper, F. Hartman, F.R. A5 Hartman, P. Leger, P.C. A6 Leger, M. Maimone, M.W. A7 Maimone, S. Maxwell, S.A. A8 Maxwell, A. Trebi-Ollennu, A. A9 Trebi-Ollennu, E. Tunstel, E.W. A10 Tunstel, and J. Wright, J.R. A11 Wright, "Mars exploration rover surface operations: driving opportunity at Meridiani Planum," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 2, pp. 63–71, 2006.
- [10] R. Lindemann, D. Bickler, B. Harrington, G. Ortiz, G.M. A4 Ortiz, and C. Voothees, C.J. A5 Voothees, "Mars exploration rover mobility development," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 2, pp. 19–26, 2006.
- [11] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, A. A4 Chase, J.-J. Hsu, J.C.-J. A5 Hsu, A. Jonsson, A. A6 Jonsson, B. Kanefsky, B. A7 Kanefsky, P. Morris, P. A8 Morris, K. R. A. K. Rajan, J. Yglesias, J. A10 Yglesias, B. Chafin, B.G. A11 Chafin, W. Dias, W.C. A12 Dias, and P. Maldague, P.F. A13 Maldague, "MAPGEN: mixed-initiative planning and scheduling for the Mars Exploration Rover mission," *Intelligent Systems, IEEE*, vol. 19, no. 1, pp. 8–12, 2004.
- [12] Y. Kunii and T. Ushioda, "Shadow casting stereo imaging for high accurate and robust stereo processing of natural environment," in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, 2008, pp. 302–307.
- [13] T. Kubota, K. Moesl, and I. Nakatani, "Map Matching Scheme for Position Estimation of Planetary Explorer in Natural Terrain," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 3520–3525.
- [14] L. Edwards, M. Sims, C. Kunz, D. Lees, and J. Bowman, "Photo-realistic Terrain Modeling and Visualization for Mars Exploration Rover Science Operations," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2, 2005, pp. 1389–1395.
- [15] M. Massari, E. Ceriani, L. Rigolin, and F. Bernelli-Zazzera, "Optimal path planning for planetary exploration rovers based on artificial vision system for environment reconstruction," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, 2005, pp. 987–992.
- [16] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 5, 2002, pp. 5–2025–5–2036 vol.5.
- [17] Y. Kunii and T. Gotoh, "Evaluation of Shadow Range Finder: SRF for planetary surface exploration," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, 2003, pp. 2573–2578 vol.2.
- [18] Y. Kunii, S. Tsuji, and M. Watari, "Accuracy improvement of Shadow Range Finder: SRF for 3D surface measurement," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, 2003, pp. 3041–3046 vol.3.
- [19] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the Mars exploration rovers - a tool to ensure accurate driving and science imaging," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 2, pp. 54–62, 2006.
- [20] CSA, "Avatar Explore Mission," <http://www.asc-csa.gc.ca/eng/sciences/avatar.asp>, 2009.
- [21] ESA, "Lunar Robotics Challenge (LRC)," http://www.esa.int/esaCP/SEMGAASHKHF_index_0.html, 2008.
- [22] USARsim, "Unified System for Automation and Robotics Simulator (USARsim)," <http://usarsim.sourceforge.net/>, 2006.
- [23] K. Pathak, N. Vaskevicius, and A. Birk, "Revisiting Uncertainty Analysis for Optimum Planes Extracted from 3D Range Sensor Point-Clouds," in *International Conference on Robotics and Automation (ICRA)*. IEEE press, 2009, pp. 1631 – 1636.
- [24] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast Plane Detection and Polygonalization in noisy 3D Range Images," in *International Conference on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE Press, 2008, pp. 3378 – 3383.
- [25] F. Prieto, T. Redarce, P. Boulanger, and R. Lepage, "CAD-based range sensor placement for optimum 3D data acquisition," in *Second International Conference on 3-D Imaging and Modeling (3DIM'99)*. Los Alamitos, CA, USA: IEEE Computer Society, 1999, p. 0128. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/IM.1999.805343>
- [26] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash lidar devices," in *International Conference on Sensing Technologies*, Palmerston North, New Zealand, November 2005. [Online]. Available: <http://www.frc.ri.cmu.edu/~alonzo/pubs/papers/icst05FlashLadar.pdf>
- [27] K. Pathak, A. Birk, S. Schwertfeger, and J. Poppinga, "3D Forward Sensor Modeling and Application to Occupancy Grid Based Sensor Fusion," in *International Conference on Intelligent Robots and Systems (IROS)*. San Diego, USA: IEEE Press, 2007, pp. 2059 – 2064.
- [28] D. S. Sivia, *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 1996.
- [29] Y. Kanazawa and K. Kanatani, "Reliability of fitting a plane to range data," *IEICE Transactions on Information and Systems*, vol. E78-D, no. 12, pp. 1630–1635, 1995.
- [30] D. Hähnel, W. Burgard, and S. Thrun, "Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 15–27, 2003.

Andreas Birk is an associate professor in Electrical Engineering and Computer Science (EECS) at Jacobs University Bremen. He started at Jacobs University in Fall 2001, building up the robotics group at this newly founded private university. Before he joined Jacobs, he held a research-mandate from the Flemish Society for Applied Research (IWT) and was a visiting Professor at the Vrije Universiteit Brussel (VUB) where he did his research in the Artificial Intelligence Lab (AI-lab). In 1995 he received his doctorate from the Universität des Saarlandes, Saarbrücken, where he previously studied Computer Science from fall 1989 to spring 1993. Andreas Birk's research focuses on autonomous systems. On the engineering side, he is working on the design and construction of complete systems. This includes the design and construction of embedded hardware and mechatronics as well as software development up to full autonomy. On the basic research side, he is interested in a constructive understanding of intelligence.

Narunas Vaskevicius has received his MSc. degree in Computer Science with specialization in Robotics from Jacobs University Bremen, Germany in 2008. He is currently pursuing a Ph.D degree in Computer Science as a member of Jacobs Robotics Lab. His research concentrates on autonomous robotics in a context of path planning and 3D mapping based on probabilistic surface feature extraction. Among others he applies his research in the fields of space, underwater, as well as safety, security and rescue robotics.

Kaustubh Pathak received his Ph.D. in Mechanical Engineering in Fall 2005 from the University of Delaware, Newark, USA, in the area of nonlinear control of underactuated and nonholonomic systems. Since 2006, he is a post-doctoral fellow with the robotics group of the department of EECS at the Jacobs University Bremen, Germany. His research interests include probabilistic feature extraction and fusion, 3D mapping, path-planning, and control of mobile robots. He has authored over 25 papers in international journals and conferences, and has been a regular participant in the rescue-league of Robocup.

Sören Schwertfeger received his Diploma in Computer Science from the University of Bremen in 2005. Currently he is pursuing the Ph.D. in Computer Science at the Jacobs University Bremen. He is working at the Jacobs Robotics Lab, developing algorithms and software for rescue and other autonomous robots. His research interests include object recognition, robot autonomy, 3D mapping algorithms and genetic algorithms.

Jann Poppinga received his Diploma in Computer Science from the University of Oldenburg in 2006. Currently he is pursuing the Ph.D. in Computer Science at the Jacobs University Bremen. He is working at the Jacobs Robotics Lab, developing algorithms and software for rescue and other autonomous robots. His research interests include 3D mapping algorithms and 3D data processing.

Heiko Bülow received his diploma(FH) in Electrical Engineering from the University of Applied Sciences in Braunschweig/Wolfenbuettel and the M.Phil degree from the School of Computing at University of Glamorgan. Currently he is pursuing the Ph.D. at the Jacobs University in Bremen. He is working at the Jacobs Robotics Lab, developing algorithms for rescue and other autonomous robots. His research interests include multi-dimensional signal processing, pattern recognition and computer vision.