amcs

RGB–D TERRAIN PERCEPTION AND DENSE MAPPING FOR LEGGED ROBOTS

DOMINIK BELTER [a,*], PRZEMYSŁAW ŁABECKI [a], PÉTER FANKHAUSER [b],
ROLAND SIEGWART [b]

[a] Institute of Control and Information Engineering
Poznań University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland
e-mail: `pl@cie.put.poznan.pl, dominik.belter@put.poznan.pl`

[b] Autonomous Systems Lab
ETH Zurich, LEE J 201, Leonhardstrasse 21, 8092 Zurich, Switzerland
e-mail: `{pfankhauser, rsiegwart}@ethz.ch`

This paper addresses the issues of unstructured terrain modeling for the purpose of navigation with legged robots. We present an improved elevation grid concept adopted to the specific requirements of a small legged robot with limited perceptual capabilities. We propose an extension of the elevation grid update mechanism by incorporating a formal treatment of the spatial uncertainty. Moreover, this paper presents uncertainty models for a structured light RGB-D sensor and a stereo vision camera used to produce a dense depth map. The model for the uncertainty of the stereo vision camera is based on uncertainty propagation from calibration, through undistortion and rectification algorithms, allowing calculation of the uncertainty of measured 3D point coordinates. The proposed uncertainty models were used for the construction of a terrain elevation map using the Videre Design STOC stereo vision camera and Kinect-like range sensors. We provide experimental verification of the proposed mapping method, and a comparison with another recently published terrain mapping method for walking robots.

Keywords: RGB-D perception, elevation mapping, uncertainty, legged robots.

1. Introduction

To operate autonomously on a previously unknown terrain, a robot has to perceive the environment, build a model, localize itself and plan its motion. Environment perception plays an important role in robot navigation. The robot needs accurate data from sensors to build a model of the environment. This model (map) can be used to adapt the gait to the terrain profile (Walas and Belter, 2011), plan the motion (Zucker *et al.*, 2010) and localize the robot (Yoon *et al.*, 2013), and should thus be designed taking into account the real-time operation of the entire navigation system.

Among many existing representations of the environment (raw point cloud, occupancy map, 2.5D elevation map, triangle mesh, octree (Hornung *et al.*, 2013), multi-level surface (MLS) maps (Pfaff *et al.*, 2007)), we work with 2.5D elevation maps. Motion

planning for walking robots is more complex than for wheeled robots. Even if the robot walks statically stable, it needs to select footholds, modify the inclination of the body to actively adapt to the terrain profile, and plan feet and body trajectories. The motion planning procedures have to send many queries about the height of the terrain at a given location. We decided to use a 2.5D elevation map because it provides constant and short access time and most of the planning methods for walking robots are designed to use elevation maps (Belter and Skrzypczyński, 2011a; Belter *et al.*, n.d.; Stelzer *et al.*, 2012). The size of the cell should be similar to that of the robot's foot to allow for foothold selection. Thus, the resolution of the map for a walking robot should be higher than for a wheeled robot. With increasing computational power, full 3D representations like octree maps (Octomap) become more interesting, but are still more challenging than a 2.5D representation.

---

*Corresponding author

In our system, the map of the environment is updated using data from two types of sensors providing dense depth measurements: a passive stereo camera and an active PrimeSense structured light sensor. The most well-known sensor of this type is Microsoft Kinect, but we use Asus Xtion and PrimeSense Carmine, which are more suitable for legged robots due to their size. As the walking robot should operate in various environments, we are focused on two complementary sensors. The PrimeSense-based sensor returns accurate depth data indoors. Because Kinect and Asus Xtion use near-infrared structured light projectors to obtain information about the distance to the objects, these sensors do not operate outdoors (Belter *et al.*, 2015). Thus, we use a stereo camera, which returns depth data even in direct sunlight conditions. On the other hand, the stereo camera encounters problems whenever the objects in the environment do not contain sufficient visual features. Plain walls and floors, which are very common indoors, are problematic for passive stereo-based perception.

The mapping method presented in this paper is general and can be used with any kind of depth sensor as input to the mapping method. We demonstrate how to use data from two common types of sensors to update the height and uncertainty of each cell of the elevation map.

1.1. Approach and contribution. In this paper we address the problem of perceiving and mapping the environment for walking robots. We present a procedure which updates the height of the cell of the elevation map. To this end, we introduce a new structure of the map. Each cell of our elevation map corresponds to a stack of voxels. This structure allows removal of erroneous measurements which are not filtered by the front-end filtering procedures (that are based on the properties of the depth sensor used for environment perception). We also present an elevation estimation procedure based on a static Kalman filter (KF). The paper also provides methods which better handle occlusions (visibility checking procedure) and erroneous data points introduced by the stereo vision system.

We additionally show how to remove spurious data from stereo images. Most of the artifacts can be found on the edges of objects. The goals of the proposed method are to remove spurious data and to preserve sharp edges. We are looking for a method that is relatively simple and fast in comparison with those known from computer graphics (Dey *et al.*, 2012; Sharf *et al.*, 2007). We show that our technique improves the accuracy of the map in comparison with the embedded filters provided by the *Small Vision System* (SVS) (Konolige, 1997) software library, and removes less useful 3D points.

We also provide uncertainty models of the sensors used for mapping. We show how the measurement error propagates through the mapping procedure to the uncertainty of each cell of the map. We present

an analytical uncertainty propagation procedure for the stereo camera and the PrimeSense-based sensor. This procedure is general enough to be applied to other types of RGB-D and vision sensors, such as time-of-flight (ToF) cameras. Moreover, we provide sensor-specific procedures for filtering the measurement data. We show how to remove incorrect measurements from the point cloud obtained when the stereo camera is used.

The last part of the paper contains an experimental verification of the mapping method. We show the properties of the mapping method, its accuracy and the role of the data filtering procedure proposed in the paper. We also provide experimental results obtained on real walking robots.

This article is a significantly extended version of the conference paper by Łabecki and Belter (2014), which builds also on the previous works of the PUT team (Belter *et al.*, 2012; Łabecki and Skrzypczyński, 2013), and the ETH/ASL team (Fankhauser *et al.*, 2014) concerning perception and mapping for legged robots. In this paper we assume that the environment is quasi-static (it does not change in close vicinity of the robot), and that the robot is well-localized relative to its environment. Therefore, we can focus on efficient estimation of geometric properties of the terrain and the influence of uncertainty in dense depth perception. The initial version of this work was presented at the special session on *Robotic perception employing RGB-D images* during the 13th National Conference on *Robotics* in Kudowa Zdrój, Poland, 2014.

2. Related work

Several papers have already addressed the application of stereo vision for terrain mapping. Poppinga *et al.* (2010) reports the usage of a stereo-on-chip (STOC) camera as a sensor for mobile robots in search and rescue tasks. However, artifacts produced by the camera were not taken into account. Stereo vision has been used for terrain modeling with walking robots so far only in few projects, mostly because typical stereo systems impose high costs of 3D points computation. In the works of Kolter *et al.* (2009) and Rusu *et al.* (2009) walking robots with stereo-based perception are shown to autonomously traverse a rugged terrain, but in both cases the computations are performed off-board, and explicit propagation of the spatial uncertainty from the stereo data to the elevation map is not taken into account. The knowledge about the elevation uncertainty of each cell in the map allows planning the path of the walking robot more efficiently, while avoiding uncertain areas (Belter and Skrzypczyński, 2011b). Finally, the uncertainty measure can be used for data fusion in a multi-sensor perception system. In this case each sensor should have its own uncertainty model, but these models have to be expressed in a common framework.

The approach of Kleiner and Dornhege (2007) as well as Fankhauser *et al.* (2014) is based on the estimation of the motion of the robot. The simplified solution of Kleiner and Dornhege (2007) assumes that the uncertainty of the robot's position and orientation is reflected in the map by linearly growing the variance of the height estimate based on the accumulated distance and angle. This approach conservatively merges an approximation of the pose uncertainty into the height variance without taking the effect of the in-plane uncertainty into account. The work of Fankhauser *et al.* (2014) extends this formulation to propagate the full pose uncertainty of the robot to the map. The mapping method presented in this paper does not account directly for the robot pose uncertainty, assuming that a precise pose estimate is available to the robot. This makes computing the elevation values simpler and faster than in the work of Fankhauser *et al.* (2014). This assumption, when applied to the local area around the legged robot which is in the field of view of the on-board RGB-D or passive camera, is quite realistic, as demonstrated by Belter and Skrzypczyński (2013). For indoor applications, precise pose estimates of a walking robot can be also obtained using RGB-D sensing (Nowicki and Skrzypczyński, 2013). On the other hand, the pose uncertainty may be easily added to the updating mechanism, as the underlying data structures are very similar.

Some of the grid-based algorithms for rough terrain mapping use *ad hoc* formulations for mapping; e.g., Ye and Borenstein (2004) apply heuristics when they fuse consecutive laser range measurements into an elevation grid: their certainty measure is a simple hit/miss counter. On the other hand, there are 2.5D mapping methods which are based on probabilistic foundations. The locus method of Kweon and Kanade (1992) applies a simple geometric approximation in order to obtain the elevation uncertainty from the model of a range sensor's beam and its interaction with the terrain. However, this method does not propagate explicitly the probabilistic model of range measurements to the uncertainty of elevation. Hebert *et al.* (1989) build a grid map for a planetary rover in which each cell represents the height of the terrain in order to approximate the surface of the environment. They use matching algorithms to find the corresponding transformation between multiple scans to build a composite elevation map. The authors do not address the issues of error propagation. More recently, a method for efficient volumetric mapping has been presented, which employed the notion of positive and negative occupancy mass to represent the environment (Dryanovski *et al.*, 2010). Although this model is efficient with regard to the map size, it is rather suited for robots that require a full 3D world model, not for walking robots, which can use much simpler 2.5D maps.
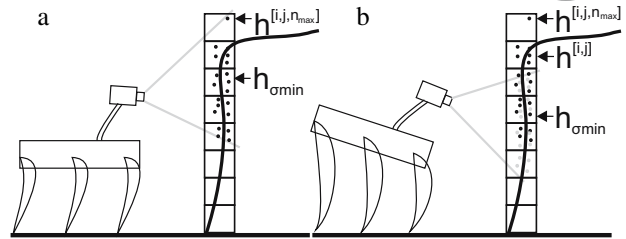

Fig. 1. Illustration of the space division concept which results in multiple hypotheses per cell.

An entirely different concept is presented by Plagemann *et al.* (2008), who propose a non-parametric continuous surface estimation method. This method is based on Gaussian process regression and applies non-stationary covariance functions for adaptation of the regression kernels to the structure of the terrain. It enables filling-in large gaps in elevation data, and provides estimates of uncertainty for the filled-in regions. However, these advantages come at the cost of a high computational burden, which can be prohibitive for a small walking robot.

So far we have not extended our mapping system towards description of multi-level or overhanging objects, such as bridges and tunnels. This can be implemented with multilevel surface maps (Pfaff *et al.*, 2007), but it is not necessary for the legged robot considered, which carries downwards-pointing RGB-D sensors or cameras, mounted in a geometric configuration optimized for terrain surface perception. We use the data from depth sensors directly to create a 2.5D model of the terrain. A similar approach is used on the DLR-Crawler walking robot (Stelzer *et al.*, 2012), which uses data from a stereo camera to model the environment and plan the motion of the robot. They applied the locus method (Kweon and Kanade, 1992), and showed that this method gives denser maps with fewer artifacts than the traditional method for creating a digital terrain model (Stelzer *et al.*, 2012). To minimize the influence of the localization error, existing values on the height map are overwritten by new values.

3. Elevation mapping method

3.1. Mapping method. The aim of the mapping method is to create an accurate model of the environment, which is suitable to be used for motion planning of a walking robot. We chose the elevation map representation of the environment because it allows us to efficiently query the map in the motion planning of the robot on a rough terrain (Belter and Skrzypczyński, 2011a; Belter *et al.*, n.d.; Saarinen *et al.*, 2013). The main problem with this mapping method for a walking robot is related to the perception system and geometrical properties of the obstacles. A mobile robot equipped with on-board

range sensors cannot measure elevation of the terrain directly (Fig. 1). When the robot is close to a high wall or its trunk is tilted down (Fig. 1(b)), it is not possible to estimate the height of the wall by measuring the top part of this obstacle. Assuming that the wall is represented in a 2.5D map, the best estimate of the elevation of the cells related to this wall is the highest measured value in the whole history of measurements for these cells $h^{[i,j,n_{\max}]}$. We refer to the (horizontal) cell index in the map with i and j, and the (vertical) voxel index with n. Note that in the example of Fig. 1 the elevation hypothesis with the lowest variance value $h_{\sigma_{\min}}$ is not a good estimate of the wall height. On the other hand, the maximal height in the history of measurements $h^{[i,j,n_{\max}]}$ is sensitive to spurious or very sparse measurements. Assuming that the maximal height is estimated solely in a probabilistic framework, like the Kalman filter, a single erroneous measurement might lead to a biased, incorrect estimate of the height of the obstacle (Fig. 1(b)). Our method removes sparse and erroneous measurements before they are fed to the probabilistic estimator, and then uses this estimator to combine the previous estimate of the cell elevation with the most recent measurements, providing an accurate estimate of the cell elevation, $h^{[i,j]}$.

The simple max height criterion originally proposed by Ye and Borenstain (2004) might cause improper height estimation even if only one measurement exists above the real height of the cell. Thus, we extended this approach, dividing the space above the 2.5D elevation cell into 3D voxels (Fig. 3(a)). This allows us to apply a state estimator, e.g., the Kalman filter, to obtain an optimal estimate of the given voxel's elevation from noisy measurements. Moreover, we apply a simple column hit counter to reduce the role of sparse and wrong measurements. Then we use another Kalman filter to estimate the elevation of the cell considering the previous estimates of this elevation. This approach separates the process of cell elevation update using measurements taken at various vantage points from the selection of the most appropriate elevation measurement at the given position of the robot. This way our procedure accounts for the qualitative-type uncertainty in the range measurements (Skrzypczyński, 2007) related to the change of the field of view when the robot observes an obstacle from different vantage points.

The diagram of the mapping procedure is presented in Fig. 2. The procedure uses point clouds as input and returns the height of each cell, $h^{[i,j]}_{(k+1)}$, and the corresponding uncertainty value, $\sigma^{2[i,j]}_{(k+1)}$. Each 3D point is used to update the map. The depth sensor measurements $([x,y,z]^T$ and related uncertainty Σ_p) are transformed to the map coordinate system using the estimated position of the robot. The method which allows computing Σ_p from a raw measurement is presented in Sections 4.2 and 4.3. In
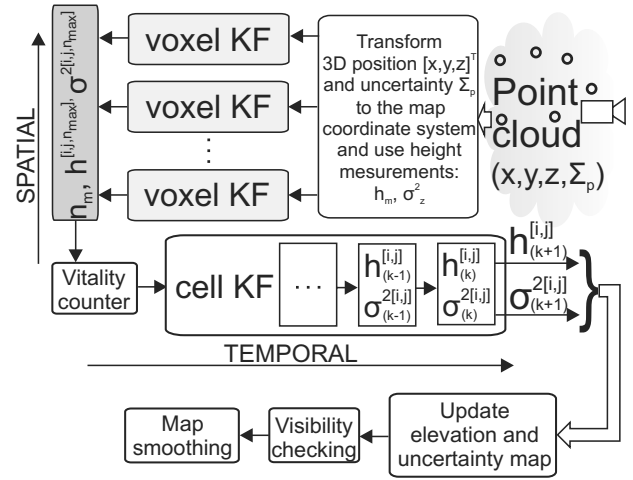

Fig. 2. General idea of the mapping procedure which shows how position $[x,y,z]^T$ and uncertainty in 3D coordinates Σ_p of each point in the cloud are transformed to height $h^{[i,j]}$ and uncertainty $\sigma^{2[i,j]}$ of each cell of the elevation and uncertainty maps. The number of points n_m used to update the voxel stack is employed to update the Vitality counter.

Section 4.1 we also show how to filter the raw point cloud obtained from the stereo camera to improve the accuracy of the elevation map.

The updated cell is selected according to the x, y coordinates of the measured point. The distance of the point above the ground, $h_{m(k+1)}$, as well as the measurement uncertainty, σ^2_z, is used to update the elevation estimate of the corresponding voxel in the column above the cell. We use n separate voxel Kalman filters to estimate the elevation of each voxel. Then, we apply another Kalman filter to estimate the final elevation of each cell (cell Kalman filter). Only the highest estimated value, $h^{[i,j,n_{\max}]}$, from the voxels updated in the previous step is provided on the input of the cell Kalman filter which is used to estimate the elevation of the cell in the elevation map. Note that the Kalman estimator cannot be applied directly to each cell of the elevation map because the mobile robot cannot directly measure the height of the cell (Fig. 1(b)). A direct measurement is possible if the range sensor measuring the elevation is located high above the ground plane, e.g., on a flying robot. In this case the sensor is much higher than the measured cells, and this configuration allows us to assume that each measurement contains information about the highest object located in each cell.

In the mapping procedure, one-dimensional static Kalman filters are used to estimate the height of the voxels. Another Kalman filter is used to estimate the height of each cell. To transform the uncertainty from the camera frame to the map coordinate system, we use the procedure described in Section 4.2. The variances of the

height measurements $\sigma^2_{z_{(k+1)}}$ are used as weights in the voxel's Kalman filters. All measurements inside the voxel considered are used to update the hypothesis about the height of that voxel. The state vector and measurement are the height of the voxel and measured height, respectively. The state transition and projection models are identity mappings. We initialize the state of the filter with the height in the center of the voxel. We update the state estimates as follows:

$$K = \sigma^{2[i,j,n]}_{h_{(k)}} \left(\sigma^{2[i,j,n]}_{h_{(k)}} + \sigma^2_{z_{(k+1)}} \right)^{-1},$$
$$h^{[i,j,n]}_{(k+1)} = h^{[i,j,n]}_k - K \left(h^{[i,j,n]}_k - h_{m_{(k+1)}} \right), \quad (1)$$
$$\sigma^{2[i,j,n]}_{h_{(k+1)}} = (1-K)\sigma^{2[i,j,n]}_{h_{(k)}},$$

where K is the Kalman gain, $\sigma^{2[i,j,n]}_{h_{(k)}}$ is the height variance in the $[i,j,n]$ voxel at time k, $h_{m_{(k+1)}}$ is the height measurement for the $[i,j,n]$ voxel at time $k+1$ and $\sigma^2_{z_{(k+1)}}$ is the variance of this measurement. The new height measurement for the voxel is $h^{[i,j,n]}_{(k+1)}$ and the new value of the variance is $\sigma^{2[i,j,n]}_{h_{(k+1)}}$.

For all updated voxels in the column considered we choose the cell with the highest elevation value, $h^{[i,j,n_{\max}]}_{(k+1)}$. This cell has the vertical index $n_{\max}$, and a corresponding variance $\sigma^{2[i,j,n_{\max}]}_{h_{(k+1)}}$. The parameters of the selected voxel, $h^{[i,j]}_{p_{(k+1)}} = h^{[i,j,n_{\max}]}_{(k+1)}$ and $\sigma^{2[i,j]}_{p_{(k+1)}} = \sigma^{2[i,j,n_{\max}]}_{h_{(k+1)}}$, are used to compute the elevation and the uncertainty of the whole column (the cell of the elevation map). The state vector and measurements of the Kalman filter in each cell are the height of the cell, $h^{[i,j]}$, and measured height, $h^{[i,j]}_p$, respectively. As in the voxel filter, the state transition model and projection model are identity mappings. The initial height of the cell (the state of the filter) is zero. The elevation, $h^{[i,j]}_{(k+1)}$, and the uncertainty, $\sigma^{2[i,j]}_{h_{(k+1)}}$, are computed as follows:

$$K_k = \sigma^{2[i,j]}_{h_{(k)}} \left(\sigma^{2[i,j]}_{h_{(k)}} + \sigma^{2[i,j]}_{p_{(k+1)}} \right)^{-1},$$
$$h^{[i,j]}_{(k+1)} = h^{[i,j]}_k - K_k \left(h^{[i,j]}_k - h^{[i,j]}_{p_{(k+1)}} \right), \quad (2)$$
$$\sigma^{2[i,j]}_{h_{(k+1)}} = (1-K_k)\sigma^{2[i,j]}_{h_{(k)}}.$$

In (2), K_k is the Kalman gain computed for the column $[i,j]$, $\sigma^{2[i,j]}_{h_{(k)}}$ is the value of the variance for the column $[i,j]$ at time k, and $h^{[i,j]}_{(k)}$ is the elevation value for the column $[i,j]$ at time k. The new elevation of the column at time $k+1$ and the new variance at time $k+1$ are $h^{[i,j]}_{(k+1)}$ and $\sigma^{2[i,j]}_{h_{(k+1)}}$, respectively.

The procedure of voxel and grid cell update is presented in Fig. 3. Each voxel is updated using new measurements (Fig. 3(b)). The measurements are aggregated for each voxel using (1), as shown in Fig. 3(c).
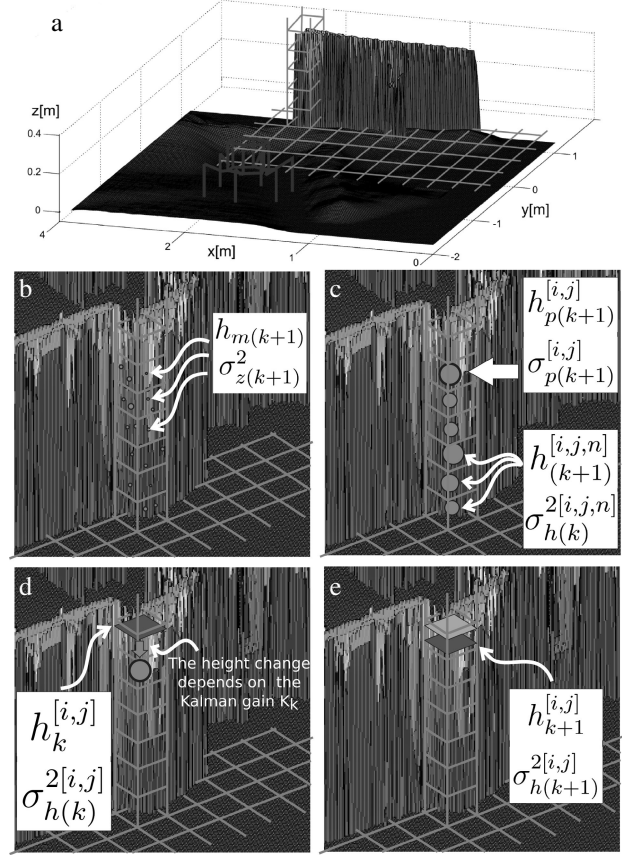

Fig. 3. Concept of the map update procedure: the map is divided into voxels (a), new measurements on the global map (b), the new measurements are aggregated for each voxel using Kalman filter (c), the highest value in the column is used to update the elevation map (d), (e).

Then, the highest value from the updated voxels for the whole column is selected (Fig. 3(d)), and used to compute the new height value in the global elevation map and the uncertainty value (Fig. 3(e)). Note that the map updating procedure might lead to height reduction of the cell (Fig. 3(e)).

We reduce the importance of isolated and old measurements using a measurements vitality counter (cf. Fig. 2). To consider the column a correct one, a minimal number of measurements in the specified time has to be collected. Whenever the sensor provides measurements of the values inside the column, the counter V for the column considered is increased by the value V_i. If the whole column is not updated during the map updating procedure, the counter is decreased by the value V_d. The starting and minimal value for each counter V is 0. If the counter value is bigger than a threshold V_t, then the counter is locked, the mapping procedure considers the column established, and the measurements are used to update the map. During the experiment we use $V_i = 3$, $V_d = 2$, $V_t = 7$. An example counter state during the map
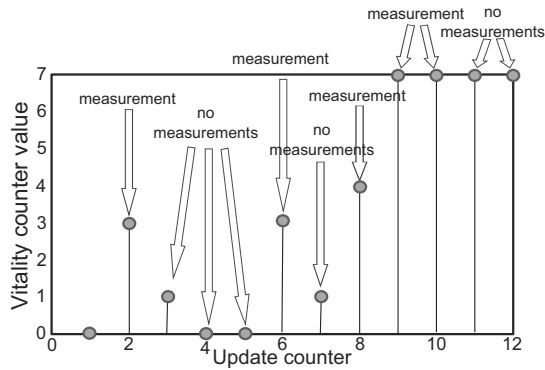
Fig. 4. Example updates of the vitality counter V with increase value $V_i = 3$, decrease value $V_d = 2$, and threshold value $V_t = 7$.

updating process is presented in Fig. 4.

3.2. Post-processing. To reduce the influence of artifacts introduced by the sensor noise, we check the visibility of each updated cell. The visibility checking procedure is performed after all the individual cells have been updated. The order of cell visitation is from the most distant to the nearest one. We assume that the cell which is updated has to be visible from the current pose of the sensor. Thus, if an existing cell $c_e^{[i,j]}$ occludes the updated cell, we assume that the existing cell contains erroneous measurements. To check the visibility constraint, we increase the height of the cells by their uncertainty, i.e., $h^{[i,j]} + 3\sigma_h^{[i,j]}$. Thus, the probability that the real height of the cell is below this threshold is 0.99. Moreover, the updated cell should be matured—the vitality counter value should be stabilized. If the vitality value of the updated cell is above the threshold, we remove the voxels from the occluding cell $c_e^{[i,j]}$ which are above the line of sight. This concept is presented in Fig. 5.

The result of the procedure on a real elevation map is presented in Fig. 6. Note that the update of a single cell might remove many existing and occluding cells. In practice, the criteria are very strict and most of the occluding obstacles are removed. We found this useful when the localization system was not precise enough. In the case of strong slippage which is not compensated by the localization system, the most recent measurements are the most accurate and they overwrite the old measurement.

Finally, median filtering is performed using a 2D sliding window on the elevation map. It allows removing small artifacts and fill-in small missing areas of the elevation grid.
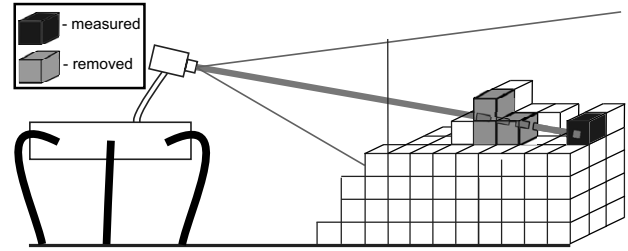

Fig. 5. Visibility checking concept: for each cell that was modified in the last data update step (black cell) the visibility constraint is checked. Cells that occlude the updated cell from the current viewpoint are removed (gray cells).
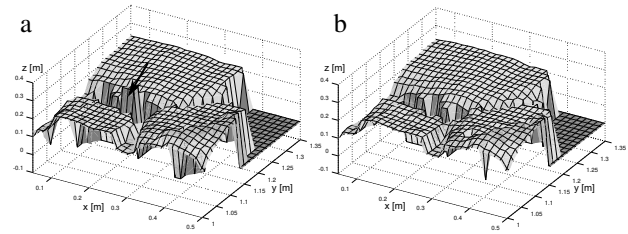

Fig. 6. Visibility checking procedure: the artifact indicated by the arrow (a) is removed in the next mapping iteration (b). The camera position is $\mathbf{p} = [0, 0, 0.8]^T$.

4. Terrain perception

The mapping method requires a point cloud as an input. In our research we use two type of sensors: a stereo camera and a PrimeSense-based device (Kinect, Carmine, Asus Xtion). These sensors have various properties, and thus we defined two separate filtering methods and uncertainty models.

4.1. Perception using a stereo camera. For outdoor experiments we prefer to work with the stereo camera. The passive camera provides data in various light conditions. We use the *Small Vision System* (SVS) software proposed by Konolige (1997). The library provides an implementation of the correlation-based stereo algorithm and returns a point cloud which corresponds to the configuration of the detected object. The SVS uses visual features to match pixels from the left and right images. Whenever the camera observes smooth surfaces without visual features, the SVS library does not return information about the distance to the obstacle. Even worse, if the camera observes objects with repeated patterns (texture), the returned data are erroneous. Fortunately, these types of objects are more common indoors. In outdoor natural environments, the obstacles are feature-rich and repeated patterns are typically rare. However, the data obtained from the passive stereo camera have do be filtered in order to remove artifacts.
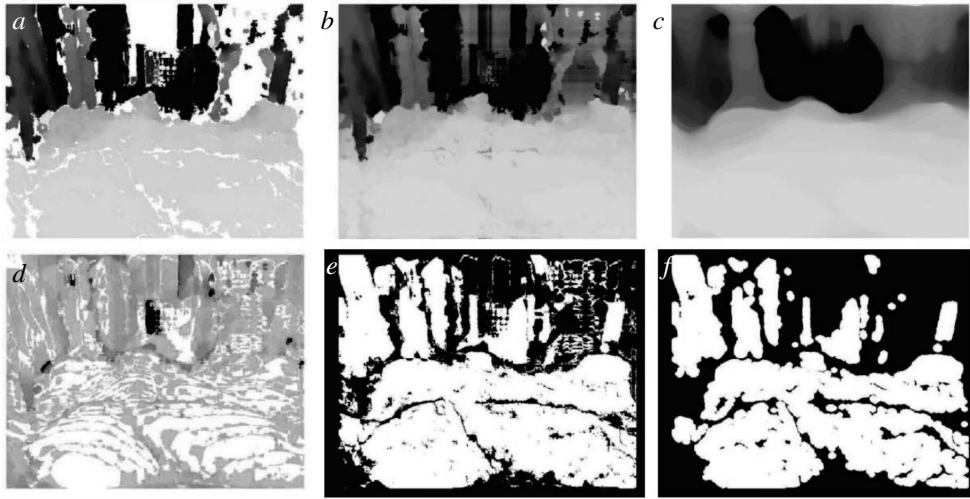
Fig. 8. Artifacts removal procedure: depth image (a), filled-in depth image (b), depth image after smoothing using a median filter (c), difference between a filtered image and the original (d), mask which defines pixels below the threshold (e), mask after morphological opening (f).
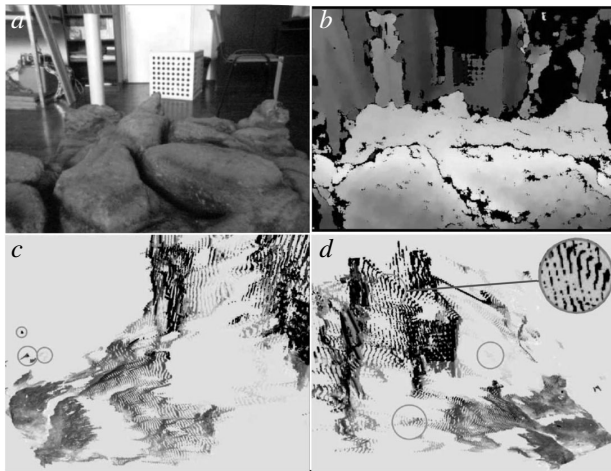

Fig. 7. Typical measurement errors: camera image (a), disparity image (b), erroneous measurements are the group of points on the reflective surfaces and out-of-horopter artifacts (marked with circles) (c), artifacts appearing on the edges of objects (marked with circles, enlarged part in the right-top corner) (d).

Because the filters embedded in the SVS do not remove all artifacts, we propose new methods which process the data obtained from the stereo camera. The artifacts which cannot be removed by the embedded filters occur whenever that of the object differs significantly from the disparity of the background (edges of the objects). As a result, halos which span the space between the edge and the background can be visible in the resulting point cloud (Fig. 7(d)). The second group of erroneous measurements consists of points on reflective surfaces and out-of-horopter artifacts. Such artifacts create a group of

points close to the current camera pose (Fig. 7(c)). We remove all of the mentioned erroneous measurements by analysis of the depth image (Fig. 7(b)). The image can be obtained by projecting the point cloud on the image surface, or directly, using the available disparity image values:

$$z = \frac{bf}{d}, \tag{3}$$

where z is the depth value, b is the baseline of the stereo camera (the distance between coordinate systems of the left and right image), f is the focal length, d is the disparity value.

The value of a pixel which does not have corresponding measurements (e.g., when the SVS did not find a correct correspondence between stereo images or the measurements are removed by the embedded filters) is 0, which is represented as a black background in the images.

To remove erroneous measurements, the empty regions in the images are filled in using the mean values of each row and column on the image (Fig. 8(b)). For each pixel $[u, v]^T$ the mean value is computed as follows:

$$a = \frac{\sum_{i=1}^{h} d(u,i) + \sum_{j=1}^{w} d(j,v)}{N_u + N_v}, \tag{4}$$

where w and h are the width and height of the image, respectively, N_u and N_v are the number of non-zero pixels in row u and column v of the depth image, $d(u,i)$ and $d(j,v)$ are the intensity values of the depth image.

After filling in the empty regions, the image is smoothed using a median filter. The window size of the filter is chosen to be 75 pixels. Image smoothing removes
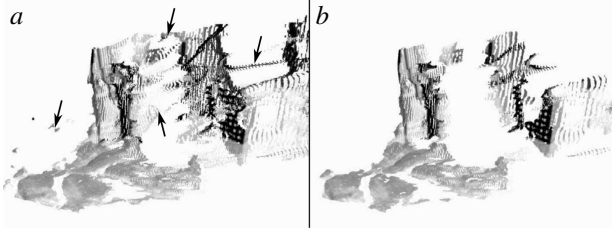
Fig. 9. Point cloud before (artifacts are indicated by the arrows) (a) and after the artifacts removal procedure (b).
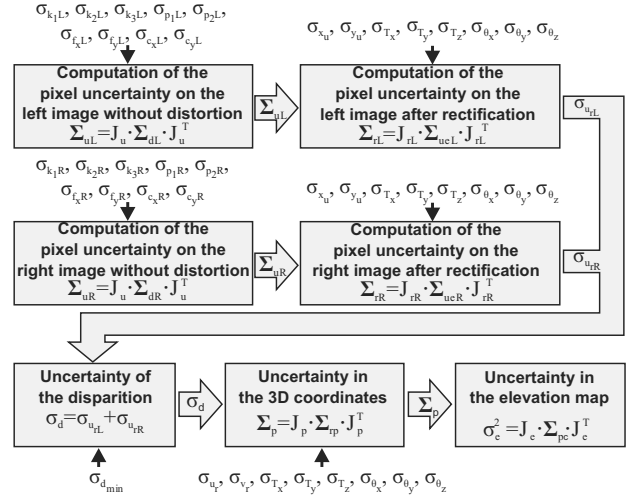

Fig. 10. Procedure of uncertainty computation for the stereo camera. Input standard deviations are located above the blocks. For each equation, the uncertainty on the right side is constructed from the uncertainty in the previous step and input standard deviations.

pixels which differ significantly from the background, which are typically erroneous pixels resulting from measurement artifacts (Fig. 8(c)). Then, the smoothed image is subtracted from the original image (Fig. 8(d)). We create a mask using the output image. The pixels which differ by less than 10% from the smoothed image are used to form the mask (Fig. 8(e)). The mask is used to remove artifacts. Occasionally, empty spaces in the image are filled in by pixels similar to those of the artifact. In this case the artifacts are similar to the background and are not removed by the procedure. In practice, such situations may arise for small groups of pixels, and they are removed by morphological openings (erosion and dilation) on the mask image (Fig. 8(f)) (Szeliski, 2011). When the mask is applied, most of the artifacts are removed from the original image. The point cloud before and after the artifact removing procedure is shown in Fig. 9.

4.2. Uncertainty model of stereo measurements.

The mapping procedure requires not only 3D measurements in the input but also their uncertainty Σ_p (Fig. 2). Because the information about the uncertainty of measurements is not returned by sensors used in this research, we present methods which allow us to determine Σ_p for two most popular depth sensors.

Analysis of the sensory data used for map building is difficult without knowledge of the underlying uncertainty of the measurements (Thrun *et al.*, 2005).

In the mapping scenario with a stereo camera as a sensor, the computation of uncertainty should be fast. The application of the robot in real environment missions requires real time operation of the mapping method. The uncertainty model should determine the influence of the camera resolution and calibration errors on the uncertainty of the coordinates of each point in the point cloud and finally the uncertainty of the cell of the elevation map. Matthies and Shafer (1987) proposed to use 3-dimensional Gaussian distributions to represent the measurement error of raster-based sensors (e.g., a visual camera). It was shown that the uncertainty model represented by the covariance matrix of 3D coordinates reduces the localization error in SLAM (Skrzypczyński, 2009). Sahabi and Basu (1996) presented a method to

compute the optimal configuration of the stereo camera, which decreases the error of the depth measurement. The optimization is possible because the uncertainty model of the camera is available.

In stereo triangulation methods, the main sources of measurement errors are related to the finite resolution of the image and wrong correspondences between the images. Because both correspondence and depth are determined on the basis of rectified images (while the rectification procedure uses the calibration parameters), the uncertainty of the calibration also influences that of the measurement. It is very difficult to analyze wrong correspondences due to their random nature, so this source of measurement errors was ignored. The remaining factors are analyzed using analytical methods (Łabecki and Skrzypczyński, 2013).

The procedure of uncertainty computation is presented in Fig. 10. First, we determine the relation between the position of the pixel in the image without distortion $[x_u, y_u]^T$ and the position of the pixel in the distorted image $[x_d, y_d]^T$:

$$[x_u, y_u]^T = f([x_d, y_d]^T). \tag{5}$$

The formula (5) is determined by the linearization of the distortion equations and depends on the distortion parameters k_1, k_2, k_3, p_1, p_2 and intrinsic camera parameters f_x, f_y, c_x, c_y found during the calibration procedure. Propagating the uncertainty, we can compute a covariance matrix on the image without distortion:

$$\boldsymbol{\Sigma}_{uL} = \mathbf{J}_u \boldsymbol{\Sigma}_{dL} \mathbf{J}_u^T, \tag{6}$$

where $\mathbf{J}_u$ is the Jacobian of (5) with partial derivatives with respect to (k_1, k_2, k_3, p_1, p_2, f_x, f_y, c_x, c_y), $\boldsymbol{\Sigma}_{dL}$ is the covariance matrix of distortion and intrinsic camera parameters.

Similarly, the uncertainty propagates through the rectification algorithm:

$$\boldsymbol{\Sigma}_{rL} = \mathbf{J}_{rL}\boldsymbol{\Sigma}_{ueL}\mathbf{J}_{rL}^T, \qquad (7)$$

where $\mathbf{J}_{rL}$ is the Jacobian of the rectification equation with respect to pixel coordinates in the image without distortion and extrinsic parameters of the camera (x_u, y_u, T_x, T_y, T_z, θ_x, θ_y, θ_z), $\boldsymbol{\Sigma}_{ueL}$ is the covariance matrix of the pixel's coordinates on the image without distortion and extrinsic parameters of the camera, $\boldsymbol{\Sigma}_{rL}$ is the covariance matrix of the pixel's coordinate on the rectified image.

Taking into account the uncertainty of the position of a pixel in the rectified images, the uncertainty of the disparity can be defined as follows:

$$\sigma_d = \sigma_{u_{rL}} + \sigma_{u_{rR}}, \qquad (8)$$

where $\sigma_{u_{rL}}$ and $\sigma_{u_{rR}}$ are standard deviations in the row u of the left and right image, respectively.

The covariance matrix of the 3D point $\mathbf{p} = [x_c, y_c, z_c]^T$ is computed as follows:

$$\boldsymbol{\Sigma}_p = \mathbf{J}_p\boldsymbol{\Sigma}_{rp}\mathbf{J}_p^T, \qquad (9)$$

where $\mathbf{J}_p$ is the Jacobian of $\mathbf{p}$ with respect to d, u_r, v_r and extrinsic parameters of the stereo camera, $\boldsymbol{\Sigma}_{rp}$ is the covariance matrix of the disparity d, the position of the pixel on the rectified image $[u_r, v_r]^T$, and extrinsic parameters of the stereo camera.

4.3. Uncertainty model of PrimeSense measurements. In our research, we also use PrimeSense-based sensors (Kinect and Asus Xtion). We employ these without any additional filtering procedures. The point cloud is used as the input of the mapping algorithm.

As with the stereo camera (Section 4.2), we apply a similar mathematical procedure to describe the uncertainty model of the PrimeSense sensor. The uncertainty model is based on the solution proposed by Park *et al.* (2012). First, the relation between the position of a pixel in the depth image and the 3D position of the point is defined:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = d \begin{bmatrix} \frac{1}{f_x} & 0 & -\frac{C_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{C_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \qquad (10)$$

where u and v represent the position of the pixel in the depth image and d is the depth measurement value. Variables f_x and f_y are the focal length parameters and C_x and C_y are optical axis parameters of the depth camera.
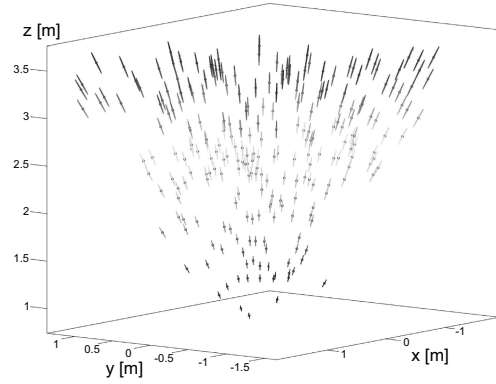

Fig. 11. Output from the uncertainty model of Kinect. Each random measured point is drawn with the uncertainty ellipsoid. The sensor is located at (0,0).

To transform the uncertainty from the camera frame to the map coordinate system, we use the procedure described in Section 4.2. Then, we can compute uncertainty of each point in the cloud:

$$\boldsymbol{\Sigma}_p = \mathbf{J}_p\boldsymbol{\Sigma}_{uvd}\mathbf{J}_p^T, \qquad (11)$$

where $\mathbf{J}_p$ is the Jacobian of (10) with respect to u, v and d, while $\boldsymbol{\Sigma}_{uvd}$ is the covariance matrix of image coordinates u, v and depth d. The values of the variances u and v are constant but the variance σ_d of depth d depends on the current depth measurement. We approximate the relation found experimentally by Khoshelham and Elberink (2012) using

$$\sigma_d = k_1 d^3 + k_2 d^2 + k_3 d + k_4, \qquad (12)$$

where $k_1, \ldots, k_4$ are constants. The obtained uncertainty model is presented in Fig. 11.

5. Results

5.1. Accuracy of the mapping procedure. The presented mapping method was used to build a map indoors in the laboratory as well as in an outdoor environment. The obstacles perceived by the system are typical for rough and unstructured environments.

A map of the rough terrain mockup created in the laboratory is shown in Fig. 12.

The stereo camera is mounted 0.8 m above the ground on a cart which can move forward. The sensor is tilted down by $35°$. Measurements are performed with a step of 0.1 m. We verified the method which removes artifacts from the stereo measurements. It is difficult to qualitatively assess the number of removed artifacts or wrong measurements. Thus, we compare the overall gain by computation of the difference between the map obtained in the experiment considered and the ground truth map. The influence of the method on the quality
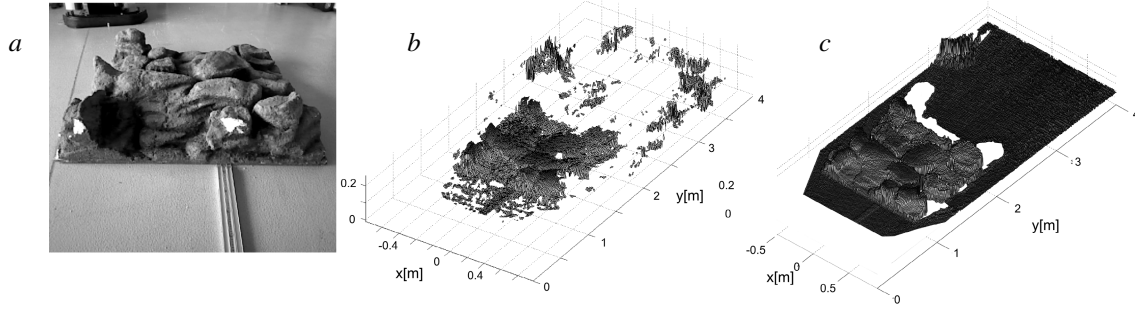
Fig. 12. Experiment with rough terrain mockup: RGB camera view (a), elevation map obtained using the stereo camera (b) and Asus Xtion (c). White cells of the map are not updated by the mapping procedure.
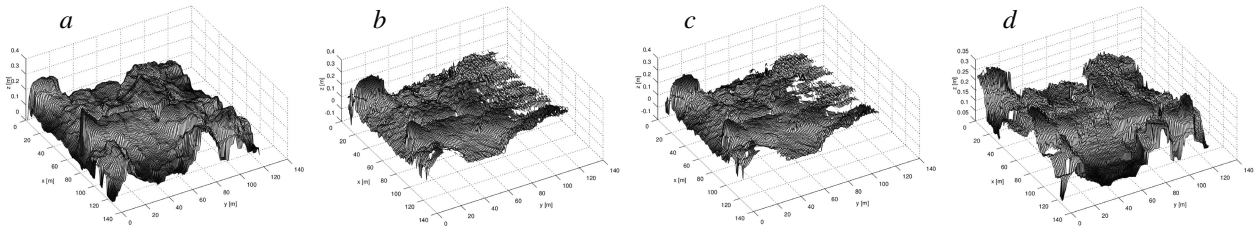

Fig. 13. Map of the rough terrain mockup: ground truth (a), map obtained from the stereo camera and the most conservative settings of the embedded filter (b), map obtained for the post-processing methods proposed in the paper, (c) map obtained for the Asus Xtion sensor (d).

of the map is measured using a modified version of the *performance index* (PI) (Ye and Borenstein, 2004):

$$\mathrm{PI} = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} o_f(i,j)\left(h_f(i,j)-h_t(i,j)\right)^2}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} o_m(i,j)\left(h_m(i,j)-h_t(i,j)\right)^2}\frac{N_{om}}{N_{of}}, \quad (13)$$

where $h_f(i,j)$ is the height of the cell i, j of the map which is built using filtered data from the stereo camera, $h_m(i,j)$ is the height of the cell i, j of the map which is built using raw data from the stereo camera, $h_t(i,j)$ is the height of the cell i, j of the ground truth map. The coefficients $o_f(i,j)$ and $o_m(i,j)$ are set to 1 if the height of $h_f(i,j)$ and $h_m(i,j)$ is measured by the sensor and to 0 if no data are available for that cell. N_{om} and N_{of} are the numbers of measured cells in raw and filtered maps, respectively. The PI value is the quotient of the mean squared error (MSE) of the two mapping methods. To obtain the ground truth map, the Hokuyo laser scanner was attached to an industrial robot arm and moved horizontally above the mockup (in x and y directions).

The resulting maps of the terrain mockup obtained by different variants of the elevation mapping method are presented in Fig. 13. The PI value for the map which is built from the stereo data obtained from the mapping method which uses the proposed data filtering procedures

is 0.91. The MSE is reduced by 9%. The PI is 0.99 for the map obtained from stereo data filtered by the embedded SVS filters with default settings. This map contains empty regions which are not visible in the map, built using the most tolerant settings of the embedded filters. This means that the embedded filters improve the mapping quality by 1% in comparison with the map built from a raw point cloud, but they remove useful data.

We also compared the use of a stereo camera (STOC) with an Asus Xtion sensor in the mapping procedure. The PI value computed for the map using Asus Xtion (Fig. 13(d)) and the stereo camera with the proposed data filtering method (Fig. 13(c)) is 0.59. This means that the MSE is reduced by 41% when Asus Xtion is used, from which we conclude that it provides much more accurate data than the stereo camera. Moreover, Asus Xtion provides depth data of regions which are plain and textureless. The floor is easily visible on the map created from the Asus Xtion data (Fig. 12(c)). The only invisible regions are caused by occlusions. On the map from the stereo camera (Fig. 12(b)) only the mockup is visible.

5.2. Influence of the localization error. A map obtained from experiments with the stereo camera mounted on a robotic manipulator is presented in Fig. 14. The robotic arm provides accurate localization of the sensor, thus we only investigate the properties of the mapping method. The maps presented in Fig. 14(a)–(f)
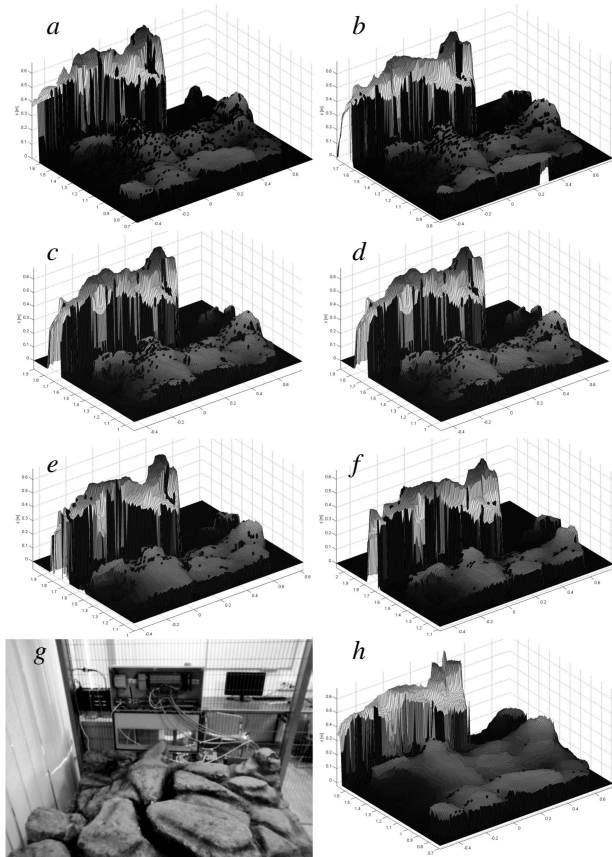
Fig. 14. Map obtained in the experiments with the stereo camera
mounted on the robotic arm. Subfigures (a)–(f) show
the maps obtained for individual snapshots, the camera
view (g), and the final map (h).

Fig. 15. Indoor experiment on the Messor 2 robot: environment
setup (a), elevation map obtained using the stereo cam-
era (b).

are obtained from the consecutive snapshots. The distance
between scans is 0.1 m. Each map in Fig. 14(a)–(f)
is created using individual snapshots. The final map
created by the mapping method proposed in this paper
is presented in Fig. 14(h). The empty regions which are
visible in Fig. 14(a)–(f) are filled in in the final map. The
main problems are connected to deeper concavities on the
mockup, which are occluded by higher obstacles. The
camera does not provide data about these regions, or data
are incorrect due to artifacts which are not removed by the
filters.

To check the influence of the localization error on
the mapping method, we computed the performance index
for two maps created using the proposed filtering and
mapping methods. The first map is created on the robotic
arm, which provides precise localization of the camera
(Fig. 14(h)), and the second is created using the rail
(Fig. 13(c)). The computed PI value is 0.86, which
means that with the more precise localization system, we
were able to reduce the mapping error by 14%. This
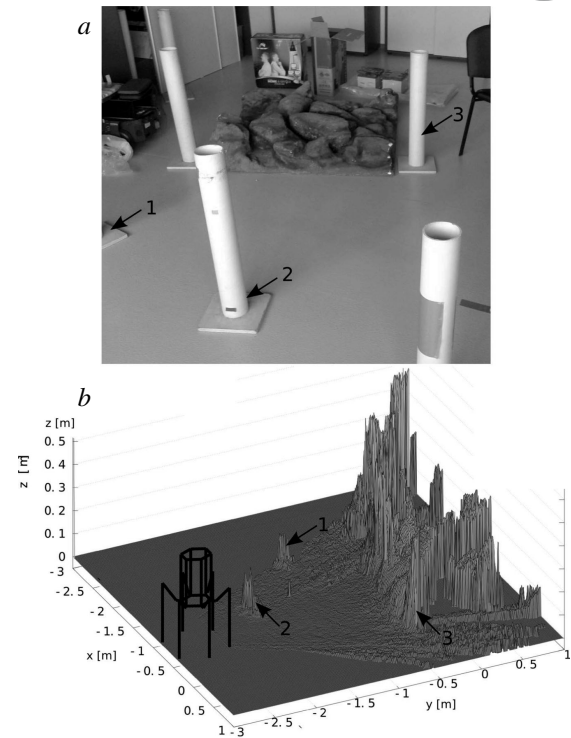shows that the proposed mapping method requires precise

localization to work properly. For this purpose, parallel
tracking and mapping method (Belter and Skrzypczyński,
2013) or RGB-D SLAM (Belter *et al.*, 2015) can be
used. Our experience shows that the localization error of
these methods is below 2 cm for small workspaces with a
sufficient number of visual features.

The results which show the quantitative improvement
of the obtained map are summarized in Table 1.

Table 1. Influence of various components in the mapping
method.

Method 1	Method 2	Error reduction
Artifacts removal	Raw SVS data	9%
Asus Xtion	Stereo camera	41%
Improved localization	Sensor on a rail	14%

5.3. Experiments on a real robot. In the first
experiment on a real robot, we conducted trials with the
six-legged Messor robot (Belter and Walas, 2014) with an
on-board stereo camera. The robot updates the map at the
end of each 10 cm step. During the experiment the robot
made 10 steps. The experimental setup and the obtained
map are presented in Fig. 15. The map contains only
partial information on the environment due to the limited

Fig. 16. Experimental setup with the quadrupedal robot StarlETH (a), laser scanner Leica Nova MS50 MultiStation (`leica-geosystems.com`) used for ground truth recording (b), overview of the experimental terrain (c).

camera viewing angle. While the robot has information about obstacles which are in front of it, obstacles which are sideways are not visible on the map. In this experiment we do not have a ground truth elevation map, and thus it is not possible to compute the mapping accuracy. However, the map contains a representation of the obstacles from the real environment (marked by the arrows in Fig. 15(b)).

In the second experiment we created an unstructured environment of ca. 4×3 m and traversed it with the quadrupedal robot StarlETH (Hutter *et al.*, 2012) (Fig. 16(a)). The robot is commanded to walk from one end to the other, turn around, and walk back to the starting point (Fig. 16(c)).

The terrain is measured and mapped continuously as the robot walks with a PrimeSense Carmine 1.09 (short range) structured light sensor which is attached at the front of the robot. Localization is limited to on-board proprioceptive sensing, and consists of the fusion of legged kinematics and inertial measurements (Bloesch *et al.*, 2013). This state estimation is prone to drift in position and yaw orientation. Comparison with motion capture data shows that the on-board state estimator accumulated a position error of 0.3 m and a yaw angle error of $20°$ for the base of the robot after the entire run. For evaluation, we recorded ground truth measurements with the Leica Nova MS50 MultiStation laser scanner (Fig. 16(b)). The terrain was scanned from three different perspectives and aligned with the built-in tools of the device.

The mapping error in this experiment is mainly caused by the localization method. The resulting maps differ significantly because of the drift of the localization

method. This can be observed, for example, with the board on the ground truth map indicated by the arrow number 4 in Fig. 17(a) being shifted by about 1 m on the obtained maps (Fig. 17(b) and (c)). For the maps obtained from this experiment, we computed the performance index to compare the one presented in this paper with the method proposed by Fankhauser *et al.* (2014).[1] The obtained value is 1.0308, which means that the MSE is bigger by 3% when our method is used. The obtained maps are similar and the difference in the PI value is insignificant.

5.4. Comparison. We compare our method with the one proposed by Fankhauser *et al.* (2014) because it is the most similar state of the art mapping method with respect to application. There is no fair quantitative comparison with very precise methods known from computer graphics, which are computationally expensive (Berger *et al.*, 2014; Dey *et al.*, 2012) or require user input for surface reconstruction (Sharf *et al.*, 2007). They cannot be applied as a mapping method for a real robot, which has limited computational resources.

Since the quantitative comparison between the presented mapping method and the one proposed by Fankhauser *et al.* (2014) does not provide clear conclusions, we provide a qualitative comparison and discussion. The first difference between these two methods is indicated by the arrow 1 in Fig. 17(b) and (c). The mapping method presented in the paper is supported by the technique which fills in empty areas in the map. In this case, we use a median filter, which is fast and efficient. To fill in larger areas, more advanced methods based on Gaussian regression can be used (Belter *et al.*, 2012). The map created by the mapping method presented in this paper contains 13730 updated cells, while the compared mapping technique contains 13403 cells. The method which fills in empty areas can nicely reconstruct the parts which are not visible to the robot due to occlusions.

The proposed updating procedure, which is based on the Kalman filter and the column of voxels, results in more realistic shapes of the obstacles on the obtained map. The obstacles indicated by the arrow 2 in Fig. 17(c) have sharp edges which are close to the shape of the real objects (cf. Fig. 16(c)). The obstacles in Fig. 17(b) have filled edges.

The last properties are related to the precision of robot localization. The mapping method proposed by Fankhauser *et al.* (2014) takes into account the uncertainty of the robot's pose. This results in generally smoother maps, as indicated with the area 5 in Fig. 17(b) and (c). Another property is that new data play a more important role when the map is updated. Due to the localization

---

[1]For this comparison we only take into account the measurement and model update steps and neglect the map fusion process of the algorithm introduced by Fankhauser *et al.* (2014).
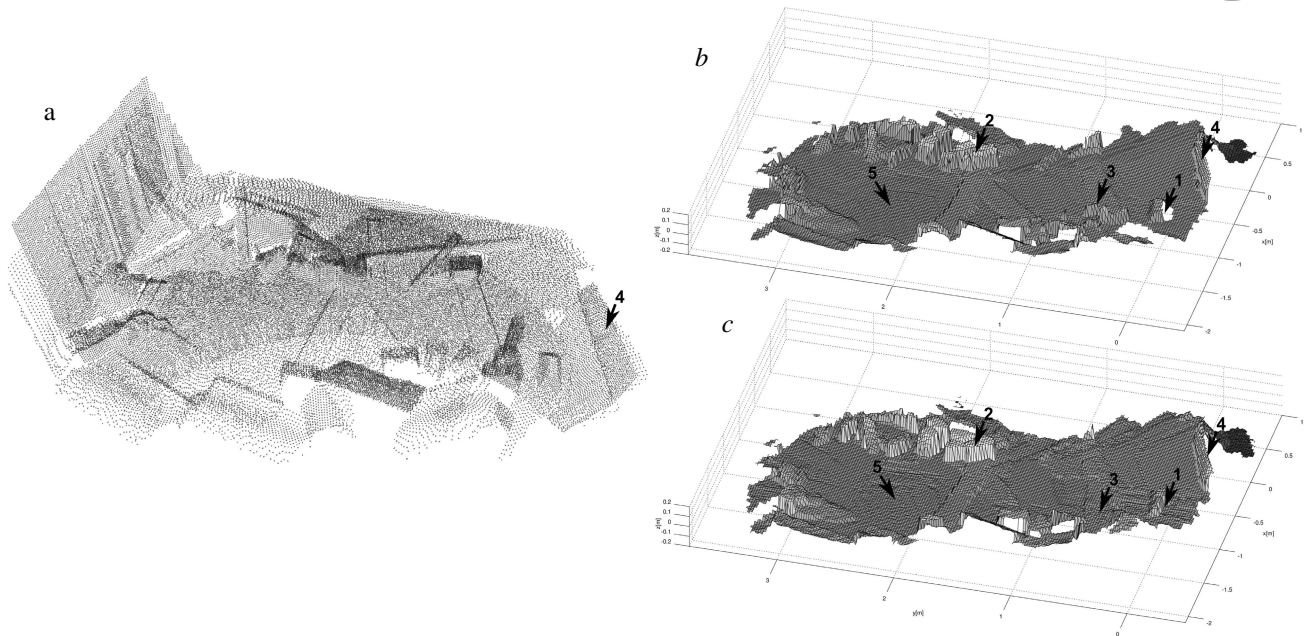
Fig. 17. Maps obtained on the StarlETH robot: ground truth map (a), map obtained using the method proposed by Fankhauser *et al.* (2014) (b), map obtained using the method presented in the paper (c).

error, the cells which were updated recently are treated as more accurate than old cells on the global map. This is advantageous as obstacles might shift in the map from the robot's perspective, and the previously gathered information needs to be updated to reflect the newly acquired data. In this experiment, parts of the terrain are observed twice: first when the robot walks in one direction, and the second time when the robot returns to its original position. As an example, the obstacle indicated by the arrow 3 on the map in Fig. 17(b) is observed at this location only as the robot passes it a second time. While the method by Fankhauser *et al.* (2014) reflects these changes, our technique (Fig. 17(c)) does not capture this obstacle. This limitation is caused by the assumption that the environment is static and the localization error is small. On the other hand, this behavior is advantageous when the localization error is significant. In this case the local map used for motion planning (Belter and Skrzypczyński, 2011a) is more accurate, despite the fact that some obstacles vanish from the global map.

Table 2. Results of quantitative comparison.

Method	Our approach	NDT-OM	Octomap
MSE [m^2]	0.063	0.127	0.524

We also compared the mapping method presented in the paper to NDT-OM (Saarinen *et al.*, 2013) and Octomap (Hornung *et al.*, 2013). We verified all three methods on the ICL-NUIM 'living room' dataset (Handa *et al.*, 2014). The dataset contains a synthetic sequence of depth and RGB images, with and without noise. It allows the creation of a perfect ground truth map for comparison. The noisy dataset is used for testing. Because our method is designed to estimate the surface of the terrain and cannot handle multi-level maps, we removed points which represent the ceiling in the dataset. Because each compared method has its own environment representation, the environment models from Octomap and NDT-OM are transformed to the elevation map and then compared with the ground truth map. We compare the elevation map obtained from a noisy dataset with the ground truth map created for the same method using a noise-free dataset.

For each method we compute the mean squared error between the ground truth map and the map obtained for noisy data. With the default configuration of mapping systems, the MSE for our method is 0.074 m^2, while for NDT-OM and Octomap the error is bigger than 1 m^2. NDT-OM and Octomap are designed to be general and store all 3D noisy measurements in the map structure. When the 3D representation is transformed to 2.5D, the voxels which contain noisy data disturb the structure of the 2.5D maps. Thus, we remove all uncertain voxels from the maps. As a threshold we use $\sigma_h^{2[i,j]}$ in our method and the occupancy value for NDT-OM and Octomap. The obtained results are presented in Table 2. Our method still performs the best of all compared methods. The MSE for our mapping method is 50% smaller than the error for NDT-OM. Also, the obtained map presented in Fig. 18(b) contains the least number of artifacts introduced by noisy data. Note that in these experiments we do not
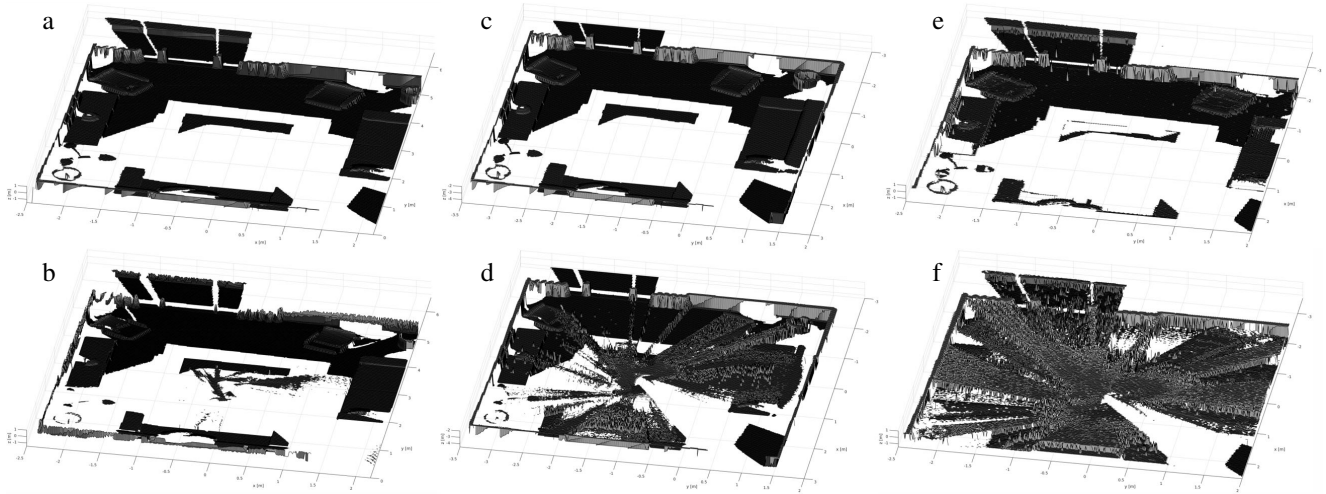
Fig. 18. Maps obtained for the ICL-NUIM 'living room' dataset using the proposed approach (a), (b), NDT-OM (c), (d) and Octomap (e), (f); first row ((a), (c), (e)) presents groundtruth maps obtained for a dataset without noise, second row ((b), (d), (f)) presents maps obtained for a dataset with noise.

use the artifact removal procedure presented in this paper. We compare mapping methods only and the same noisy sequence of point clouds is provided to the input of each of the mapping methods.

6. Conclusions

This article presents a mapping method for walking robots based on 3D depth sensors. The main contributions of this work are as follows:

- a real-time mapping method with a unique updating procedure which uses the Kalman filter for cell height estimation and a stack of voxels to accumulate measurements;

- a set of procedures which remove erroneous measurements and cells from the map (the vitality counter and the visibility checking procedure);

- a data filtering procedure which improves the quality of point clouds obtained from the stereo camera;

- an uncertainty model for the stereo camera and PrimeSense-based measurements; we show how the uncertainty of the measurement propagates to the uncertainty of the cell height;

- extended experimental verification of mapping, which demonstrates the importance of the particular properties of the mapping method with respect to the elevation map quality, as well as the role of the selected sub-procedures.

In this article we also provide the results of the experiments on two different walking robots. We compare the presented method with that recently proposed by

Fankhauser *et al.* (2014) and discuss the differences between these two methods. We point out the advantages of the method on the maps obtained in the experiment with the real robot and also show its limitations.

In the future we are going to take into account the uncertainty of the robot's pose in the updating procedure to reduce the influence of localization drift on the map quality. This will be helpful in assessing the quality of data in situations where the robot has not observed a region and has moved away from/over it. We are also going to integrate our mapping method with the terrain classification algorithm (Walas and Nowicki, 2014) to create an object-aware navigation system for legged robots.

References

Belter, D., Łabecki, P. and Skrzypczyński, P. (2012). Estimating terrain elevation maps from sparse and uncertain multi-sensor data, *IEEE 2012 International Conference on Robotics and Biomimetics, Guangzhou, China*, pp. 715–722.

Belter, D., Łabecki, P. and Skrzypczyński, P. (n.d.). Adaptive motion planning for autonomous rough terrain traversal with a walking robot, *Journal of Field Robotics*, (in print).

Belter, D., Nowicki, M., Skrzypczyński, P., Walas, K. and Wietrzykowski, J. (2015). Lightweight RGB-D SLAM system for search and rescue robots, in M.K.R. Szewczyk and C. Zieliński (Eds.), *Recent Advances in Automation, Robotics and Measuring Techniques*, Advances in Intelligent Systems and Computing, Vol. 351, Springer, Cham, pp. 11–21.

Belter, D. and Skrzypczyński, P. (2011a). Integrated motion planning for a hexapod robot walking on rough terrain, *18th IFAC World Congress, Milan, Italy*, pp. 6918–6923.

Belter, D. and Skrzypczyński, P. (2011b). Rough terrain mapping and classification for foothold selection in a walking robot, *Journal of Field Robotics* **28**(4): 497–528.

Belter, D. and Skrzypczyński, P. (2013). Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping, *Industrial Robot: An International Journal* **40**(3): 229–237.

Belter, D. and Walas, K. (2014). A compact walking robot—flexible research and development platform, in M.K.R. Szewczyk and C. Zieliński (Eds.), *Recent Advances in Automation, Robotics and Measuring Techniques*, Advances in Intelligent Systems and Computing, Vol. 267, Springer, Cham, pp. 343–352.

Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A. and Silva, C. (2014). State of the art in surface reconstruction from point clouds, in S. Lefebvre and M. Spagnuolo (Eds.), *Eurographics 2014—State of the Art Reports*, The Eurographics Association, Geneve.

Bloesch, M., Gehring, C., Fankhauser, P., Hutter, M., Hoepflinger, M.A. and Siegwart, R. (2013). State estimation for legged robots on unstable and slippery terrain, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, pp. 6058–6064.

Dey, T.K., Ge, X., Que, Q., Safa, I., Wang, L. and Wang, Y. (2012). Feature-preserving reconstruction of singular surfaces, *Computer Graphics Forum* **31**(5): 1787–1796.

Dryanovski, I., Morris, W. and Xiao, J. (2010). Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan*, pp. 1553–1559.

Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M. and Siegwart, R. (2014). Robot-centric elevation mapping with uncertainty estimates, *International Conference on Climbing and Walking Robots (CLAWAR), Poznań, Poland*, pp. 433–440.

Handa, A., Whelan, T., McDonald, J. and Davison, A. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM, *IEEE International Conference on Robotics and Automation, ICRA, Hong Kong, China*, pp. 1524–1531.

Hebert, M., Caillas, C., Krotkov, E. and Kweon, I. (1989). Terrain mapping for a roving planetary explorer, *Proceedings of the IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA*, pp. 997–1002.

Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C. and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots* **34**(3): 189–206.

Hutter, M., Gehring, C., Bloesch, M., Hoepflinger, M.A., Remy, C.D. and Siegwart, R. (2012). StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion, *International Conference on Climbing and Walking Robots (CLAWAR), Baltimore, MD, USA*, pp. 483–490.

Kleiner, A. and Dornhege, C. (2007). Real-time localization and elevation mapping within urban search and rescue scenarios, *Journal of Field Robotics* **24**(8–9): 723–745.

Khoshelham, K. and Elberink, S. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications, *Sensors* **12**(2): 1437–1454.

Kolter, J., Kim, Y. and Ng, A. (2009). Stereo vision and terrain modeling for quadruped robots, *Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan*, pp. 1557–1564.

Konolige, K. (1997). Small vision systems: Hardware and implementation, *8th International Symposium on Robotics Research, Monterey, CA, USA*, pp. 111–116.

Kweon, I. and Kanade, T. (1992). High-resolution terrain map from multiple sensor data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2): 278–292.

Łabecki, P. and Belter, D. (2014). RGB-D based mapping method for a legged robot, in C.Z.K. Tchoń (Ed.), *Zeszyty Naukowe Politechniki Warszawskiej*, Warsaw University of Technology Press, Warsaw, pp. 297–306, (in Polish).

Łabecki, P. and Skrzypczyński, P. (2013). Spatial uncertainty assessment in visual terrain perception for a mobile robot, in J. Korbicz and M. Kowal (Eds.), *Intelligent Systems in Technical and Medical Diagnostics*, Advances in Intelligent Systems and Computing, Vol. 230, Springer-Verlag, Berlin, pp. 357–368.

Matthies, L. and Shafer, S. (1987). Error modeling in stereo navigation, *International Journal of Robotics and Automation* **3**(3): 239–248.

Nowicki, M. and Skrzypczyński, P. (2013). Combining photometric and depth data for lightweight and robust visual odometry, *European Conference on Mobile Robots*, Barcelona, Spain, pp. 125–130.

Park, J.-H., Shin, Y.-D., Bae, J.-H. and Baeg, M.-H. (2012). Spatial uncertainty model for visual features using a kinect sensor, *Sensors* **12**(7): 8640–8662.

Pfaff, P., Triebel, R. and Burgard, W. (2007). An efficient extension to elevation maps for outdoor terrain mapping and loop closing, *International Journal of Robotics Research* **26**(2): 217–230.

Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N. and Burgard, W. (2008). Learning predictive terrain models for legged robot locomotion, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*, pp. 3545–3552.

Poppinga, J., Birk, A. and Pathak, K. (2010). A characterization of 3D sensors for response robots, *in* J. Baltes *et al.* (Eds.), *RoboCup 2009*, Lecture Notes in Artificial Intelligence, Vol. 5949, Springer, Berlin, pp. 264–275.

Rusu, R., Sundaresan, A., Morisset, B., Hauser, K., Agrawal, M., Latombe, J.-C. and Beetz, M. (2009). Leaving flatland: Efficient real-time three-dimensional perception and motion planning, *Journal of Field Robotics* **26**(10): 841–862.

Saarinen, J., Andreasson, H., Stoyanov, T. and Lilienthal, A.J. (2013). 3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments, *International Journal of Robotics Research* **32**(14): 1627–1644.

Sahabi, H. and Basu, A. (1996). Analysis of error in depth perception with vergence and spatially varying sensing, *Computer Vision and Image Understanding* **63**(3): 447–461.

Sharf, A., Lewiner, T., Shklarski, G., Toledo, S. and Cohen-Or, D. (2007). Interactive topology-aware surface reconstruction, *ACM Transactions on Graphics* **26**(3), Article No. 43.

Skrzypczyński, P. (2007). Spatial uncertainty management for simultaneous localization and mapping, *Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy*, pp. 4050–4055.

Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach, *International Journal of Applied Mathematics and Computer Science* **19**(4): 575–588, DOI: 10.2478/v10006-009-0045-z.

Stelzer, A., Hirschmuller, H. and Gorner, M. (2012). Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain, *International Journal of Robotics Research* **31**(4): 381–402.

Szeliski, R. (2011). *Computer Vision, Algorithms and Applications*, Springer, London.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, Cambridge, MA.

Walas, K. and Belter, D. (2011). Supporting locomotive functions of a six-legged walking robot, *International Journal of Applied Mathematics and Computer Science* **21**(2): 363–377, DOI: 10.2478/v10006-011-0027-9.

Walas, K. and Nowicki, M. (2014). Terrain classification using Laser Range Finder, *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA*, pp. 5003–5009.

Ye, C. and Borenstein, J. (2004). A novel filter for terrain mapping with laser rangefinders, *IEEE Transactions on Robotics and Automation* **20**(5): 913–921.

Yoon, S., Hyung, S., Lee, M., Roh, K., Ahn, S., Geeb, A., Bunnunb, P., Calwayb, A. and Mayol-Cuevas, W. (2013). Real-time 3D simultaneous localization and map-building for a dynamic walking humanoid robot, *Advanced Robotics* **27**(10): 759–772.

Zucker, M., Bagnell, J., Atkeson, C. and Kuffner, J. (2010). An optimization approach to rough terrain locomotion, *IEEE International Conference on Robotics and Automation, Anchorage, AK, USA*, pp. 3589–3595.

Dominik Belter graduated from the Poznań University of Technology (2007). He received the Ph.D. degree in robotics from the same university in 2012. Since 2012 he has been an assistant professor at the Institute of Control and Information Engineering (ICIE) there and a member the Mobile and Walking Robots Team at the ICIE. Dominik Belter is the author or a co-author of over 40 technical papers in the field of robotics and computer science. His research interests include the control of walking robots, machine learning, and soft computing.

Przemysław Łabecki received the M.Sc. and Ph.D. degrees in robotics from the Poznań University of Technology in 2008 and 2014, respectively. During his studies, he focused on environment modelling and mapping methods for legged robots. His research interests include software engineering, image processing and computer vision.
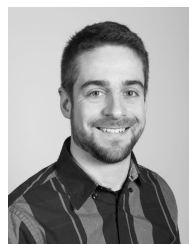
Péter Fankhauser has been a Ph.D. student at the Autonomous Systems Lab (ASL) at ETH Zurich since 2013. His current work focuses on mapping and motion planning for legged robots in a rough terrain. He received his B.Sc. in mechanical engineering and his M.Sc. in robotics, systems and control from ETH Zurich in 2010 and 2012, respectively. During his studies, he worked on system modeling and control of UAVs and led the development of Rezero, a robot that drives on a single ball while interacting with people.

Roland Siegwart is the founding co-director of Wyss Zurich and a professor of autonomous mobile robots at ETH Zurich. He studied mechanical engineering at ETH, spent ten years as a professor at EPFL (1996–06), was a vice president of ETH Zurich (2010–14), and held visiting positions at Stanford University and NASA Ames. He has been the coordinator of multiple European projects and a co-founder of half a dozen spin-off companies. He is a recipient of the IEEE RAS Inaba Technical Award, an IEEE fellow and an officer of the International Federation of Robotics Research (IFRR). He is on the editorial board of multiple journals on robotics and a general chair of several conferences in robotics, including *IROS 2002, AIM 2007, FSR 2007* and *ISRR 2009*. His interests are in the design and navigation of wheeled, walking and flying robots operating in complex and highly dynamical environments.