


# Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization

**Journal Article****Author(s):**

Fankhauser, Péter; Bloesch, Michael; [Hutter, Marco](#) 

**Publication date:**

2018-10

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000272110>

**Rights / license:**

[In Copyright - Non-Commercial Use Permitted](#)

**Originally published in:**

IEEE Robotics and Automation Letters 3(4), <https://doi.org/10.1109/lra.2018.2849506>

# Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization

Péter Fankhauser<sup>1</sup>, Michael Bloesch<sup>2</sup>, and Marco Hutter<sup>1</sup>

**Abstract**—Mobile robots build on accurate, real-time mapping with onboard range sensors to achieve autonomous navigation over rough terrain. Existing approaches often rely on absolute localization based on tracking of external geometric or visual features. To circumvent the reliability-issues of these approaches, we propose a novel terrain mapping method which bases on proprioceptive localization from kinematic and inertial measurements only. The proposed method incorporates the drift and uncertainties of the state estimation and a noise model of the distance sensor. It yields a probabilistic terrain estimate as a grid-based elevation map including upper and lower confidence bounds. We demonstrate the effectiveness of our approach with simulated datasets and real-world experiments for real-time terrain mapping with legged robots and compare the terrain reconstruction to ground truth reference maps.

**Index Terms**—Mapping, Field Robots, Legged Robots

## I. INTRODUCTION

UNMANNED ground vehicles (UGVs) require knowledge of the surrounding to safely and efficiently navigate through an environment. In rough terrain, an accurate terrain model is essential for a robot to plan a motion over and around obstacles. Equipped with onboard range sensors (e.g., laser range, time-of-flight, and stereo camera sensors), a robot can collect distance measurements and perceive the geometry of its surrounding terrain. Because these vehicles are moving systems, the sweeping motion of the range sensor over the terrain and the associated measurement errors must be carefully accounted for [1].

Much work has focused on obtaining a *globally*-consistent map [2], where an accurate absolute localization of a system is presumed. These methods are often sensitive to environmental conditions such as sufficient lighting, the availability of visual or geometric features, or external signals such as GPS. To relieve us from these constraints, we present a novel approach for terrain mapping independent of a global localization method. In our approach, the mapping relies on the *relative* localization based on proprioceptive sensing (kinematic and inertial measurements) such as wheeled or legged odometry. In previous work [3], we have shown that using proprioceptive sensing only, the position and yaw-rotation of a mobile robot

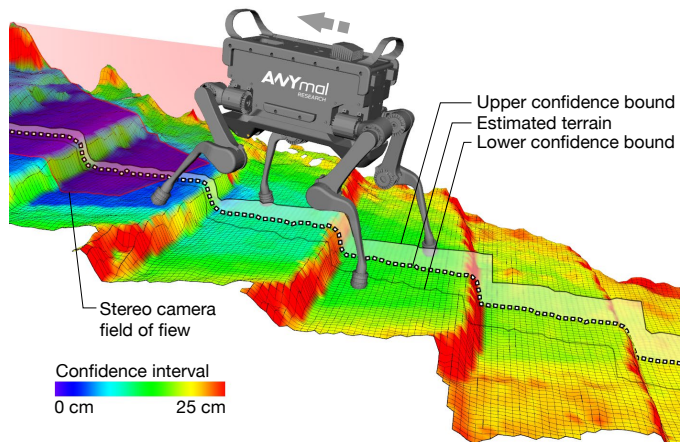


Fig. 1. The presented robot-centric mapping framework enables mobile robots to create consistent elevation maps of the terrain. Measurements from onboard range sensors are fused with the proprioceptive state estimation while taking its drift of pose into account. The mapping method estimates the terrain profile including a lower and upper confidence bound. This figure shows a snapshot of the quadrupedal robot ANYmal [4] using the generated map to climb a set of stairs [5]. The map is generated in real-time based on depth measurements from a forward-facing stereo camera. A video is available at <https://youtu.be/vSveQrJLRTo>.

remain unobservable. This means that the estimates of the position and yaw-rotation drift over time in comparison to their real values.

In a classical, *world-centric* mapping approach, the terrain map is associated with an inertial frame. Depending on whether uncertainties are considered or not, the use of drifting state estimation in a world-centric formulation will inevitably lead to either blurry or inconsistent maps. To account for the pose estimate drift probabilistically, we introduce a formulation which is expressed in a *robot-centric* formulation. In this formulation, the terrain map is associated with the current pose of the robot. At any time, the robot-centric elevation map is a local representation of the surrounding terrain, meaning that the observed regions in front of the robot have the highest accuracy, while ‘older’, previously seen parts of the map accumulate uncertainty and ‘melt’ into each other (see Fig. 1).

## II. RELATED WORK

In the context of rough terrain locomotion, many approaches (e.g., [6, 7, 8, 9]) rely on a grid-based, 2.5D height map representation of the terrain, where each cell in the map represents the terrain height at that position. While this type of map comes with certain limitations when compared to full 3D maps, its simple structure and better scalability allow for efficient data access and processing. Early work on the generation of elevation maps for mobile robots was presented

Manuscript received: February, 23, 2018; Revised May, 9, 2018; Accepted May, 31, 2018. This paper was recommended for publication by Editor Cyril Stachniss upon evaluation of the Associate Editor and Reviewers’ comments. This work has been conducted as part of ANYmal Research, a community to advance legged robotics, and was supported by the National Centre of Competence in Research Robotics.

<sup>1</sup>Péter Fankhauser and Marco Hutter are with the Robotics Systems Lab (RSL), ETH Zurich, Switzerland, peterfankhauser@me.com, mahutter@ethz.ch

<sup>2</sup>Michael Bloesch is with the Dyson Robotics Lab, Imperial College London, United Kingdom, bloesch.michael@gmail.com

by [10, 11]. Both authors used feature matching algorithms to find the corresponding transformation between multiple scans to build a composite elevation map, but they did not address the issues of error propagation as a result of inaccuracies during matching. In the approaches presented by [6, 12], a local terrain map is used that surrounds the robot and is moved along with its motion. While we use a similar setup, their approach relies on an accurate and reliable pose tracking algorithm and does not address the issue of a drifting pose estimation. Our approach is similar to the work by [13], where the elevation map is updated based on the motion of the robot. In their work, 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 in-plane uncertainty into account.

This paper contributes with a probabilistic, local terrain mapping approach for a mobile robot with uncertain localization from proprioceptive sensing. A key aspect of our work is how the local uncertainty and neighboring cells are processed to estimate an upper and lower bound of the resulting terrain map. Note that in this approach no corrections from the mapping process are fed back to the localization of the robot. This robustifies the framework by preventing any mapping errors from influencing the state estimate and hence control of the robot. This paper is based on our previous work [14] and extends it with the following key contributions: We have reformulated our original method for the map updates from the robot's motion (Section III). To this end, we work with the full three-dimensional covariance of the map representation and present a complete derivation of the map update error propagation. We also introduce a new map fusion formulation which builds on the full map covariance to achieve a more accurate terrain estimation. Additionally, an extension based on visibility checking is presented that allows the application to dynamic environments. Finally, we demonstrate the effectiveness of our approach with the evaluation of a synthetic dataset, a real dataset in comparison with ground truth measurements, and a comparison to alternative approaches (Section IV).

### III. METHOD

The processing steps of our elevation mapping method are illustrated in Fig. 2. In the *data collection* part, the *map update from range measurements* (Section III-B) method processes the range sensor data and updates all map cells to which new measurements are associated. As the robot moves, the uncertainty of the robot's motion is propagated to the map data to comply with the robot-centric formulation. This is accomplished in the *map update from robot motion* (Section III-C) for all cells in the map. These two processing steps are used to create a robot-centric elevation map in real-time. Because the computational costs rise for an increasing map size, we limit the size of the map to the local environment of the robot.

Whenever the elevation map is required for further processing, such as collision checking, a request triggers the *map fusion* (Section III-D) in the *data processing* part of the

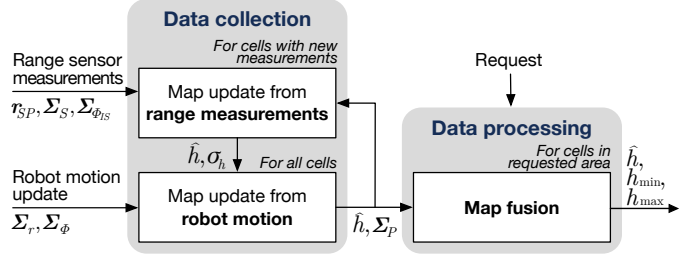


Fig. 2. The robot-centric elevation mapping is structured in a real-time *data collection* and a parallelized *data processing* step. The data collection consists of the processing steps of the range sensor measurements and the robot motion updates to create a probabilistic terrain map. When the map is needed for further processing, the map fusion process creates a consistent estimate of the terrain map together with an uncertainty interval.

approach. For each cell, the map fusion step considers the data from its surrounding cells to retrieve a consistent map representation with lower and upper confidence estimates. The requested map is typically a submap of the entire map, and the map fusion is reduced to the desired area. To continue the processing of new range measurements during the fusion step, the data processing part can be run in parallel to the data collection.

#### A. Definitions

To describe the framework, we introduce four coordinate frames, namely the inertial frame  $I$ , the robot base frame  $B$ , the sensor frame  $S$ , and the map frame  $M$  (see Fig. 3). The inertial frame  $I$  is fixed to the environment, and we assume that the real terrain is stationary relative to this frame. The base frame  $B$  and sensor frame  $S$  are body-fixed frames attached to the center of the base and the distance sensor, respectively. We assume that there exists a known transformation  $(r_{BS}, \Phi_{BS})$  between base frame  $B$  and distance sensor frame  $S$ .<sup>1</sup> With this, we can transform any formulation between the two frames  $B$  and  $S$ , and we use them interchangeably in the following formulations.

The base frame  $B$  is related to the inertial frame  $I$  through the translation  $r_{IB}$  and rotation  $\Phi_{IB}$ . This transformation is obtained through the pose estimation and characterized by the six-dimensional pose covariance matrix  $\Sigma_{IB} = \text{cov}(r_{IB}, \Phi_{IB})$ . We can split the rotation between the inertial and base frame  $\Phi_{IB}$  as

$$\Phi_{IB} = \Phi_{I\tilde{B}}(\psi) \circ \Phi_{\tilde{B}B}(\theta, \varphi), \quad (1)$$

where  $\Phi_{I\tilde{B}}(\psi)$  describes the rotation around the vertical axis  $e_z^I$  with yaw angle  $\psi$  for the frame  $\tilde{B}$  and  $\Phi_{\tilde{B}B}(\theta, \varphi)$  describes the tilt rotation between  $\tilde{B}$  and  $B$  with pitch and roll angles  $\theta$  and  $\varphi$ , respectively.

Finally, the elevation map frame  $M$  is defined through a relation to the base frame  $B$  (or sensor frame  $S$ ) with translation  $r_{BM}$  and rotation  $\Phi_{BM}$ , which are specified by the user. We choose the rotation  $\Phi_{BM}$  such that the  $z$ -axis of the map frame  $M$  and the inertial frame  $I$  remain always aligned ( $e_z^I = e_z^M$ ).

<sup>1</sup>The term  $\Phi_{BA} \in \text{SO}(3)$  is used to represent the relative orientation of a coordinate system  $B$  w.r.t. the coordinate system  $A$ . Applied to a vector  ${}_A r_{BC}$ , the rotation  $\Phi_{BA}$  maps the representation of the vector from frame  $A$  to frame  $B$ , e.g.,  ${}_B r_{BC} = \Phi_{BA}({}_A r_{BC})$ . The operator  $\circ$  is used to indicate a concatenation of rotations, e.g.,  $\Phi_{CA} = \Phi_{CB} \circ \Phi_{BA}$  [15].

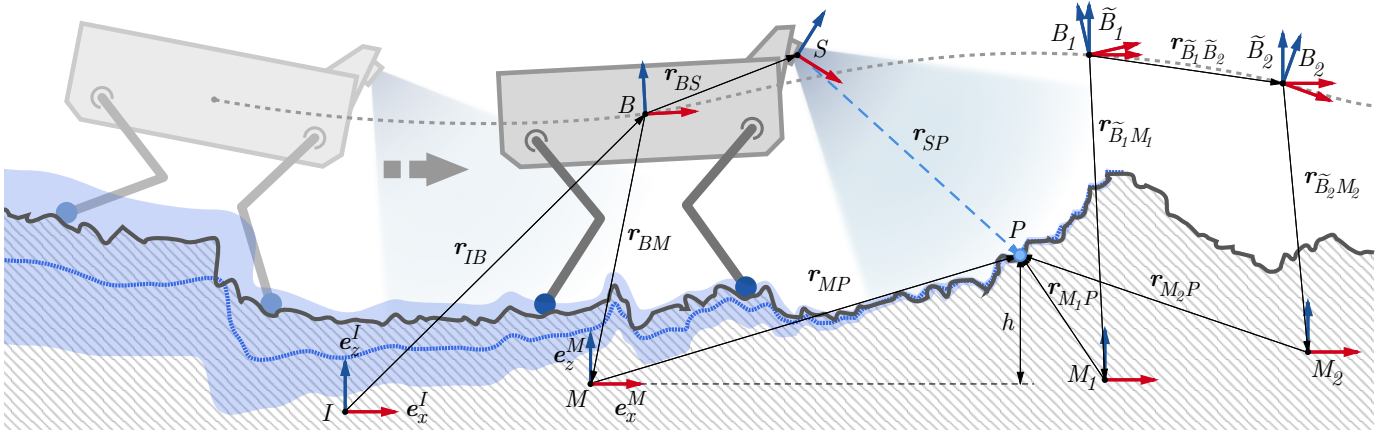


Fig. 3. Illustration of the coordinate frames used for elevation mapping. The relation between the inertial frame  $I$  and base frame  $B$  is given by the state estimation of the robot. The sensor frame  $S$  is associated to the base  $B$  through a fixed transformation. Note that in our robot-centric formulation, the map frame  $M$  is defined relative to the base frame  $B$  (as opposed to the inertial frame  $I$ ). The point  $P$  is measured in the sensor frame  $S$  and incorporated in the map in frame  $M$ . The base frames  $\tilde{B}$  are aligned with  $B$  in position and yaw-angle, but are not tilted as defined in (1) with  $e_z^I = e_z^{\tilde{B}}$ .

The last degree of freedom, which corresponds to the yaw angle  $\psi$  between  $I$  and  $M$  (rotation around  $e_z^I$ ), is chosen to match the yaw angle between  $I$  and  $B$ . The three-dimensional position of a grid cell  $i$  of the map is given by  $P_i = (x_i, y_i, \hat{h}_i)$  where  $x_i$  and  $y_i$  are defined by position of grid cell  $i$ , and  $\hat{h}_i$  is the estimated height of the terrain at cell  $i$ .

### B. Map Update from Range Measurements

New measurements from the distance sensor are handled as points in space and are mapped to the elevation map. This results in a new height measurement  $\tilde{p}$  at a cell  $(x, y)$  in the height map (we assume that one cell is updated per range measurement point). In the elevation map frame  $M$ , the height measurements are approximated by a Gaussian probability distribution as  $\tilde{p} \sim \mathcal{N}(p, \sigma_p^2)$  with mean  $p$  and variance  $\sigma_p^2$ . A single measurement, given as the position  ${}_S r_{SP}$  in the sensor frame  $S$ , can be transformed to the corresponding height measurement  $p$  with

$$p = \mathbf{P} (\Phi_{SM}^{-1}({}_S r_{SP}) - {}_M r_{SM}). \quad (2)$$

The projection matrix  $\mathbf{P} = [0 \ 0 \ 1]$  maps the three-dimensional measurement to the scalar height measurement  $p$  (in map frame  $M$ ). To obtain the variance of the height measurement  $\sigma_p^2$ , we derive the Jacobians for the sensor measurement  $\mathbf{J}_S$  and the sensor frame rotation  $\mathbf{J}_\Phi$  from (2) as

$$\mathbf{J}_S = \frac{\partial p}{\partial {}_S r_{SP}} = \mathbf{P} \mathbf{C}(\Phi_{SM})^\top, \quad (3)$$

$$\mathbf{J}_\Phi = \frac{\partial p}{\partial \Phi_{SM}} = \mathbf{P} \mathbf{C}(\Phi_{SM})^\top {}_S r_{SP}^\times, \quad (4)$$

where  $\mathbf{C}(\Phi)$  is used to describe the mapping to the corresponding rotation matrix (refer to [15] for a definition of  $\mathbf{C}(\Phi)$ ). The error propagation for the variance  $\sigma_p^2$  is given as

$$\sigma_p^2 = \mathbf{J}_S \Sigma_S \mathbf{J}_S^\top + \mathbf{J}_\Phi \Sigma_{\Phi_{IS}} \mathbf{J}_\Phi^\top, \quad (5)$$

where  $\Sigma_S$  denotes the covariance matrix of the range sensor model. The values for  $\Sigma_S$  are obtained from range sensor noise models such as [16]. The term  $\Sigma_{\Phi_{IS}}$  denotes the covariance matrix of the sensor rotation (submatrix of  $\Sigma_{IS}$ ). It is important

to note that the uncertainties of the position of the sensor  $\Sigma_{r_{IS}}$  do not need to be taken into account in this step because of our choice for the definition of the elevation map frame  $M$  (Section III-A). Furthermore, an influence of the uncertainty of the yaw-rotation of the sensor  $\psi$  (rotation around  $e_z^I$ ) on the measurement is excluded because of the use of the projection matrix  $\mathbf{P}$  and our definition of the map frame  $M$  (Section III-A).

The height measurement  $(\tilde{p}, \sigma_p^2)$  is fused with the existing elevation map estimation  $(\hat{h}, \sigma_h^2)$  by means of a one-dimensional Kalman filter:

$$\hat{h}^+ = \frac{\sigma_p^2 \hat{h}^- + \sigma_h^{2-} \tilde{p}}{\sigma_p^2 + \sigma_h^{2-}}, \quad \sigma_h^{2+} = \frac{\sigma_h^{2-} \sigma_p^2}{\sigma_h^{2-} + \sigma_p^2}, \quad (6)$$

where estimates before an update are denoted with a  $-$  superscript, respectively with a  $+$  superscript if the update has already taken place.

If multiple measurements with different heights fall into the same cell (as it is in the case of a vertical wall), we employ an update rule similarly to the one presented by [13]. Based on the Mahalanobis distance, the rule fuses the measurements for the highest elevation and drops measurements that fall below a certain distance from the current estimate.

Note that this method to combine multiple height measurements is identical in cases when the range measurements follow consecutively for the same area or when an area is re-traversed (e.g., when the robot turns or moves backward). In the latter case, ‘old’ areas will have accumulated uncertainty from the robot’s motion (see Section III-C), which causes (6) to quickly converge to the new measurements.

### C. Map Update from Robot Motion

As the elevation map frame  $M$  is defined relative to the pose of the sensor/robot (Section III-A), the elevation map data needs to be updated whenever a motion of the robot relative to the inertial frame  $I$  has occurred. This is necessary because the real terrain is stationary in the inertial frame  $I$  and we want to estimate the terrain in the moving map frame  $M$ . Therefore, the mean  $\hat{h}$  and variance values  $\sigma_h^2$  need to be

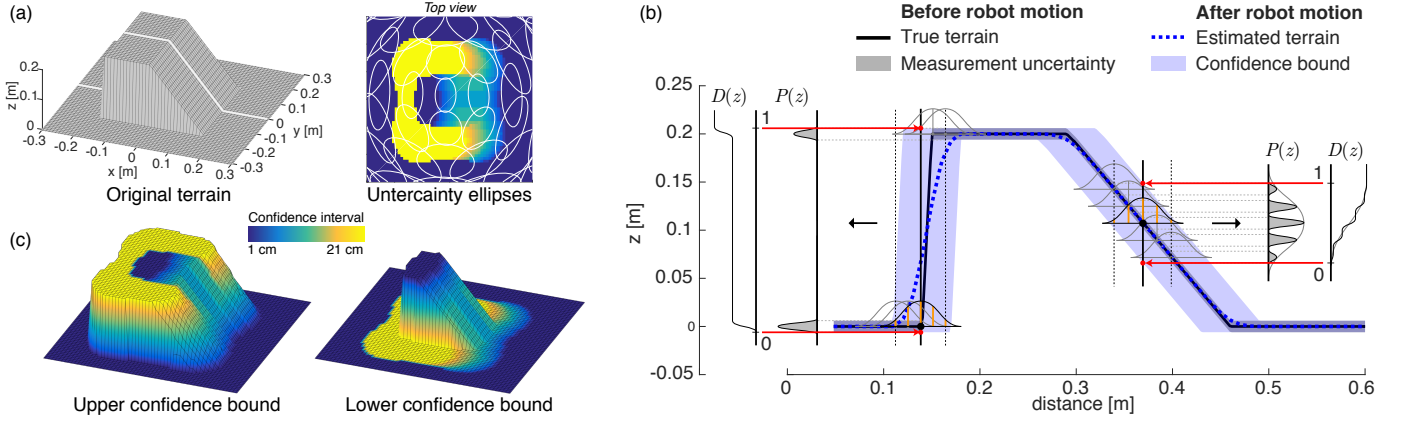


Fig. 4. The map fusion process is illustrated for a reference terrain (a) with a profile section (b). Based on the spatial covariance  $\Sigma_{P_i}$  (uncertainty ellipses in the top view), a lower and upper confidence bound can be estimated for every cell in the map (c). This map fusion process is computed with the generation of an empirical cumulative distribution function  $D(z)$  for each grid cell based on its neighboring cells.

updated according to the change of the pose estimate of the robot. Ideally, the variance and mean of each cell are updated depending on the uncertainty of the motion as well as on the estimates of the surrounding cells. However, performing such an update for each cell of the map is computationally prohibitive. Instead, we extend the elevation map structure with information about the spatial covariance matrix  $\Sigma_{P_i} \in \mathbb{R}^{3 \times 3}$  relative to the true terrain for each cell  $i$ . Herewith, we can gather the full three-dimensional uncertainty for each point of the elevation map, and postpone the computationally expensive fusion (Section III-D) to the time when the map is needed by the user or another algorithm.

In the map frame  $M$ , the robot can measure the terrain visible to the sensor accurately because  $M$  is defined relative to current pose of the robot (Section III-A). Hence, if a grid cell  $i$  receives a measurement update, we set its covariance to

$$\Sigma_{P_i} = \begin{bmatrix} \sigma_{x,\min}^2 & 0 & 0 \\ 0 & \sigma_{y,\min}^2 & 0 \\ 0 & 0 & \sigma_{h_i}^2 \end{bmatrix} \quad (7)$$

with the variance of the height estimate  $\sigma_{h_i}^2$  computed from Section III-B. The values  $\sigma_{x,\min}^2$  and  $\sigma_{y,\min}^2$  are an approximation of the horizontal uncertainty from the grid discretization and are computed as  $\sigma_{x,\min}^2 = \sigma_{y,\min}^2 = (d/2)^2$  for a square grid cell size with side length  $d$ . If the cell does not receive new sensor measurement updates, we continuously update the covariance  $\Sigma_{P_i}$  based on the relative motion of the robot from its previous to its current pose. We derive the robot motion based map update with a setup as shown in Fig. 3. For the derivation, a map reference frame  $M_k$  is associated with the robots current reference frame  $\tilde{B}_k$  through the transform  $(\mathbf{r}_{\tilde{B}_k M_k}, \Phi_{\tilde{B}_k M_k})$  for each time instance  $k$ .<sup>2</sup> The estimated position of a point  $P$  in the map coordinate frame  $M_2$  for time  $k=2$ ,  $\hat{\mathbf{r}}_{M_2 P}$ , can be expressed in relation to the map coordinate frame  $M_1$  at time  $k=1$  as

$$\hat{\mathbf{r}}_{M_2 P} = -\mathbf{r}_{\tilde{B}_2 M_2} - \hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2} + \mathbf{r}_{\tilde{B}_1 M_1} + \hat{\mathbf{r}}_{M_1 P}. \quad (8)$$

<sup>2</sup>We use here the yaw-aligned but not tilted reference frame  $\tilde{B}$  as defined in Section III-A because we are interested in the states unobservable by the robot state estimator.

Expressing this relation w.r.t. reference frame  $M_2$ , it can be written as

$$\begin{aligned} M_2 \hat{\mathbf{r}}_{M_2 P} &= -M_2 \mathbf{r}_{\tilde{B}_2 M_2} - \Phi_{\tilde{B}_2 M_2}^{-1} (\tilde{B}_2 \hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2}) \\ &+ \hat{\Phi}_{M_1 M_2}^{-1} (M_1 \mathbf{r}_{\tilde{B}_1 M_1} + M_1 \hat{\mathbf{r}}_{M_1 P}). \end{aligned} \quad (9)$$

Without loss of generality, we can choose the pose of reference frame  $M_2$  as

$$\Phi_{\tilde{B}_2 M_2} = \hat{\Phi}_{\tilde{B}_1 \tilde{B}_2}^{-1} \circ \Phi_{\tilde{B}_1 M_1}, \quad (10)$$

$$M_2 \mathbf{r}_{\tilde{B}_2 M_2} = -\Phi_{\tilde{B}_2 M_2}^{-1} (\tilde{B}_2 \hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2}) + \hat{\Phi}_{M_1 M_2}^{-1} (M_1 \mathbf{r}_{\tilde{B}_1 M_1}) \quad (11)$$

to align the reference frames  $M_1$  and  $M_2$  with properties

$$\hat{\Phi}_{M_1 M_2} = \Phi_{\tilde{B}_1 M_1}^{-1} \circ \hat{\Phi}_{\tilde{B}_1 \tilde{B}_2} \circ \Phi_{\tilde{B}_2 M_2} = \Phi_I, \quad (12)$$

$$\hat{\mathbf{r}}_{M_1 M_2} = -\mathbf{r}_{\tilde{B}_1 M_1} + \hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2} + \mathbf{r}_{\tilde{B}_2 M_2} = \mathbf{0}. \quad (13)$$

This way, we can refer to a common reference frame  $M$  and do not have to move any data in the map which is efficient in the implementation. The propagation of the covariance from time  $k=1$  to  $k=2$  can be written with (9) as follows for  $\mathbf{r}_{M_k P} \sim \mathcal{N}(\hat{\mathbf{r}}_{M_k P}, \Sigma_{P,k})$ :

$$\Sigma_{P,2} = \mathbf{J}_P \Sigma_{P,1} \mathbf{J}_P^T + \mathbf{J}_r \Sigma_r \mathbf{J}_r^T + \mathbf{J}_\Phi \Sigma_\Phi \mathbf{J}_\Phi^T. \quad (14)$$

The covariance  $\Sigma_{P,1}$  for time  $k=1$  is either initialized with (7) or known from the previous update. The covariances  $\Sigma_r$  and  $\Sigma_\Phi$  express the uncertainty for the estimated motion of the robot reference frames  $\tilde{B}_1$  and  $\tilde{B}_2$  with

$$\mathbf{r}_{\tilde{B}_1 \tilde{B}_2} \sim \mathcal{N}(\hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2}, \Sigma_r), \quad (15)$$

$$\Phi_{\tilde{B}_1 \tilde{B}_2} \sim \mathcal{N}(\hat{\Phi}_{\tilde{B}_1 \tilde{B}_2}, \Sigma_\Phi). \quad (16)$$

The transformation of the relative covariances  $\Sigma_r$  and  $\Sigma_\Phi$  from the inertial covariance  $\Sigma_{IB}$  is derived in the Appendix. The Jacobians can be evaluated to

$$\mathbf{J}_P = \frac{\partial M_2 \hat{\mathbf{r}}_{M_2 P}}{\partial M_1 \hat{\mathbf{r}}_{M_1 P}} = \mathbf{C}(\hat{\Phi}_{M_1 M_2})^T = \mathbf{I}, \quad (17)$$

$$\mathbf{J}_r = \frac{\partial M_2 \hat{\mathbf{r}}_{M_2 P}}{\partial \tilde{B}_2 \hat{\mathbf{r}}_{\tilde{B}_1 \tilde{B}_2}} = -\mathbf{C}(\Phi_{\tilde{B}_2 M_2})^T, \quad (18)$$

$$\mathbf{J}_\Phi = \frac{\partial M_2 \hat{\mathbf{r}}_{M_2 P}}{\partial \hat{\Phi}_{\tilde{B}_1 \tilde{B}_2}} = -(M_1 \hat{\mathbf{r}}_{\tilde{B}_1 M_1} + M_1 \hat{\mathbf{r}}_{M_1 P}) \times \mathbf{C}(\Phi_{\tilde{B}_1 M_1})^T. \quad (19)$$

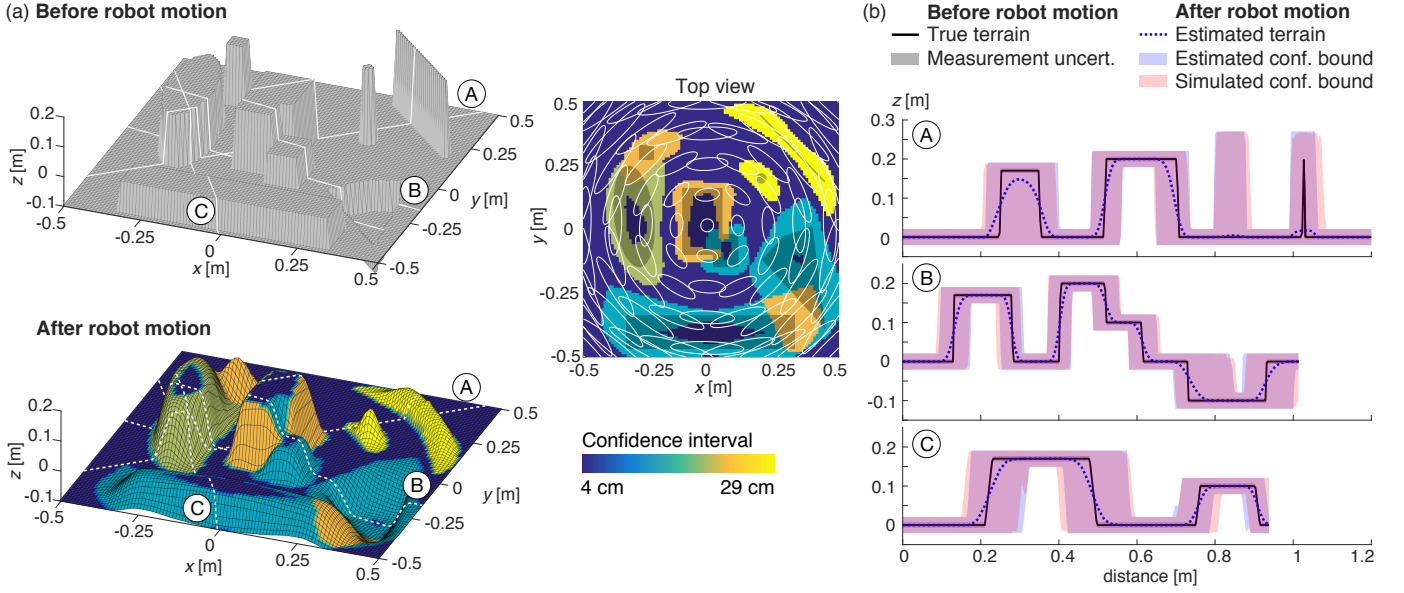


Fig. 5. A simulated terrain (a) is used to verify the map fusion process (Section III-D) for a terrain perceived under uncertainty (*after robot motion*). The three terrain profile sections A–C (b) show that the presented map fusion approximates the true (simulated) confidence bounds accurately.

We can rewrite the error propagation (14) for the case where (12) and (13) are met: To update the robot pose uncertainty from time  $k$  to  $k+1$ , evaluate the covariance of point  $P_i$  for every grid cell  $i$  of the map for the common reference frame  $M$  as

$$\Sigma_{P_i, k+1} = \Sigma_{P_i, k} + \mathbf{J}_r \Sigma_r \mathbf{J}_r^T + \mathbf{J}_\Phi \Sigma_\Phi \mathbf{J}_\Phi^T \quad (20)$$

with

$$\mathbf{J}_r = -\mathbf{C}(\Phi_{\hat{B}_{k+1}M})^T \quad \text{and} \quad \mathbf{J}_\Phi = -M \hat{\mathbf{r}}_{\hat{B}_k P_i}^\times \mathbf{C}(\Phi_{\hat{B}_k M})^T. \quad (21)$$

Note that while the change of the full robot position covariance matrix  $\Sigma_r$  is propagated on the elevation map variances, only the variances for the yaw-rotation  $\psi$  from the sensor rotation covariance matrix  $\Sigma_\psi$  (rotation around  $e_z^I = e_z^M$ ) are propagated. This is complementary to the measurement update (Section III-B), where the sensor position covariance and the sensor yaw-rotation uncertainties are excluded from the update step.

#### D. Map Fusion

Whenever required, we transform the elevation map data structure  $(\hat{h}_i, \Sigma_{P_i})$  to the representation  $(\hat{h}_i, h_{i,\min}, h_{i,\max})$  for every cell  $i$ . We refer to this step as the *map fusion* process. The mean height estimate  $\hat{h}_i$  is computed as the weighted mean from all cells within the  $2\sigma$  confidence ellipse in the  $xy$ -plane of cell  $i$ .<sup>3</sup> The values  $h_{i,\min}$  and  $h_{i,\max}$  represent the lower and upper confidence bounds of the height estimation such that the real height is within the 95% confidence interval  $h_i \in [h_{i,\min}, h_{i,\max}]$ . We infer  $h_{i,\min}$  and  $h_{i,\max}$  from the data of the surrounding cells within the  $2\sigma$  confidence ellipse (95%). This process is illustrated in Fig. 4: For each cell  $i$ , we create a weighted empirical probability density function (PDF)  $P(z)$

<sup>3</sup>Depending on the application, the height estimate  $\hat{h}_i$  could be also taken over from the original data without averaging to retain the surface structure information.

and its corresponding cumulative distribution function (CDF)  $D(z)$  from the height  $\hat{h}_j$  and variance  $\sigma_{h_j}^2$  for all cells  $j$  in the confidence ellipse. The bivariate CDF [17] in the  $xy$ -plane for cell  $i$  determines the weight for each cell  $j$  evaluated at its position relative to cell  $i$ . The lower and upper confidence bounds  $h_{i,\min}$  and  $h_{i,\max}$  can now be determined by sampling minimum and maximum values of the CDF  $D(z)$ .

#### E. Extension to Dynamic Environments

As expected, the height update rule (6) converges after multiple measurements to a high-confidence estimate. However, this approach works poorly in dynamic environments because many measurement updates are required to reflect the new shape of the terrain, causing a significant mapping delay. To this end, we introduce an extension to the map update process which consists of two parts: The first part adds a constant noise value  $\bar{\sigma}^2$  as

$$\sigma_h^{2+} = \sigma_h^{2-} + \bar{\sigma}^2. \quad (22)$$

to map cells for which new measurements (based on the timestamp) are recorded that fall below the current height estimate and exceed a Mahalanobis distance threshold. This improves the map adaptation time in regions where the real terrain is *lowered*, which is the case when for example an obstacle is moved away. With the choice of the value  $\bar{\sigma}^2$ , the user trades off between fast adaptation to a changing terrain and an increased uncertainty for cells at edges in the terrain.

In a second part, a visibility check is performed based on ray tracing as shown in Fig. 6. We create a *visibility map* from the rays connecting the point of the height measurement  $p + 3\sigma_p$  to the depth sensor's position at this measurement. This map reflects the maximal height  $h_{\max}$  that each cell can have based on the visibility constraint. Consequently, cells with  $\hat{h} - 3\sigma_h > h_{\max}$  violate the visibility constraint and are removed. As this visibility check is computationally intensive, it is only

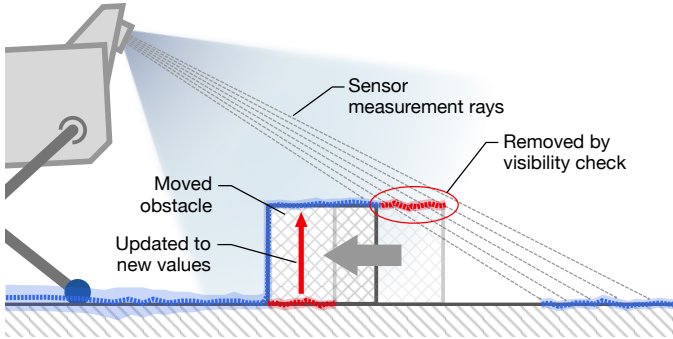


Fig. 6. The mapping process adapts to dynamic environments through removal of cells that violate the visibility constraint. In this figure, the obstacle was moved from right to left. The previous map (red) is updated and the visibility is checked with ray tracing resulting in an updated map (blue).

performed at a lower rate (e.g., 1 Hz) and in parallel to the data collection process. In this sense, it is complementary to the first part of the adaptation to dynamic environments presented in this section.

#### IV. RESULTS

The described elevation mapping algorithm is implemented as a C++ library with an interface to the Robot Operating System (ROS) and is available open-source<sup>4</sup>. For efficient data handling and operations, the software builds upon the Grid Map library [18]. Our implementation features measurement noise models for popular range sensors from the product families Intel RealSense, Microsoft Kinect, Hokuyo, and Velodyne. In the following, we discuss the presented mapping framework with results from simulation and experiments with a robotic platform.

##### A. Evaluation in Simulation

We compare the map fusion process (Section III-D) for a artificial dataset to a reference map. Fig. 5a shows the original and resulting terrain map. The reference map was created through exhaustively disturbing and sampling the original map with the given uncertainty. Fig. 5b shows the estimated and simulated confidence intervals for three slices A–C through the map. This comparison shows that the estimated confidence intervals describe the position of the actual terrain at high fidelity. One important observation is that large flat regions remain at a low confidence interval. In contrast, areas at the edges of objects become more uncertain in their height estimation with high confidence intervals. This is a critical feature which can be exploited in motion planning for collision avoidance and contact planning (e.g., foothold planning for legged robots [19]).

##### B. Evaluation on Hardware

We have validated our approach in various environments on our legged robots where the real-time generated maps serve as the basis for motion planning [5]. To evaluate the effectiveness of the proposed robot-centric mapping approach, we compare the resulting terrain reconstruction to ground

truth measurements. The robot StarLETH [20] walks over obstacles and localizes itself based on a proprioceptive state estimation based on kinematic and inertial measurements [3]. With this setup, the position and yaw-rotation of the robot are unobservable and drift over time as the robot moves. The robot carries a PrimeSense Carmine 1.08 structured-light depth sensor and processes the sensor data (data collection) at 20 Hz at a resolution of  $1 \times 1$  cm per map cell. As ground truth, we tracked the real position of the robot with an optical motion capture system and recorded a reference terrain with a stationary Leica Nova MS50 MultiStation geomatic laser scanner. Fig. 7 shows a snapshot of the experiment. The robot’s estimated and reference position are shown in Fig. 7a together with a top view of the onboard terrain reconstruction. We analyze the resulting terrain map in comparison with the reference map for a terrain profile section as shown in Fig. 7b. As expected, the drift of the state estimation causes the terrain estimates to diverge from the true terrain for areas which have not been updated with range measurements. For these areas, we can observe an increase of the error ellipses in Fig. 7a. Even though the unobserved regions diverge, our mapping method correctly captures the uncertainty, and the actual terrain remains within the confidence bounds as shown in Fig. 7b.

##### C. Comparison

The approach presented in this paper is compared to two alternative approaches, namely to our previous contribution Fankhauser et al., 2014 [14] and to the work of Kleiner et al., 2007 [13]. Fig. 8 shows the result of the different mapping methods when applied to terrain under the uncertainty of a robot’s motion. For the presented approach (a), the estimated terrain confidence bounds match closely with the true (sampled) bounds. For our earlier approach [14] (b), the confidence bounds partially over and underestimate the real distribution. On the one side, the two-dimensional covariance data is a conservative approximation of the true uncertainty. On the other side, the approach in [14] insufficiently captures the multi-modal distribution in the map fusion process, leading to ‘overshooting’ artifacts at edges. Finally, the approach used in [13] (c) neglects the effect of motion uncertainty in the horizontal plane. Instead, the vertical uncertainty  $\sigma_h^2$  is accumulated, which leads to strong over- and underestimation in flat and structured regions of the terrain, respectively. Furthermore, no indication of the terrain-structure influenced uncertainty can be given with this method.

#### V. CONCLUSION

We have presented a novel approach to elevation mapping that addresses the problem of localization drift as it often occurs on mobile robots with proprioceptive state estimation. The presented method estimates the elevation map in a robot-centric coordinate frame such that the process of integrating new measurements into the map is only affected by the range sensor noise and the uncertainty of the observable roll- and pitch angles. The data in the map is updated based on the uncertainty of the incremental motion as the robot moves

<sup>4</sup>[http://github.com/ethz-asl/elevation\\_mapping](http://github.com/ethz-asl/elevation_mapping)

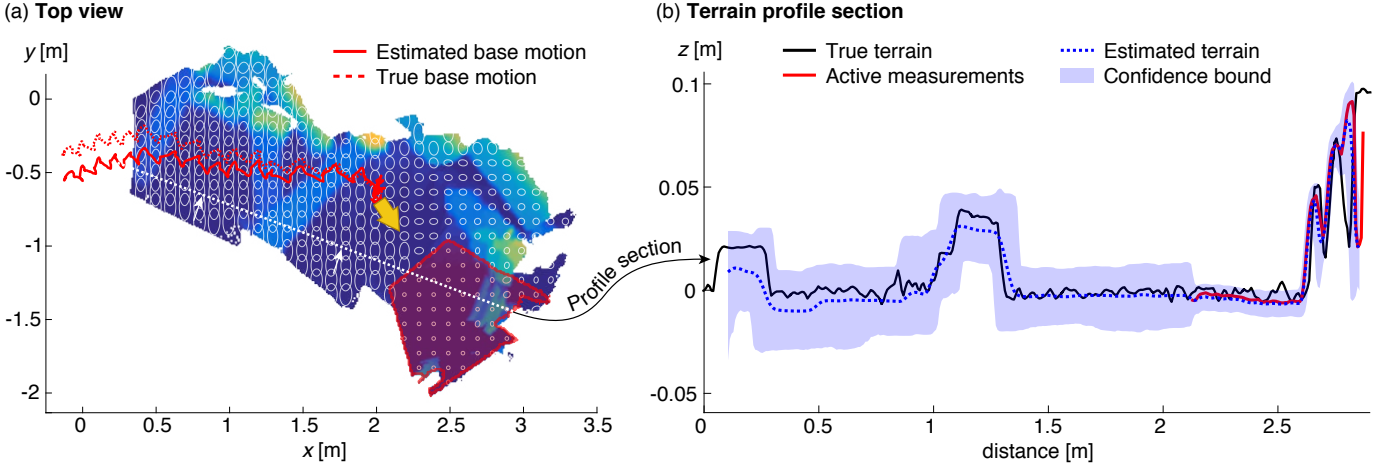


Fig. 7. This figure shows the mapping process with a legged robot walking over obstacles. The robot uses proprioceptive state estimation to localize itself, which is affected by drift as the system moves. The top view (a) shows the onboard map creation and the comparison of the estimated robot trajectory to the true motion. The terrain profile section (b) show that the true terrain remains within the confidence bounds.

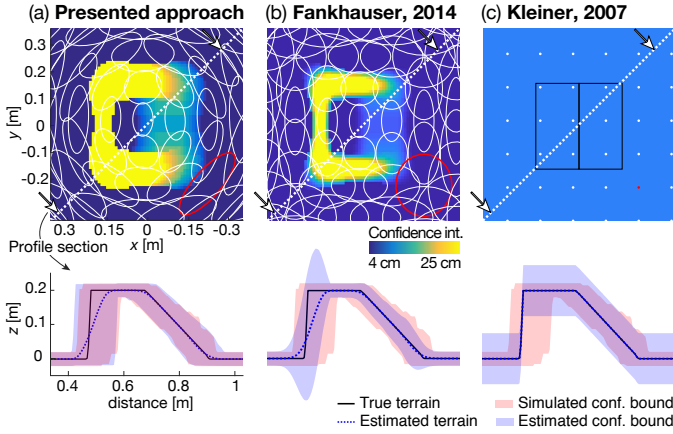


Fig. 8. We compare the mapping method presented in this paper (a) with the approaches used in [14] (b) and [13] (c). The top row shows a top view on the map (same terrain as in Fig. 5) where the colors indicate the confidence interval and the ellipsoids depict the (horizontal) uncertainty. The bottom row shows a diagonal profile section of the map and compares the estimated confidence bounds (blue) with the true confidence bounds (red).

through the environment. This gives the robot at any point in time an estimate of the terrain from its local perspective. We lower the computation burden of the mapping procedure by splitting the method into a data collection (map update from range measurements and robot motion) and a map fusion step. This enables reliable real-time elevation mapping independent of an exteroceptive localization method. We illustrate how the mapping process estimates the terrain profile with a corresponding confidence interval and show how the estimated distribution matches the ground truth reference in simulated and real-world experiments.

A current limitation of this work is prominent when used with legged robots. In cases when the robot takes many steps in place, the terrain map below the robot increases in uncertainty due to drift of the localization, which poses a difficulty for the terrain-aware control of the robot. In our current work, we tackle this challenge with downward facing depth sensors, more accurate localization, and matching of footholds with the terrain map to provide feedback for the localization.

## APPENDIX

Localizations methods typically specify the uncertainty of the robot's pose  $B$  relative to an inertial frame  $I$  as

$$\mathbf{x}_{IB} = ({}^I\mathbf{r}_{IB}, \Phi_{IB}), \quad \Sigma_{IB}^{6 \times 6} = \text{cov}(\mathbf{x}_{IB}). \quad (23)$$

However, in our formulation of the robot motion update process (Section III-C), the uncertainty of the pose is expressed relative from a pose at time  $k$  to time  $k+1$  as

$$\mathbf{x}_{\tilde{B}\tilde{B}} = (\tilde{B}_{k+1} \mathbf{r}_{\tilde{B}_k \tilde{B}_{k+1}}, \Psi_{\tilde{B}_{k+1} \tilde{B}_k}), \quad \Sigma_{\tilde{B}\tilde{B}}^{4 \times 4} = \text{cov}(\mathbf{x}_{\tilde{B}\tilde{B}}). \quad (24)$$

In the following, we derive the relative covariance  $\Sigma_{\tilde{B}\tilde{B}}$  for the case that the absolute covariance  $\Sigma_{IB}$  is available.

### A. State Reduction

In a first step, we need to reduce the state representation of the covariance matrix from the full robot pose  $B$  with state vector  $\mathbf{x}_{IB} = (\mathbf{r}_{IB}, \Phi_{IB})$  to the  $z$ -aligned robot pose  $\tilde{B}$  with state vector  $\mathbf{x}_{\tilde{B}} = (\mathbf{r}_{\tilde{B}}, \psi)$  with  $\mathbf{r}_{IB} = \mathbf{r}_{\tilde{B}}$  and  $\psi = \Psi_{\tilde{B}} = \Psi_{IB}$ . We can achieve this with

$$\Sigma_{\tilde{B}}^{4 \times 4} = \mathbf{J} \Sigma_{IB}^{6 \times 6} \mathbf{J}^T \quad (25)$$

$$\text{with } \mathbf{J} = \begin{bmatrix} \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{0}^{1 \times 3} & \frac{\partial \psi}{\partial \Phi} \end{bmatrix} \quad (26)$$

$$\text{and } \frac{\partial \psi}{\partial \Phi_{IB}(\varphi, \theta, \psi)} = \begin{pmatrix} \cos \psi \sin \theta & \sin \theta \sin \psi & 1 \\ \cos \theta & \cos \theta & 0 \end{pmatrix}. \quad (27)$$

### B. Covariance Transformation

We assume that the state estimate can be modeled as Gaussian random walk, which is the case for example for odometry-based localization methods:

$$\begin{aligned} \mathbf{x}_{\tilde{B},k+1} &= \begin{pmatrix} {}^I\mathbf{r}_{\tilde{B},k+1} \\ \psi_{k+1} \end{pmatrix} \\ &= \begin{pmatrix} {}^I\mathbf{r}_{\tilde{B},k} \\ \psi_k \end{pmatrix} + \begin{bmatrix} \mathbf{C}_{\tilde{B}}(\psi_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{pmatrix} \tilde{B}\mathbf{v}_k + \mathbf{n}_{v,k} \\ \omega_k + \mathbf{n}_{\omega,k} \end{pmatrix} \Delta t \end{aligned} \quad (28)$$

with translational velocity of the robot  $\mathbf{v}$  and rotational velocity  $\omega$  around the  ${}^I\mathbf{e}_z = \tilde{B}\mathbf{e}_z$  axis with zero mean multivariate



Gaussian noise vector  $\mathbf{n} = (\mathbf{n}_v, n_\omega)^\top$ . The covariance propagation can be written as

$$\Sigma_{\bar{I}\bar{B},k+1} = \mathbf{F} \Sigma_{\bar{I}\bar{B},k} \mathbf{F}^\top + \mathbf{G} \mathbf{Q}_{k+1} \mathbf{G}^\top, \quad (29)$$

with

$$\mathbf{F} = \frac{\partial \mathbf{x}_{\bar{I}\bar{B},k+1}}{\partial \mathbf{x}_{\bar{I}\bar{B},k}} = \begin{bmatrix} \mathbf{I}^{3 \times 3} & \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^\times C_{\bar{I}\bar{B}}(\psi_k) \bar{\mathbf{v}}_k \Delta t \\ \mathbf{0}^{1 \times 3} & 1 \end{bmatrix}, \quad (30)$$

$$\mathbf{G} = \frac{\partial \mathbf{x}_{\bar{I}\bar{B},k+1}}{\partial \mathbf{n}_k} = \begin{bmatrix} C_{\bar{I}\bar{B}}(\psi_k) \Delta t & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \Delta t \end{bmatrix}. \quad (31)$$

The relative motion of robot from time  $k$  to  $k+1$  can be expressed in the  $\bar{B}$  frame as

$$\begin{aligned} \mathbf{x}_{\bar{B}\bar{B}} &= \begin{pmatrix} \bar{\mathbf{b}}_{k+1} \mathbf{r}_{\bar{\mathbf{B}}_k \bar{\mathbf{B}}_{k+1}} \\ \psi_{\bar{\mathbf{B}}_{k+1} \bar{\mathbf{B}}_k} \end{pmatrix} \\ &= \begin{pmatrix} C_{\bar{B}\bar{I}}(\psi_k) (\mathbf{I} \mathbf{r}_{\bar{I}\bar{B},k+1} - \mathbf{I} \mathbf{r}_{\bar{I}\bar{B},k}) \\ \psi_{k+1} - \psi_k \end{pmatrix} \end{aligned} \quad (32)$$

$$\text{with (28)} \quad = \begin{pmatrix} \bar{\mathbf{b}} \mathbf{v}_k + \mathbf{n}_{v,k} \\ \omega_k + n_{\omega,k} \end{pmatrix} \Delta t.$$

With this we can express  $\Sigma_{\bar{B}\bar{B}}$  as

$$\Sigma_{\bar{B}\bar{B}} = \text{cov}(\mathbf{x}_{\bar{B}\bar{B}}) = \mathbf{Q}_{k+1} \Delta t^2. \quad (33)$$

With (29) we get

$$\mathbf{Q}_{k+1} = \mathbf{G}^{-1} (\Sigma_{\bar{I}\bar{B},k+1} - \mathbf{F} \Sigma_{\bar{I}\bar{B},k} \mathbf{F}^\top) (\mathbf{G}^\top)^{-1} \quad (34)$$

and finally

$$\Sigma_{\bar{B}\bar{B}} = \mathbf{G}^{-1} (\Sigma_{\bar{I}\bar{B},k+1} - \mathbf{F} \Sigma_{\bar{I}\bar{B},k} \mathbf{F}^\top) (\mathbf{G}^\top)^{-1} \Delta t^2 \quad (35)$$

with

$$\mathbf{G}^{-1} = \begin{bmatrix} \frac{C_{\bar{I}\bar{B}}(\psi_k)^\top}{\Delta t} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \frac{1}{\Delta t} \end{bmatrix}. \quad (36)$$

This can be simplified assuming the property  $\mathbf{v}_k \Delta t = \mathbf{r}_{\bar{I}\bar{B},k+1} - \mathbf{r}_{\bar{I}\bar{B},k}$  to

$$\Sigma_{\bar{B}\bar{B}} = \hat{\mathbf{G}} (\Sigma_{\bar{I}\bar{B},k+1} - \hat{\mathbf{F}} \Sigma_{\bar{I}\bar{B},k} \hat{\mathbf{F}}^\top) \hat{\mathbf{G}}^\top \quad (37)$$

with

$$\hat{\mathbf{F}} = \begin{bmatrix} \mathbf{I}^{3 \times 3} & \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^\times C_{\bar{I}\bar{B}}(\psi_k) (\mathbf{r}_{\bar{I}\bar{B},k+1} - \mathbf{r}_{\bar{I}\bar{B},k}) \\ \mathbf{0}^{1 \times 3} & 1 \end{bmatrix}, \quad (38)$$

$$\hat{\mathbf{G}} = \begin{bmatrix} C_{\bar{I}\bar{B}}(\psi_k)^\top & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & 1 \end{bmatrix}. \quad (39)$$

For this to be used in (20), we need to get  $\Sigma_r$  and  $\Sigma_R$  from  $\Sigma_{\bar{B}\bar{B}}$ . From the definition (24),  $\Sigma_r$  can be read directly from

$$\Sigma_{\bar{B}\bar{B}} = \begin{bmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \sigma_{\psi_{\bar{\mathbf{B}}_{k+1} \bar{\mathbf{B}}_k}}^2 \end{bmatrix}. \quad (40)$$

The expression for  $\Sigma_R$  can be retrieved as

$$\Sigma_R = \mathbf{J}_\psi \sigma_{\psi_{\bar{\mathbf{B}}_{k+1} \bar{\mathbf{B}}_k}}^2 \mathbf{J}_\psi^\top \quad (41)$$

with

$$\mathbf{J}_\psi = \frac{\partial \Phi(\varphi, \theta, \psi)}{\partial \psi} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (42)$$

## REFERENCES

- [1] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *International Symposium on Robotic Research*, pp. 467–474, 1987.
- [2] S. Thrun, "Robotic mapping: A Survey," in *Exploring artificial intelligence in the new millennium* (G. Lakemeyer and B. Nebel, ed.), 2002.
- [3] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6058–6064, 2013.
- [4] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer, "ANYmal - Toward Legged Robots for Harsh Environments," *Advanced Robotics*, vol. 31, pp. 918–931, sep 2017.
- [5] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust Rough-Terrain Locomotion with a Quadrupedal Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [6] D. Belter, P. Labecki, and P. Skrzypczyński, "Estimating terrain elevation maps from sparse and uncertain multi-sensor data," in *International Conference on Robotics and Biomimetics*, 2012.
- [7] A. Roennau, T. Kerscher, M. Ziegenmeyer, J. M. Zoellner, and R. Dillmann, "Six-legged walking in rough terrain based on foot point planning," in *International Conference on Climbing and Walking (CLAWAR)*, pp. 591–598, Singapore, World Scientific, 2009.
- [8] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and Foothold Optimization using Low-Dimensional Models for Rough Terrain Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1096–1103, 2017.
- [9] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.
- [10] M. Herbert, C. Caillas, E. Krotkov, I. Kweon, and T. Kanade, "Terrain mapping for a roving planetary explorer," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 997–1002, IEEE, 1989.
- [11] S. Kweon and T. Kanade, "High resolution terrain map from multiple sensor data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, 1992.
- [12] D. Wooden, M. Malchano, K. Blankespoor, A. Howard, A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4736–4741, 2010.
- [13] A. Kleiner and C. Dornhege, "Real-time localization and elevation mapping within urban search and rescue scenarios," *Journal of Field Robotics*, vol. 24, pp. 723–745, aug 2007.
- [14] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-Centric Elevation Mapping with Uncertainty Estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, (Poznan, Poland), 2014.
- [15] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nützi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart, "A Primer on the Differential Calculus of 3D Orientations," tech. rep., 2016.
- [16] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling," in *International Conference on Advanced Robotics (ICAR)*, pp. 388–394, IEEE, 2015.
- [17] R. V. Hogg, J. McKean, and A. T. Craig, *Introduction to Mathematical Statistics*. Upper Saddle River, NJ, United States of America: Pearson Education, 6 ed., 2005.
- [18] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)* (A. Koubaa, ed.), ch. 5, Springer, 2016.
- [19] P. Fankhauser, *Perceptive Locomotion for Legged Robots in Rough Terrain*. Doctoral thesis, ETH Zurich, 2018.
- [20] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "StarLETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2012.