Autonomous Robots (2021) 45:843–857
https://doi.org/10.1007/s10514-021-10013-w

Navigating by touch: haptic Monte Carlo localization via geometric sensing and terrain classification

Russell Buchanan[1] · **Jakub Bednarek**[2] · **Marco Camurri**[1] · **Michał R. Nowicki**[2] · **Krzysztof Walas**[2] · **Maurice Fallon**[1]

Received: 17 November 2020 / Accepted: 23 July 2021 / Published online: 27 August 2021

Abstract
Legged robot navigation in extreme environments can hinder the use of cameras and lidar due to darkness, air obfuscation or sensor damage, whereas proprioceptive sensing will continue to work reliably. In this paper, we propose a purely proprioceptive localization algorithm which fuses information from both geometry and terrain type to localize a legged robot within a prior map. First, a terrain classifier computes the probability that a foot has stepped on a particular terrain class from sensed foot forces. Then, a Monte Carlo-based estimator fuses this terrain probability with the geometric information of the foot contact points. Results demonstrate this approach operating online and onboard an ANYmal B300 quadruped robot traversing several terrain courses with different geometries and terrain types over more than 1.2 km. The method keeps pose estimation error below 20 cm using a prior map with trained network and using sensing only from the feet, leg joints and IMU.

Keywords Legged robots · Proprioceptive localization · Terrain classification · Tactile sensing

1 Introduction

Recent advances in the maturity and robustness of quadrupedal robots have made them appealing for dull and dirty industrial operations, such as routine inspection and monitoring. Automating these operations in underground mines and sewers is particularly challenging due to darkness, in-air dust, dirt, and water vapor, which can significantly impair a robot's vision system (Kolvenbach et al. 2020). Additionally, camera or laser sensor failure may leave only proprioceptive sensors (i.e.,IMU and joint encoders) at the robot's disposal (Fig. 1).

Blind quadrupedal locomotion has achieved impressive levels of reactive robustness without requiring vision sensors (Focchi et al. 2020; Lee et al. 2020). However, without the ability to also *localize* proprioceptively, a robot would still be incapable of completing missions or inspections.

1.1 Contributions

In this paper, we significantly extend our prior work on proprioceptive localization (Buchanan et al. 2020) with the following contributions:

- A novel proprioceptive legged robot localization system that, in contrast to Buchanan et al. (2020), fuses both terrain geometry and semantic information. To the best of our knowledge, this is the first localization system using semantics when completely blind.
- A terrain classification method employing signal masking in the 1D convolutional modules, making it possible to process variable length signals from footsteps without

✉ Russell Buchanan
russell@robots.ox.ac.uk

Jakub Bednarek
jakub.bednarek@put.poznan.pl

Marco Camurri
mcamurri@robots.ox.ac.uk

Michał R. Nowicki
michal.nowicki@put.poznan.pl

Krzysztof Walas
krzysztof.walas@put.poznan.pl

Maurice Fallon
mfallon@robots.ox.ac.uk

[1] Oxford Robotics Institute, University of Oxford, Oxford, UK

[2] Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland

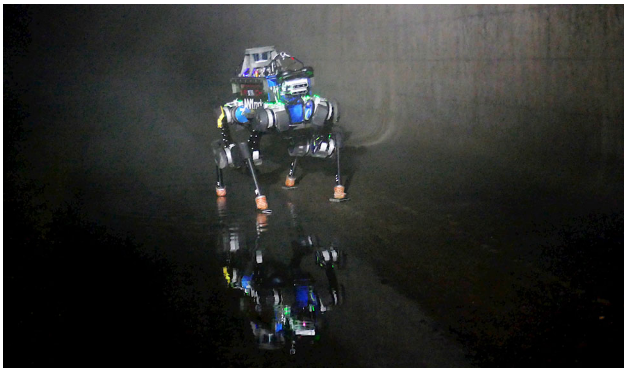


Fig. 1 An ANYmal robot (Hutter et al. 2016) in a sewer with two feet in a slippery, wet depression and two feet in a dry, elevated area. With prior information about terrain type and geometry, it is possible for the robot to localized in the world using only touch. This would be extremely useful in dark and foggy environments (Image courtesy of RSL/ETH and from Kolvenbach et al. (2020))

the need to truncate or pad them. This enables our method to work on uneven terrain and at differing walking speeds, unlike in Bednarek et al. (2019b).

- Extensive additional testing on an ANYmal B300 quadruped robot (Hutter et al. 2016) including different geometries and terrain types, for a total duration of 2.5 h and more than 1.2 km of traveled distance. The pose estimation error is kept down to 10 cm on geometrically feature rich terrain and on average below 20 cm on all terrain while exploiting terrain semantics. We also demonstrate convergence after only five steps from an unknown initial pose.

The remainder of this document is structured as follows: Sect. 2 summarizes relevant research in the fields of terrain classification, in-hand tactile localization and legged haptic localization; Sect. 3 defines the mathematical background of the legged haptic localization problem; Sect. 4 describes our proposed haptic localization algorithm; Sect. 5 describes the implementation details to deploy our algorithm on a quadruped platform; Sect. 6 presents the experimental results collected using the ANYmal robot; Sect. 7 provides an interpretation of the results and discusses the limitations of the approach; finally, Section 8 concludes with final remarks.

2 Related works

Pioneering work which exploits a robot's legs, not just for locomotion, but also to infer terrain information such as friction, stiffness and geometry has been presented by Krotkov (1990). This idea has recently been revisited to perform terrain vibration analysis or improve locomotion parameter selection via terrain classification. Since we are interested in using terrain classification for localization, we cover the most relevant works applied to legged robots in Sect. 2.1. Works on proprioceptive localization in manipulation and legged robots are described in Sects. 2.2 and 2.3, respectively.

2.1 Tactile terrain classification

The first tactile terrain classification method for walking robots was presented by Hoepflinger et al. (2010) and concerned experiments with a single leg detached from a robot's body. In this work, force measurements and motor currents were used to successfully distinguish between four terrain types. These results paved the way for the application of terrain classification methods on complete legged robots.

More recently, Kolvenbach et al. (2019) used a single leg on a real, standing ANYmal robot to differentiate between four different types of soil. Two types of feet (point and planar) were used to collect force, torque and IMU measurements that were processed by an SVM classifier. The system showed that the tactile information could be used to differentiate between visually similar soils. However, their method required a pre-determined probing action which is impractical as it forces the robot to stop walking.

A terrain classification system that could operate during locomotion was presented by Wellhausen et al. (2019). At start, their legged robot was trained to assign a terrain negotiation cost based on force/torque sensors. Once operating, their system assigned a terrain negotiation cost from images based on previous feet-to-image correspondences and terrain classification based on proprioceptive sensors. The ability to predict the terrain negotiation based on images was then used to plan the robot's motion and avoid high cost terrains.

In complete darkness, which is our intended domain, vision-based sensors are of limited use for terrain classification, so we focus on purely proprioceptive sensing. In our previous work (Bednarek et al. 2019a), we showed how deep learning models can be used to increase the terrain classification accuracy. The system showed 98% classification accuracy from force-torque measurements against six terrain classes during a statically stable walk. However, this approach was limited to fixed-length input signals and thus could not generalize to aperiodic gaits, different speeds, or uneven terrain. In our work, we overcome these limitations with a novel masking mechanism in the convolutional layer, which allows us to process variable length signals.

More recently, Lee et al. (2020) showed an end-to-end approach to terrain classification for locomotion. Their deep learning controller was based on proprioceptive signals to adapt the gait to rough terrains. Even though tactile terrain classification was not explicitly performed, an internal representation of the terrain type was implicitly stored inside the network's memory. In our work, we opt for a modular approach that explicitly returns a terrain class, which can then be used by the localization estimator.

2.2 Tactile localization in manipulation

Tactile localization involves the estimation of the 6-Degree of Freedom (DoF) pose of an object (of known shape) in the robot's base frame by means of kinematics of the robot's fingers and its tactile sensors. Since the object can have any shape, the probability distribution of its pose given tactile measurements can be multimodal. For this reason, tactile localization has typically been addressed using Sequential Monte Carlo (SMC) methods, a subfamily of which are called *particle filters* (Fox et al. 2001). SMC methods are sensitive to the dimension of the state space, which should be low enough to avoid combinatorial explosions or particle depletion. State-of-the-art methods aim to reduce this dimensionality and also to sample the state space in an efficient manner. For example, Koval et al. (2016) reduced the state space of the pose of an in-hand object to the observable contact manifold.

Chalon et al. (2013) proposed a particle filtering method for online in-hand object localization that tracks the pose of an object while it is being manipulated by a fixed base arm. The estimated pose was subsequently used to improve the performance of pick and place tasks. The particle weights were updated by penalizing finger/object co-penetration and the distance between the object and the fingertip in contact.

Manuelli and Tedrake (2016) approached a slightly different problem, using a particle filter to estimate external contacts on a rigid body robot using only force/torque sensors in the joints of the robot. Particles were distributed around the robot's body and particle weights were computed from how well the contact point explained an external torque.

Vezzani et al. (2017) proposed an algorithm for tactile localization using the Unscented Particle Filter (UPF) on a iCub robot with sensorized fingertips to localize four different objects in the robot's reference frame. The algorithm was recursive and could process data in real-time. The object and the robot's base were assumed to be static, allowing the pose to be estimated as a fixed parameter. For legged haptic localization, the assumption of both a static robot and terrain does not always hold and more general methods are required.

2.3 Haptic localization of legged robots

The first example of haptic localization applied to legged robots is from Chitta et al. (2007). In their work, they presented a proprioceptive localization algorithm based on a particle filter for LittleDog, a small electric quadruped. The robot was commanded to perform a statically stable gait over a known irregular terrain course, using a motion capture system to feed the controller. While walking, the algorithm approximated the probability distribution of the base state with a set of particles. The state included three DoF: linear position on the xy-plane and yaw. Each particle was sampled from the uncertainty of the odometry, while the weight of a particle was determined by the L2 norm of the height error between the map and the feet contact location. The algorithm was run offline on eight logged trials of 50 s each.

Schwendner et al. (2014) demonstrated haptic localization on a wheeled robot with protruding spikes. The spikes detected contacts with the ground, which were compared to a prior 2.5D elevation map. Each wheel enabled multiple contact measurements, which they used to perform plane fitting against the prior map and improve localization over larger, flatter terrain. They also performed terrain classification, but with a camera, which we do not require in our proposed work. They demonstrated an average position error 39 cm in five experiments, of approximately 100 m each.

In Buchanan et al. (2020), we presented an SMC method that estimated the past trajectory (instead of the latest pose) at every step. Furthermore, the localization was performed for the full 6-DoF of the robot, instead of just the x, y and yaw dimensions as in Chitta et al. (2007) and Schwendner et al. (2014). The localization system was experimentally demonstrated online and onboard an ANYmal robot and used in a closed loop navigation system. When walking on flat areas, the localization uncertainty increased due to the lack of constraints on the xy-plane. We are therefore motivated to use terrain classification techniques described in Sect. 2.1 to incorporate more information into the SMC.

3 Problem statement

Let $\boldsymbol{x}_k \in SE(3)$ be a robot's pose at time k. We use the notation $\tilde{\boldsymbol{x}}$ to represent a pose estimate from an external estimator, and $\boldsymbol{x}^*$ to represent its most likely estimate from our SMC filter.

3.1 Quadruped state definition

We assume that for each timestep k, an estimate of the robot pose $\tilde{\boldsymbol{x}}_k$ and its covariance $\Sigma_k \in \mathbb{R}^{6\times 6}$ are available from an inertial-legged odometric estimator, such as Bloesch et al. (2018), Fink and Semini (2020), Hartley et al. (2020). The uncertainties for the rotation manifold are maintained in the Lie tangent space, as in Forster et al. (2017). We also assume that the location of the robot's end effectors in the base frame $\mathcal{D}_k = (\mathbf{d}_{\text{LF}}, \mathbf{d}_{\text{RF}}, \mathbf{d}_{\text{LH}}, \mathbf{d}_{\text{RH}}) \in \mathbb{R}^{3\times 4}$ are known from forward kinematics. The forces acting on each foot $\mathcal{F}_k = (\mathbf{f}_{\text{LF}}, \mathbf{f}_{\text{RF}}, \mathbf{f}_{\text{LH}}, \mathbf{f}_{\text{RH}}) \in \mathbb{R}^{3\times 4}$ are measured by foot sensors (when available) or inferred from inverse dynamics. Finally, the binary contact states $\mathcal{K}_k = (\kappa_{\text{LF}}, \kappa_{\text{RF}}, \kappa_{\text{LH}}, \kappa_{\text{RH}}) \in \mathbb{B}^4$ are inferred from $\mathcal{F}_k$.

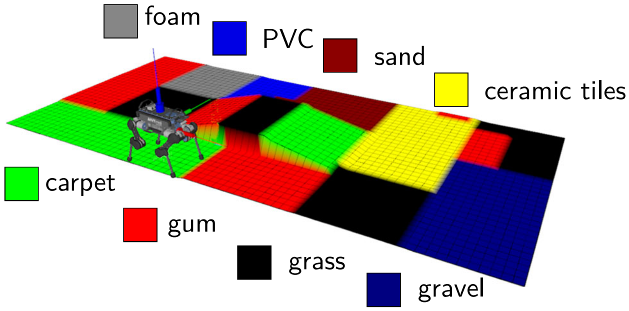


Fig. 2 Terrain map used in Experiment 3, showing terrain categories

For simplicity, we assume errors due to joint encoder noise or limb flexibility to be negligible. Therefore, the propagation of the uncertainty from the base to the end effectors is straightforward to compute. For brevity, the union of the aforementioned states (pose, contacts, and forces) at time k will be referred as the *quadruped state* $\mathcal{Q}_k = \{\tilde{\boldsymbol{x}}_k, \Sigma_k, \mathcal{D}_k, \mathcal{K}_k, \mathcal{F}_k\}$.

3.2 Prior map

Our approach can localize against 2.5D terrain elevation maps as well as full 3D maps. Terrain classification is meant to be carried out while the robot is walking, with no dedicated probing actions, therefore 2.5D maps are augmented with a terrain class category for each cell. This enables our method to overcome the degeneracy caused by featureless geometries (e.g., flat grounds, which are uninformative about the robot position on the xy-plane). Point clouds are used for 3D maps and only contain geometric information. To distinguish between the three types of map, we will refer to $\mathcal{M}$ when 2.5D only, $\mathcal{M}_3$ when 3D, and $\mathcal{M}_c$ when 2.5D augmented with class information, respectively. An example of an $\mathcal{M}_c$ map colorized by terrain class is shown in Fig. 2.

3.3 Estimation objective

Our goal is to use a sequence of quadruped states to estimate the most likely sequence of robot poses up to time k:

$$\mathcal{X}^*_k = [\boldsymbol{x}^*_0,\ \boldsymbol{x}^*_1,\ \ldots,\ \boldsymbol{x}^*_k] \tag{1}$$

such that the likelihood of the contact points to be on the map is maximized. Additionally, we assume $\boldsymbol{x}^*_0$ is known.

4 Proposed method

To perform localization, we sample a predefined number of particles at regular intervals from the pose distribution provided by the odometry (as described in Sect. 4.2) and we compute the likelihoods of the measurements by comparing each particle to the prior map, so as to update the weights of the particle estimator (see Sect. 4.3). For convenience, we give a brief summary of SMC theory in Sect. 4.1.

4.1 Sequential Monte Carlo localization

In SMC Localization, the objective is to approximate the posterior distribution of the state $\boldsymbol{x}_k$ given a history of measurements $z_0, \ldots, z_k = z_{0:k}$ as follows:

$$p\left(\boldsymbol{x}_k|z_{0:k}\right) = \sum_i w^i_{k-1} p\left(z_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}\right) \tag{2}$$

where w^i is the *importance weight* of the i-th particle; $p\left(z_k|\boldsymbol{x}_k\right)$ is the measurement likelihood function and $p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}\right)$ is the motion model for the i-th particle state. Since $p\left(\boldsymbol{x}_k|z_{0:k}\right)$ is typically unknown, the state $\boldsymbol{x}_k$ is typically sampled from $p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}\right)$, yielding:

$$p\left(\boldsymbol{x}_k|z_{0:k}\right) = \sum_i w^i_{k-1} p\left(z_k|\boldsymbol{x}^i_{k-1}\right) \delta\left(\boldsymbol{x}_k - \boldsymbol{x}^i_k\right) \tag{3}$$

where $\delta(\cdot)$ is the Dirac delta function. Over repeated sampling steps, the particles will spread out of the whole state space with weights approaching zero. To avoid this "impoverishment" of the particles, an additional re-sampling from the mostly likely state is used. When the re-sampling is done at every step, the method is known as particle filtering. We use a different strategy that merges likelihoods from a history of states and so refer to our method with the more general term SMC.

4.2 Locomotion control and sampling strategy

Without a very robust reactive controller, blind locomotion requires conservative footstep placement, hence we opt for a statically stable gait, which guarantees stability at all times even when the motion is stopped mid flight phase. Since only one leg can be moved at the time, as soon as the swing leg touches the ground the robot enters into a four-support phase; at this time, the quadruped state estimate $\mathcal{Q}_k$ and the estimated terrain class $\tilde{c}$ for a given foot position are collected. Then, a new set of particles is sampled in a manner similar to Chitta et al. (2007):

$$p(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}) = \mathcal{N}(\boldsymbol{x}_k, \Delta\tilde{\boldsymbol{x}}_k \boldsymbol{x}^i_{k-1}, \Sigma_k) \tag{4}$$

where $\Delta\tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}^{-1}_{k-1}\tilde{\boldsymbol{x}}_k$ is the pose increment measured by the onboard state estimator at times $k-1$ and k.

At time k, the new particles $\boldsymbol{x}^i_k$ are thus sampled from a normal distribution centered at the pose estimated from the

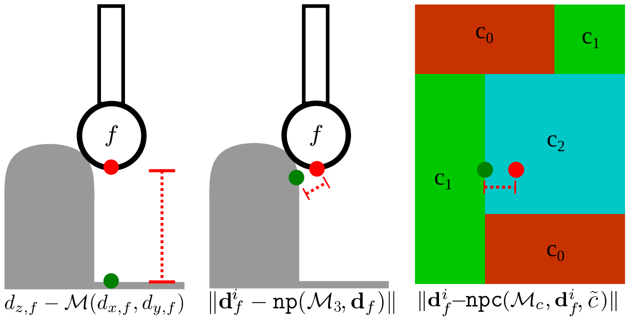

f
f
c$_0$
c$_1$
c$_1$
c$_2$
c$_0$
$d_{z,f} - \mathcal{M}(d_{x,f}, d_{y,f})$
$\|\mathbf{d}^i_f - \texttt{np}(\mathcal{M}_3, \mathbf{d}_f)\|$
$\|\mathbf{d}^i_f - \texttt{npc}(\mathcal{M}_c, \mathbf{d}^i_f, \tilde{c})\|$

Fig. 3 Comparison between contact measurements for 2.5D (left), 3D (middle) and terrain class (right) map representations. The red dots indicate the contact point as sensed by the robot, while the green dots indicate the corresponding location returned by the map. The red line shows the magnitude of the measurement. In the terrain classification case, the view is from a top down perspective. The robot's foot sensed a contact in the c_2 region but a classification of c_1 was detected. The nearest c_1 point was returned by the map (Color figure online)

odometry with its corresponding covariance. Since roll and pitch angles are observable from inertial-legged estimators, their estimates have low uncertainty. In practice, this allows us to reduce the number of necessary particles along these two dimensions, while still retaining the ability to compensate for the errors of the state estimator due to IMU biases, which are observable by exteroceptive sensors only.

4.3 Measurement likelihood model for 2.5D data

The measurement likelihood is modeled as a univariate Gaussian centered at the local elevation of each cell, as done in Buchanan et al. (2020). The variance σ_z was manually set to 1 cm. Given a particle state $\boldsymbol{x}^i_k$, the estimated position of a contact in world coordinates for an individual foot f is defined as the concatenation of the estimated robot base pose and the location of the end effector, in base coordinates:

$$\mathbf{d}^i_f = (d^i_{xf}, d^i_{yf}, d^i_{zf},) = \boldsymbol{x}^i_k \mathbf{d}_f \tag{5}$$

Thus, the measurements and their relative likelihood functions for the i-th particle and a specific foot f are (Fig. 3, left):

$$z_k = d^i_{zf} - \mathcal{M}(d^i_{xf}, d^i_{yf}) \tag{6}$$

$$p(z_k|\boldsymbol{x}^i_k) = \mathcal{N}(z_k, 0, \sigma_z) \tag{7}$$

where d^i_{zf} is the vertical component of the estimated contact point location in world coordinates of foot f, according to the i-th particle; $\mathcal{M}(d^i_{x,f}, d^i_{y,f})$ is the corresponding map elevation at the xy coordinates of $\mathbf{d}^i_f$.

4.4 Measurement likelihood model for 3D data

Our method can incorporate contact events from 3D probing. This is useful for areas where the floor does not provide enough information to localize. In this case, the robot can probe walls and 3D objects with its feet. To better model this situation, we represent the prior map $\mathcal{M}_3 \in \mathbb{R}^{3\times N}$ by a 3D point cloud with N points. The likelihood of a particular contact point is computed using the Euclidean distance between the foot and the nearest point in the map. This likelihood is again modeled as a zero-mean Gaussian evaluated at the Euclidean distance between the estimated contact point $\mathbf{d}^i_f$ and its nearest neighbor on the map, with variance σ_z:

$$z_k = \|\mathbf{d}^i_f - \texttt{np}(\mathcal{M}_3, \mathbf{d}^i_f)\| \tag{8}$$

$$p(z_k|\boldsymbol{x}^i_k) = \mathcal{N}(z_k, 0, \sigma_z) \tag{9}$$

where $\texttt{np}(\mathcal{M}_3, \mathbf{d}^i_f)$ is the function that returns the nearest point of $\mathbf{d}^i_f$ on the map $\mathcal{M}_3$, computed from its k–d tree (Fig. 3, middle). In our tests, point clouds were sufficiently small to make the k-d tree search time negligible. For larger scale environments, more compact representations based on Truncated Sign Distance Fields (Oleynikova et al. 2017) or Octrees (Vespa et al. 2018) could be used to allow for faster search and less memory usage.

4.5 Terrain classification

Let $f\colon \mathcal{S} \mapsto \mathcal{C}$ be the haptic terrain classification function that associates an element from the signal domain $\mathcal{S}$ to an integer from the class counter-domain $\mathcal{C}$. The set $\mathcal{S}$: $\{s \in \mathbb{R}^{l(s)\times 6}\}$ includes sequences of variable length force and torque signals s (of length $l(s)$) generated by a foot touchdown event. The set $\mathcal{C}$ is defined as the integers from 0 to $n-1$, where n is the total number of terrain classes that the robot is expected to be walking on (in our case, $n = 8$). The terrain classes were chosen to have a different haptic response, ensuring the existence of the function f.

Given the problem definition above, we introduce the following:

- a classification method $f'\colon \mathcal{S} \mapsto \mathcal{C}$, which approximates the function f. As an implementation of f' we used a neural network;
- a dataset consisting of a list of pairs $d\colon [(s, c)]$, where $s \in \mathcal{S}, c \in \mathcal{C}$. Such a dataset was divided into two subsets, training and validation, with a ratio of 80:20;
- a training process formulated as an approximation of the function f using the function f' by the minimization of cross-entropy between the probability distributions generated by these functions.

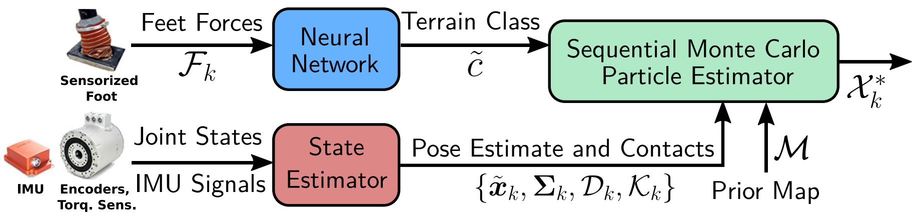


Fig. 4 Information flow diagram of the overall system

4.6 Terrain class measurement likelihood model

The prior map is represented as a 2-dimensional grid whose cells are associated with a terrain class (an example is provided by Fig. 2). Measurements are represented as a piece-wise cost function. If the estimated terrain class $\tilde{c}$ for a given foot position $\mathbf{d}_f^i$ does match the class c, at that location in the prior map $c = \mathcal{M}_c(d_{xf}^i, d_{yf}^i)$ the probability is a constant value corresponding to the maximum value of a zero-mean univariate Gaussian with manually selected variance σ_c of 5 cm (which is the width of the foot).

If the estimated class does not match the expected class in the map, the same Gaussian distribution is used to model the likelihood, where the input z_k^c is the distance to the closest point in the map with the expected class. This function is shown as

$$p(z_k|\boldsymbol{x}_k^i) = \begin{cases} \frac{1}{\sigma_c\sqrt{2\pi}} & \tilde{c} = c \\ \mathcal{N}(z_k, 0, \sigma) & \tilde{c} \neq c, \end{cases} \tag{10}$$

where

$$z_k^c = \|\mathbf{d}_{xyf}^i - \mathtt{npc}(\mathcal{M}_c, \mathbf{d}_{xyf}^i, \tilde{c})\|_2. \tag{11}$$

The function $\mathtt{npc}(\mathcal{M}_c, \mathbf{d}_{xyf}^i, \tilde{c})$ returns the nearest 2D point with class $\tilde{c}$ to the 2D foot position $\mathbf{d}_{xyf}^i$ in the map $\mathcal{M}_c$. This last case is shown in Fig. 3, right.

We assume elevation and terrain class measurements (z_k, z_k^c) are conditionally independent. Therefore, their joint probability can be computed as

$$p(z_k, z_k^c|\boldsymbol{x}_k^i) = p(z_k|\boldsymbol{x}_k^i)p(z_k^c|\boldsymbol{x}_k^i). \tag{12}$$

5 Implementation

The block diagram of our system is shown in Fig. 4. The internal estimator on the ANYmal robot, TSIF (Bloesch et al. 2018), provides the odometry for the particle estimator, while the neural network estimates the class. This information is compared against the prior map to provide an estimate of the robot's trajectory, $\mathcal{X}_k^*$.

Pseudocode for the particle estimator (green block in Fig. 4) is listed in Algorithm 1. At time k, the estimates of the terrain class $\tilde{c}$ and the robot pose $\tilde{\boldsymbol{x}}_k$ are collected. The pose estimate is used to compute the relative motion $\Delta\tilde{\boldsymbol{x}}_{k-1:k}$, propagate forward the state of each particle $\boldsymbol{x}_k^i$, and draw a sample from the distribution centered in $\Delta\tilde{\boldsymbol{x}}_k\boldsymbol{x}_{k-1}^i$ with covariance $\Sigma_k = (\sigma_{x,k}, \sigma_{y,k}, \sigma_{z,k})$.

```
$\boldsymbol{x}_0^i \sim \mathcal{N}(\boldsymbol{x}_0, \tilde{\boldsymbol{x}}_0, \Sigma_0) \quad \forall i \in N$
foreach four-support phase k do
    $\Delta\tilde{\boldsymbol{x}}_k \leftarrow \tilde{\boldsymbol{x}}_{k-1}^{-1}\tilde{\boldsymbol{x}}_k$
    foreach particle $i \in N$ do
        $\boldsymbol{x}_k^i \sim \mathcal{N}(\boldsymbol{x}_k, \Delta\tilde{\boldsymbol{x}}_k\boldsymbol{x}_{k-1}^i, \Sigma_k)$
        $w_k^i \leftarrow w_{k-1}^i$
        foreach foot f do
            if 3D then
                $z_k \leftarrow \|\mathtt{np}(\mathcal{M}_3, \mathbf{d}_f^i) - \mathbf{d}_f^i\|$
                $w_k^i \leftarrow w_k^i p(z_k|\boldsymbol{x}_k^i)$
            else
                $z_k \leftarrow d_{zf}^i - \mathcal{M}(d_{xf}^i, d_{yf}^i)$
                $z_k^c \leftarrow \|\mathbf{d}_{xyf}^i - \mathtt{npc}(\mathcal{M}_c, \mathbf{d}_{xyf}^i, \tilde{c})\|$
                $w_k^i \leftarrow w_k^i p(z_k, z_k^c|\boldsymbol{x}_k^i)$
            end
        end
        $x_k^* \leftarrow \mathrm{WeightedMean}(x_k^0 \ldots x_k^N, w_k^0, \ldots w_k^N)$
        $\mathcal{X}_k^* \leftarrow [\boldsymbol{x}_0^j, \ldots, \boldsymbol{x}_k^j]$
        if $Var(w_k^i) > threshold$ then
            resample$(x_k^i)$
        end
    end
end
```

Algorithm 1: Haptic Sequential Monte Carlo Localization

The weight of a particle w^i is then updated by multiplying it by the likelihood that each foot is in contact with the map and the terrain class. In our implementation, we modify the likelihood functions from Eq. 7 as:

$$p(z_k|\boldsymbol{x}_k^i) = \min(\rho, \mathcal{N}(z_k, 0, \sigma_z)) \tag{13}$$

where ρ is a minimum weight threshold, so that outlier contact measurements do not immediately lead to degeneracy.

Re-sampling is triggered when the variance of the weights rises above a certain threshold. This is necessary to avoid dispersion of the particle set across the state space, with many particles with low weight. By triggering this process when the variance of the weights increases, the particles can first track the dominant modes of the underlying distribution.

5.1 Particle statistics

The pose estimate for the k-th iteration, $\boldsymbol{x}_k^*$, is computed from the weighted mean of all the particle poses. However, as we showed in our previous work, the particle distribution is often multi-modal. This motivated us to selectively update different dimensions of the robot pose. If the variance of the particle positions in the x and y axes are low (i.e.,$\ll \sigma_{x,k}, \sigma_{y,k}$), we assume a well defined estimate and update the robot's full pose. However, if they are high, we update only the z component of the robot's location, which is always low as the robot keeps contact with the ground.

In practice, with the terrain classification we found the terrain course was sufficiently detailed to keep particle position variance in the xy-plane low, therefore z-only updates were rare. The threshold we used was a standard deviation of 10 cm, which corresponds to the typical uncertainty in our experiments.

To avoid particle degeneracy, importance sampling can be done in areas with higher likelihood. For example, if a grass terrain is detected, some particles could be injected in every grass terrain in the map. Further investigation on the benefits of importance sampling are left for future work.

5.2 Terrain classifier network

Ou neural network architecture is shown in Fig. 5 and consists of three components: convolutional, recurrent, and predictive. Both the convolutional and recurrent components must process variable-length data. Therefore, for the convolution part, masking of the signal is required to prevent padded values from affecting the forward-pass result.

The first component of our network consists of two residual layers (ResLay). The ResLay used in our work is an adaptation of the one by He et al. (2016) with 2D convolutions replaced by 1D and support for masking. The recurrent component uses two bidirectional layers (Bidir) with two Gated Recurrent Units (GRU) (Cho et al. 2014) in each. The output of the recurrent component is an average of two resulting hidden states of the last Bidir. The final output of the neural network is produced by the predictive component, which takes the recurrent component's output, and using two fully connected (FC) layers produces a probability distribution from which the terrain class is inferred.

The number of convolutional layers and their sizes were chosen empirically to produce the best possible features before applying the compressed signal to the recursive part. To ensure better error propagation for the convolution part, we employed so-called skip-connections between layers thus reducing dimensionality.

The consecutive layers of the model are presented in Fig. 5. Each convolution block executes the following operations: batch normalization (Ioffe Szegedy 2015), dropout (Srivastava et al. 2014), and Exponential Linear Unit (ELU) activation function (Clevert et al. 2016). All are modified to support masking. We used kernel size of 5 in each convolution layer. The output from each ResLay block is two times smaller as a result of stride in convolutions (marked as /2). Dropout is also used in every Bidir and FC (which also use batch normalization).

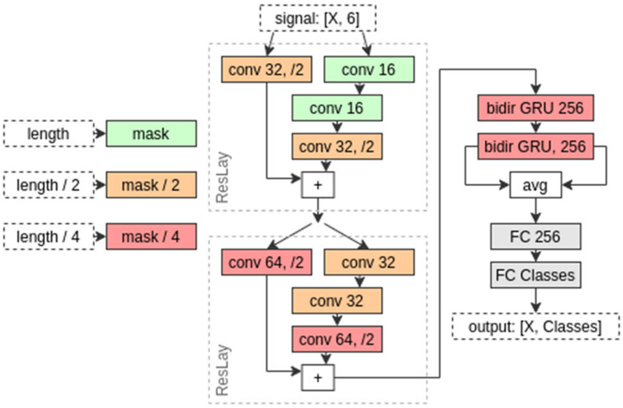


Fig. 5 Neural Network Structure used for Terrain Classification. The main blocks of neural architecture are convolutional (conv), recursive (GRU bidir), and fully connected (FC) blocks. The masking mechanism used in variable-length signal processing blocks takes appropriate masks, marked in green, orange, and red. These masks correspond to the individual signal lengths, taking into account the initial length and the reduction of size by convolution with stride equal to 2 (Color figure online)

The model uses a dropout rate of 0.3 and a batch normalization momentum of 0.6. The proposed neural network consists of 1,374,920 trainable parameters.

5.2.1 Training

The learning process was carried out using the k-fold cross-validation methodology, with $k = 5$. The AdamW optimizer from Loshchilov and Hutter (2019) was used to minimize the loss function, with the following parameters:

- learning rate: 5e−4 with exponential decay,
- weight decay: 1e−4 with cosine decay.

Training was performed for 1000 epochs after which the training continued until no progress was made for 100 consecutive epochs. The size of each mini-batch was 256.

6 Experimental results

We extensively evaluated the performance of our algorithm in three different experiments, each one targeting a different type of localization. These are described in more detail in Sects. 6.2, 6.3, and 6.4.

6.1 Evaluation protocol

There are three different modalities in our algorithm: HL-G (Haptic Localization Geometric), which uses only geometric information, HL-C (Haptic Localization Class), using only class information, and HL-GC (Haptic Localization Geometric and Class), which uses both geometric and terrain class information. HL-G was tested in Experiments 1 and 2 using 2.5D and 3D prior maps, respectively. HL-G, HL-C and HL-GC were tested in Experiment 3 with 2.5D maps augmented with terrain class information.

6.1.1 Evaluation metrics

We quantitatively evaluated localization performance by computing the the mean of the Absolute Translation Error (ATE) as described by Sturm et al. (2012):

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathrm{trans}\left(\mathbf{T}_i^{-1}\hat{\mathbf{T}}_i\right)\| \qquad (14)$$

where $\mathbf{T}_i$ and $\hat{\mathbf{T}}_i$ are the robot's ground truth and estimated poses, respectively. In contrast to Sturm et al. (2012), we do not perform the alignment of trajectories, as ground truth and estimated poses are represented in the same coordinate system.

A qualitative evaluation was also performed for Experiment 1 and 2 by assessing the ability of the robot to reach its planned goals or end effector targets while using the localization online. This demonstrated the benefit of the localization when used in the loop with the onboard motion planner.

6.1.2 Ground truth and prior map

The ground truth trajectories were collected by motion capture systems at 100 Hz. The pose of both the robot and the terrain course reference frame were accurately measured with mm accuracy via reflective markers installed on them.

At start of the experiment, the relative position of the robot within the map was measured using ground truth and used for initialization only. Thereafter, the pose of the robot was estimated using the particle filter. To account for initial errors, particles at the start were sampled from a Gaussian centered at the initial robot pose with a covariance of 20 cm.

The prior maps were captured with survey grade laser scanners (Leica BLK-360 and SURPHASER 100HSX) which provided point clouds with sub-centimeter accuracy.

6.2 Experiment 1: 2.5D terrain course

In this experiment, the robot was commanded to navigate between four walking goals at the corners of a rectangle. One of the edges required crossing a 4.2 m terrain course composed of a 12° ascending ramp, a 13 cm high chevron pattern, an asymmetric composition of uneven square blocks and a 12° descending ramp (Fig. 6). After crossing the wooden course, the robot returned to the starting position across a portion of flat ground, so as to test the system behavior in feature-deprived conditions.

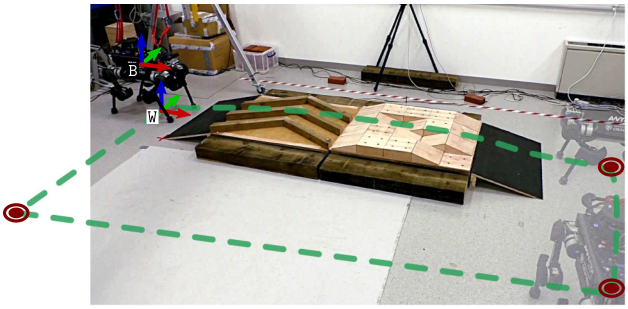


Fig. 6 Experiment 1: ANYmal haptic localization experiments. The robot traverses the terrain, turns 90 deg right and comes back to the initial position passing trough the flat area. The goals given to the planner are marked by the dark red disks, while the planned route is a dashed green line (one goal is out of the camera field of view). The world frame `W` is fixed to the ground, while the base frame `B` is rigidly attached to the robot's chassis. The mutual pose between the robot and the terrain course is bootstrapped with the motion capture (Color figure online)

Table 1 Experiment 1: Estimation performance

Mean Absolute Translation Error (ATE)				
Trial	Dist. (m)	Time (s)	TSIF (m)	HL-G (m)
1	66.42	525	0.63	**0.13**
2	145.31	1097	2.57	**0.40**
3	55.67	557	0.52	**0.19**
4	68.71	604	0.65	**0.32**
5	172.65	1606	2.00	**0.61**

TSIF, Two-State Implicit Filter (Bloesch et al. 2018); HL-G, Haptic Localization with Geometry only

While blind reactive locomotion has been developed by a number of groups including (Di Carlo et al. 2018; Focchi et al. 2020), unfortunately our blind controller was not sufficiently reliable to cross this terrain course, so we resorted to use of the statically stable gait from Fankhauser et al. (2018) which used a depth camera to aid footstep planning. However, the pose estimation was performed without access to any camera information.

To demonstrate repeatability, we performed five trials of this experiment, for a total distance traveled of more than 0.5 km and 1 h 13 min duration. A summary of the results is presented in Table 1, where HL-G shows an overall improvement between 50 and 85% in the ATE compared to the onboard state estimator. ATE is 33 cm on average, which reduces to 10 cm when evaluating only the feature-rich portion of the experiments (i.e.,the terrain course traversal).

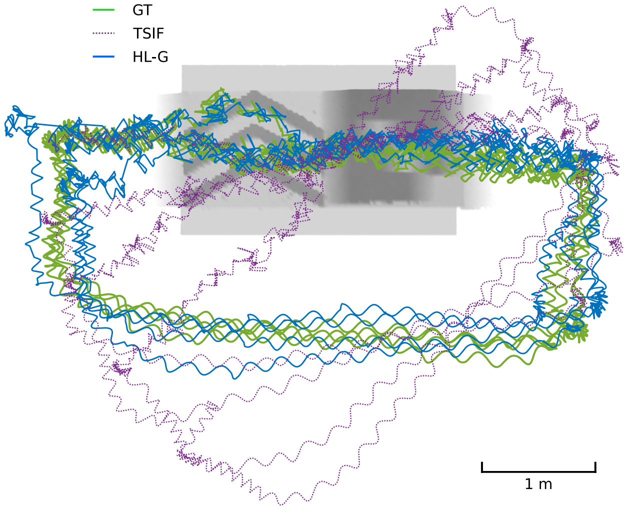


Fig. 7 Experiment 1, Trial 2: Top view of the estimated trajectories from TSIF (dashed purple), haptic localization (blue) and ground truth (green) (Color figure online)

For trials 1 and 2, the robot was manually operated to traverse the terrain course, completing two and four loops, respectively. In trials 3–5, the robot was commanded to follow the rectangular path autonomously. In these trials, the haptic localization algorithm was run online in closed-loop and effectively guided the robot towards the goals (Fig. 7). Using only the prior map and the contact events only, the robot stayed localized in all the runs and successfully tracked the planned goals. This can be seen in Fig. 7, where the estimated trajectory (in blue) diverges from ground truth on the xy-plane when the robot is walking on the flat ground. This is due to growing uncertainty from lack of geometric information, however the covariance reduces significantly and the cluster mean re-aligns with the ground truth when the robot reaches the terrain.

Figure 8 shows in detail the estimator performance for each of the three linear dimensions and yaw. Since position and yaw are unobservable, the drift on these states is unbounded. In particular, the error on the odometry filter [TSIF (Bloesch et al. 2018), purple dashed line] is dominated by upward drift (due to kinematics errors and impact nonlinearities, see third plot) and yaw drift (due to IMU gyro bias, see bottom plot). This drift is estimated and compensated for by the haptic localization (blue solid line), allowing accurate tracking of the ground truth (green solid line) in all dimensions. This can be noted particularly at the four peaks in the z-axis plot, where the estimated trajectory and ground truth overlap. These times coincide with the robot is at the top of the terrain course.

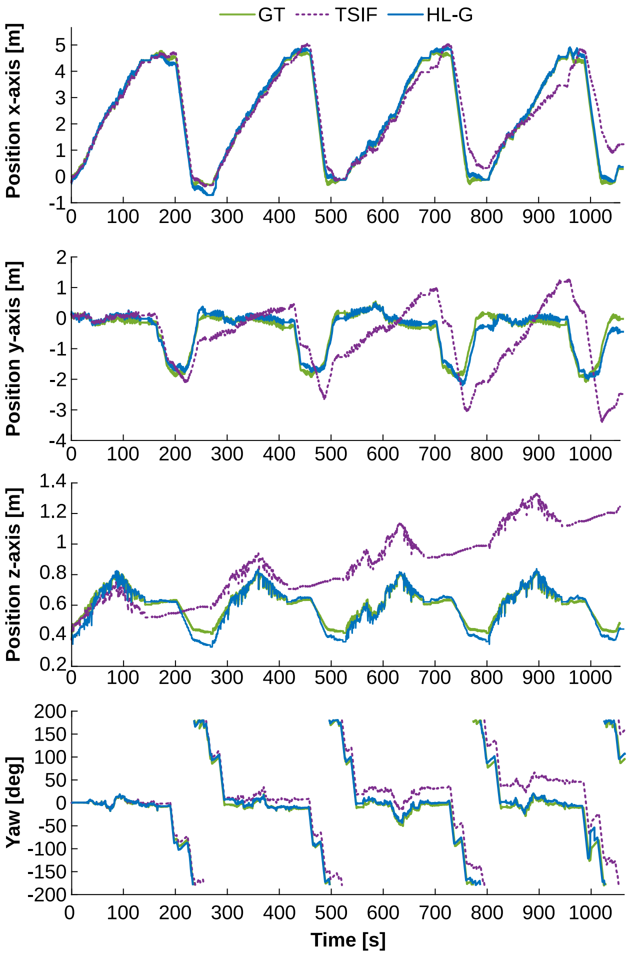


Fig. 8 Experiment 1, Trial 2: Comparison between the estimated position from TSIF (dashed purple) and haptic localization (blue) against ground truth (green). After 200 s, the estimation error in TSIF has drifted significantly upward and in yaw. In particular, the upward drift is noticeable in the third plot, where the values grow linearly. The drift is eliminated by the re-localization against the prior map (Color figure online)

6.3 Experiment 2: online haptic exploration on vertical surfaces

The second experiment involved a haptic wall following task with the robot starting in front of a wall but with an uncertain location. The particles were again initialized with 20 cm position covariance. To test the capability to recover from an initial error, we applied a 10 cm offset in both x and y from the robot's true position in the map. At start, the robot was commanded to walk 1 m to the right (negative y direction) and press a button on the wall, whose location in the prior map was known. To accomplish the task, the robot needed to "feel its way" by alternating probing motions with its right front foot and walking laterally to localize inside the room. The fixed number of probing motions was pre-scripted so with each step to the right, the robot probed both in front and

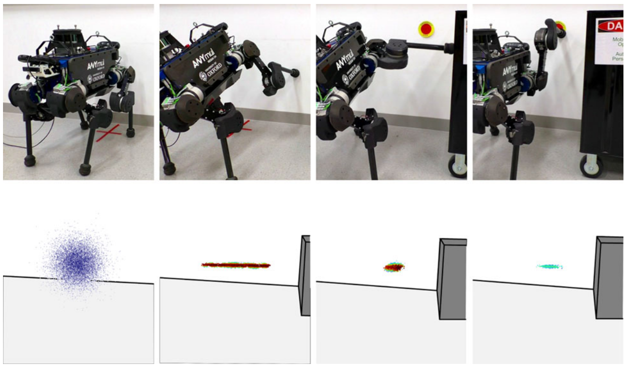


Fig. 9 Experiment 2: Haptic probing experiment in 3-dimensions. The top row shows the robot performing the experiment while the bottom row shows the particle distribution of position estimates. The particle set is colorized by normalized weight according to the jet colormap (i.e., dark blue = lowest weight, dark red = highest weight). First, on the bottom left, the robot has an initial distribution with poses equally weighed. The robot makes a forward probe and then moves to the right. Now the particles are distributed as an ellipse with high uncertainty to the left and right of the robot. Then, the robot makes a probe to the right and touches an obstacle; the particle cloud collapses into a tight cluster. Since the robot is now localized, it is able to complete the task of pressing the button on the wall (Color figure online)

to its right. The whole experiment was executed blindly with the static controller from Fankhauser et al. (2018).

As shown in Fig. 9 the robot was able to correct its localization and complete the task of touching the button. The initial probe to the front reduced uncertainty in the robot's x and z directions, which reduces the particles to an ellipsoidal elongated along y. As the robot moves, uncertainty in the x direction increases slightly. By touching the wall on the right, the robot re-localized in all three dimensions in much the same way as a human following a wall in the dark would. The re-localization allows the robot to press the button, demonstrating the generalization of our algorithm to 3D. This would enable a robot to localize by probing a known piece of mining machinery, allowing it to perform maintenance tasks. The final position error was: [7.7, −3.7, −0.2] centimeters in the x, y and z directions.

6.4 Experiment 3: terrain classification

In the third experiment, we demonstrate the localization using terrain semantic information, which has been tested on a custom designed terrain course. Multiple $1 \times 1\ \mathrm{m}^2$ tiles of different terrain materials were placed on a $3.5 \times 7\ \mathrm{m}^2$ area. The course includes a 20 cm high platform with two ramps with different terrain materials, as shown in Fig. 10. The different terrain types used were: gum, carpet, PVC, foam, sand, artificial grass, ceramic and gravel.

For training, we gathered an additional dataset of the robot walking on the different patches consisting of 8773 steps with a quasi-static walk gait. Examples of data collection is shown in Fig. 10 Top. During the data collection, the maximum base displacement and rotation were enforced to 0.21 m and 0.23 rad, respectively. These limits ensured a stable walk at all times. The dataset was split into 7018 training and 1775 testing samples. To minimize the impact of unbalanced data on the learning process (e.g., more steps on a specific class), the loss function was weighted based on the number of steps taken on each terrain. The network was trained as described in Sect. 5.2.1 and the mean and standard deviation of the accuracy was estimated from k-fold cross-validation to be 94% and 0.09, respectively.

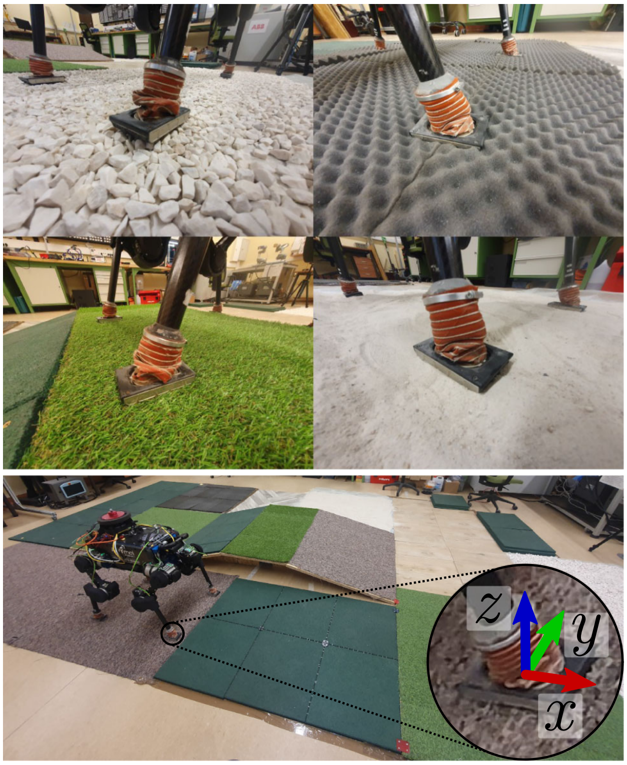


Fig. 10 Top: Examples of foot positioning when collecting data on different terrains under controlled laboratory conditions. Bottom: Experiment 3: ANYmal with sensorized flat feet standing on the multiple terrain type course. Close up of the foot is provided. An IMU and fore/torque sensor is located in the sole with coordinate frame shown

The robot was equipped with sensorized feet which feature high quality 6-axis force/torque sensors (Valsecchi et al. 2020). These feet are necessary to provide the signals for terrain classification as in Bednarek et al. (2019b). The robot autonomously walked between pre-programmed waypoints placed over the entire course, including several passes over the ramp. Large sections of the trajectory were only on the flat terrain tiles, forcing the algorithm to rely mostly on terrain classification for localization. Unlike Experiment 1, the robot was able to walk completely blind and no exteroception was used for footstep planning. A statically stable gait was used such that one foot was in the air at a time.

To demonstrate repeatability, we have performed three trials of this type, for a total distance traveled of more than 0.7 km and 1 h 7 min duration. We compare results produced using HL-G (Geometry), HL-C (Terrain Class) and HL-GC (Geometry and Terrain Classification). As the majority of the terrain course is flat, there is not enough information for geometry only localization to be continuously accurate. Only when using terrain class information as well as geometry can the robot localize in all parts of the terrain course.

A summary of the experiments is presented in Table 2, where HL-GC shows an overall improvement between 14 and 56% in the ATE compared to HL-G. Using only the prior knowledge of the terrain geometry and class, the robot stayed localized in all the runs and bounded the linearly growing drift of the state estimator. This can be seen in Fig. 11, where the estimated trajectory (in red) is able to stay near the ground truth trajectory (green). In areas where there are large patches of the same material, such as the gravel (dark blue) and ceramic (yellow), there is not enough information to localize in the xy-plane and the pose estimate drifts. When the robot crosses the boundary into a new terrain type the localization is able to correct.

Figure 13 shows in detail the estimator performance for each of the three linear dimensions and yaw. As in Experiment 1, the error on the odometry filter (TSIF, purple dashed line) of the robot is dominated by upward and yaw drift. This drift is estimated and compensated for by the haptic localization (red solid line), allowing accurate tracking of the ground truth (green solid line) in all dimensions.[1]

7 Discussion

The results presented in Sects. 6.2 and 6.3 demonstrate that terrain with a moderate degree of geometric complexity (such as Fig. 6) already provides enough information to bound the uncertainty of the robot's location. The effectiveness of a purely geometric approach is obviously limited by the actual terrain morphology in a real world situation, which would need to contain enough features such that all the DoF of the robot are constrained once the robot has touched them.

In the case where there is not enough geometric information, we have shown in Sect. 6.4 that terrain semantics can be used to localize. With sufficiently diverse terrain types (as shown in Fig. 10), boundary crossings from one terrain to another provide enough information to correct for drift in the xy-plane.

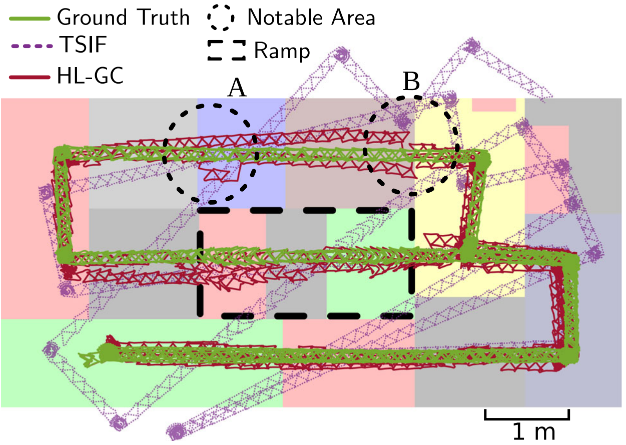


Fig. 11 Experiment 3, Trial 2: Top down view of the state estimator (dashed purple), HL-GC estimated trajectory (red) and ground truth (green). Trajectories are overlaid on terrain map with ramp shown in a dashed box. Two dotted circles show notable areas in the trajectory. **A** At a boundary crossing, the particle mean diverges from the ground truth. However, as the particle cloud nears the ramp, the geometric information gives higher likelihood to the particles in the center of the terrain. **B** Another boundary crossing which in two separated crossings triggers good localization updates (Color figure online)

7.1 Analysis of particle distribution on geometric terrain

Figure 14 Top shows the evolution of the particles up to the first half of the terrain course for Experiment 1, Trial 2. As the robot walks through, the particle cluster becomes concentrated, indicating good convergence to the most likely robot pose.

In the third subfigure, it can be noted how the probability distribution over the robot's pose follows a bimodal distribution, which is visible as two distinct clusters of particles. This situation justifies the use of particle filters, as they are able to model non-Gaussian distributions which can arise from a particular terrain morphology. In this case, the bimodal distribution is caused by the two identical gaps in between the chevrons. In such situations, a weighted average of the particle cluster would lead to a poor approximation of the true pose distribution. Therefore, the particle evolution illustrated in Sect. 5.1 is crucial to reject such an update.

Figure 14 Middle shows the particle distribution over flat ground. While not transitioning between terrain types, our method must rely on geometric information only and therefore is constrained in z but not x and y. In these cases we only update the drift estimate in the z direction which keeps the particle distribution near the ground but spread out.

7.2 Analysis of particle distribution on terrain class

Figure 12 shows data from Experiment 3, Trial 1. We compare results from HL-G, HL-GC and HL-C. We can see that

[1] A video showing all of these experiments is attached as supplementary material.

Table 2 Experiment 3: HL-G = Haptic Localization with only geometry; HL-C = Haptic Localization with only terrain class; HL-GC = Haptic Localization with both geometry and terrain class

Mean absolute translation error (ATE)						
Trial	Dist. (m)	Time (s)	TSIF (m)	HL-G (m)	HL-C (m)	HL-GC (m)
1	191	1114	0.64	0.23	0.63	**0.14**
2	331	1850	1.28	0.25	0.73	**0.11**
3	193	1090	0.72	0.21	0.61	**0.18**

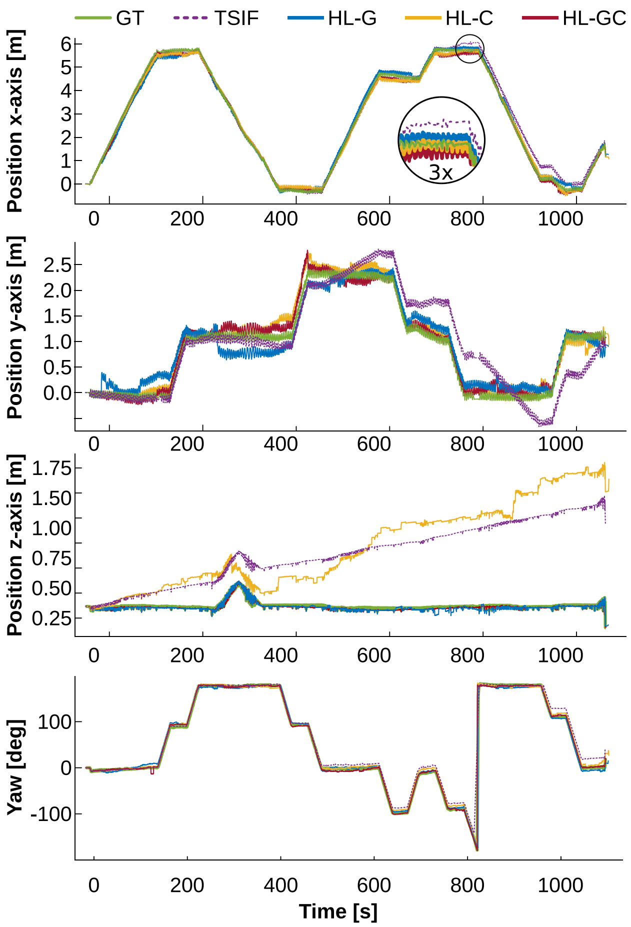


Fig. 12 Experiment 3, Trial 2: Here we show the difference in results when using only geometric information HL-G (blue), only terrain class information HL-C (yellow) and both HL-GC (red). We compare these to the state estimator (dashed purple) and ground truth (green) (Color figure online)

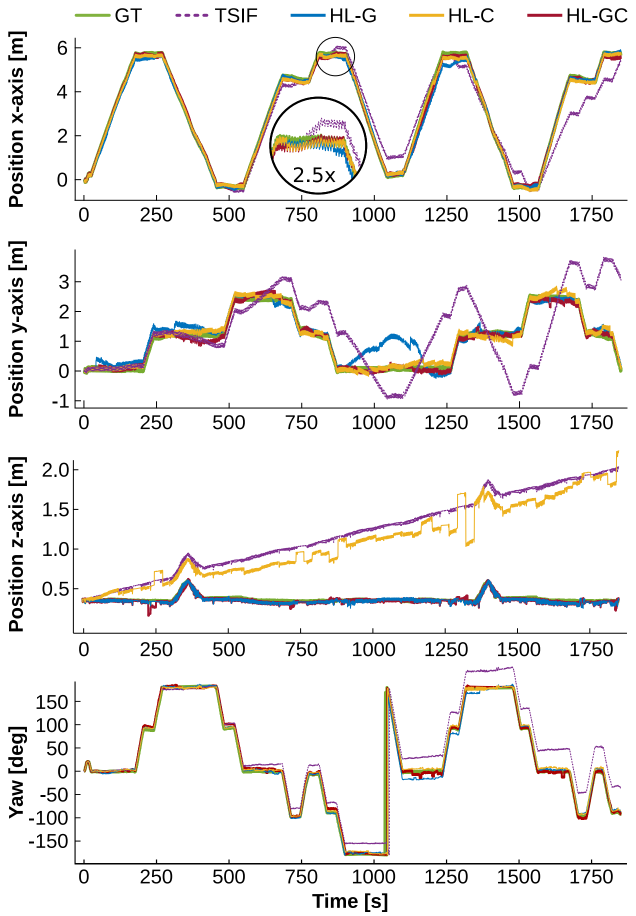


Fig. 13 Experiment 3, Trial 2: Here we show the difference in results when using only geometric information HL-G (blue), only terrain class information HL-C (yellow) and both HL-GC (red). We compare these to the state estimator (dashed purple) and ground truth (green) (Color figure online)

even with only class information, this method is able to keep pose estimation error bounded in the x–y plane (mean ATE for the class only trajectory was 0.63 m). In the third subplot from the top, the class only trajectory drifts upward in a similar way to TSIF in Fig. 13. This is because of the absence of any measurement in the z, hence our method relies on the proprioceptive state estimator.

Further analysis of the effect of terrain class on localization is shown in Fig. 14 middle. Here, we show the evolution of particles from HL-GC in Experiment 3, Trial 3. The particles, which initially are normally distributed around the starting position, quickly converge along the z axis as the floor elevation information is used. Once the boundary transition from green to red occurs, the particles correct for drift in the x direction.

Finally, in Fig. 14 bottom we show the behavior of HL-GC when the particles are initialized evenly across the entire map. Within 5 footsteps the distribution has converged to the green section in the bottom left. This is because there are only two green terrains, and the bottom left is the only one on flat ground.

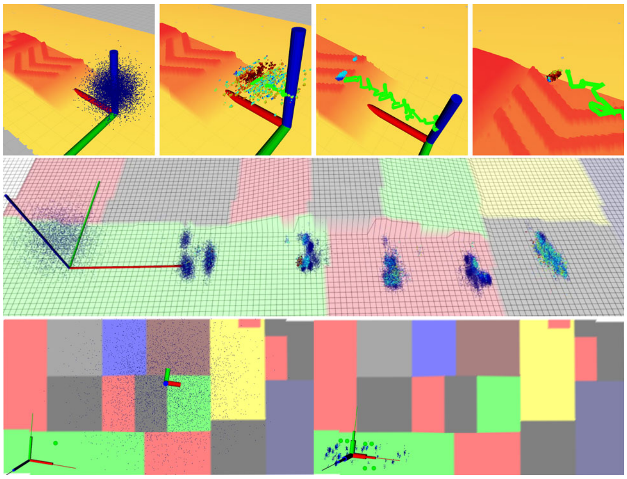


Fig. 14 Evolution of particle distributions during experiments. Particles are colorized by normalized weight according to the jet colormap (i.e., dark blue = lowest weight, dark red = highest weight). **Top (Experiment 1):** The green line indicates the ground truth trajectory. **A** At start, all the particles have the same weight and are normally distributed at the starting position. **B** After a few steps on the ramp, the robot pose is well estimated on x and z directions, but there is uncertainty on y. **C** When the robot approaches the chevron the particle set divides into two clusters, indicating two strong hypotheses as to the robot pose. **D** After a few more steps on the chevron, the robot is fully localized and the particles are tightly clustered. **Middle (Experiment 3):** As the robot walks from left to right, the particle cloud makes two terrain class transitions. As the robot crosses the first transition, the cloud becomes more narrow in the x direction as error along this axis is corrected. **Bottom (Experiment 3):** Top-down perspective with the robot estimate initialized in the middle of the map and the initial particle distribution spread out over the whole map. Within 5 steps the particle distribution has re-sampled over the correct green terrain patch. The green circles are the classified footsteps

8 Conclusion

We have presented a haptic localization algorithm for quadrupedal robots based on Sequential Monte Carlo methods. The algorithm can fuse geometric information (in 2.5D or 3D) as well as terrain semantics to localize against a prior map. We have demonstrated that even using only geometric information, walking over a non-degenerate terrain course containing slopes and interested geometry can reduce localization error to 10 cm. Our method also works if the robot probes vertical surfaces, measuring its environment in full 3D. Finally, we have shown how in areas of even sparser geometric information, terrain semantics can be used to augment this geometry.

The proposed approach demonstrated an average of 20 cm position error over all areas of a terrain course with different terrain classes and geometries. The ability to localize purely proprioceptively is valuable for repetitive autonomous tasks in vision-denied conditions, such as inspections of sewage systems. This method could also serve as a backup localization system in case of sensor failure—enabling a robot to complete its task and return to base.

8.1 Future work

The main limitation to our method is the need for sufficiently informative terrain. To mitigate this, we intend to incorporate other terrain properties such as slope or friction coefficient. Additionally, incorporating the network uncertainty as a prior on the terrain classification measurement would improve fusion with geometric information. Finally, we intend to generalize to more dynamic gaits and remove the costly dependency on sensorized feet.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10514-021-10013-w.

Acknowledgements This research has been conducted as part of the ANYbotics research community. It was part funded by the EU H2020 Project THING (Grant ID 780883) and a Royal Society University Research Fellowship (Fallon).

References

Bednarek, J., Bednarek, M., Kicki, P., & Walas, K. (2019a). Robotic Touch: Classification of materials for manipulation and walking. In *IEEE international conference on soft robotics (RoboSoft)* (pp. 527–533). https://doi.org/10.1109/ROBOSOFT.2019.8722819

Bednarek, J., Bednarek, M., Wellhausen, L., Hutter, M., & Walas, K. (2019b). What am I touching? Learning to classify terrain via haptic sensing. In *IEEE international conference on robotics and automation (ICRA)* (pp 7187–7193). https://doi.org/10.1109/ICRA.2019.8794478

Bloesch, M., Burri, M., Sommer, H., Siegwart, R., & Hutter, M. (2018). The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters, 3*(1), 573–580. https://doi.org/10.1109/LRA.2017.2776340.

Buchanan, R., Camurri, M., & Fallon, M. (2020). Haptic sequential Monte Carlo localization for quadrupedal locomotion in vision-denied scenarios. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*

Chalon, M., Reinecke, J., & Pfanne, M. (2013). Online in-hand object localization. In *IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE* (pp. 2977–2984). https://doi.org/10.1109/IROS.2013.6696778

Chitta, S., Vernaza, P., Geykhman, R., & Lee, D. (2007). Proprioceptive localization for a quadrupedal robot on known terrain. In *IEEE international conference on robotics and automation (ICRA)* (pp. 4582–4587). https://doi.org/10.1109/ROBOT.2007.364185

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Conference on empirical methods in natural language processing (EMNLP)*. https://doi.org/10.3115/v1/D14-1179.

Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In *International conference on learning representations (ICLR)*. http://arxiv.org/abs/1511.07289

Di Carlo, J., Wensing, P. M., Katz, B., Bledt, G., & Kim, S. (2018). Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1–9). https://doi.org/10.1109/IROS.2018.8594448

Fankhauser, P., Bjelonic, M., Dario Bellicoso, C., Miki, T., & Hutter, M. (2018). Robust rough-terrain locomotion with a quadrupedal robot. In *IEEE international conference on robotics and automation (ICRA)* (pp. 1–8). https://doi.org/10.1109/ICRA.2018.8460731

Fink, G., & Semini, C. (2020). Proprioceptive sensor fusion for quadruped robot state estimation. In *IEEE international conference on intelligent robots and systems (IROS)*

Focchi, M., Orsolino, R., Camurri, M., Barasuol, V., Mastalli, C., Caldwell, D. G., & Semini, C. (2020). *Heuristic planning for rough terrain locomotion in presence of external disturbances and variable perception quality* (pp. 165–209). Springer. https://doi.org/10.1007/978-3-030-22327-4_9

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017). On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics, 33*(1), 1–21. https://doi.org/10.1109/TRO.2016.2597321.

Fox, D., Thrun, S., Burgard, W., & Dellaert, F. (2001). *Particle filters for mobile robot localization* (pp. 401–428). Springer. https://doi.org/10.1007/978-1-4757-3437-9_19

Hartley, R., Ghaffari, M., Eustice, R. M., & Grizzle, J. W. (2020). Contact-aided invariant extended Kalman filtering for robot state estimation. *The International Journal of Robotics Research, 39*(4), 402–430. https://doi.org/10.1177/0278364919894385.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). http://arxiv.org/abs/1512.03385

Hoepflinger, M. H., Remy, C. D., Hutter, M., & Siegwart, R. (2010). Haptic terrain classification on natural terrains for legged robots. In *International conference on climbing and walking robots (CLAWAR)* (pp. 785–792). https://doi.org/10.1142/9789814329927_0097

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., & Hoepflinger, M. (2016). ANYmal—A Highly mobile and dynamic quadrupedal robot. In *IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE* (pp. 38–44). https://doi.org/10.1109/IROS.2016.7758092

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of machine learning research* (Vol. 37, pp. 448–456). Lille, France: PMLR.

Kolvenbach, H., Bärtschi, C., Wellhausen, L., Grandia, R., & Hutter, M. (2019). Haptic inspection of planetary soils with legged robots. *IEEE Robotics and Automation Letters, 4*(2), 1626–1632. https://doi.org/10.1109/LRA.2019.2896732.

Kolvenbach, H., Wisth, D., Buchanan, R., Valsecchi, G., Grandia, R., Fallon, M., & Hutter, M. (2020). Towards autonomous inspection of concrete deterioration in sewers with legged robots. *Journal of Field Robotics, 37,* 1314–1327. https://doi.org/10.1002/rob.21964.

Koval, M. C., Pollard, N. S., & Srinivasa, S. S. (2016). *Manifold representations for state estimation in contact manipulation* (pp. 375–391). Springer. https://doi.org/10.1007/978-3-319-28872-7_22

Krotkov, E. (1990). Active perception for legged locomotion: Every step is an experiment. In *Proceedings of the 5th IEEE international symposium on intelligent control 1990* (Vol. 1, pp. 227–232). https://doi.org/10.1109/ISIC.1990.128462

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., & Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science Robotics*. https://doi.org/10.1126/scirobotics.abc5986.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International conference on learning representations*. http://arxiv.org/abs/1711.05101

Manuelli, L., & Tedrake, R. (2016). Localizing external contact using proprioceptive sensors: The contact particle filter. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5062–5069).

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., & Nieto, J. (2017). Voxblox: Incremental 3d Euclidean signed distance fields for on-board MAV planning. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Schwendner, J., Joyeux, S., & Kirchner, F. (2014). Using embodied data for localization and mapping. *Journal of Field Robotics, 31*(2), 263–295. https://doi.org/10.1002/rob.21489.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(56), 1929–1958.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A Benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 573–580). https://doi.org/10.1109/IROS.2012.6385773

Valsecchi, G., Grandia, R., & Hutter, M. (2020). Quadrupedal locomotion on uneven terrain with sensorized feet. *IEEE Robotics and Automation Letters, 5*(2), 1548–1555. https://doi.org/10.1109/LRA.2020.2969160.

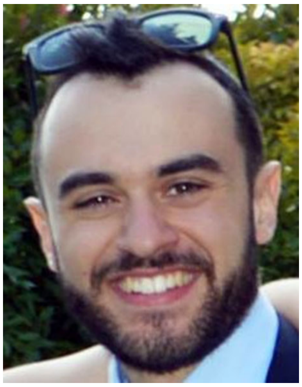
Vespa, E., Nikolov, N., Grimm, M., Nardi, L., Kelly, P. H. J., & Leutenegger, S. (2018). Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters, 3*(2), 1144–1151. https://doi.org/10.1109/LRA.2018.2792537.

Vezzani, G., Pattacini, U., Battistelli, G., Chisci, L., & Natale, L. (2017). Memory unscented particle filter for 6-DOF tactile localization. *IEEE Transactions on Robotics, 33*(5), 1139–1155. https://doi.org/10.1109/TRO.2017.2707092.

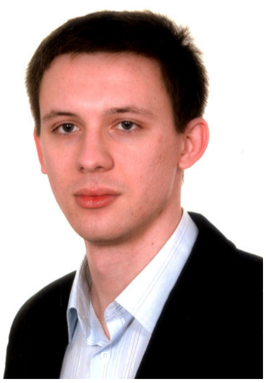
Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., & Hutter, M. (2019). Where should i walk? Predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters, 4*(2), 1509–1516. https://doi.org/10.1109/LRA.2019.2895390.

Russell Buchanan is a Ph.D. candidate at the University of Oxford in the Oxford Robotics Institute, Dynamic Robot Systems Group. In 2016 he received a Bachelor of Engineering (Honours) degree in Electrical Engineering from McGill University in Montréal, Canada. Then, in 2018 he completed an MSc in Robotics, Systems and Control at ETH Zurich in Zurich, Switzerland. This included a research visit to the Robotics and Autonomous Systems Group of CSIRO in Brisbane, Australia. His research interests are in proprioceptive localization and state estimation for legged robots.

Jakub Bednarek graduated from Poznan University of Technology (PUT) receiving BSc and MSc in Computer Science. He is a Ph.D. student at the Faculty of Computer Science at PUT, conducting research on artificial intelligence, and currently, he is a Research Assistant in the Institute of Robotics and Machine Intelligence at PUT. His research focuses on the development of artificial intelligence methods, including research on how to explain their operation, and its application in robotics.



Marco Camurri received the B.Eng. and M.Eng. degrees in Computer Engineering from the University of Modena and Reggio Emilia (Modena, Italy) in 2009 and 2012, respectively. He received his Ph.D. degree in Advanced Robotics in 2017 from the Istituto Italiano di Tecnologia (IIT) in Genoa, Italy. From 2017 to 2020 he was postdoc at IIT (one year) and University of Oxford (two years). Since 2021, he is Senior Research Associate at the Oxford Robotics Institute, University of Oxford. His research interests include legged state estimation, mobile robot perception, sensor fusion, mapping, and mobile autonomous navigation.



Michal R. Nowicki graduated from Poznan University of Technology (PUT) in Poland receiving BSc in Computer Science and both BSc and MSc in Automatic Control and Robotics. He received (with honors) Ph.D. in Robotics in 2018 for his thesis concerning information fusion using factor graphs in SLAM. Currently, he is a Research Assistant Professor in the Institute of Robotics and Machine Intelligence at PUT, Poland. His research interests are related to optimization-based methods for multi-sensor fusion including problems related to localization, mapping, and sensory setup calibration.

Krzysztof Walas graduated from Poznan University of Technology (PUT) in Poland receiving MSc in Automatic Control and Robotics. He received (with honours) Ph.D. in Robotics in 2012 for his thesis concerning legged robots locomotion in structured environments. Currently, he is an Assistant Professor in the Institute of Robotics and Machine Intelligence at PUT, Poland. His research interests are related to robotic perception for physical interaction applied both to walking and grasping tasks.

Maurice Fallon received his B.Eng (electrical engineering) from University College Dublin, Ireland and his Ph.D. (acoustic source tracking) from University of Cambridge, UK. From 2008–2012 he was a post-doc and research scientist in MIT Marine Robotics Group working on SLAM. Later, he was the perception lead of MIT's team in the DARPA Robotics Challenge. From 2017 he has been a Royal Society University Research Fellow and Associate Professor at University of Oxford, UK. He leads the Dynamic Robot Systems Group in Oxford Robotics Institute. His research focuses on probabilistic methods for localization, mapping, dynamic motion planning, and multi sensor fusion.