

# MONTE CARLO LOCALIZATION FOR HUMANOID ROBOT NAVIGATION IN COMPLEX INDOOR ENVIRONMENTS

ARMIN HORNUNG

STEFAN OSSWALD

DANIEL MAIER

MAREN BENNEWITZ

*Department of Computer Science, University of Freiburg,  
Georges-Koehler-Allee 74, 79110 Freiburg, Germany*

*{hornunga, osswald, maierd, maren}@informatik.uni-freiburg.de*

Received 20 June 2013

Revised 12 December 2013

Accepted 23 February 2014

To appear in: Int. Journal of Humanoid Robotics (IJHR). **Author's preprint.**

Accurate and reliable localization is a prerequisite for autonomously performing high-level tasks with humanoid robots. In this article, we present a probabilistic localization method for humanoid robots navigating in arbitrary complex indoor environments using only onboard sensing, which is a challenging task. Inaccurate motion execution of biped robots leads to an uncertain estimate of odometry, and their limited payload constrains perception to observations from lightweight and typically noisy sensors. Additionally, humanoids do not walk on flat ground only and perform a swaying motion while walking, which requires estimating a full 6D torso pose. We apply Monte Carlo localization to globally determine and track a humanoid's 6D pose in a given 3D world model, which may contain multiple levels and staircases. We present an observation model to integrate range measurements from a laser scanner or a depth camera as well as attitude data and information from the joint encoders. To increase the localization accuracy, e.g., while climbing stairs, we propose a further observation model and additionally use monocular vision data in an improved proposal distribution. We demonstrate the effectiveness of our methods in extensive real-world experiments with a Nao humanoid. As the experiments illustrate, the robot is able to globally localize itself and accurately track its 6D pose while walking and climbing stairs.

*Keywords:* 3D, Localization, Navigation, Range Sensing, Laser, RGB-D, Vision

## 1. Introduction

The capability to robustly navigate in complex indoor environments, possibly consisting of different levels connected by steps and staircases, is a prerequisite for humanoid robots to fulfill high-level tasks such as delivery, home-care, or even disaster relief as motivated in the current DARPA robotics challenge.

One main problem of biped robots is the inaccurate execution of motions due to their kinematic structure and joint backlash. While walking, this typically leads to



Fig. 1. Nao humanoid with a laser range finder navigating in a complex environment containing multiple levels, a staircase, and a ramp (Environment I). Using our localization approach, the robot is able to accurately estimate its 6D pose while walking and climbing stairs.

foot slippage and consequently to only noisy and inaccurate odometry estimation. A second challenge is the limited payload of humanoid robots. Lightweight sensors that are typically noisy have to be used for perception and robust techniques for localizing a humanoid robot under these constraints are needed.

Humanoids usually cannot be assumed to move on a plane to which their sensors are parallel due to their swaying walking motion. Additionally, their capability to step over or onto objects needs to be taken into account when planning in environments containing staircases or obstacles on the floor. Thus, a 2D grid map representation is not sufficient for navigation with humanoid robots. Accordingly, 2.5D models that also represent the height of objects are often used in the humanoid robotics community.<sup>1,2,3</sup> However, for arbitrary environments containing several levels (e.g., as shown in Fig. 1), a full 3D representation is needed to store free and occupied areas in a volumetric way.<sup>4</sup> For reliably performing navigation tasks, a robot must be able to globally determine its pose in such a model and accurately track it over time. When operating in non-planar, multi-level environments, a robot hence needs to estimate a 6D state that contains the height above the ground as well as the roll and pitch angles in addition to the 2D position and yaw angle.

We apply Monte Carlo localization (MCL) to estimate the robot's 6D torso pose in a 3D environment representation using range data, e.g., from a laser or depth camera. Furthermore, our system integrates data provided by an attitude sensor and information from the joint encoders of the robot. To increase the localization accuracy in critical situations, e.g., when climbing stairs, we additionally use data from a monocular camera in an improved proposal distribution within the particle filter. As we show in extensive real-world experiments with a Nao humanoid, our

approach enables the robot to determine its global 6D pose and accurately track it while walking and climbing stairs. As a further contribution, our implementation is available open source as a ROS package.<sup>a</sup>

This work presents a unified approach of our previous localization frameworks<sup>5,6,7</sup> with a thorough evaluation of different sensors and observation models, and provides a method for odometry calibration.

The remainder of this article is structured as follows. We first discuss related work in the next section. Sec. 3 describes our range data based 6D localization technique including the odometry calibration. In Sec. 4, we present an observation model for vision data and our technique to combine it with range data by means of an improved proposal distribution within the particle filter. Finally, the experiments presented in Sec. 5 demonstrate the robustness and accuracy of our localization approach in a number of experiments.

## 2. Related Work

In the last few years, many approaches for tracking the pose of humanoids in the two-dimensional space have been presented. For example, Ido *et al.*<sup>8</sup> apply a vision-based approach and compare the current image to previously recorded reference images in order to estimate the location of the robot. Oßwald *et al.*<sup>9</sup> and Bennwitz *et al.*<sup>10</sup> compare visual features to a previously learned 2D feature map during pose tracking. Pretto *et al.*<sup>11</sup> track visual features over time for estimating the robot's odometry. Cupec *et al.*<sup>12</sup> detect objects with given shapes and colors in the local environment of the humanoid and determine its pose relative to these objects. Seara and Schmidt<sup>13</sup> use stereo vision and known landmarks for estimating the humanoid's foot positions in the 2D plane with a Kalman filter while walking. By actively controlling the gaze, the authors reduce the error of the estimated poses. Furthermore, techniques using laser range data have also been developed. Stachniss *et al.*<sup>14</sup> presented an approach to learn accurate 2D grid maps of large environments with a humanoid equipped with a Hokuyo laser scanner. Such a map was subsequently used by Faber *et al.*<sup>15</sup> for humanoid localization in 2D. Similarly, Tellez *et al.*<sup>16</sup> developed a navigation system for such a 2D environment representation. Their SLAM approach builds 2D maps from odometry and laser range finders located in the humanoid's feet.

Since a 2D map is often not sufficient for humanoid navigation, several methods use 2.5D grid maps that additionally store a height value for each cell. Thompson *et al.*<sup>1</sup> track the 6D pose of a humanoid equipped with a Hokuyo URG-04LX laser scanner in such a representation. However, they assume that the robot is only walking on flat ground, constraining height, roll, and pitch within fixed thresholds. In further approaches, authors use only odometry data to estimate the robot's pose. E.g., Chestnutt *et al.*<sup>2</sup> and Nishiwaki *et al.*<sup>17</sup> construct a local 2.5D height

<sup>a</sup>[http://www.ros.org/wiki/humanoid\\_localization](http://www.ros.org/wiki/humanoid_localization)

map from a tilting laser range finder located in a humanoid's hip based on the estimated motion. Gutmann *et al.*<sup>18</sup> build a local height map for obstacle avoidance from stereo data. To avoid problems resulting from the accumulated error due to odometry drift, old data is discarded after a short period of time in these approaches. In recent work, Ahn *et al.*<sup>19</sup> propose to fuse kinematic and visual odometry with IMU data in order to estimate the pose of a humanoid walking with a heel-toe motion. While this improves the odometry estimate, it is still subject to drift over time. With such local tracking based on odometry, it is also not possible to globally determine the robot's initial position in a given map.

Michel *et al.*<sup>3</sup> localize a humanoid with respect to a close object. The authors apply a model-based approach to track the 6D pose of a manually initialized object relative to the camera. Stasse *et al.*<sup>20</sup> and Alcantarilla *et al.*<sup>21</sup> proposed an approach to simultaneously localizing the robot and mapping the environment. The authors combine vision and motion information to estimate the pose and velocities of the camera as well as visual feature positions in 3D while the robot is walking on a small circle. Similarly, Kwak *et al.*<sup>22</sup> discussed grid-based particle filter SLAM for the humanoid HRP-2. In their approach, they build a grid map of 10 cm resolution with point clouds from the robot's stereo vision cameras.

Finally, there exist navigation systems for humanoid robots which use external sensors to track the robot's pose, e.g., as used by Michel *et al.*<sup>23</sup>, Stilman *et al.*<sup>24</sup>, or Chestnutt *et al.*<sup>2</sup> However, external sensing is usually not practical to employ outside a lab setting.

In contrast to all of these approaches, we present a system which is able to accurately determine the complete 6D pose of a humanoid robot in a 3D representation of a complex, multi-level environment using only on-board sensors. The use of range sensors leads to a high accuracy of the estimated pose. We hereby extend our previous work on localization<sup>5,6,7</sup> with a unified approach for range measurements and vision data, compare different sensors and observation models, and improve the performance by means of odometry calibration.

### 3. 6D Localization for Humanoid Robots

Our approach estimates the six-dimensional pose  $\mathbf{x} = (x, y, z, \varphi, \theta, \psi)$  of the robot, i.e., it maintains a belief about the 3D position  $(x, y, z)$  and the roll, pitch, and yaw angles  $(\varphi, \theta, \psi)$  of the robot's body reference frame in the global 3D map of the environment. This reference frame is located in the center of the humanoid's torso, which is also the origin of all of its kinematic chains. For estimating the robot's 6D state, we apply Monte Carlo localization<sup>25</sup>, which we briefly recapitulate in the following.

### 3.1. Monte Carlo Localization (MCL)

MCL is a Bayes filtering technique that recursively estimates the posterior about the robot's pose  $\mathbf{x}_t$  at time  $t$ :

$$p(\mathbf{x}_t | m, \mathbf{o}_{1:t}, \mathbf{u}_{1:t}) = \eta \cdot \overbrace{p(\mathbf{o}_t | m, \mathbf{x}_t)}^{\text{observation model}} \cdot \int_{\mathbf{x}_{t-1}} \underbrace{p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)}_{\text{motion model}} \cdot \underbrace{p(\mathbf{x}_{t-1} | \mathbf{o}_{1:t-1}, \mathbf{u}_{1:t-1})}_{\text{recursive term}} d\mathbf{x}_{t-1} \quad (1)$$

Here,  $\eta$  is a normalization constant resulting from Bayes' rule,  $\mathbf{u}_{1:t}$  denotes the sequence of all odometry readings up to time  $t$ ,  $\mathbf{o}_{1:t}$  is the sequence of all observations, and  $m$  the environment model or map. The term  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$  is called motion model and denotes the probability that the robot ends up in state  $\mathbf{x}_t$  given it executed the motion  $\mathbf{u}_t$  in state  $\mathbf{x}_{t-1}$ . The observation or sensor model  $p(\mathbf{o}_t | m, \mathbf{x}_t)$  denotes the likelihood of obtaining observation  $\mathbf{o}_t$  given the robot is at pose  $\mathbf{x}_t$  in the map  $m$ .

In MCL, the belief distribution over the robot's current state is approximated by a set of  $n$  weighted samples or pose hypotheses  $\{\langle \mathbf{x}_t^{(1)}, w_t^{(1)} \rangle, \dots, \langle \mathbf{x}_t^{(n)}, w_t^{(n)} \rangle\}$ . Here, each  $\mathbf{x}_t^{(i)}$  is one pose hypothesis and  $w_t^{(i)}$  is the corresponding weight. The weight of a particle is proportional to the likelihood that the robot is in that state. The update of the belief, also called particle filtering, consists of the following steps:

- (1) **Prediction:** The current set of particles is propagated according to the motion model  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ .
- (2) **Correction:** The importance weight of each particle is computed according to the observation model  $p(\mathbf{o}_t | m, \mathbf{x}_t)$  given the map.
- (3) **Resampling:** New particles at time  $t+1$  are drawn with replacement from the distribution proportional to the particle weights  $w_t^{(i)}$ . Afterwards, their weights are reset to  $w_{t+1}^{(i)} = \frac{1}{n}$ . This step ensures that the filter converges to pose hypotheses with high likelihoods.

The filter is either initialized with a distribution of equally weighted samples around an initial pose estimate for tracking, or with a uniform distribution over all possible hypotheses for global localization.

### 3.2. Motion Model

In the prediction step of MCL, a new pose is drawn for each particle according to the motion model  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ . In the approach presented in this article,  $\mathbf{u}_t$  corresponds to the incremental motion of the humanoid's torso in a local coordinate frame during navigation. In general, the humanoid's estimated odometry pose as 6D rigid body transform at time  $t$  can then be computed as a concatenation of the incremental motions

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} \otimes T(\mathbf{u}_t) \quad (2)$$

in an arbitrary odometry coordinate frame.  $T(\mathbf{u}_t)$  denotes the 6D rigid body transform of the incremental motion  $\mathbf{u}_t$ .  $\mathbf{u}_t$  is computed with forward kinematics from the current stance foot while walking. This kinematic odometry leads to an accurate 6D pose of the humanoid under ideal conditions. However, the feet of the robot typically slip on the ground due to inaccurate execution of motions. Additionally, gear backlash in individual joints can aggravate the problem. Hence, in practice, only noisy estimates of the odometry are available. Consequently, a particle filter has to account for that noise with a higher variance, requiring a higher number of particles and thus more computational power for successful pose estimation.

An alternative source of odometry could be provided by visual odometry.<sup>11,19</sup> By using visual cues from a monocular camera to estimate the relative pose changes, the estimated motions are less affected by drift. However, visual odometry requires additional computational power to detect and match features.

### 3.2.1. Motion Model Calibration

By learning or calibrating the parameters of the particle filter motion model, the localization performance can be increased in terms of both computational efficiency and accuracy. Here, we consider the most general case of any kind of 3D positional and rotational displacement (omnidirectional walking motion). For calibration, we assume that systematic drift affects the motion reported by odometry in the 2D plane, i.e., only  $(\tilde{x}, \tilde{y}, \tilde{\psi})$  are affected from  $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{\varphi}, \tilde{\theta}, \tilde{\psi})$ . This is not a strong restriction as long as the humanoid walks on a solid surface, since its motion is constrained by this surface and it cannot leave the ground. Even when climbing challenging terrain such as stairs, the drift of the motion occurs in the 2D plane of the stance foot as long as the robot does not fall or slide down a slope. General noise in the kinematic estimate of the humanoid's height above the ground does not lead to a systematic drift. For odometry calibration, we will refer to the reduced state vectors  $\mathbf{x}' = (x, y, \psi)^\top$  containing the 2D position and orientation.

Corresponding to Eq. (2),  $\mathbf{u}'_t = (u_{x,t}, u_{y,t}, u_{\psi,t})^\top$  estimates the displacement between two poses reported by odometry

$$\tilde{\mathbf{x}}'_t = \tilde{\mathbf{x}}'_{t-1} + \mathbf{u}'_t. \quad (3)$$

To calibrate the drift of  $\mathbf{u}'_t$ , we require a ground truth pose  $\mathbf{x}'$  available in a prior learning phase, e.g., from a motion capture system, scan matching, or visual odometry. Based on the deviations from ground truth, values of a calibration matrix  $\mathbf{M} \in \mathbb{R}^{3 \times 3}$  can be determined to correct the 2D drift of odometry, such that

$$\mathbf{x}'_t = \mathbf{x}'_{t-1} + \mathbf{M} \mathbf{u}'_t. \quad (4)$$

We describe how such a calibration matrix can be learned in the next section.

To account for general motion noise, the particle prediction step adds multivariate Gaussian noise to  $\mathbf{u}'_t$  for each particle  $i$ , whereas the systematic drift is accounted for by the mean of the Gaussian distribution  $\mathcal{N}$ . As Eliazar and Parr<sup>26</sup>, we assume

independent noise sources for the drift in forwards and sideways direction, as well as orientation. This results in a noise model in which the mean depends linearly on  $\mathbf{u}'_t$  and the variance quadratically. Thus, the prediction step samples for particle  $i$  a new pose according to:

$$\mathbf{x}'_t^{(i)} = \mathbf{x}'_{t-1}^{(i)} + \mathcal{N}(\mathbf{M} \mathbf{u}'_t, \mathbf{\Sigma} \mathbf{u}'_t{}^2). \quad (5)$$

Here,  $\mathbf{\Sigma}$  is a matrix containing vectors of noise parameters and  $\mathbf{u}'_t{}^2 = (u_{x,t}^2, u_{y,t}^2, u_{\psi,t}^2)^\top$ . The first element of  $\mathbf{x}'_t^{(i)}$ , for example, results in

$$x_t^{(i)} = x_{t-1}^{(i)} + \mathcal{N}(M_{xx}u_{x,t} + M_{xy}u_{y,t} + M_{x\psi}u_{\psi,t}, \sigma_{xx}^2u_{x,t}^2 + \sigma_{xy}^2u_{y,t}^2 + \sigma_{x\psi}^2u_{\psi,t}^2). \quad (6)$$

By filling the remaining values for  $z$ , roll, and pitch we obtain the final motion model  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$  in 3D. Since these dimensions are constrained by the ground plane when walking, we assume zero-mean noise variances that depend on the traveled distance since the last odometry reading.

### 3.2.2. Learning the Motion Model Parameters

In this section, we introduce our approach to learn the motion model parameters so that the Gaussian distribution in Eq. (5) represents the actual drift and noise by the walking humanoid.

Let  $\mathbf{U} \in \mathbb{R}^{N \times 3}$ ,  $U_i = (u_{x,i}, u_{y,i}, u_{\psi,i})$  contain the incremental odometry motion estimates for each of the  $N$  timesteps acquired while recording the ground truth data for calibration, e.g., with an external tracking system. Following the idea of least-squares odometry estimation by Eliazar and Parr<sup>26</sup>,  $\mathbf{M}_x = (M_{xx}, M_{xy}, M_{x\psi})^\top$  is the column vector containing the elements of the mean calibration matrix  $\mathbf{M}$  that influence  $x$ , and  $\mathbf{X}_x \in \mathbb{R}^{N \times 1}$  contains the ground truth measurements of the incremental displacements in  $x$  direction. We can now solve the overdetermined system

$$\mathbf{U} \mathbf{M}_x = \mathbf{X}_x \quad (7)$$

for  $\mathbf{M}_x$  with least squares to determine the first row of  $\mathbf{M}$ . The remaining rows  $\mathbf{M}_y$  and  $\mathbf{M}_\psi$  can be solved analogously.

To estimate the variance parameters  $\mathbf{\Sigma}$ , we define  $\mathbf{U}^2 \in \mathbb{R}^{N \times 3}$  as a vector of squared odometry motions  $U_i^2 = (u_{x,i}^2, u_{y,i}^2, u_{\psi,i}^2)$ ,  $\mathbf{\Sigma}_x = (\Sigma_{xx}, \Sigma_{xy}, \Sigma_{x\psi})^\top$  as the column vector of variances in  $x$ , and  $\mathbf{X}_{x^2} \in \mathbb{R}^{N \times 1}$  as the measurements on the variances such that  $X_{x^2,i} = (U_i \mathbf{M}_x - X_{x,i})^2$ . We can now solve the linear equation

$$\mathbf{U}^2 \mathbf{\Sigma}_x = \mathbf{X}_{x^2}, \quad (8)$$

and the corresponding ones for  $\mathbf{\Sigma}_y$  and  $\mathbf{\Sigma}_\psi$ .

Fig. 2 shows the difference between a calibrated and an uncalibrated motion model for a humanoid walking straight and on an arc. The calibrated motion model properly captures the drift since the particle distribution is close to the ground

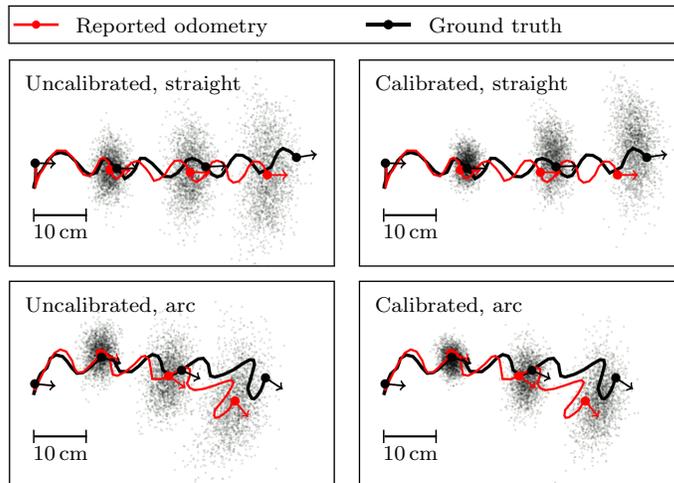


Fig. 2. Comparison between an uncalibrated odometry motion model (left) and a calibrated one (right) for walking straight and on an arc with a Nao humanoid. The uncalibrated motion model requires a larger variance for the particles to account for the systematic drift. 2000 particles were used in each iteration for this visualization.

truth. Compared to that, the uncalibrated motion model requires larger variances, and thus more particles, to account for systematic drift.

### 3.3. Observation Model

The belief about the humanoid’s 6D state is updated based on three different sources of sensor information contained in one observation  $\mathbf{o}_t$ . First, range measurements  $\mathbf{r}_t$  provided by a laser range finder or depth camera are integrated. Second, we regard the height  $\tilde{z}_t$  of the humanoid’s torso above the current ground plane as a measurement of its joint encoders and also integrate the angles for roll  $\tilde{\varphi}_t$  and pitch  $\tilde{\theta}_t$  as estimated by the noisy IMU. Since all these measurements are independent, the observation model decomposes to the product

$$\begin{aligned}
 p(\mathbf{o}_t \mid m, \mathbf{x}_t) &= p(\mathbf{r}_t, \tilde{z}_t, \tilde{\varphi}_t, \tilde{\theta}_t \mid m, \mathbf{x}_t) \\
 &= p(\mathbf{r}_t \mid m, \mathbf{x}_t) \cdot p(\tilde{z}_t \mid m, \mathbf{x}_t) \cdot p(\tilde{\varphi}_t \mid \mathbf{x}_t) \cdot p(\tilde{\theta}_t \mid \mathbf{x}_t).
 \end{aligned} \tag{9}$$

#### 3.3.1. 3D Environment Representation

For humanoid navigation in non-planar, possibly multi-level environments, a full 3D occupancy grid map is necessary since the map needs to encode both occupied and free volumes for arbitrary structures. In our system, we employ OctoMap, an efficient implementation of 3D occupancy grids based on octrees.<sup>4</sup> This enables our system to use map resolutions as small as 1 cm for a complete 3D indoor map. In this work, we assume a volumetric 3D representation of the environment as given

and leave map building and SLAM for future research.

### 3.3.2. Range Measurements

To integrate range measurements, usually the endpoint model (also called likelihood field) or raycasting (also called beam model) are used for  $K$  beams  $r_{t,k}$  of a range measurement  $\mathbf{r}_t$  that consists of a complete scan from a laser or a depth image from a camera.

In the endpoint model<sup>27</sup>, the likelihood of a single range measurement  $r_{t,k}$  depends on the distance  $D(r_{t,k})$  of the corresponding hypothetical beam endpoint to the closest obstacle represented in the map, i.e.,

$$p(r_{t,k} | m, \mathbf{x}_t) = \phi(D(r_{t,k}), \sigma_r), \quad (10)$$

with the Gaussian distribution

$$\phi(d, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d^2}{2\sigma^2}\right). \quad (11)$$

Here,  $\sigma_r$  is the standard deviation of the sensor noise. For a given 3D map, the distances  $D$  can be pre-computed for all 3D cells with a Euclidean distance transform.<sup>28</sup> Since the lookup can then be implemented as a single query, the endpoint model is a popular and efficient choice for laser-based localization.<sup>29</sup> However, it neglects the beam characteristics of range sensors, which cannot pass obstacles between the sensor origin and its endpoint. This potentially leads to a less accurate localization especially in three-dimensional environment structures.

In the ray casting model, a ray is cast from the origin of the sensor in the current beam direction until it hits an obstacle. This returned distance  $\tilde{d}(r_{t,k})$  is then compared with the actually measured beam distance  $d(r_{t,k})$ :

$$p(r_{t,k} | m, \mathbf{x}_t) = \phi\left(\tilde{d}(r_{t,k}) - d(r_{t,k}), \sigma_r\right). \quad (12)$$

While the ray casting operation is computationally demanding particularly in 3D, it is better informed of three-dimensional structures, which can result in an improved localization accuracy. In Sec. 5, we will compare the performance of raycasting with the endpoint model. For efficiency, we rely on a parallelized raycasting implementation in our 3D mapping framework OctoMap.<sup>4</sup>

As presented by Thrun *et al.*<sup>30</sup>, we extend the single probabilities  $p(r_{t,k})$  to a full probabilistic model including terms for maximum range measurements to account for sensor errors and random measurements to account for unmodeled objects in the sensor beam.

The integration of a full scan  $\mathbf{r}_t$  is then computed as the product of the beam likelihoods

$$p(\mathbf{r}_t | m, \mathbf{x}_t) = \prod_{k=1}^K p(r_{t,k} | m, \mathbf{x}_t) \quad (13)$$

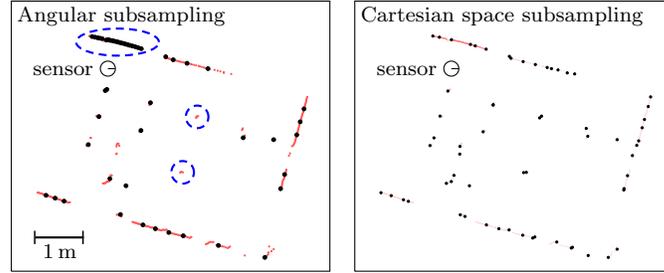


Fig. 3. Comparison of different techniques for selecting a subset of endpoints (black dots) from a 2D laser scan (light red dots): Angular subsampling takes every  $n$ -th point (left), whereas our method subsamples the end points based on their distance in Cartesian space (right). Highlighted by dashed ellipses are potential problems in angular subsampling: Close obstacles receive a higher weighting by more endpoints and thin obstacles can be missed.

with the common assumption of conditional independence between the beams. In practice, the independence can be assured by using only a subset of all the available range measurements. A common practice is to use every  $n$ -th beam, e.g. at intervals of  $5^\circ$ . While this angular subsampling is straightforward to implement, it exhibits a number of problems, as highlighted in Fig. 3. First, endpoints on close obstacles are spaced close to each other while obstacles further away only have few points on them. This results in an overconfidence on close-range readings, since they are no longer conditionally independent. Second, thin obstacles that are observed only with few endpoints but could provide unique features for localization can be missed. To overcome these problems, we implement a subsampling of the endpoints in the Cartesian space. All points falling into one cell of an equally-spaced grid are approximated by their centroid. As can be seen in Fig. 3 (right), our Cartesian space subsampling strategy results in regularly-spaced endpoints that represent the actual scan much closer with the same number of points. The same technique can be applied to 3D point clouds, e.g., from depth or stereo cameras.

### 3.3.3. Roll, Pitch, and Height Measurements

Furthermore, we need to integrate the torso height  $\tilde{z}_t$  above the ground plane as computed from the values of the joint encoders and the roll  $\tilde{\varphi}_t$  and pitch  $\tilde{\theta}_t$  provided by the IMU. Here, we evaluate the difference between the quantities predicted according to the motion model and their measured values. Similar to the endpoint model for range measurements, their likelihoods are computed according to the distance function in Eq. (11):

$$p(\tilde{z}_t \mid m, \mathbf{x}_t) = \phi(z_{t,\text{ground}} - \tilde{z}_t, \sigma_z) \quad (14)$$

$$p(\tilde{\varphi}_t \mid \mathbf{x}_t) = \phi(\varphi_t - \tilde{\varphi}_t, \sigma_\varphi) \quad (15)$$

$$p(\tilde{\theta}_t \mid \mathbf{x}_t) = \phi(\theta_t - \tilde{\theta}_t, \sigma_\theta), \quad (16)$$

where  $\sigma_z$ ,  $\sigma_\phi$ , and  $\sigma_\theta$  are given by the noise characteristics of the joint encoders and the IMU, and  $z_{t,\text{ground}}$  is computed from the height difference between  $z_t$  and the closest ground level in the map.

Finally, the complete measurement update step of the localization can be combined to the product of Eq. (13)-(16) according to Eq. (9) by choosing appropriate weighting factors for the individual likelihoods.

### 3.4. Global Localization and Relocalization in Multi-level Environments

When the robot has no initial guess about its pose, it needs to estimate the pose globally. In MCL, this is done by initially distributing the pose hypotheses over possible robot poses in the whole environment, in our case also over all levels. To efficiently draw robot pose hypotheses, we sample  $x$ ,  $y$ , and  $\psi$  uniformly within free areas of the map and  $z$  from the probability distribution given by Eq. (14). For each sampled 2D position  $(x, y)$ , we hereby consider the initial height measurement  $\tilde{z}_0$  at all represented ground levels. Similarly, roll and pitch are sampled according to Eq. (15) and (16) for an initial IMU measurement  $(\tilde{\varphi}_0, \tilde{\theta}_0)$ .

Obviously, global localization requires more particles than pose tracking. However, once the initial particle cloud has converged, the robot's pose can be tracked using fewer particles. Convergence of the particle cloud can be detected by the spread of the particle poses.

Overconfidence in the robot's pose could cause the localization estimate to lose track of the actual robot pose. This problem, and the related *kidnapped robot* problem, can be solved by globally injecting random particles.<sup>30</sup>

## 4. Improved Proposals for Highly Accurate Pose Tracking

During challenging whole-body motions such as stair climbing, odometry information may become highly unreliable and noisy, leading to a flat proposal distribution of the motion model. In contrast, the observation likelihood is peaked so only a small number of particles have high weights and cover the meaningful areas of the target distribution. Hence, a large number of particles is required to sufficiently represent the posterior distribution. In order to achieve more focused particle sets, which require fewer particles to represent the posterior, we therefore use an improved proposal that takes into account the latest range observations that we fuse with vision observations.<sup>7</sup>

### 4.1. Observation Model for Vision Data

While our range-based localization is highly accurate when walking on the ground, translational errors can be larger while climbing demanding terrain such as staircases due to inaccurate motion execution and slippage on the steps. The placement of a laser range finder may inhibit directly observing the area in front of the feet,

which may be crucial for accurate positioning. Similarly, stereo and RGBD-sensors have a dead spot at close ranges due to their cameras' baseline or properties of the active illumination projector. We therefore propose to augment the observation model in the particle filter with vision data in the form of detected edges, which are typically present in all kinds of complex environments and can be reliably detected in monocular images.

The new observation model for vision data  $p(C_t | m, \mathbf{x}_t)$  defines the likelihood of capturing the scene in a set of images  $C_t = \{c_{t,1}, c_{t,2}, \dots\}$  given a 3D edge model of the environment  $m$  and the estimated pose  $\mathbf{x}_t$  of the robot. We assume that the individual image observations are conditionally independent:

$$p(C_t | m, \mathbf{x}_t) = \prod_{c_t \in C_t} p(c_t | m, \mathbf{x}_t), \quad (17)$$

which is a reasonable assumption when there is only little or no overlap between them.

Our approach for estimating the observation likelihood is based on chamfer matching and relies on a consistent matching of the given edge model of the staircase to lines detected in the camera images.<sup>7</sup> The edge model can be specified by the user, computed from the volumetric 3D map, or learned from 3D range observations.<sup>31</sup>

First, the algorithm applies the Canny edge detection algorithm and a probabilistic Hough transform in order to extract line segments. Afterwards, a distance transformation is applied to the detected lines so that the value of each pixel indicates the Euclidean distance between the pixel and the nearest detected line. Similarly, we compute an orientation transformation that maps each pixel to the orientation of the nearest detected line.

To determine the observation likelihood for a given estimate of the robot's pose, the algorithm projects the edges of the given staircase model from the camera pose onto the image. By iterating over all visible pixels  $\mathbf{q} = (u, v)$  of the model edges  $l$  projected onto the camera image, the algorithm computes the cost function

$$cost = \sum_{l \in m} \sum_{\mathbf{q} \in l} (\alpha \cdot \text{dist}(\mathbf{q}, d(\mathbf{q})) + \beta \cdot \angle(d(\mathbf{q}), l)). \quad (18)$$

Here,  $d(\mathbf{q})$  denotes the detected line nearest to pixel  $\mathbf{q}$ ,  $\text{dist}(\mathbf{q}, d(\mathbf{q}))$  denotes the Euclidean distance between the pixel  $\mathbf{q}$  and the nearest detected line,  $\angle(d(\mathbf{q}), l)$  is the absolute angle between the nearest detected line and the projected model line, and  $\alpha, \beta$  are constant weighting factors.

Since the costs in Eq. (18) are within the semi-infinite interval  $[0, \infty)$ , we apply the exponential distribution for the observation likelihood

$$p(c_t | m, \mathbf{x}_t) = \lambda \exp(-\lambda \cdot cost), \quad (19)$$

where the distribution parameter  $\lambda$  was determined experimentally.

#### 4.2. Improved Proposal Distribution

According to Doucet *et al.*<sup>32</sup>, the following distribution is the optimal proposal in terms of minimizing the variance of the importance weights:

$$p(\mathbf{x}_t \mid m, \mathbf{x}_{t-1}^{(i)}, \mathbf{o}_t, \mathbf{u}_{t-1}) = \frac{p(\mathbf{o}_t \mid m, \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1})}{\underbrace{p(\mathbf{o}_t \mid m, \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1})}_{=: \eta^{(i)}}} \quad (20)$$

However, computing this proposal analytically requires to evaluate the integral

$$\eta^{(i)} = \int p(\mathbf{o}_t \mid m, \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}) \, d\mathbf{x}_t, \quad (21)$$

for which there is no closed-form solution in the general case.

Following the approach proposed by Grisetti *et al.*<sup>33</sup>, we approximate the integral as a finite sum

$$\eta^{(i)} \simeq \sum_{j=1}^K p(\mathbf{o}_t \mid m, \mathbf{x}_j^{(i)}) \cdot p(\mathbf{x}_j^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}), \quad (22)$$

where  $\{\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_K^{(i)}\}$  is a set of sample points drawn around the particle's pose. This sampling technique can be used to efficiently approximate the proposal if the sampled points cover meaningful regions of the proposal and these regions are sufficiently small. The proposal distribution typically only has one mode, which we determine by scan matching the point cloud of the range sensor in the 3D environment model. As the most likely pose returned by the scan matcher already provides a good estimate, we expect that the meaningful regions of the proposal distribution will be in the vicinity of the mode. We can thus cover the meaningful area of the proposal distribution by drawing a low number of samples from a uniform distribution within a fixed radius around the computed mode and finally approximate the distribution by fitting a Gaussian to the weighted sample points.

#### 4.3. Improved Proposals for Range and Vision Observations

In our case of combining ranges  $\mathbf{r}_t$  and camera images  $C_t$  as observation  $\mathbf{o}_t$ , the proposal distribution is

$$\frac{1}{\eta^{(i)}} \cdot p(\mathbf{r}_t, C_t \mid m, \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}), \quad (23)$$

and we evaluate it in points sampled in  $x$ ,  $y$ , and  $\psi$  direction. According to Eq. (22), this results in an approximation of  $\eta^{(i)}$ :

$$\begin{aligned} \eta^{(i)} &:= p(\mathbf{r}_t, C_t \mid m, \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}) \\ &\simeq \sum_{j=1}^K \left( p(\mathbf{r}_t, C_t \mid m, \mathbf{x}_j^{(i)}) \cdot p(\mathbf{x}_j^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}) \right) \end{aligned} \quad (24)$$



Fig. 4. Experimental Environment II

At each sample point  $\mathbf{x}_j^{(i)}$ , the algorithm evaluates the observation likelihood  $p(\mathbf{r}_t, C_t | m, \mathbf{x}_j^{(i)})$  based on the current range measurement  $\mathbf{r}_t$  and the set of camera images  $C_t$ . We assume that the range measurement and the measurements from the individual images are conditionally independent and compute the range measurement observation  $p(\mathbf{r}_t | m, \mathbf{x}_j^{(i)})$  according to the raycasting model (Sec. 3.3.2) and the vision observations  $p(c_t | m, \mathbf{x}_j^{(i)})$  according to chamfer matching (Sec. 4.1).

Finally, the algorithm draws a new particle pose from a Gaussian fitted to the weighted samples and computes the corresponding importance weight according to

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot \eta^{(i)} \cdot p(\tilde{z}_t | m, \mathbf{x}_t^{(i)}) \cdot p(\tilde{\varphi}_t | \mathbf{x}_t^{(i)}) \cdot p(\tilde{\theta}_t | \mathbf{x}_t^{(i)}) \quad (25)$$

with the observation models defined in Eq. (14)–(16).

## 5. Experiments

In a set of localization experiments, we evaluated various aspects of our localization approach with two different Nao humanoids. The experimental environments are shown in Fig. 1 and Fig. 4. A volumetric 3D map of these environments was manually constructed.

We used the weighted mean of the particle distribution as estimated localization pose of the robot. To provide ground truth data, we used a *MotionAnalysis Raptor-E* motion capture system in both environments. For the tracking experiments, we initialized the particle filter pose at the ground truth pose returned by the motion capture system.

### 5.1. Humanoid Robot Platform

As a robotic platform in our experiments, we used the Aldebaran Robotics Nao humanoid, shown in Fig. 5. The robot is 58 cm tall, weighs 4.8 kg and has 25 degrees of freedom.



Fig. 5. Nao humanoids used for the evaluation of our work, using a head-mounted 2D laser range finder (left) and an RGBD-camera (right).

We first performed experiments with a Nao equipped with a laser head, i.e., we used the range data from a Hokuyo URG-04LX laser range finder. The 2D range finder provides a field of view of  $240^\circ$  with a maximum range of 5.6 m. The second Nao humanoid is equipped with a consumer-grade Asus Xtion Pro Live RGB-D camera. The camera has a field of view of  $58^\circ$  horizontally and  $45^\circ$  vertically. It is mounted on the robot's head in such a way that its optical axis faces the floor in a  $30^\circ$  angle while walking. Since the error of this sensor grows quadratically<sup>34</sup>, we limited the maximum range to 5.6 m. We also adjusted  $\sigma_r$  in the range measurement models of Eq. (10) and Eq. (12) to quadratically depend on the measurement range  $r$ . Finally, to sample rays from the depth camera data, our system separates all end points of the corresponding point cloud into ground and non-ground parts by means of a RANSAC-based plane detection. Our approach uses this information for sampling rays from non-ground points uniformly over the Cartesian space with a higher density than ground points. Thus, we compensate for the fact that beams hitting the floor can provide no information for estimating the translation in the horizontal plane, which is typically more important than height or pitch and roll.

In order to obtain measurements of its joint positions, Nao is equipped with Hall effect sensors which measure the angle of each joint. Using the joints of the support leg, an estimate of the robot's torso position and orientation can be obtained through forward kinematics at any time. Additionally, an inertial measurement unit (IMU) yields an estimate about the robot's orientation. Measurements from a two-axis gyroscope and a three-axis accelerometer are integrated in order to obtain an estimate of the robot's torso orientation around the world  $x$  and  $y$ -axis (roll and pitch, respectively). The measurements of this small and lightweight IMU are quite noisy compared to the IMUs often used in robotics. However, especially while walking, these values are more accurate than the roll and pitch obtained through kinematics of the measured support leg joint angles, because the robot's feet may

not always precisely rest on the ground.

In all our experiments, Nao was teleoperated with an omnidirectional velocity input. From that, the default walking engine (API version 1.12) generated the gait for walking straight, on arcs, sideways, and for turning.

In practice, odometry and other sensor data do not arrive at discrete timesteps but asynchronously and with different update rates. To solve this problem and achieve time synchronization, we update the MCL filter based on range sensor data, interpolating odometry and IMU sensor data between two valid measurements. A second problem stems from the fact that a range image is not generated instantaneously but over a certain amount of time in which the robot may be moving. To this end, we apply temporal uncertainty sampling as introduced by Thompson *et al.*<sup>1</sup> For each particle, odometry data is interpolated to a timestamp which is sampled uniformly around the current measurement timestamp in an interval corresponding to the time needed for a complete scan (0.1 s).

Throughout our experiments, we used a Cartesian sampling distance for beam endpoints of 30 cm and integrated new observations only after the robot has traveled a distance of 15 cm or changed its orientation for more than  $23^\circ$  since the last sensor integration.

## 5.2. Run Time Performance

We first evaluated the run times of the range sensor model. It constitutes the most computationally expensive part of MCL, but can be efficiently parallelized. We compared the endpoint model to raycasting in a map of Environment I (Fig. 1) with 1 cm resolution on a standard desktop CPU (Intel Core i7-2600, 3.4 GHz). The average number of considered end points here was 60, with an average beam length of 1.68 m. Through parallelization with four threads, we achieved average run times as fast as 0.013 ms per particle for the endpoint model, and 0.448 ms per particle for raycasting (single-threaded: 0.024 ms / 1.172 ms). This results in a maximum realtime update rate of 11.2 Hz for 200 particles with raycasting and parallelization. Note that in practice, sensor measurements are only integrated after the robot has moved for a certain distance to ensure conditional independence between updates. As the raycasting time directly depends on the map resolution and number of end points, this can be further optimized depending on the scenario at hand.

## 5.3. Laser-Based Pose Tracking

With the laser range finder, we evaluated the performance of our localization approach in Environment I for different sensor and motion models against ground truth from the motion capture system. Since any Monte Carlo method is susceptible to the effects of pseudo-random number generators, we evaluated the errors as mean and 95% confidence interval over ten differently seeded localization runs of the same recorded sensor data.

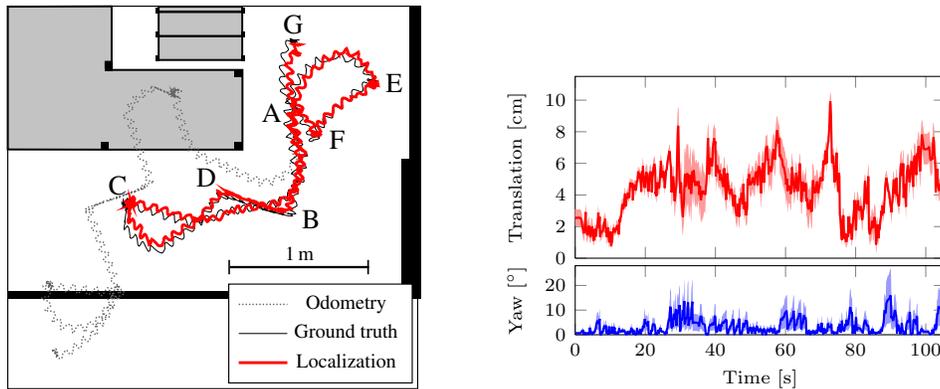


Fig. 6. *Left*: Trajectory estimated with our localization approach compared to ground truth and odometry on the lower level of Environment I. 200 particles were used with a laser range finder using raycasting and a calibrated odometry motion model. The robot started walking forward at (A), then walked on a right turn at (B) towards (C) where it turned left on the spot and walked on an arc to (D). From there, it walked sideways to (B) and continued forwards over (A) to (E). At (E) it turned on the spot, walked forward to (F), turned again, and walked backwards to the goal at (G). *Right*: Mean translational and yaw error with 95% confidence interval for  $N = 10$  runs in the same setting.

To calibrate the odometry motion model using the method described in Sec. 3.2.2, the humanoid executed a set of sample movements such as walking forward, sideways, or turning on the spot before performing the actual experiments. We then used the data of the incremental odometry motions and the ground truth from motion capture to obtain the systematic drift and noise parameters of the motion model. In the uncalibrated model, the drift is accounted for with larger noise parameters.

The complete trajectory for the evaluation of the localization performance has a length of approximately 5 m and includes parts of walking forward, on arcs, sideways, backwards, and turning on the spot. Fig. 6 shows a single trajectory and the error over time for 200 particles. The mean translational error for this run was 3.9 cm and the mean yaw error  $1.7^\circ$ . Roll and pitch errors are not shown but generally below  $3^\circ$ . Pose estimation based on odometry quickly diverges while our localization approach is able to track the ground truth closely through all walking motions. Fig. 7 shows the aggregated results for different numbers of particles. Note that the angular errors behave similarly but are not shown for clarity. As can be seen, the endpoint model, while computationally more efficient, is not able to capture the full 3D structure of the environment and results in a significantly higher localization error (t-test, 99% confidence). A calibrated odometry motion model generally improves the localization performance when using only few particles. This improvement is statistically significant up to 200 particles (t-test, 95%). Hence, we used raycasting and the calibrated motion model in the next experiments.

Estimating only  $x, y$ , and  $\psi$  in a 2D map with raycasting and odometry cali-

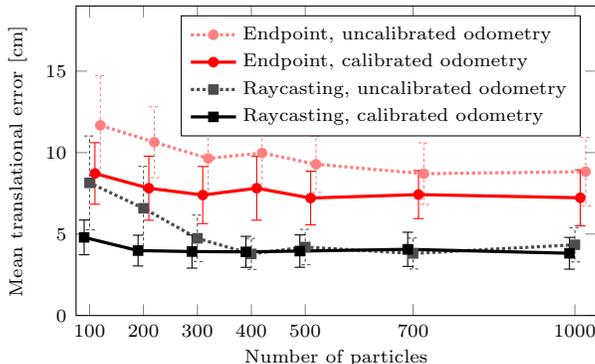


Fig. 7. Mean translational error with 95% confidence interval for laser-based localization on the lower level of Environment I ( $N = 10$  runs each). Raycasting leads to a significantly lower error. A calibrated odometry motion model requires less particles to achieve the same performance as an uncalibrated one.

bration leads to a higher average error of 4.6 cm, whereas our method results in 3.9 cm when using 200 particles. While this difference is not statistically significant, the individual deviations can be much higher, potentially leading to collisions with obstacles when navigating with an inaccurate pose estimate. In all trajectories, the maximum error was 14.5 cm (translation) /  $22^\circ$  (yaw) for 2D localization and 9.5 cm /  $9^\circ$  for our 3D localization, both using 200 particles with raycasting and calibrated odometry. This demonstrates that the humanoid’s swaying motion of up to  $5^\circ$  in each direction needs to be considered and indeed a full 6D pose estimation results in the highest accuracy.

A video demonstrating our laser-based localization approach is available online at <http://www.youtube.com/watch?v=uiIi2rSKWAU>.

#### 5.4. Laser-Based Global Localization

Fig. 8 shows the evolution of the particle distribution for global localization with 50 000 particles. Initially, the complete area of Environment I was covered with pose hypotheses. After three sensor updates, the particles converged to the ground truth. Throughout 10 runs in the same setting, our global localization approach always converged to a pose within 10 cm of the ground truth requiring at most four sensor updates, which demonstrates the reliability of our approach for global localization. As discussed in Sec. 3.4, once the particle poses have converged, the localization changes into a tracking mode and resamples the distribution using fewer particles.

#### 5.5. Comparing Laser-Based With Depth Camera-Based Localization

In this experiment, we evaluated the localization performance when using an affordable consumer-level depth camera instead of the Hokuyo laser range finder. Both

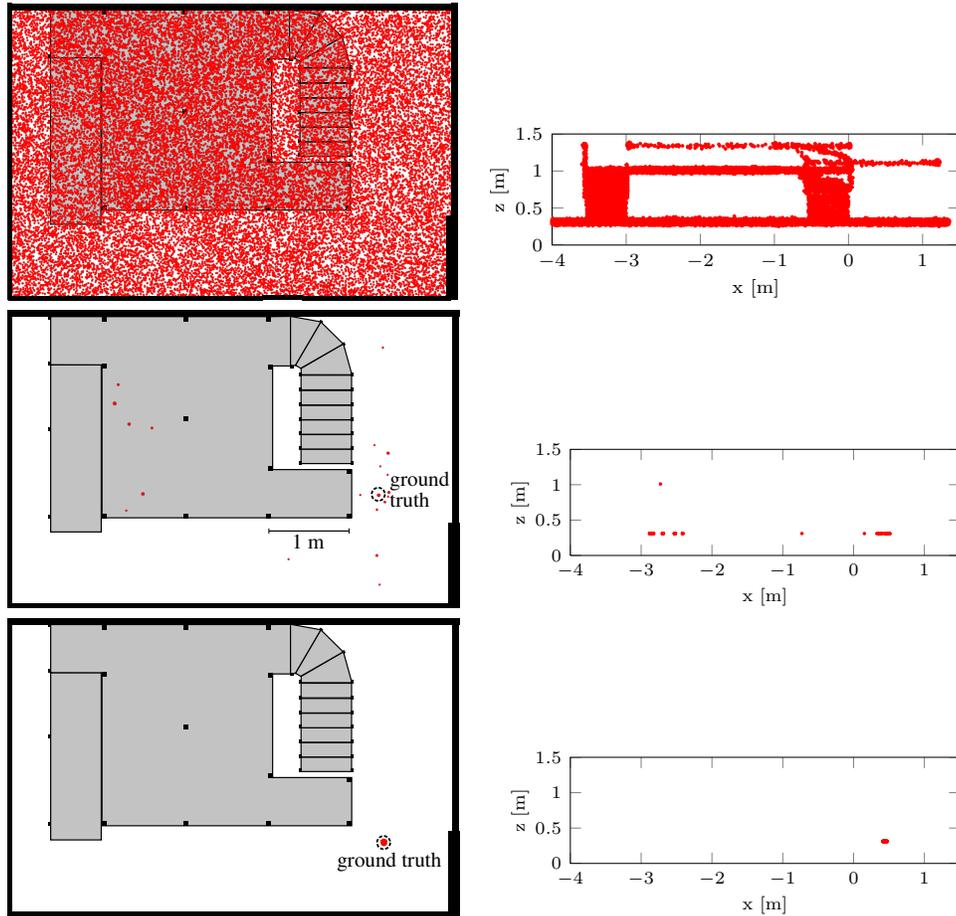


Fig. 8. Particle distribution over time for global localization with 50 000 particles in Environment I. The left column shows a projection into the  $xy$ -plane, while the right column shows a side view. Initially (top row), particles were distributed uniformly over all surfaces including the ramp and staircase. The first sensor update (middle row) resulted in a few clusters, one on the upper level at  $z=1$  m. After three sensor updates, the particles converged to the ground truth (bottom row).

Nao robots are nearly identical except for the installed range sensors (see Fig. 5). We calibrated the motion model parameters for both robots as explained in Sec. 5.3. Both robots were teleoperated to follow the same path of approximately 8 m length through Environment II (Fig. 4), while they were tracked with the motion capture system. We compared ten localization runs using 60 beams on average of each sensor for a localization update. As can be seen in Fig. 9, the mean error increases when using the depth camera due to its narrow horizontal field of view and increasing noise for longer measurements. The difference is statistically significant for 100 particles (t-test, 99%). Maximum errors over all 10 localization runs were 13.4 cm

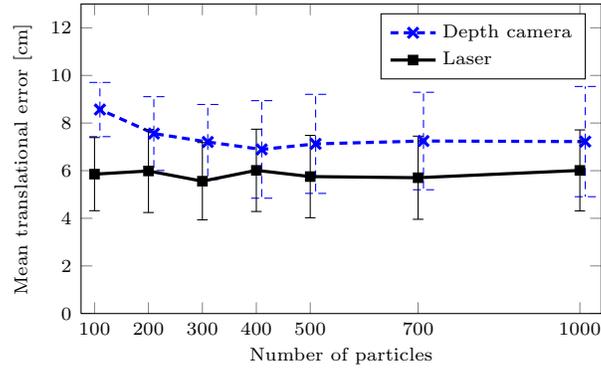


Fig. 9. Mean translational error with 95% confidence interval for laser-based and depth camera-based localization in Environment II ( $N = 10$  runs each).

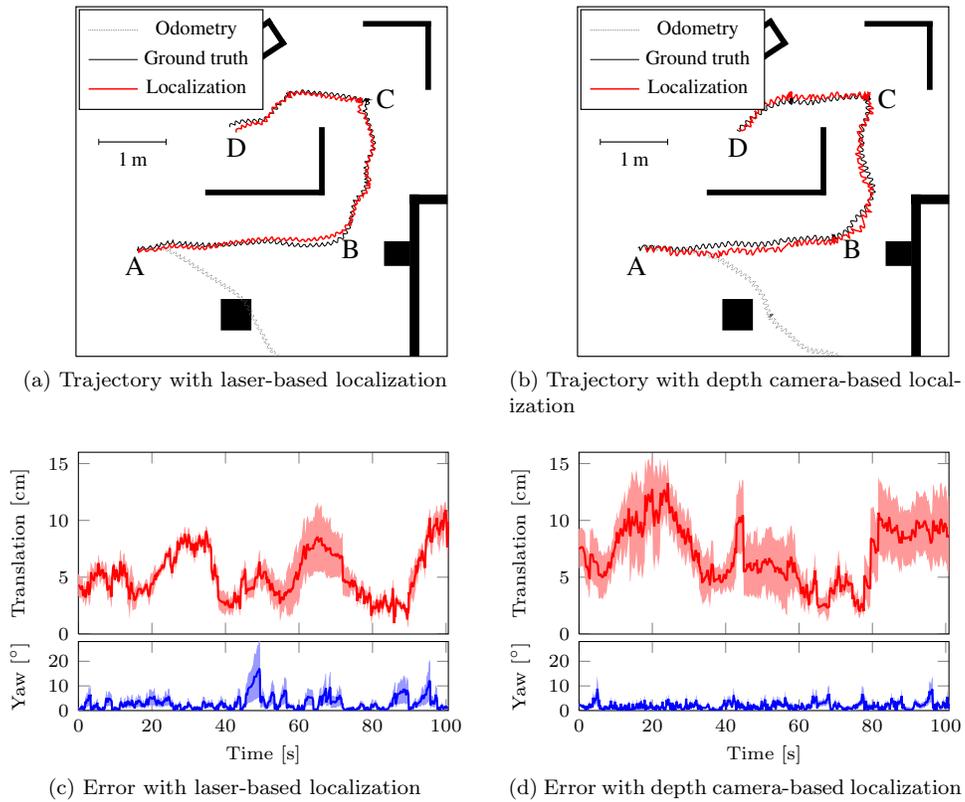


Fig. 10. Comparison of the trajectories and error over time for laser-based and depth camera-based localization with 300 particles in Environment II ( $N = 10$  runs each). The robots started at (A), walked forward to (B), then on a curve to (C). There, they turned on the spot and continued forward to the goal (D).



Fig. 11. Nao humanoid climbing a spiral staircase using our approach of combining vision and laser range data for localization to correct for motion noise.

(translation) /  $7^\circ$  (yaw) when using the laser, and 21.4 cm /  $8^\circ$  when using the depth camera. As visible in the error over time in Fig. 10, the error for depth camera based localization is highest when the robot has only few obstacles to observe nearby, e.g. between locations (A) and (B) (10 – 30s) or when approaching (D) (82 – 100s). Nevertheless, both sensor configurations are able to accurately track the humanoid's motions while odometry quickly diverges.

### 5.6. Improved Proposals Localization for Climbing Stairs

We then evaluated the performance of our localization approach extended with vision information for the scenario of climbing stairs. Here, the robot had to climb the ten steps of the spiral staircase connecting the two levels of Environment I using the laser sensor and a monocular camera for localization (Fig. 11). In addition to the volumetric 3D map, the robot here made use of the known edge model of the staircase.

Based on the localization estimate, the robot aligns with the step in front of it and then proceeds to climb the step with a learned whole-body motion. Once the humanoid reaches the top of the step, it integrates a sensor measurement for pose estimation and then corrects for motion drift by aligning with the next step. For integrating vision data into our improved proposals approach, the robot assumes an upright posture and acquires three images with its bottom-facing camera, looking to the left, center, and right. This combination allows the humanoid to capture most of the area in front of its feet, despite the narrow field of view of the internal camera. The robot stands still while capturing an image to avoid motion blur.

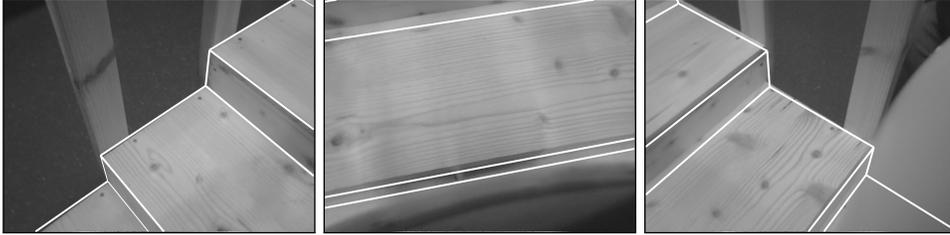


Fig. 12. Camera view of the robot when looking to the left, center, and right. Overlaid in white is the projected stair model from the best particle pose after integrating all observations. The matching model demonstrates the high accuracy of the localization.

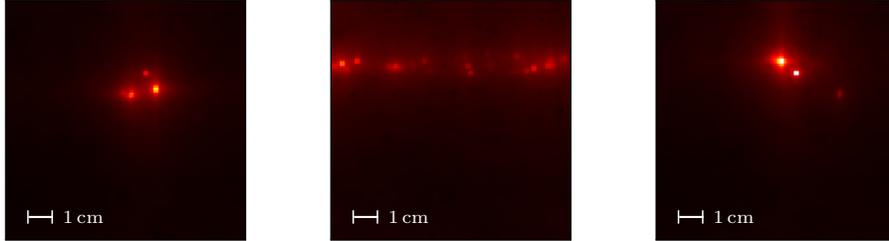


Fig. 13. Observation likelihood  $p(c_t | m, \mathbf{x}_j)$  for left, center, and right camera image in the horizontal plane as returned by chamfer matching. Brighter areas correspond to a higher likelihood.

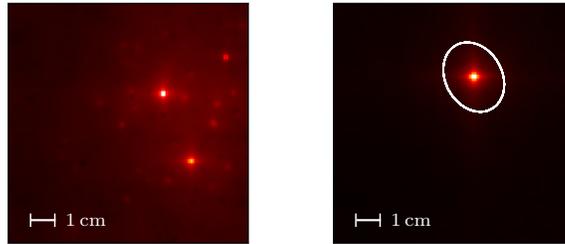


Fig. 14. *Left*: Observation likelihood  $p(\mathbf{r}_t | m, \mathbf{x}_j)$  of the laser data. *Right*: Final improved proposal distribution by combining laser and vision observations (Fig. 13) with the 95% confidence ellipse of the Gaussian approximation (white). Brighter areas correspond to a higher likelihood.

### 5.6.1. Measurement Integration for a Single Step

As an illustrating example of integrating vision data according to Sec. 4, Fig. 12 shows the robot's view of the staircase when standing on the second step and looking to the left, center, and right with its bottom-facing camera. Fig. 13 shows the resulting observation likelihoods in the horizontal plane for these camera observations and Fig. 14 (left) shows the likelihood of the corresponding laser observation with raycasting. The coordinate system is identical for all distributions and is cen-

tered at the odometry pose. The distributions of the left and right camera image are both focused and have similar modes while the distribution of the center camera image is spread in horizontal direction due to the higher uncertainty in lateral direction. The final improved proposal and fitted Gaussian distribution resulting from combining the distributions for vision and laser is shown in Fig. 14 (right). As can be seen, the pose estimate is highly focused. Fig. 12 shows the edge model projected from the best particle's pose after localizing on this step. The projected lines on the left and right camera image closely fit the corresponding edges of the staircase; the errors in lateral direction and in the orientation are only small. The error in forward direction is slightly higher, which is caused by the strong lines in the wood texture parallel to the stair edge.

### 5.6.2. Localization Accuracy

As a performance measure for the quality of the localization, we evaluated the pose returned by the localization approaches with and without integrating visual observations and compared it to the ground truth from the motion capture system on each single step. To this end, we aggregated the results over the number of successful steps the robot climbed in six runs ( $N = 23$  for laser-only localization,  $N = 60$  for the laser and vision). Note that our combined approach lead to a success rate of 100% for climbing all 10 steps of the spiral staircase, whereas laser-based localization resulted in a fall in five out of six runs.

For the standard laser-based localization, the mean translational error in the horizontal plane on each successful step was  $2.56 \text{ cm} \pm 0.63 \text{ cm}$  (95% confidence interval) and a mean orientation error was  $1.3^\circ \pm 0.4^\circ$ . In comparison to that, our combined approach using improved proposals with chamfer matching resulted in a mean error in the orientation of only  $0.6^\circ \pm 0.2^\circ$ . The difference of the angular error is highly significant (t-test, 99.9%). The improved proposals approach also decreased the mean translational error to  $1.09 \text{ cm} \pm 0.14 \text{ cm}$ , which is also statistically significant (t-test, 95%). Note that the accuracy is in many cases higher than the map resolution (1 cm) used for laser-based MCL, which additionally demonstrates the advantages of integrating visual information.

Our approach is robust against unmodeled objects and texture on the surfaces, as the vision observation model only tests if the predicted edges are present in the camera images. The robot can detect self-occlusions and exclude the affected regions from the observation model. However, the localization performance may degrade if other unmodeled objects occlude significant parts of the predicted model edges, or if the texture contains strong lines running parallel to the modeled edges.

Fig. 11 shows the robot climbing the spiral staircase using our approach. A video of the complete sequence is available online at <http://www.youtube.com/watch?v=U9118y3Svkw>.

## 6. Conclusions

In this article, we presented an approach to probabilistic localization for humanoid robots in a 3D representation of the environment. Our system is able to deal with all challenges occurring during humanoid robot navigation. This includes highly inaccurate odometry information, inherently noisy sensor data, and the violation of the flat world assumption. We apply Monte Carlo localization to globally determine and reliably track a humanoid's 6D pose, consisting of the 3D position and the three rotation angles. Hereby we integrate range data from a 2D laser or an RGBD-camera, as well as attitude estimates from an IMU and measurements from the joint encoders. We extended this approach to additionally integrate vision data. In this way, the localization performance can be further improved.

We thoroughly evaluated and discussed the presented methods. As we show in our experiments with a Nao humanoid robot, our method is able to accurately estimate the 6D pose of the humanoid's torso while walking and climbing stairs. Our approach is generally applicable to humanoid robots equipped with range sensors given a 3D model of the environment.

Our localization system currently relies on the availability of a 3D map to localize in, which must be acquired by the robot itself or supplied by the user beforehand. We intend to use a Simultaneous Localization and Mapping (SLAM) approach in future research in combination with our sensor and motion models.

## Acknowledgments

This work has been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 and within the Research Training Group 1103. The authors would like to thank Rainer Kümmerle and Slawomir Grzonka for fruitful discussions, and Jörg Müller for assistance with the motion capture system.

## References

1. S. Thompson, S. Kagami, and K. Nishiwaki, "Localisation for autonomous humanoid navigation," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2006.
2. J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami, "Biped navigation in rough environments using on-board sensing," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.
3. P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.
4. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189–206, 2013, software available at <http://octomap.github.com>.
5. A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

6. D. Maier, A. Hornung, and M. Bennewitz, "Real-time navigation in 3D environments based on depth camera data," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2012.
7. S. Oßwald, A. Hornung, and M. Bennewitz, "Improved proposals for highly accurate localization using range and vision data," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, October 2012.
8. J. Ido, Y. Shimizu, Y. Matsumoto, and T. Ogasawara, "Indoor navigation for a humanoid robot using a view sequence," *Int. Journal of Robotics Research (IJRR)*, vol. 28, no. 2, pp. 315–325, 2009.
9. S. Oßwald, A. Hornung, and M. Bennewitz, "Learning reliable and efficient navigation with a humanoid," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
10. M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke, "Metric localization with scale-invariant visual features using a single perspective camera," in *European Robotics Symposium 2006*, ser. STAR Springer tracts in advanced robotics, vol. 22, 2006.
11. A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.
12. R. Cupec, G. Schmidt, and O. Lorch, "Experiments in vision-guided robot walking in a structured scenario," in *Proc. of the IEEE Int. Symp. on Industrial Electronics*, 2005.
13. J. Seara and G. Schmidt, "Intelligent gaze control for vision-guided humanoid walking: methodological aspects," *Robotics & Autonomous Systems*, vol. 48, no. 4, pp. 231–248, 2004.
14. C. Stachniss, M. Bennewitz, G. Grisetti, S. Behnke, and W. Burgard, "How to learn accurate grid maps with a humanoid," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.
15. F. Faber, M. Bennewitz, C. Eppner, A. Goeroeg, A. Gonsior, D. Joho, M. Schreiber, and S. Behnke, "The humanoid museum tour guide Robotinho," in *Proc. of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.
16. R. Tellez, F. Ferro, D. Mora, D. Pinyol, and D. Faconti, "Autonomous humanoid navigation using laser and odometry data," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2008.
17. K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *Int. Journal of Robotics Research (IJRR)*, vol. 31, no. 11, pp. 1251–1262, 2012.
18. J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *The International Journal of Robotics Research (IJRR)*, vol. 27, no. 10, pp. 1117–1134, 2008.
19. S. Ahn, S. Yoon, S. Hyung, N. Kwak, and K. Roh, "On-board odometry estimation for 3D vision-based SLAM of humanoid robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.
20. O. Stasse, B. Verrelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves, and K. Yokoi, "Integrating walking and vision to increase humanoid autonomy," *Int. Journal of Humanoid Robotics (IJHR)*, special issue on Cognitive Humanoid Robots, vol. 5, no. 2, pp. 287–310, 2008.
21. P. F. Alcantarilla, O. Stasse, S. Druon, L. M. Bergasa, and F. Dellaert, "How to localize humanoids with a single camera?" *Autonomous Robots*, vol. 34, no. 1-2, pp.

- 47–71, 2013.
22. N. Kwak, O. Stasse, T. Foissotte, and K. Yokoi, “3D grid and particle based slam for a humanoid robot,” in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2009.
  23. P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, “Online environment reconstruction for biped navigation,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006.
  24. M. Stilman, K. Nishiwaki, S. Kagami, and J. Kuffner, “Planning and executing navigation among movable obstacles,” *Springer Journal of Advanced Robotics*, vol. 21, no. 14, pp. 1617–1634, 2007.
  25. F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo localization for mobile robots,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.
  26. A. I. Eliazar and R. Parr, “Learning probabilistic motion models for mobile robots,” in *Proc. of the International Conference on Machine Learning (ICML)*, 2004.
  27. S. Thrun, “A probabilistic online mapping algorithm for teams of mobile robots,” *Int. Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
  28. B. Lau, C. Sprunk, and W. Burgard, “Efficient grid-based spatial representations for robot navigation in dynamic environments,” *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116 – 1130, 2013.
  29. R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard, “Monte Carlo localization in outdoor terrains using multilevel surface maps,” *Journal of Field Robotics (JFR)*, vol. 25, pp. 346–359, 2008.
  30. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT-Press, 2005.
  31. S. Oßwald, J.-S. Gutmann, A. Hornung, and M. Bennewitz, “From 3D point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids,” in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.
  32. A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
  33. G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
  34. K. Khoshelham and S. Oude Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications,” *Sensors: Journal on the Science and Technology of Sensors and Biosensors*, vol. 12, pp. 1437–1454, 2012.



**Armin Hornung** is a research scientist and PhD student in the Humanoid Robots Laboratory at the University of Freiburg. In 2009, he received his Diplom (Master's) degree in computer science with specialization in artificial intelligence and robotics from University of Freiburg, Germany. His main research focus lies in humanoid robot navigation in complex indoor environments.



**Stefan Oßwald** studied computer science at the University of Freiburg and received his Master of Science degree in 2012. Currently, he is a PhD student in the Autonomous Intelligent Systems Laboratory at the University of Freiburg. His research activities focus on cooperation and coordination between humans and robots in exploration tasks.



**Daniel Maier** studied at the University of Freiburg and the University of Washington. He received his Diplom (M.S.) in computer science from the University of Freiburg in 2010. Currently, he is a PhD student in the humanoid robots laboratory at the University of Freiburg and member of the research training group “embedded microsystems”. His research is focused on autonomous navigation and perception for resource-constrained systems.



**Maren Bennewitz** is an assistant professor for Computer Science at the University of Freiburg in Germany. She got her Ph.D. in Computer Science from the University of Freiburg in 2004. From 2004 to 2007, she was a Postdoc in the humanoid robots laboratory at the University of Freiburg which she heads since 2008. The focus of her research lies on robots acting in human environments. In the last few years, she has been developing novel solutions for intuitive human-robot interaction and navigation of robots in complex indoor environments.