Accurate Indoor Localization for RGB-D Smartphones and Tablets given 2D Floor Plans

Wera Winterhalter Freya Fleckenstein Bastian Steder Luciano Spinello Wolfram Burgard

***Abstract*— Accurate localization in indoor environments is widely regarded as a key opener for various location-based services. Despite tremendous advancements in the development of innovative sensor concepts, the most effective and accurate solutions to this problem make use of a map computed from sensory data. In this paper, we present an efficient approach to localize an RGB-D smartphone or tablet that only makes use of a two-dimensional outline of the environment as a map as it is typically available from architectural drawings. Our technique employs a particle filter to estimate the 6DoF pose. We propose a sensor model that robustly estimates the likelihood of measurements and accommodates the disagreements between floor plans and real world data. In extensive experiments, we demonstrate that our approach is able to globally localize a user in a given 2D floor plan using a Google Tango device and to accurately track the user in such an environment.**

I. Introduction

Accurate localization of people in indoor environments is widely regarded as a key opener for various services that require location awareness. This includes user localization, robot navigation and search and rescue. During the previous decades, the robotics and computer vision community proposed a set of accurate, robust and efficient localization methods. Despite all these efforts, accurate and metric localization has been mainly achieved by phrasing the problem as a pose estimation task in a map that is computed beforehand from sensor data. This implies that the robot or the person have to visit the environment *before* the localization run.

In this paper, we present an approach to estimate the 6DoF position of a smartphone or tablet that is equipped with an RGB-D camera, such as a Google Tango device. As map we only require a two-dimensional outline of the environment, like a floor plan. A floor plan is the depiction of the "naked" environment, without furniture, fixtures, windows, or clutter. It contains no data actually measured in the environment, which makes visual or existing range-based techniques difficult to apply directly. Such floor plans are readily available for most buildings in the form of their original blueprint from the time they were designed. Fig. 1 depicts an example application scenario for our approach.

Our system uses a Monte-Carlo particle filter localization approach that runs online while the user is walking through the environment. The calculation is done off-board by transmitting the sensor data to a server. Note that the idea to use the cloud for time-consuming processes is a quite common approach for various smartphone services including such requiring speech understanding, route planning or image processing. The odometry information needed for the filter is computed onboard the device, using the RGB-D and IMU data. For the implementation of the particle filter we designed a sensor model that handles the special case of localizing with six degrees of freedom in a two-dimensional map. Our likelihood model assigns a probability to the measurement, using a function of the distance between the measurement and the floor plan, also taking into account measurements generated from floor and ceiling or obstacles not present in the map. It also considers physical constraints, i.e., the user cannot move through walls.

All authors are with the Department of Computer Science of the Faculty of Engineering at the University of Freiburg, Germany

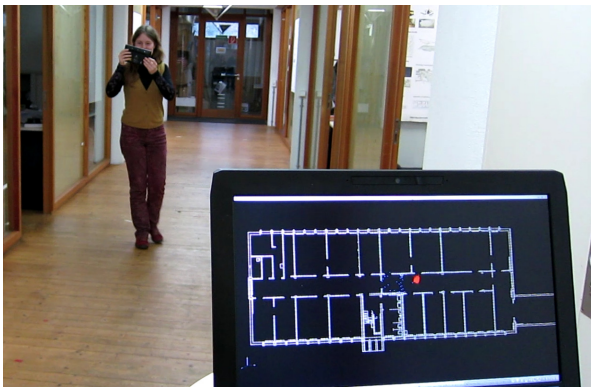

Fig. 1. Localization in 6DoF using only a 2D outline of the environment and an RGB-D Google Tango device. The computer screen shows the floor plan in white and the particle cloud representing the current pose estimate in red.

We evaluate our approach with respect to three different use cases: global localization, position tracking and coarsely initialized position tracking. The latter is an initialization that is typical for coarse WiFi position estimates from access point signal magnitude or a user manually marking the starting position in the floor plan. We demonstrate the applicability of our approach in two real-world settings using two busy office buildings during working hours. Additionally, we provide a precise quantification of the accuracy of our approach using a custom-built indoor environment set up in a motion capture studio. All results demonstrate that our approach robustly converges to the correct solution and provides accurate position estimates. Our system is applicable for online use, even when performing global localization. Please refer to the attached video for a demonstration of the latter.

We imagine our system to be of relevance, e.g., in search and rescue scenarios. A person entering an unknown building

can wear the necessary device on a helmet or on the back of a jacket. A team on the outside can then track the progress of the person on the floor plan and give directions. While floor plans are available for most buildings, in such a scenario, other indoor localization methods like WiFi-based systems might not be available due to power outages or a lack of knowledge about the access points in the building. In this case, the filter can be approximately initialized at the entrance door without the need for global localization.

II. Related Work

Mobile localization is a well studied field in robotics. Especially probabilistic approaches for estimating the pose of a mobile robot given a map have proven to be robust in practice. Successful techniques rely on the extended Kalman filter (EKF) [18], histogram filters [14], or particle filters, often referred to as Monte-Carlo localization (MCL) [7]. There are also approaches combining ideas of MCL and histogram-based methods [17]. Commonly used sensors for vehicle localization are laser range scanners [7, 23], cameras [1, 2, 9, 20], RFID or wireless receivers using radio signal strength [8, 11], or GPS receivers. Vision-based MCL for mobile robots was first introduced by Dellaert et al. [6].

Fallon et al. [10] propose a very efficient RGB-D 6DoF Monte-Carlo localization approach for indoor environments. Their technique requires a preliminary mapping phase, where only major plane segments are recorded. For evaluating the likelihood, the algorithm synthesizes RGB-D camera views at the location of particle poses. Similar to our work, they make use of visual odometry to estimate the motion between two poses. Paton and Kosecka [19] propose a global localization approach that makes use of SIFT correspondences and scan matching in depth data to obtain a correct pose. Shotton et al. [21] tackle the problem of localizing an RGB-D camera relative to a previously recorded 3D scene, given only a single acquired image. For this, they develop regression forests that are able to infer the correspondence of the camera frame to the scene world coordinate frame.

A very common approach to indoor localization is to make use of WiFi or bluetooth sources in the environment [3, 4, 12]. These techniques tend to be sensitive to the WiFi source placement, the accuracy of the signal strength map and the amount of clutter and dynamics in the environment. Additionally, the number of WiFi sources in the environment plays an important role.

Grzonka et al. [15] present an approach to keep track of the position of a person by employing an IMU-based motion capture suit and detecting specific motion patterns as landmarks to avoid the accumulation of drift.

Localization in floor plans has received very little attention from the scientific community. Ito et al. [16] address this problem by using RGB-D data and an additional map of WiFi signal strengths. The latter is computed by using data from a previous mapping run. Spasso [22] obtains the trajectory of a pedestrian using an inertial navigation system, which provides information about the user's position, heading and velocity. The system then uses a map-matching technique to associate a user's trajectory with a 2D or 3D floor plan. Comblet et al. [5] additionally use knowledge of accessible positions in the vicinity of the previous position to improve the estimate of the user's trajectory.

The main advantage of our method is that we do not require an initial mapping phase with the device used for localization and that we acquire full 6DoF poses. The methods described above require the user to map the environment either in advance or online during the run, to acquire range data, visual information, or signal strength distributions. In contrast, our method relies only on prior information given by a floor plan, which is typically available from the time the building was constructed, for example in the form of an architectural drawing.

III. 6DoF Localization in 2D Floor Plans

In this section, we describe how we apply the Monte-Carlo localization (MCL) algorithm to achieve accurate 6DoF localization on a 2D floor plan using a mobile RGB-D camera. To achieve this, we propose a dense-depth sensor model that is able to handle floor plans as maps. Furthermore, we present a motion model that leverages the typically high accuracy of the onboard visual-inertial odometry.

A. Monte-Carlo Localization with KLD Sampling

We estimate the 6DoF pose x_t of the device at time t using the Monte-Carlo localization (MCL) as proposed by Dellaert et al. [7]. This approach estimates a posterior about the pose in the following recursive fashion:

$$p(x_t \mid z_{1:t}, u_{1:t}, m) \propto p(z_t \mid x_t, m) \cdot \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}, m) \, \mathrm{d}x_{t-1}. \tag{1}$$

Here, $u_{1:t}$ is the sequence of the executed motions, $z_{1:t}$ is the sequence of observations, and m is the provided map of the environment. The motion model $p(x_t \mid x_{t-1}, u_t)$ denotes the probability to end up in state x_t given the motion command u_t in state x_{t-1}. The sensor model $p(z_t \mid x_t, m)$ denotes the likelihood of making the observation z_t given the pose x_t and the map m.

MCL uses a set of random samples, also called particles, to represent the belief about the system state and updates the belief by sampling from the motion model when receiving odometry information. MCL computes an importance weight for each particle, which is proportional to the observation likelihood of the measurement given the corresponding state of the particle. In the so-called resampling operation particles survive with a probability proportional to the importance weight.

To optimize the performance of the MCL, we apply the KLD sampling approach by Fox [13], which adjusts the number of particles by limiting the error introduced by the sample-based representation. It computes this error using the Kullback-Leibler divergence between the sampled distribution and a discrete distribution computed over the whole map. Thus, particles are generated on demand. For example, during global localization or in areas where localization is

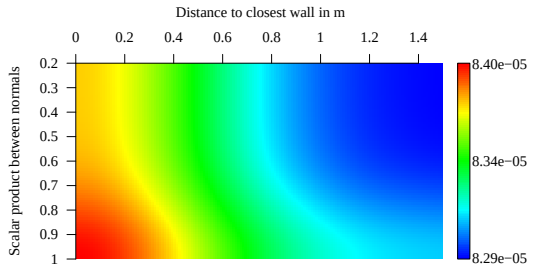

Fig. 2. The likelihood function of the sensor model, depending on the distance from a measured endpoint to the closest wall in meters (x-axis) and the scalar product of the measured and the expected normal at that endpoint (y-axis).

difficult, it creates new particles, whereas it keeps the particle set small when tracking the pose.

B. Sparse Depth-Beams Sensor Model

The task of the sensor model is to determine the likelihood $p(z \mid x, m)$ of a measurement z, given the pose x in the map m. In our approach, the measurements are range data extracted from a dense-depth image. The map is internally represented as a 3D environment, generated from a 2D floor plan and a given ceiling height: we model all walls from the floor plan as vertical planes and represent the floor and the ceiling as two horizontal planes, whose distance is assumed to be given.

Each time a new image arrives, K range measurements are randomly sampled from the dense-depth image and converted into a 3D point cloud $\mathcal{Z}$. Let z_j be the j-th measurement of $\mathcal{Z}$. In our system, we apply the endpoint model [24]. The endpoint model computes the likelihood of z_j based on the scan point z'_j corresponding to z_j transformed into the map coordinate frame according to the pose x and on the closest corresponding point in the map $m_j \in m$:

$$p(z \mid x, m) \quad = \quad f((z'_1, m_1), \ldots, (z'_K, m_K)). \tag{2}$$

Under the assumption that the beams are independent, we can rewrite this as

$$f((z'_1, m_1), \ldots, (z'_K, m_K)) \quad \propto \quad \prod_{j=1}^{K} f(z'_j, m_j). \tag{3}$$

Please note that we use a log-likelihood representation in our implementation to circumvent problems with floating point precision. We model the likelihood for each observed point as a mixture of two Gaussians, each combined with a uniform distribution:

$$\begin{aligned} f(z'_j, m_j) \quad = \quad & (\mathcal{N}(z'_j, m_j, \sigma_d) \; + \; c_d) \cdot \\ & (\mathcal{N}(|n_z \cdot n_m|, 1, \sigma_s) \; + \; c_s), \end{aligned} \tag{4}$$

where $\sigma_d, \sigma_s \in \mathbb{R}^+$ are the standard deviations and $c_d, c_s \in \mathbb{R}^+$ are constants representing the uniform distributions. The term n_z is the normal vector corresponding to the measurement z' and n_m is the one computed from the map at position m_j. The dot product of the two is the cosine of the included angle and therefore $|n_z \cdot n_m|$ is 1 if the normal vectors are identical and 0 if they are orthogonal. This favors measurements that are oriented with the floor plan. The constants c_d and c_s keep the filter from becoming overconfident in situations in which there is little or no overlap between the map and the observation. This is typically caused by objects not present in the map blocking the view of the sensor. This sensor model allows for comparatively low correspondence between measured endpoints and wall-points, given high similarity for the normals, and vice versa (see Fig. 2). This makes the particle filter more robust to errors regarding the estimated orientation, thereby facilitating global localization. We determined the parameters σ_d, σ_s, c_d and c_s experimentally by evaluating the measurement error on a dataset with ground truth sensor poses in an office environment. In such an environment, where the view of the sensor is often blocked by furniture, we determined $\sigma_d \simeq 0.52\,\text{m}$, $\sigma_s \simeq 0.18$, $c_d \simeq 80$, and $c_s \simeq 800$.

Another measure we take for an improved estimate of the particle weights is to take into account physical constraints and the potential inaccuracy of the map. More precisely, we penalize every particle that moves through a wall. For this we check the movement of every individual particle in every timestep and compare it to the map. When the movement traverses an occupied cell in the map, we multiply the particle weight with an $\varepsilon \in [0, 1]$, thereby enforcing physical world constraints and feasible position estimates (see also Thrun et al. [24]).

C. Visual Odometry Motion Model

In the prediction step of MCL it is of key importance to compute an accurate estimate of the motion between x_{t-1} and x_t, especially as we cannot rely on accurate odometry information as it would be available on a mobile robot. On an RGB-D device, one can use 3D scan matching, RGB-based visual odometry, or make use of the RGB and depth sensors together, e.g., by adding the range measurements into a visual odometry framework, thereby not purely relying on structure from motion. In addition, most smartphones or tablets include IMUs that can be used to substantially reduce the orientation uncertainty. The Google Tango device comes with a natively integrated, high performance 6DoF motion estimation software that makes use of all the integrated sensors. This visual-inertial odometry system computes robust and accurate motion estimates in realtime. Unfortunately, no uncertainty measure for the odometry estimate is provided. We therefore determined the scaling factors Σ_o and the standard deviations σ_o empirically in an offline calibration experiment. We corrupt the estimated motion Δx with the determined noise for each dimension: $x_t = x_{t-1} + \mathcal{N}(\Delta x, \Sigma_o \cdot \Delta x) + \mathcal{N}(0, \sigma_o)$.

IV. Experiments

We evaluated the performance of our approach in two settings. First, a metric accuracy quantification in a motion capture studio and second, a real world localization scenario. For all the experiments, we used a Google Tango device and employed its depth camera for acquiring range measurements from the environment.

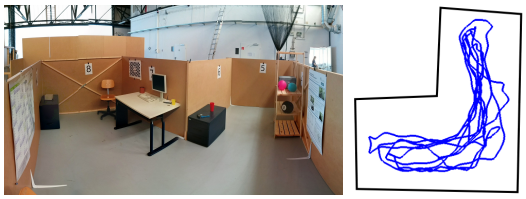

Fig. 3. For accurate metric evaluation, we built an L-shaped and furnished indoor environment in a motion capture hall. Left: a photo of the setup. Right: the corresponding floor plan with the walked ground truth trajectory in blue.

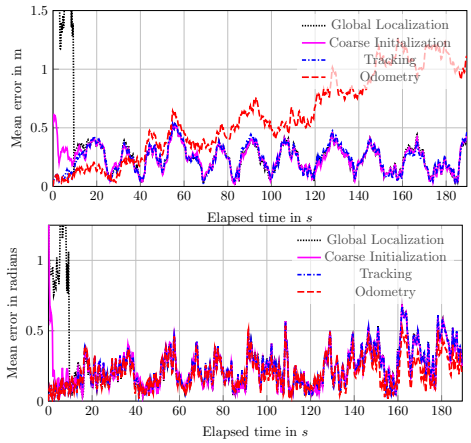

Fig. 4. Accuracy evaluation in the motion capture hall of the 6DoF localization. The top plot shows the error in the x/y plane and the bottom plot the error in orientation.

A. Accuracy Quantification using a Motion Capture Studio

This experiment is designed to quantify the localization accuracy of our technique. For this, we prepared an experimental L-shaped indoor environment within a motion capture hall. We used wooden panels as walls and furnished the environment with tables, chairs and cupboards (see Fig. 3(a)). The idea is to create an environment that is similar to typical indoor scenes. The motion capture system consists of 10 high-framerate infrared cameras that provide submillimeter localization accuracy of dedicated passive infrared markers. We created the floor plan of this environment by taking measurements from reflective markers installed on the top of the walls and corners. The resulting 2D floor plan (see Fig. 3(b)) was the only information given to our algorithm. In the experiment, a user walked through the environment carrying the Google Tango device with attached reflective markers while other people were also present in the environment.

For evaluating the accuracy we performed three different experiments: position tracking, coarse pose initialization, and global localization. For each of these scenarios we performed 10 runs and determined the weighted mean pose estimate over all particles, which we compared to the ground truth position given by the motion capture system. Fig. 4 shows the corresponding results. As can be seen from the figure, the error of the odometry, initialized at the ground-truth position, tends to increase steadily over time, which is the expected behavior.

In the position tracking experiment, we initialized the filter with 5k particles at the ground truth position so that the filter started with zero error. At first the particle cloud spread out because of the noise in the motion model and then stayed at an error around $0.2 - 0.5\,\text{m}$ in x/y and around $5° - 20°$ in orientation. The orientation error is very similar to the results of the odometry. The value is relatively high, most likely because of a slight misalignment of the coordinate frame of the motion capture studio and the local frame of the device. Regarding the translation, our filter has a higher error than the odometry in the very beginning (because of a conservative motion model and the noise (furniture etc.) in the environment) but does not diverge over time. The error in z (which is not plotted) is between 0.2 and $0.4\,\text{m}$. This error is not well restricted because the experimental setup does not include a ceiling and the floor was not visible in the range measurements most of the time. For the second part of the experiment we assumed to have a method to coarsely determine the initial x/y position of the track. This represents additional information, e.g., from a WiFi-based localization system or an approximate starting position provided by the user by clicking on a point in the map. The filter started with 10k particles. We sampled the particle positions with a normal distribution ($3\,\text{m}$ standard deviation), centered around a point randomly determined within a $1\,\text{m}$ distance from the ground truth. The yaw angle is uniformly sampled between $-180°$ and $180°$. Pitch and roll were sampled with low noise around the values provided by the IMU of the device. After the initially poor accuracy, the filter reached a low error, similar to the tracking scenario, after about $10 - 15\,\text{s}$. For global localization, we initialized the particle filter with 200k particles, uniformly distributed over the whole floor plan in x and y. The z-value is uniformly sampled between floor and ceiling. Roll, pitch, and yaw were sampled in the same way as described above. In this scenario, the filter converged approximately at the same time as in the scenario above, which can be explained by the small environment.

B. Localization using Real Floor Plans

To estimate the performance of our approach in real-world settings, we ran experiments in two office buildings on our university campus. We refer to them as *geb74* and *geb79*. They have different footprints and sizes and are furnished differently. We captured the data during typical office hours with people and clutter present in the environment. During the trials a user held the device and walked through the environment as shown in Fig. 1. Dataset *geb74* is $240\,\text{s}$ long, while dataset *geb79* is $85\,\text{s}$ long. We fixed the ceiling-to-floor distance in both environments to $3\,\text{m}$.

Unfortunately, no ground truth trajectory is available for these datasets. We therefore computed a reference trajectory in the environment by initializing our method with a manually determined, highly accurate starting position and performed tracking on this. Even though it is not a real ground truth, it closely represents the real trajectory while providing useful information to metrically evaluate all the

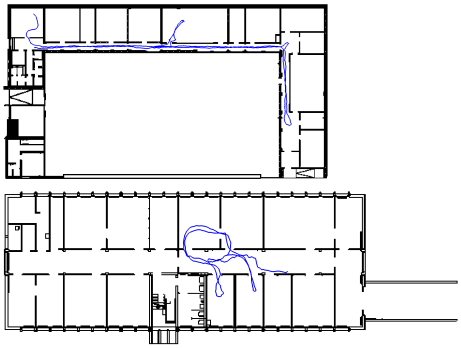

Fig. 5. Floor plans of *geb74* (top) and *geb79* (bottom) with reference trajectories in blue.

other methods. We checked the correctness of this trajectory by careful visual inspection of individual scans projected onto the map. Fig. 5 shows the corresponding floor plans overlaid with the reference trajectories.

For both datasets, we evaluated the approximate initialization scenario with 10k particles and global localization with 200k. We initialized the filter as described in IV-A. The resulting errors compared to the reference trajectory are plotted in Fig. 6 (*geb74*) and Fig. 7 (*geb79*). Fig. 8 depicts a qualitative view of the global localization process. In *geb74* the user first walked along the long corridor, which shows repetitive and ambiguous range appearance. The particles therefore first spread out along the corridor and developed into a multimodal distribution (see Fig. 8 top middle), with two blocks going into opposite directions. Eventually, when the ambiguity was resolved, the filter rapidly converged to a solution similar to the reference trajectory. In *geb79* the user started in the corridor and quickly moved into different offices. Therefore the particles start to form clusters inside the similarly structured rooms (see Fig. 8 bottom middle). The global localization was very close to convergence after 40 s. Yet, until 75 s the distribution stayed bimodal, because of a small cloud of particles representing an alternative pose estimate. The behavior of the filter is similar for both considered scenarios (approximate initialization and global localization), even though the global localization naturally has higher errors and needs longer to converge.

In the experiments above, we performed multiple runs for each scenario. The standard deviation for the translation over these runs was always around 0.03 m for the translational error and around 0.08° for the orientation error.

Our approach initially requires a large set of particles for two reasons. First, there is a large disagreement between the floor plan and the sensor data; second, the state space to be explored is substantially larger than in typical applications with robots navigating in a plane. Nevertheless, our technique is able to converge and yields a steady-state error. It is important to notice that, when this happens, the filter uses less particles, with a minimum amount of 5,000. A situation where our system might fail is when the environment does not provide enough unique features, as for example in a perfectly symmetrical, rectangular environment.

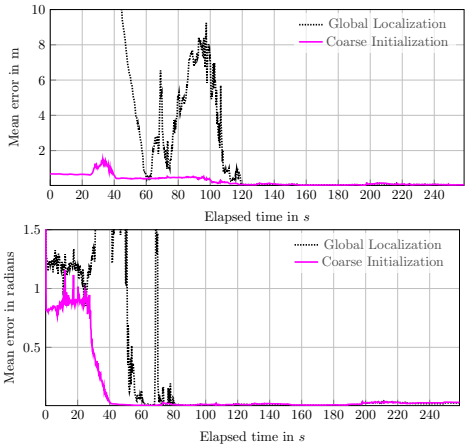

Fig. 6. Development of the error compared to the reference trajectory for *geb74*. The top plot shows the error in the x/y plane and the bottom plot the error in orientation.

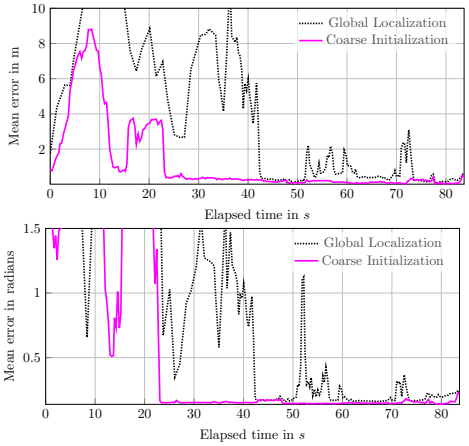

Fig. 7. Development of the error compared to the reference trajectory for *geb79*. The top plot shows the error in the x/y plane and the bottom plot the error in orientation.

To enable other researchers to compare their methods with ours, we will make the datasets used in our experiments publicly available [25].

V. Runtime

The off-board processing for the filter was carried out on an Intel i7 laptop computer. All experiments were performed in online settings and the filter was allowed to drop frames when necessary. The time for the integration of a new range image varied between 1.3 s for global localization (200k particles) and 0.04 s for tracking (5k particles). Dropping frames for online processing did not influence the filters ability to converge to the correct solution, since the visual odometry information is good enough to bridge these small gaps.

VI. Conclusions

This paper presents a 6DoF localization approach for RGB-D smartphones that takes as input RGB-D data and only uses a 2D outline of the environment, such as a floor plan, as the map. In contrast to alternative methods, our approach does not require a dedicated mapping phase and

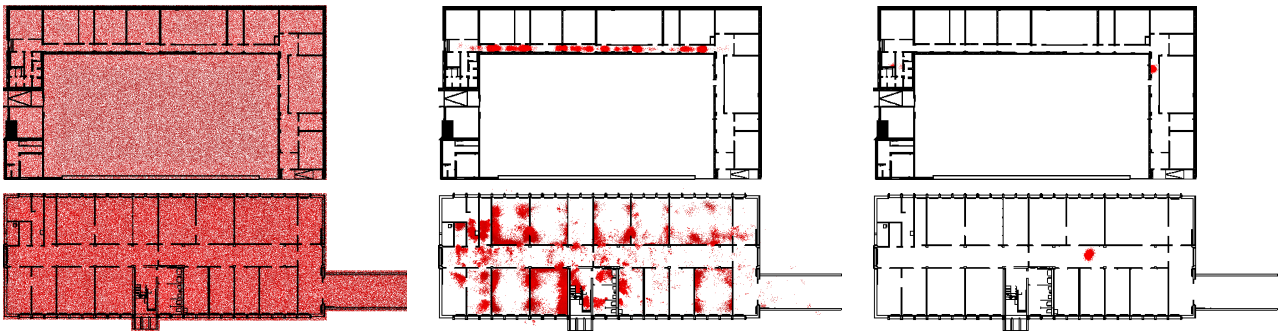

Fig. 8. Global localization for *geb74* (top) and *geb79* (bottom), view from above. These environments contain furniture and we recorded the datasets during office hours. Even though there is a large disagreement between the floor plan and the sensor data, the particle filter converges. From left to right: the filter is initialized (left), shows a multimodal distribution (middle) and then focuses the particles onto the correct position (right).

rather relies on 2D outlines typically available as architectural drawings. We introduce a novel and specific sensor model to deal with the lack of information in the floor plans compared to the real world. Our approach has been implemented and can be executed online on a remote server wirelessly connected to the smartphone. We present the results of extensive experiments carried out in two office buildings and in a motion capture studio. We evaluate our approach in different use cases, which are position tracking, localization with a coarse initial position estimate, and global localization. We show that our approach can be used online and yields accurate results. Furthermore, the filter converges quickly even in environments with long corridors or ambiguous locations.

References

[1] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2005.

[2] 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*, volume 22, 2006.

[3] J. Biswas and M. Veloso. Wifi localization and navigation for autonomous indoor mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2010.

[4] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proc. of the Int. Conf. on Mobile Comp. and Netw.*, 2010.

[5] F. Comblet, F. Peyret, C. Ray, J. Bonnin, and Y.-M. Le Roux. The locoss project of gis its bretagne: status and perspectives. In *International Conference on ITS Telecommunications (ITST)*, 2007.

[6] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the Condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 1999.

[7] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 1999.

[8] F. Duvallet and A. Tews. Wifi position estimation in industrial environments using Gaussian processes. In *Proc. of the IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS)*, 2008.

[9] P. Elinas and J. Little. σMCL: Monte-Carlo localization for mobile robots with stereo vision. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[10] M. F. Fallon, H. Johannsson, and J. J. Leonard. Efficient scene simulation for robust monte carlo localization using an rgb-d camera. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2012.

[11] B. Ferris, D. Haehnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[12] B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM using gaussian process latent variable models. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.

[13] D. Fox. Adapting the sample size in particle filters through kld-sampling. *Int. Journal of Robotics Research*, 22, 2003.

[14] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal on Artificial Intelligence Reserach (JAIR)*, 11, 1999.

[15] S. Grzonka, A. Karwath, F. Dijoux, and W. Burgard. Activity-based Estimation of Human Trajectories. *IEEE Transactions on Robotics (T-RO)*, 8(1):234–245, 2 2012.

[16] S. Ito, F. Endres, M. Kuderer, G. D. Tipaldi, C. Stachniss, and W. Burgard. W-RGB-D: Floor-plan-based indoor global localization using a depth camera and wifi. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2014.

[17] P. Jensfelt, D. Austin, O. Wijk, and M. Andersonn. Feature based condensation for mobile robot localization. In *Proc. of the IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2000.

[18] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(4):376–382, 1991.

[19] M. Paton and J. Kosecka. Adaptive rgb-d localization. In *Computer and Robot Vision (CRV)*, 2012.

[20] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *Proc. of the IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS)*, 2002.

[21] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. of the IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2013.

[22] I. Spasso. *Algorithms for map-aided autonomous indoor pedestrian positioning and navigation*. PhD thesis, École Polytechnique Federale de Lausanne, 2007.

[23] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128, 2001.

[24] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[25] W. Winterhalter, F. Fleckenstein, B. Steder, L. Spinello, and W. Burgard. Datasets for RGB-D localization in 2D floor plans. www.informatik.uni-freiburg.de/∼steder/datasets.html.