Monte Carlo Localization in Hand-Drawn Maps

Bahram Behzadian Pratik Agarwal Wolfram Burgard Gian Diego Tipaldi

***Abstract*— Robot localization is one of the most important problems in robotics. Most of the existing approaches assume that the map of the environment is available beforehand and focus on accurate metrical localization. In this paper, we address the localization problem when the map of the environment is not present beforehand, and the robot relies on a hand-drawn map from a non-expert user. We addressed this problem by expressing the robot pose in the pixel coordinate and simultaneously estimate a local deformation of the hand-drawn map. Experiments show that we can successfully identify the room in which the robot is located in 80% of the tests.**

I. Introduction

Localization, the problem of estimating a robot pose in the environment, is probably one of the most studied and understood problems in mobile robotics. Several solutions have been presented in the last decades, most of them based on probabilistic inference over the space of possible robot configurations [22]. Although the existing approaches have been demonstrated to be robust, efficient and very accurate [3, 6, 16], they mostly rely on one major assumption: The existence of an already available map of the environment, built beforehand with the same sensing modality of the robot.

In some circumstances, however, building such a map could be a nuisance for the user. This is the case, for instance, of vacuum cleaners, lawn mowers, pool cleaners and many other service robots. Often, when the user buys such a robot, he or she wants to use it immediately without waiting for an expensive and time-consuming mapping routine to complete. In some other cases, building an a-priori map is not even possible, for instance in environments that may harm humans, e.g., a minefield or a toxic factory. Moreover, mapping algorithms may result in local minima and the resulting maps might be unusable for navigation. Although automatic tools to detect such inconsistencies exist [14], they require an expert user to analyze the data to correct the map.

In this paper, we address the localization problem when no map of the environment is available beforehand. We propose an algorithm that solely requires a hand-drawn map of the environment, sketched by the user. We believe that drawing such a map puts no burden on the user and is an intuitive task. Furthermore, we do not assume that the map is metrically accurate nor proportional up to a single scale. Objects might be missing, and the deformation of the map might change at different locations. This reduces the ability to accurately localize the robot in a metric sense, since the map is *bended* in different ways and distances are not respected. To address these problems, we extend the Monte Carlo localization algorithm in two ways. First, we express the robot pose in the pixel coordinate of the drawing. This resolves the metrical issues and provides a sort of normalization with respect to the deformation. Second, we augment the state space of the robot with a deformation parameter and track the local deformation of the drawing over time.

All authors are with the University of Freiburg, Institute of Computer Science, 79110 Freiburg, Germany. This work has partly been supported by the European Commission under FP7-610532-SQUIRREL.

Fig. 1. Occupancy grid of the dataset FR079 (top), built using a SLAM algorithm, a hand-drawn map used to localize the robot (middle) and their overlay (bottom).

II. Related work

Robot localization is a widely studied problem [22] and several approaches have been proposed in the past, such as Markov localization [7], Monte Carlo localization (MCL) [3], and multiple hypothesis tracking (MHT) [9]. Those approaches rely on the existence of a prior map and a static environment. Some researchers extended those approaches to be able to handle changes in the environment over time [23, 12, 1]. However, they still rely on metrical maps and some prior information of the environment to be

built beforehand.

Few works have been done with respect to localization with weak prior maps that could be incomplete or metrically inconsistent. Koenig and Simmons [11] propose a localization approach, where the user provides only a topological sketch of the environment. Some authors used floor plan maps available from construction blueprints. Ito et al. [8] propose a localization algorithm that relies on blueprints of the environment. They employ an RGB-D sensor to estimate the walls of the environment and match them to the floor plan. To improve the initialization, they rely on a map of WiFi signal strengths. Setalaphruk et al. [17] employ a multi-hypothesis filter on blueprints. They first compute a Voronoi diagram of the floor plan and store its intersections as landmarks. Second, they use the Voronoi intersection from the current readings and match them to the one computed from the map. Contrary to our approach, they assume that the blueprint of the environment is metrically correct, and its scale is known.

Most of the works using hand-drawn maps exploit them as a mean of communication between the robot and a human operator. Kawamura et al. [10] present an approach where an operator first sketches a rough map of the environment with a set of waypoints to follow. They use a library of known objects to associate perceptions to the sketch map and triangulate the robot position using angle measurements. Skubic et al. [21] propose a sketch interface for guiding a robot. The user draws a sketch of both the environment and instructs the robot to go to certain locations. The system computes a path in the hand-drawn map, and the robot executes it in an open-loop controller, without localizing the robot. Shah et al. [18] propose a similar approach. The focus of the work, however, is on how to extract qualitative instructions for the robot. They translate the instructions to robot commands and localize the robot using an overhead camera system. Yun and Miura [24] proposed and evaluated a quantitative measure of navigability on hand-drawn maps. The work only focuses on qualitative aspects of navigability but does not address the ability to localize in them. Moreover, the sketch maps that they considered are made of line segments and are automatically generated. Skubic et al. [20] propose an approach for qualitative reasoning about sketch maps. The authors are able to extract information such as an obstacle is on the right of the robot and give qualitative commands as turn left or go straight. The work has been extended by Chronis and Skubic [2] with the use of reactive controllers to guide the robot. Forbus et al. [5] developed nuSketch, a qualitative reasoning framework exploiting topological properties and Voronoi diagrams. They provide answers to query like finding places with certain properties or properties of paths. Our work is orthogonal to them, and the proposed localization algorithm can be integrated with any of those control interfaces.

Localization using hand-drawn maps has received very little attention from the robotics community. Parekh et al. [15] presented a technique to perform scene matching between a map of the environment and a sketch using spatial relations. They assume a set of objects is present in both the sketch and the map with known segmentation. They then use particle swarm optimization (PSO) techniques to compute the alignment and the sketch deformation. Matsuo and Miura [13] extended the previous work in a simultaneous localization and mapping (SLAM) framework. They assume, however that the sketch map is composed of rectangles corresponding to building in the scene. Shah and Campbell [19] present an algorithm for controlling a mobile robot using a qualitative map consisting of landmarks and path waypoints. They assume that both the sketch map and the real environment is made of point landmarks and also assume known data associations between them. In contrast to them, our approach does not make any assumption on the format of the sketch map and treats it as a raster image. Moreover, we do not attempt to *transform* the hand-drawn map to reflect the real world but we directly localize the robot in the hand-drawn map. To the best of our knowledge, the proposed approach is the first attempt to localize a robot from a generic hand-drawn map with no further assumptions.

III. Localization in Hand-Drawn Maps

In this section, we describe the extension we made in the original Monte Carlo localization algorithm [3] for localizing in hand-drawn maps. We propose two main extensions. First, we augment the state space of the robot with an additional variable that represents the local deformation of the map. Second, we localize the robot in the pixel coordinate frame of the map, instead of the world coordinate frame. In order to do so, we extended both the motion and the observation model to be locally projected onto the hand-drawn map.

A. Monte-Carlo Localization in Pixel Coordinates

The goal of our work is to estimate the pose of the robot $\mathbf{x}_t \in SE(2)$ and a local scale $s \in \mathbb{R}$ at time t, given the history of odometry measurements $\mathbf{u}_{1:t}$ and observations $\mathbf{z}_{1:t}$. Formally, this is equivalent to recursively estimate the following posterior:

$$p(\mathbf{x}_t, s_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, m) = \eta p(\mathbf{z}_t \mid \mathbf{x}_t, s_t, m) \int_{\mathbf{x}_{t-1}, s_{t-1}} p(s_t \mid s_{t-1}) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, s_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1}, s_{t-1} \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, m) \, \mathrm{d}\mathbf{x}_{t-1} \mathrm{d}s_{t-1}, \quad (1)$$

where η is a normalization factor and m is the hand-drawn map. The motion model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, s_{t-1}, u_t)$ denotes the probability that the robot ends up in state $\mathbf{x}_t$ given it executes the odometry readings $\mathbf{u}_t$ in state $\mathbf{x}_{t-1}$. The distribution $p(s_t \mid s_{t-1})$ represents the transition process for the scale parameter and the observation model $p(\mathbf{z}_t \mid \mathbf{x}_t, s_t, m)$ denotes the likelihood of making the observation $\mathbf{z}_t$ given the robot's pose $\mathbf{x}_t$, the local scale s_t, and the map m.

Following the MCL approach, we approximate the distribution as a weighted sum of Dirac delta functions. The recursive estimation is performed using the sequential importance resampling algorithm [4]. For the proposal, we sample the pose and the scale from the motion model and the scale transition process, respectively. Under the chosen proposal

distribution, we compute the weight of the particle according to the observation model and the recursive term. The particle set is then resampled, at each iteration, according to the weight of the particles.

To compute the number of particles needed, one can use the KLD sampling approach of Fox [6]. The algorithm estimates a discrete approximation of the target distribution using the weighted set of particles. During resampling, it computes the Kullback-Leibler divergence each time a new particle is resampled and stops the process when the divergence is below a confidence level.

B. Proposal Distribution

The purpose of the proposal distribution is to provide a mean for sampling a new set of particles given the current one. In the original MCL algorithm, the proposal distribution is the robot motion model. In our work, we need to provide a proposal distribution for both the robot position $\mathbf{x}$ and the local scale s. We modified the original motion model describe in the MCL algorithm to project the motion of the robot in the image coordinates. Let $\mathbf{x}_t^i$ and s_t^i the pose and scale associated with the i-particle at time t and $\mathbf{u}_t$ the incremental odometry measurement. The new pose of the particle is computed as follow

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i \oplus \mathbf{S}^{-1}(\mathbf{u}_t \oplus \hat{\mathbf{e}}) \qquad \mathbf{S} = \begin{bmatrix} s_t^i & 0 & 0 \\ 0 & s_t^i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

where $\mathbf{u}_t$ represents the odometry reading and $\hat{\mathbf{e}}$ is a sample from the noise term. We sample $\hat{\mathbf{e}} = \begin{bmatrix} \mathbf{q}_t^i & \theta_t^i \end{bmatrix}^T$ from the normal distribution and a wrapped normal distribution, respectively for the translation $\mathbf{q}_t^i$ and the rotational θ_t^i part.

$$\mathbf{q} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{q}}) \quad (3)$$

$$\theta_t \sim \mathcal{WN}(0, \sigma_\theta^2), \quad (4)$$

where $\Sigma_{\mathbf{q}}$ and σ_θ are the covariance matrix for the translation and the standard deviation for the rotation. We modeled the evolution of the scale similarly to a Brownian motion

$$s_{t+1}^i = s_t^i + \epsilon^i \qquad \epsilon \sim \mathcal{N}(0, \sigma_s^2), \quad (5)$$

where ϵ^i is a sample from a normal distribution and σ_s is the standard deviation of the scale increment. Note that we include a standard deviation term in the Wiener process. This is to account for smaller variations than its original formulation. One can formulate Eq. 5 using the standard formulation of the Wiener process by including an additional scaling term to the ϵ^i. The scale sampling has been chosen to be locally close to the scale of the previous step, with small isotropic variations. The Brownian motion was a natural choice for that, given its statistical properties.

Intuitively, given the pose of a particle and its estimated scale, we first sample a zero mean, $SE(2)$ transformation from the odometry noise and perturb the odometry measurement accordingly. We then project the perturbed odometry on the hand-drawn map and apply the projected transformation to the robot pose.

C. Observation Model

After we sample the particles according to the proposal distribution, we follow the importance sampling principle and compute the weights of the particle set. With our choice of proposal, the weight of each particle must be proportional to the observation model $p(\mathbf{z} \mid \mathbf{x}, s, m)$, where we omitted the time index for notational simplicity. Intuitively, this model describes the likelihood of the measurement $\mathbf{z}$, given the hand-drawn map m, the local scale s and the pose of the robot in the image coordinates $\mathbf{x}$. In this work, we consider 2D laser range finders as sensors. The measurements $\mathbf{z} = [z_1, \cdots, z_K]^T$ consist of a set of range values along a set of corresponding directions $\mathbf{a} = [\alpha_1, \cdots, \alpha_K]^T$. We employ the beam based model [22] and modify it to project the observations in the hand-drawn map.

Formally, let z_i be the i-th range measurement along the angle α_i. Let us trace a ray on the map m from the robot pose $\mathbf{x}$ to the closest obstacle in the map and be $\hat{z}$ the measured distance. The original formulation of the beam based model considers $\hat{z}$ as being expressed in world coordinates and describes the measurement distribution as a mixture of four components

$$p(z_i \mid \hat{z}_i) = \begin{bmatrix} w_{\text{hit}} \\ w_{\text{dyn}} \\ w_{\max} \\ w_{\text{rnd}} \end{bmatrix}^T \begin{bmatrix} f_{\text{hit}}(z_i, \hat{z}_i) \\ f_{\text{dyn}}(z_i, \hat{z}_i) \\ f_{\max}(z_i, \hat{z}_i) \\ f_{\text{rnd}}(z_i, \hat{z}_i) \end{bmatrix}, \quad (6)$$

where f_{hit} models the measurement noise, f_{dyn} models the unexpected obstacles not present in the map, $f_{\max}$ models the sensor maximum range, and f_{rnd} models a uniformly distributed random measurement. The functions are defined as following

$$f_{\text{hit}}(z, \hat{z}) = \mathcal{N}(z; \hat{z}, \sigma_z) \quad (7)$$

$$f_{\text{dyn}}(z, \hat{z}) = \text{TEXP}(z; \lambda, \hat{z}) \quad (8)$$

$$f_{\max}(z, \hat{z}) = \mathcal{U}(z; 0, z_{\max}) \quad (9)$$

$$f_{\text{rnd}}(z, \hat{z}) = \mathcal{U}(z; z_{\max} - \delta, z_{\max} + \delta). \quad (10)$$

Here, $\text{TEXP}(x; \lambda, a)$ denotes a truncated exponential distribution with parameter λ and support between 0 and a. $\mathcal{U}(x; b, c)$ denotes a uniform distribution between b and c and δ is a window parameter. To account for the deformations, we need to project the real measurements coming from the sensor to the image coordinate frame by applying the estimated transformation. In our case, this entails to scale all the ranges according to estimated scale s. The resulting observation model is

$$p(z_i \mid s, \hat{z}_i) = p(\frac{z_i}{s} \mid \hat{z}_i) \quad (11)$$

All the parameters of the model have been learned from real data. To collect the data for the learning phase, we positioned the robot at fixed locations in the environment and draw few different sketches for each location. Then, for each sketch and location, we performed grid search to find the best scale. Given the scale, we computed the maximum-likelihood estimate of the parameters, following the approach described in [22].

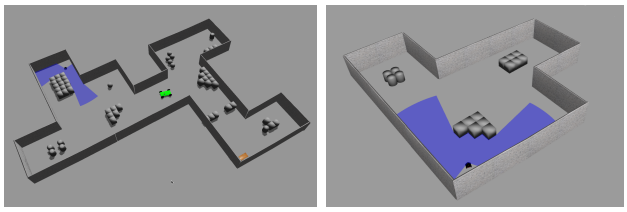

Fig. 2. Simulated Apartment (left) and Room (right) environment in Gazebo. The area covered by the simulated laser scanner is shown in blue.

IV. Experimental Results

We evaluated our approach in both simulated and real environments. For the simulation, we used the Gazebo simulator available in ROS. We created two virtual environments, resembling a simulated room (Room) and a simulated apartment (Apartment). Figure 2 depicts the two simulated environments together with a simulated robot. In the real world experiment, we tested our algorithm in our building (FR079), whose map and a full dataset are publicly available. We chose this dataset for two reasons. First it contains very challenging aspects of localization algorithms, given the presence of many, similarly looking rooms. Second, similar data is publicly available online, and other researchers can use that to replicate our results. Since not everyone has the same *artistic capabilities*, we asked nine people to walk in the environment and draw a sketch. Figure 5 shows all the nine sketches together.

We use the same parameters for all our experiments and the simulated robot. For the proposal distribution, we have $\Sigma_{\mathbf{q}} = 0.1I$ as covariance matrix for the translational component, $\sigma_\theta = 0.05$ for the rotational noise, and $\sigma_s = 0.1$ for the scale noise. With respect to the observation model, we set $\sigma_z = 0.1$ according to our sensor specification, and we estimated the rest of the parameters from data. The resulting estimated values are $\lambda = 0.1$ for the exponential distribution, $\delta = 0.01$ for the max range, and $w_{\text{hit}} = 0.005$, $w_{\text{dyn}} = 0.5$, $w_{\max} = 0.3$, $w_{\text{rnd}} = 0.4$ for the mixture weights. We also subsampled the range measurements to only 10 beams per scan, to compensate for the independence assumption of the laser beams. We initialized the filter by drawing the particles uniformly over a region of the map drawn by the user, for the position, uniformly between $-\pi$ and π for the orientation, and uniformly between 0.01 and 1 for the scale. The maximum number of particles being used in these experiments is 100K. However, in most of experiment half or even a quarter of this amount is sufficient. The square region was about the size of a room, simulating a plausible initial guess from the user.

A. Simulated Experiments

For the simulation, we used the Room environment as a proof of concept scenario. We let the robot start in the lower right corner of the room and performed 4 different paths. We simulated a weak prior on the initial robot position by sampling the particles uniformly in a square of 150×150 centered at the robot position. We obtained a success rate of 100% in localizing the robot. Some videos of the whole experiment are available on Youtube[1].

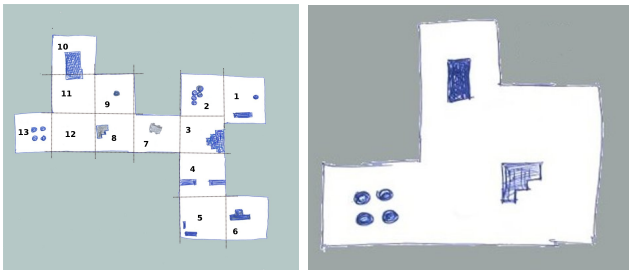

Fig. 3. Hand-drawn map of the Apartment (left) and the Room (right) environment. The dashed squares represents the rooms we used in our experiment.

Fig. 4. Overlay of the hand-drawn map over the Apartment environment. The hand-drawn map has been manually aligned. Note the non uniform scaling and the distortions present.

For the second experiment in the Apartment environment, we simulated a series of navigation tasks, where the robot starts in a certain location and is instructed to reach a specific room. We believe this is the natural application of our approach, where the user sketches the map, a rough location of the robot and ask the robot to reach a particular location, marking it on the map. We set up our experiments in the following way. After we draw a sketch of the environment, we subdivided it into small squares, each representing a room to be reached. We then randomly generated 10 different navigation tasks, in the form of go from room A to room B. For each sequence, we performed 10 runs of our localization algorithm with different random seeds. We considered a sequence as a success if the robot, at the end of the trajectory, is localized in the correct room.

Figure 3 shows the hand-drawn map used in this experiments, together with our subdivision. To understand the differences between the real map and the hand-drawn one, Figure 4 shows an overlay of the two, where we manually rotate, scaled and aligned the two maps. Even under manual alignment, one can see that the scaling of the sketch is not uniform with respect to the real map and that many distortions are present. Table I shows the results of the experiment, together with the sequences we used. Our approach has an overall success rate of 93%. We only had a few failures in

[1] https://goo.gl/mCOsCK

room a → b	Chance of Success
1 → 6	100%
1 → 10	100%
6 → 1	100%
6 → 10	100%
8 → 1	100%
8 → 6	100%
10 → 1	100%
10 → 6	70%
13 → 6	80%
13 → 10	80%
Total	93%

TABLE I

SUCCESS RATE FOR THE APARTMENT ENVIRONMENT

the paths from 10 to 6 and 13 to 6. Note that this are the most challenging paths, since are the longest and traverse the whole map. We also had some problems from 13 to 10. This was due to the ambiguity in the two corners, where the robot was mistakenly localized in the wrong one.

All the trajectories for this experiment are publicly available on youtube[2].

B. Real World Experiments

Figure 5 shows the hand-drawn maps of Building 079 used for this experiment. The numbers in the figure represent the different rooms we identified in the environment. In a way similar to the simulated experiments, we randomly generate 7 navigation sequences and, for each sequence, we performed 10 runs of our localization algorithm with different random seeds. Table II shows the localization results with the real data for each run. The ratio difference denotes the absolute difference between the ratio (length/width) of the original occupancy grid map and each hand-drawn map. Figure 6 illustrates the success rate as a function of the difference in ratios. We see that localization has a high success rate, almost 80%, when the difference in the ratio of the hand-drawn map is relatively low. The success rate of localization decline, when this difference increases. The table shows the highest failure in the test run from room 9 to 12. Room number 9 is fully occupied with furniture that heavily distorted the image of the walls in the laser scan, therefore, the robot was not able to localize properly in the beginning. The robot randomly localized itself in any of the other rooms looking alike. The lowest successful rate was obtained when using the map No. 2. The user has drawn the doors in an unusual way that the robot can not recognize the entrance and exit properly. The localization results for AIS map-0[3] and map-3[4] and is publicly available on youtube.

In addition to FR079, we also tested our method on the Intel dataset. The videos from the Intel dataset are publicly available on youtube[5].

We believe the reason our approach failed in more extreme ratio differences is due to the estimation of a single scale.

[2]https://goo.gl/D0Oou7
[3]https://goo.gl/XkCG1s
[4]https://goo.gl/XVQA5Q
[5]https://goo.gl/UPFo1q

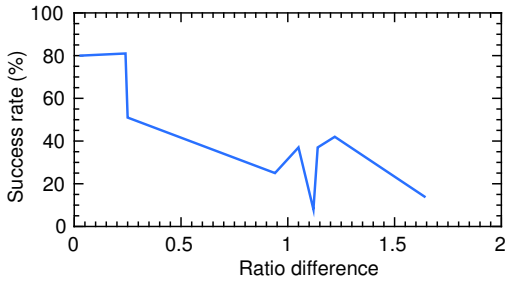

Fig. 6. Percentage of FR079 runs successfully localized.

The framework here presented can be straightforwardly extended to account for multiple scales and shearing. A proper exploration of different kind of transformation is subject to future work.

V. CONCLUSIONS

In this paper, we addressed the problem of robot localization when no accurate map of the environment is available and the robot has to solely rely on a hand-drawn map sketched by the user. To do so, we extended the classical Monte Carlo localization algorithm in two ways. First, we propose to localize with respect to the image coordinate frame. Second, we track, together with the pose of the robot, a local deformation of the hand-drawn map. Since no metric information is available on the hand-drawn map, we propose to evaluate the localization in terms of coarse localization at the level of rooms of the environment. We evaluated our approach in both simulated and real environments and achieved a correct localization, up to the room level, of about 80% of the cases when the ratio of the sketch map resembles the real environment. We expect to achieve this kind of performance also on blueprints, which are metrically accurate, even when the scale of blueprint is not known in advance. We believe this is a starting point that addresses a very challenging problem with potential applications. In future, we plan to extend our approach to incorporate more sophisticated distortion models and employ it for navigation purposes.

REFERENCES

[1] Joydeep Biswas and Manuela Veloso. Episodic non-markov localization: Reasoning about short-term and long-term features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[2] George Chronis and Marjorie Skubic. Robot navigation using qualitative landmark states from sketched route maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[3] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

[4] Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

[5] Kenneth D Forbus, Jeffrey Usher, and Vernell Chapman. Qualitative spatial reasoning about sketch maps. *AI magazine*, 25, 2004.

[6] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *International Journal of Robotics Research*, 22, 2003.

[7] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[8] Seigo Ito, Felix Endres, Markus Kuderer, Gian Diego Tipaldi, Cyrill Stachniss, , and Wolfram Burgard. W-rgb-d: Floor-plan-based indoor

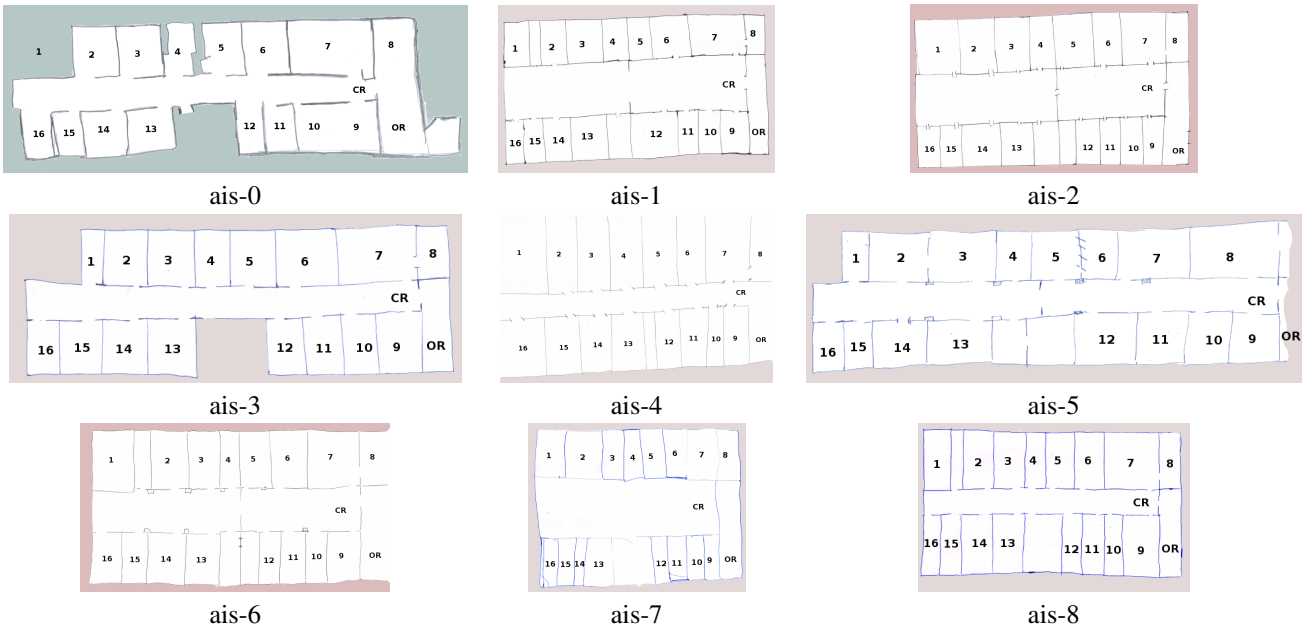

Fig. 5. Sketch maps for the FR079 environment used for the experiments.

room-to-room	ais-3	ais-0	ais-5	ais-6	ais-1	ais-2	ais-8	ais-4	ais-7
ratio difference	0.03	0.24	0.25	0.94	1.05	1.12	1.14	1.22	1.64
15 to 16	100	100	100	80	100	60	100	20	40
14 to 13	80	100	0	40	60	0	60	100	0
4 to 6	100	100	100	0	20	0	0	80	0
9 to 12	0	0	60	0	0	0	0	0	0
5 to 11	100	100	0	0	0	0	0	0	20
CR to 12	80	100	100	0	0	0	100	0	0
11 to OR	100	70	0	60	80	0	0	100	40
success rate %	80	81	51	25	37	8	37	42	14

TABLE II

RESULTS OF THE REAL WORLD EXPERIMENTS.

global localization using a depth camera and wifi. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[9] Patric Jensfelt and Steen Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17, 2001.

[10] Kazuhiko Kawamura, A Bugra Koku, D Mitchell Wilkes, Richard Alan Peters, and A Sekmen. Toward egocentric navigation. *International Journal of Robotics and Automation*, 17, 2002.

[11] Sven Koenig and Reid G Simmons. Passive distance learning for robot navigation. In *ICML*, 1996.

[12] Tomáš Krajnık, Joao Santos, Bianca Seemann, and Tom Duckett. Froctomap: an efficient spatio-temporal environment representation. *Advances in Autonomous Robotics Systems*, 2014.

[13] Keisuke Matsuo and Jun Miura. Outdoor visual localization with a hand-drawn line drawing map using fastslam with pso-based mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[14] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. A Statistical Measure for Map Consistency in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[15] Gaurav Parekh, Marjorie Skubic, Ozy Sjahputera, and James M. Keller. Scene matching between a map and a hand drawn sketch using spatial relations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

[16] Jörg Röwekämper, Christoph Sprunk, Gian Diego Tipaldi, Stachniss Cyrill, Patrick Pfaff, and Wolfram Burgard. On the Position Accuracy of Mobile Robot Localization based on Particle Filters Combined with Scan Matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[17] Vachirasuk Setalaphruk, Atsushi Ueno, Izuru Kume, Yasuyuki Kono, and Masatsugu Kidode. Robot navigation in corridor environments using a sketch floor map. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2003.

[18] Danelle Shah, Joseph Schneider, and Mark Campbell. A robust sketch interface for natural robot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[19] Danelle C Shah and Mark E Campbell. A qualitative path planner for robot navigation using human-provided maps. *International Journal of Robotics Research*, 32, 2013.

[20] Marjorie Skubic, Sam Blisard, Andy Carle, and Pascal Matsakis. Hand-drawn maps for robot navigation. In *AAAI Spring Symposium, Sketch Understanding Session*, page 23, 2002.

[21] Marjorie Skubic, Derek Anderson, Samuel Blisard, Dennis Perzanowski, and Alan Schultz. Using a hand-drawn sketch to control a team of robots. *Autonomous Robots*, 22, 2007.

[22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[23] Gian Diego Tipaldi, Daniel Meyer-Delius, and Wolfram Burgard. Lifelong localization in changing environments. *International Journal of Robotics Research*, 32, 2013.

[24] Jooseop Yun and Jun Miura. A quantitative measure for the navigability of a mobile robot using rough maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.