Robot Navigation in Hand-Drawn Sketched Maps

Federico Boniardi Bahram Behzadian Wolfram Burgard Gian Diego Tipaldi
Autonomous Intelligent Systems Group, University of Freiburg, Germany
{boniardi, behzadb, burgard, tipaldi}@informatik.uni-freiburg.de

***Abstract*—Many navigation systems for mobile robots use a metrical map for localization and trajectory planning. In several situations, however, building such a map upfront is hard or even impossible. In this paper, we present a combined approach for robot localization and navigation solely relying on a hand-drawn sketch of the environment. In our work we model the sketch as a two-dimensional manifold with an unknown metric. We use Monte Carlo Localization for estimating the state of the robot in combination with the local deformation of the sketch. Accordingly, our approach can track both, the pose of the robot and the local metric properties of the drawing. Furthermore, we couple the pose tracker with a path planner that uses the global information of the sketch and the estimated local metric to robustly perform navigation tasks in real world environments.**

I. INTRODUCTION

Several approaches to mobile robot navigation rely on a metrically consistent map where desired paths are executed combining online pose estimation and trajectory planning. Although such methods have been shown to be remarkably robust as well as highly efficient, they all rely on the assumption that an accurate map of the scenario is available beforehand. Typically, such a map is retrieved with human-driven or active exploration. This can however be hard or even impossible in many circumstances: rescue or harmful operations are just few examples where teleoperating a robot could be undesirably time-consuming or difficult. Furthermore, both research and industry are increasingly investigating new service applications in contexts where robots are designed to operate and interact with non-expert people. Consequently, an intuitive representation of maps and environments could be helpful to not overburden the user with preliminary tedious operations.

In this work we address the problem of performing autonomous navigation tasks only relying upon a hand-drawn sketch of indoor environments whose metrical description is not available. The navigation system proposed here models the sketch as a two-dimensional metric manifold. We base our work on the assumption that the local metric on the sketch can be approximated by means of suitable scales that account for the local deformation along a particular reference frame of the drawing. Consequently, a metric conversion between the real world and the robot's sensors can be applied in order to match the measurements with the map. On top of this, we develop a path planner that combines the global connectivity of the sketch with the scaled sensor measurements (Fig. 1). As a consequence, inconsistencies due to the approximate sketch as well as unmapped obstacles can be safely avoided. The work presented here expands upon the ideas proposed by Behzadian *et al.* [1], where hand-drawn maps together with an enhanced Monte Carlo method have been employed for robot localization in indoor environments.

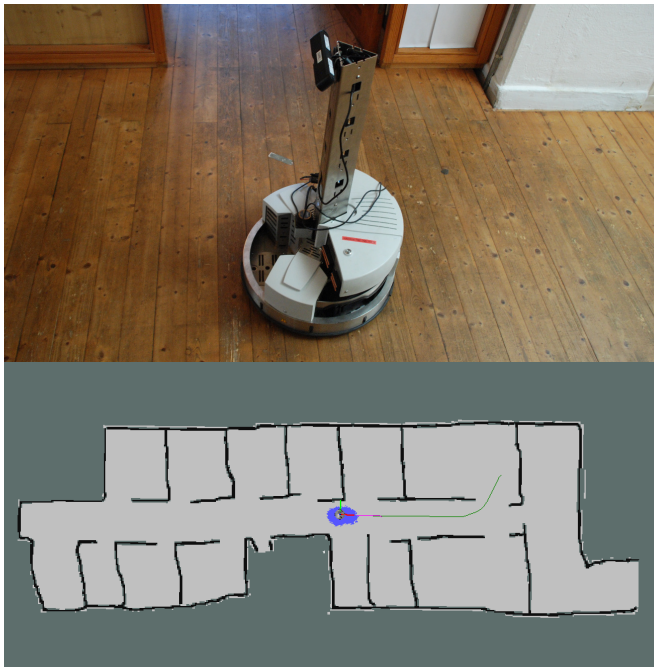

Fig. 1. The navigation system proposed in this paper and the robot used for the experiments. The robot performs navigation only with a laser rangefinder.

In the remainder of the paper we first discuss the related work in Sec. II. Then, in Sec. III, we present the proposed distortion model and its integration within both the Monte Carlo Localization and the path planning algorithms. Finally, we present experimental results and evaluations in Sec. IV.

II. RELATED WORK

Thus far, there has been limited research in the context of robot navigation without metrically consistent maps. An early attempt to perform localization and navigation only relying on the topological structure of the environment has been proposed by Koenig and Simmons [2]. The authors analyzed the problem as a Partially Observable Markov Decision Process (POMDP) and developed an extension of the Baum-Welch algorithm to learn the metric conversion between the topological map and real world. Navigational actions are consequently planned according to the underlying POMDP model. In this work, however, only simple maps with regular patterns have been studied. Konolige *et al.* [3], proposed a hybrid method where a topological map obtained from a SLAM graph is used to plan

global paths. They encode those paths as a coarse sequence of way points on the graph. The robot heads towards the way points following metrically consistent trajectories output by a local planner that uses the current laser scans. Setalaphruk *et al.* [4] considered indoor navigation tasks relying on a blueprint of a building, which is converted into a topological map using a Voronoi decomposition. The authors only localize the robot in a topological sense by adopting Multi-Hypotheses Tracking. Autonomous navigation is achieved by exploiting the metric properties of the blueprint coupled with a local path planner for simple obstacle avoidance.

Using hand-drawn sketches of environments to enhance the spatial cognition of robots has been already studied by both the robotics and AI communities. Research so far has mainly investigated the topic from the perspective of human-robot interaction, focusing on the usability of sketches as means for interoperating with robots. Indeed, many works have approached the problem by emulating a human-like navigation based on landmark matching. Skubic *et al.* [5], for instance, focused on a human-robot interface to communicate navigation tasks to the robot. A user provides a sketch of the scene and a trajectory. During navigation, a set of rules is used to determine the position of the robot along the path and selects an appropriate behavior-based controller to follow it. Similar approaches and interfaces have been studied in the past [6], [7], [8] where fuzzy methods substitute deterministic state controllers. The corresponding authors also investigated the interactive control of a team of robots [9].

A slightly different perspective has been suggested by Shah and Campbell [10]. Their work focuses on creating a path planner that outputs trajectories that are consistent with a hand-drawn map in terms of spatial relationships. A further optimization layer is responsible for avoiding collisions. Kawamura *et al.* [11] developed a navigational system on sketch maps where trajectories are defined by a set of way points. A topological estimation of the robot pose is computed by triangulating the landmarks perceived by the robot's sensors (the *Sensory Egosphere*) with the qualitative landscape observable from the path's way points (the *Landmark Egosphere*). The robot navigates heading towards those way points that output the best matching between the two egospheres.

Yun and Miura [12] aimed to provide a quantitative measure of the navigability of sketched maps. To do so, they employed a localization system based on Multi-Hypotheses-Tracking and a basic planner that executes a set of primitive motions. According to the success rate of goal-reaching tasks, they fit a navigability function dependent upon shape and dimension of the sketch as well as uncertainty of the robot tracked position and landmarks' existence. Although this seems to be the only approach so far that designed an objective measure of the quality of the sketch, only simulated and simplified scenarios have been used in the experiments. Parekh *et al.* [13] proposed a method to match occupancy grid maps with sketches by means of particle swarm optimization (PSO) techniques. In Li *et al.* [14] visual segmentation is used to match landmarks with a sketch using a database of objects. The robot is given only a sketched path together with an approximate distance between the starting point and the goal position. The path is then followed by triangulating the observed landmarks.

Although navigation in hand-drawn maps has been recently addressed in research, very few works have investigated the possibility of extending a standard navigational system that relies on robot pose tracking in the hand-drawn map. Matsuo and Miura [15] suggested a method to track the robot state using Monte Carlo localization. During navigation, they combine a fastSLAM algorithm with Particle Swarm Optimization to fit the hand-drawn map to a local occupancy grid obtained from stereo-data. Albeit the authors' claims, they assumed a very simple configurations of sketches with only rectangular shapes representing buildings. Conversely, the methods described in this paper aims to provide a general approach to navigation in sketch map relying only on minimal assumptions, namely, topological consistency and moderate deformations. Moreover, the approach described here does not attempt to fit the sketch to the real world, on the contrary, we try to convert sensors' observations and measurements to the sketch. Such approach has been used in [1] and seems, according to our investigation, to be the first attempt towards this direction.

III. NAVIGATING IN SKETCHED MAPS

In this section we describe how we tackle the problem of metrical inconsistencies between the sketch map and the real world. We model the problem in terms of estimating a set of suitable scale factors that convert the standard euclidean metric into the local metric of the sketch.

In order to mathematically formalize this, assume to be given a sketched map $\mathscr{S} := (O_{\mathscr{S}}, \mathrm{R}_{\mathscr{S}})$ encoded as a rasterized image $O_{\mathscr{S}}$ together with a reference frame $\mathrm{R}_{\mathscr{S}}$. The sketch describes a real world environment $\mathscr{W} := (O_{\mathscr{W}}, \mathrm{R}_{\mathscr{W}})$, again encoded as a rasterized image. We assume that the sketch satisfies the following consistency property: there exists a diffeomorphism $\Phi : O_{\mathscr{W}} \subset \mathbb{R}^2 \longrightarrow O_{\mathscr{S}} \subset \mathbb{R}^2$ which transforms pixelwise the free space of the related occupancy grid maps. Thus, any robot trajectory $(\mathbf{x}_t^{\mathscr{W}})_{t\geq 0} = ([x_t^{\mathscr{W}}, y_t^{\mathscr{W}}, \theta_t^{\mathscr{W}}])_{t\geq 0} \subseteq \mathrm{SE}(2)$ in the metrically consistent map $\mathscr{W}$ can be described in the sketched map in terms of composition of the path with the transformation Φ, that is, $([\Phi(x_t^{\mathscr{W}}, y_t^{\mathscr{W}}), \theta_t^{\mathscr{W}}])_{t\geq 0} \subseteq \mathrm{SE}(2)$. As a consequence, with $\partial\Phi$ being the Jacobian operator,

$$\int_{\mathbf{x}_0^{\mathscr{S}}}^{\mathbf{x}_t^{\mathscr{S}}} d\mathbf{x}^{\mathscr{S}} = \mathrm{T}_{\mathscr{S}\to\mathscr{W}} \int_{\mathbf{x}_0^{\mathscr{W}}}^{\mathbf{x}_t^{\mathscr{W}}} \begin{bmatrix} \partial\Phi(x_t^{\mathscr{W}}, y_t^{\mathscr{W}}) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} d\mathbf{x}^{\mathscr{W}}, \qquad (1)$$

where $\mathrm{T}_{\mathscr{S}\to\mathscr{W}}$ is the transformation that aligns the origin of the sketch to the origin of the world. Multiplications such as the one in the right-hand term of (1), where only planar components of $\mathrm{SE}(2)$ are affected, will be henceforth shorthanded as $\langle M, \cdot \rangle$, with $M \in \mathbb{R}^{2\times 2}$. According to (1) and observing that the infinitesimal increment $d\mathbf{x}_t^{\mathscr{W}}$ can be approximated by means of the odometry reading $\mathbf{u}_t$ as $\Delta\mathbf{x}_t^{\mathscr{W}} := \mathbf{x}_t^{\mathscr{W}} - \mathbf{x}_{t-1}^{\mathscr{W}} = (\mathbf{x}_{t-1}^{\mathscr{W}} \oplus \mathbf{u}_{t-1}) - \mathbf{x}_{t-1}^{\mathscr{W}}$, it is natural to describe the motion of a robot in a sketched map in terms of a matrix valued function $\Sigma : O_{\mathscr{W}} \subset \mathbb{R}^2 \longrightarrow \mathrm{GL}(2)$ which defines the local deformation between the real world and the sketch. In terms of manifold representation, we can think of a sketched map as a two-dimensional Riemannian manifold in $\mathbb{R}^2$ and we characterize the motion of the robot on the manifold in terms of the Riemannian metric induced by the positive definite matrix $[\partial\Phi(x^{\mathscr{W}}, y^{\mathscr{W}})]^T \partial\Phi(x^{\mathscr{W}}, y^{\mathscr{W}})$. We observe that, we used the assumption of the existence of a diffeomorphism between the sketch and the real world map to provide a mathematical

insight on the role of the scales. However this assumption can be relaxed due to the discrete nature of the rasterized images $O_{\mathscr{S}}$ and $O_{\mathscr{W}}$.

In this work, we assume that the sketch respects the orthogonality and parallelism of the real world, up to minor inaccuracies. We believe that this is not a strong assumption, since people in general have a good understanding of orthogonality and parallelism of walls in indoor environments. This assumption also relies on the empirical evidence that an indoor environments commonly have parallel and perpendicular walls. Under these hypotheses, we can simplify our model so that the global transformation Φ can be described in terms of composition of a metric and orientation preserving transformation and a local distortion with approximately no shearing. Formally, we suppose that

$$\partial\Phi \approx \mathcal{R}(\phi)S : O_{\mathscr{W}} \subset \mathbb{R}^2 \longrightarrow GL(2), \tag{2}$$

where, $\mathcal{R}(\phi) \in \mathrm{SO}(2)$ is a constant rotation matrix and S is a diagonal matrix-valued function. The matrix $\mathcal{R}(\phi)$ can be thought as the rotational component of the transformation between $R_{\mathscr{W}}$ and a suitable reference frame $R_{\mathscr{S}}$ in the sketch. Consequently, $\mathcal{R}(\phi)$ can be absorbed in the rotational component of the transformation between the world reference frame and the sketch $\mathrm{T}_{\mathscr{S}\to\mathscr{W}}$, i.e., $\partial\Phi \approx [\mathrm{T}^{rot}_{\mathscr{S}\to\mathscr{W}}]S$. Henceforth, we assume this to be valid.

In the rest of this section we outline the two main components of the navigation stack, namely the localization algorithm and the path planning routine. As it is reported below, we show how Monte Carlo Localization can be employed to estimate the deformation factors together with the current robot's pose.

A. Monte Carlo Localization and Scale Estimation

To track the robot pose $\mathbf{x}^{\mathscr{S}}_t = [x_t, y_t, \theta_t]^{\mathscr{S}}$ while estimating the local scale operator S, we employ an extension of Monte Carlo Localization algorithm [16]. We enhance the current robot state with two scale parameters $a, b \in \mathbb{R}_+$ which encodes the local deformation along the reference axes of $\mathrm{R}_{\mathscr{S}}$. More precisely, we can approximate the scaling operator in (2) as

$$\begin{aligned}\partial\Phi(x^{\mathscr{W}}_t, y^{\mathscr{W}}_t) \approx &[\mathrm{T}^{rot}_{\mathscr{S}\to\mathscr{W}}]\begin{bmatrix} a_t & 0 \\ 0 & b_t \end{bmatrix} =: \\ =&:[\mathrm{T}^{rot}_{\mathscr{S}\to\mathscr{W}}]S_t =: \Sigma_t.\end{aligned} \tag{3}$$

Such approach is a straightforward extension to the method described by Behzadian *et al.* [1] where a single, frame independent scale is estimated together with the robot's pose.

Accordingly, we can apply the standard Bayes filter to the augmented state $\xi_t := (\mathbf{x}^{\mathscr{S}}_t, a_t, b_t)$ conditioning on the history of commands $\mathbf{u}_{1:t}$ and sensor measurements $\mathbf{z}_{1:t}$. That is, the joint posterior can be estimated applying the recursive relation

$$\begin{aligned}p(\xi_t \mid \mathbf{u}_{1:t}, \mathbf{z}_{1:t}) \propto\ & p(\mathbf{z}_t \mid \xi_t) \cdot \\ &\cdot \int p(\xi_t \mid \xi_{t-1}, \mathbf{u}_t) p(\xi_{t-1} \mid \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})\, d\xi_{t-1}.\end{aligned} \tag{4}$$

Monte Carlo Localization represents the state belief with a set of weighted samples, usually referred to *particles*, which approximates the state distribution as a weighted sum of Dirac's delta distributions. At each iteration step, particles are propagated according to an evolutionary model of the robot state, and finally resampled with *importance sampling* with respect of weights defined by the likelihood of the observations.

1) Model for the Evolution of the State: We assume conditional independence of both the local scales from the robot pose [1] and the local scales between each others. Using this assumption and (1), the posterior distribution of the state ξ_t in the integral term of (4) can be approximated as follows:

$$\begin{aligned}&p(\xi_t \mid \xi_{t-1}, \mathbf{u}_t) \approx \\ &\quad p(\mathbf{x}^{\mathscr{S}}_{t-1} + \langle \Sigma_{t-1}, \Delta\mathbf{x}^{\mathscr{W}}_t \rangle) p(a_t \mid a_{t-1}) p(b_t \mid b_{t-1}).\end{aligned} \tag{5}$$

Thus, owing to (5), it is natural to choose a *proposal distribution* that combines the probabilistic counterpart of the motion model in (1) with two independent martingales that describe the evolution of the local scales, formally

$$\begin{cases} \mathbf{x}^{\mathscr{S}}_t := \mathbf{x}^{\mathscr{S}}_{t-1} \oplus \langle \Sigma_{t-1}, \mathbf{u}_{t-1} + \boldsymbol{\varepsilon}_{t-1} \rangle, \\ a_t := a_{t-1}\eta_{t-1}, \\ b_t := b_{t-1}\zeta_{t-1}, \end{cases} \tag{6}$$

where the independent noise terms in (6) are chosen so that $\boldsymbol{\varepsilon}_t \sim [\mathcal{N}_{0,\sigma_j}]^3_{j=1}$ is Gaussian-distributed zero-mean random vector with independent components (Wrapped Gaussian for the orientation component). The multiplicative noise η_t, ζ_t can be chosen to be Gamma-distributed with unitary expected value so the noise ranges in the entire scale space $\mathbb{R}_+$. $\Gamma(\sigma_i^{-2}, \sigma_i^2)$ is a natural choice for multiplicative white noise as it provides a parameter for the variance, while the mean is constant. Martingales are also a natural choice for propagating the scales because the local changes of the deformations may be completely arbitrary and no prior bias should be expected.

2) Observation Model: Given the endpoints $\mathbf{z}_t := (z_{i,t})_{i=1}^N$ related to the sensor measurements at time t, the likelihood of sensor measurements $p(\mathbf{z}_t \mid \xi_t)$ is computed by converting the readings according to the metric in the sketch. More precisely, we consider a generalization of the *likelihood fields model* for range finders described in [16]. Set $\mathrm{T}_{\mathscr{S}\to\mathscr{R}}$ to be the transformation between the sketch and the robot reference frame and defined $o_{i,t}$ to be the closest obstacle to the beam endpoint $z_{i,t}$ and $z'_{i,t} := \mathrm{T}^{tran}_{\mathscr{S}\to\mathscr{R}} + S_t[\mathrm{T}^{rot}_{\mathscr{S}\to\mathscr{R}}]z_{i,t}$ the scaled scan, we employ the following approximation for the sensors' likelihood model,

$$p(\mathbf{z}_t \mid \xi_t) \approx \mathcal{L}_{\lambda,a_t}(a'_t)\mathcal{L}_{\nu,b_t}(b'_t)\prod_{i=1}^{N}\mathcal{N}_{o_{i,t},\sigma^2}(z'_{i,t}). \tag{7}$$

In (7), $\mathcal{N}_{o_{i,t},\sigma^2}$ is a Gaussian kernel centered in $o_{i,t}$ with variance σ^2, $\mathcal{L}_{\beta,s}$ is a Laplace distribution centered in s with scale parameter β and (a'_t, b'_t) are virtual measurements for the scales obtained by raytracing the sketch from the predicted pose of the best particle. More formally, we infer (a'_t, b'_t) by solving the following least squares problem

$$\min_{A,B\in\mathbb{R}} \sum_{i=1}^{N} [\pi_a(\mathbf{p}_i - A z_{i,t})]^2 + [\pi_b(\mathbf{p}_i - B z_{i,t})]^2, \tag{8}$$

where $\mathbf{p}_i$ is the raytraced endpoint and π_a, π_b represent the orthogonal projections on a reference frame parallel to $\mathrm{R}_{\mathscr{S}}$ and centered in the robot position. The indexes a and b refer to the directions of the related scales. Such least-square problem, can be easily decoupled in two subproblems related to each

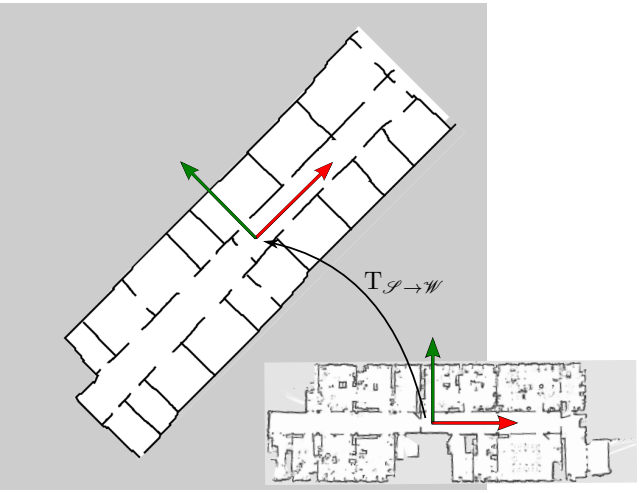

Fig. 2. The scaling directions computed using the method described in Sec. III-B to a sketch of Building 079. The rotation angle between the world's frame $R_{\mathscr{W}}$ aligned with the pixels coordinates and the detected frame $R_{\mathscr{S}}$ is $\phi \approx 0.777634$ rad ≈ 44.555 deg.

scale, then the solution can be obtained by projecting the beam and raytraced endpoints on that reference system and computing the mean of the ratios of the raytraced value and the real sensor measurement along the a,b-directions. We add this further term in the likelihood model in order to reduce the effect of "looking through walls" typical of likelihood fields models. Indeed, in our case, the presence of scaling factor can increase the chance of matching wrong obstacles, such as walls behind the actual walls detected. In other terms, the state of the robot is underdetermined in the sense that, given a pose, multiple configurations of the scales could be equally likely.

Finally, we observe that the choice of using a Laplace distribution for the scales is motivated by the fact that raytracing on the sketch is generally unstable due to the imprecision of the drawing. Therefore, we need a kernel function that does not excessively suppress outliers

Roewekaemper *et al.* [17] have shown that the likelihood field models are robust even when unmapped obstacles and dynamical objects interact with the robot, which is the natural scenario we envision for our work.

B. Choosing the Direction for Scaling

To select a suitable reference frame so that (2) and (3) can be satisfied, we apply the heuristic that walls in indoor environments are mainly orthogonal. Accordingly, it is natural to assume a world's reference frame $R_{\mathscr{W}}$ aligned to the pixel coordinates and to choose $R_{\mathscr{S}}$ so that one of the axes is parallel to the most frequent direction of the walls sketched in the map. This procedure can be easily performed by detecting the line segments in the sketch and applying a clustering procedure to the set of directions. Similarly to [18], in our work we applied the Progressive Probabilistic Hough Transform [19] after having preprocessed the sketch with Canny's algorithm for edge detection [20]. We obtain a set of segments whose slopes with respect to the pixels' coordinate system are $\{\theta_i\}_{i=1}^{K} =: \Theta \subset \mathbb{R}$. We further run k-means on Θ. The rotation angle of $T^{rot}_{\mathscr{S}\to\mathscr{W}}$ is then defined as the centroid of the biggest cluster. Please note that, in general, this procedure does not compute the actual rotation angle ϕ in (2) since the direction of the lines are determined up to a rotation of π. As a consequence, ϕ is matched up to a rotation of $k\pi$ ($k \in \mathbb{Z}$). However, this does not affect our model as the scaling acts along the direction of the axes. Indeed, this only results in a flipped behavior of the scaling factors.

This procedure captures the qualitative direction of the walls as the hand-drawn walls are affected by local bending which are therefore filtered. Indeed, the Hough Transform detects every wall as superimposition of shorter consecutive lines. Fig. 2 shows an example sketch map with the computed directions.

C. Path planning in sketches

The sketch carries only few information about the real displacement of obstacles in the vicinity of the robot. Consequently, we need to merge the high level knowledge provided by the sketch with the actual configuration of the environment, which can be retrieved by the sensor data. To do so, we incorporate two planners based on Dijsktra's shortest path algorithm, namely:

- A *global connectivity planner* that computes trajectories solely on the sketch map.
- A *metric local planner* that plans a local collision free path only with respect to the scaled readings $(z'_{i,t})_{i=1}^{N}$ and the current robot pose.

Whereas *connectivity planner* computes trajectories that are consistent with the connectivity properties of the sketched map, that is, captures a high level description of the topology of the real world, the local planner enforces the robot to avoid unmapped obstacles as well as to correct local inconsistencies between the real world and the sketch such as misplaced doors or walls.

To coordinate the planners, given a time stamp t during the navigation, we define a local window around the current robot position $W_\rho(\mathbf{x}_t^{\mathscr{S}}) \subset O_{\mathscr{S}}$ to be the circle with radius ρ centered in $\mathbf{x}_t^{\mathscr{S}}$. The local window will be the local area where the scaled scans are account for planning. Hence, we first compute a global path $\{\mathbf{x}_i^{\mathscr{S}}\}_{i=1}^{N}$ from $\mathbf{x}_t^{\mathscr{S}}$ to a target pose, by means of the Dijkstra algorithm on the sketch map $O_{\mathscr{S}}$. Then, we select a local target pose $\mathbf{x}_*^{\mathscr{S}}$ defined as the farthest way point of the global path that lies in the local window $W_\rho(\mathbf{x}_t^{\mathscr{S}})$, formally:

$$\mathbf{x}_*^{\mathscr{S}} := \arg\max \left\{ \left\| \mathbf{x} - \mathbf{x}_t^{\mathscr{S}} \right\|_{\mathbb{R}^2} \middle| \, \mathbf{x} \in W_\rho(\mathbf{x}_t^{\mathscr{S}}) \cap \{\mathbf{x}_i^{\mathscr{S}}\}_{i=1}^{N} \right\}.$$

Such a way point is in general not uniquely defined. Accordingly, we pick the first way point with respect to the order defined by the indexing. Please note that we use the standard distance on $\mathbb{R}^2$ although the natural choice would have been the local metric of the sketch $\|\cdot\|_{S_t}$. The reason is that points are planned uniformly as the global scaling operator S is not known, as a consequence a local contraction and dilatation does not result in finer or coarser set of way points. Using the local metric $\|\cdot\|_{S_t}$ the set of way points in the local window could reduce to the solely current robot position.

Finally, we compute the local path from $\mathbf{x}_t^{\mathscr{S}}$ to $\mathbf{x}_*^{\mathscr{S}}$ with respect to the scaled sensors, that is, on the local grid map

defined by the scaled scans. This path is the actual trajectory tracked by the robot.

IV. EXPERIMENTAL EVALUATION

To assess the performance of our navigation system we performed a set of sequential goal-reaching tasks in two environments, namely the Building 079 at the University of Freiburg and a scenario created in our laboratory. All the sketches provide only information about the static obstacles (walls) since we believe that a real application scenario should not overburden the user by requiring information other than the high level description.

We employed a Festo Robotino omnidirectional robot provided with a Hokuyo URG-04LX laser rangefinder (see Fig. 1). To control the robot we used the differential drive controller provided by the manufacturer.

We assumed the initial pose of the robot to be known and provided by a user, while the initial scales are estimated from the sketch applying the least-square approximation described in Sec. III-A2. The initial set of particles was thereby sampled according to the state model in (6). In all the experiments, the initial estimation of the robot pose was provided by the user only at the beginning of the first navigation task. Every time the robot lost track of its current pose a new guess was provided and the task was marked as a failure.

Since metrical information are no longer reliable on the sketch, we are interested in evaluating the navigation capability from a qualitative perspective. We instruct the robot to reach positions in the environment, such as a room or a enclosed space, and we considered a task successful if, given a starting pose, the robot is able to access them. Furthermore, we required the executed trajectories to be topologically consistent with the sketch, up to minor inconsistencies in the drawing.

A. Experimental results

For the experiments in the real building we took four sketches of Building 079 drawn by different students. We select a sequence of rooms and command the robot to enter them. Approximate starting and goal positions are annotated in Fig. 3.

As shown in Table I, the performance of the proposed method are mainly affected by the quality of the sketch. Ranging from almost 87% for the best trail, to 26% in the worst case. In the table we reported the approximate length of the paths in the real world for each navigation task, computed using a metrically consistent map.

We identified two main causes for the failures. On the one hand, the limited field of view of the range sensors is reducing the effectiveness of the filter in tracking the scales. Such a phenomenon is apparent, for instance, whenever the robot is navigating along a corridor with only lateral walls observable. This results in a wrong estimation of the scale along the direction of the corridor. This is evident in sketches *A* and *D*, which result in a different successful rate. Similarly, occlusions due to clutter and unmapped obstacles can result in wrong estimation of the scales as the filter can update the scale so that clutter is matched with the obstacles in the sketch. On the other hand, local magnitude of the scales could be significantly

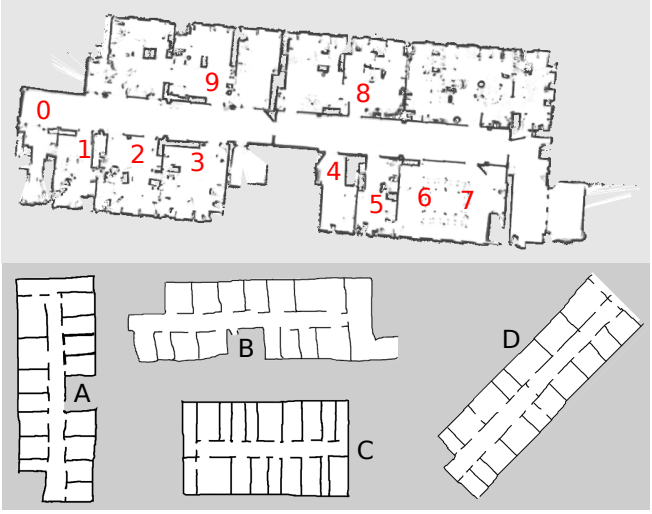

Fig. 3. The sketches employed in the experiments (bottom) and the occupancy grid (top) of Building 079, computed using the CARMEN framework [21]. Rooms 6 and 7 are now split, as shown in the sketches. In red we annotated the starting and goal positions.

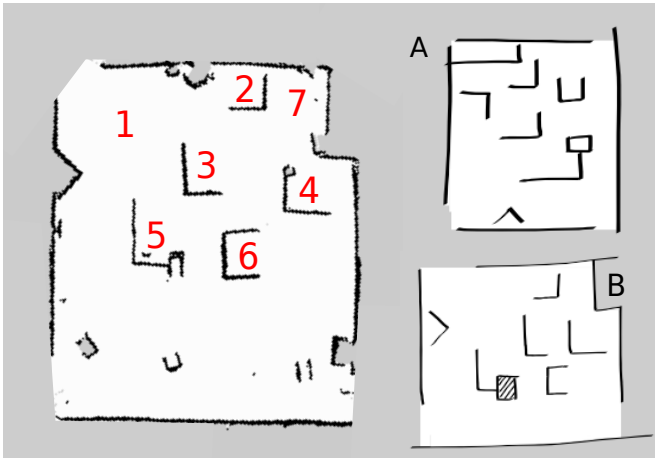

Fig. 4. Sketches used for the experiments in the scenario created in our laboratory. A metrical map based on SLAM is reported on the left. In red we annotated the starting and goal positions.

varying within the area perceived by the rangefinder so that the robot has a false perception of the obstacles in the scene.

To confirm our analysis, we ran the same experiments in the scenario reported in Fig. 4 where a part of the area is discontinuously down-scaled (bottom region) with respect of the other half (top region). We observed that in all experiments the failures occurred when the robot tried to access region 6. Indeed, approaching that region, the gross inconsistency between the sensor measurements related to the two sides of the area results in a wrong localization. Results of the experiments are reported in Table II.

V. CONCLUSION AND DISCUSSION

In this work we proposed a robot navigation system that solely relies on a rough sketch, hand-drawn by a user. The

TABLE I. EXPERIMENTS IN BUILDING 079 AT THE UNIVERSITY OF FREIBURG.

		Success rate (success/attempts)			
Path	Length (apprx.)	A	B	C	D
$0 \rightarrow 1$	$6.4m$	10/10	10/10	8/10	10/10
$1 \rightarrow 2$	$9.7m$	10/10	10/10	7/10	8/10
$2 \rightarrow 3$	$9.5m$	9/10	10/10	6/10	0/10
$3 \rightarrow 9$	$7.1m$	9/10	0/10	9/10	0/10
$9 \rightarrow 4$	$13.6m$	6/10	0/10	2/10	0/10
$4 \rightarrow 5$	$8.5m$	8/10	0/10	6/10	0/10
$5 \rightarrow 8$	$7.9m$	7/10	9/10	2/10	1/10
$8 \rightarrow 6$	$9.6m$	10/10	10/10	1/10	0/10
$6 \rightarrow 7$	$8.2m$	10/10	0/10	9/10	5/10
Average	$8.94m$	87.7%	54.5%	55.5%	26.6%

TABLE II. EXPERIMENTS IN THE LABORATORY SCENARIO.

		Success rate (success/attempts)	
Path	Length (apprx.)	A	B
$1 \rightarrow 2$	$3.8m$	9/10	8/10
$1 \rightarrow 4$	$6.7m$	9/10	10/10
$2 \rightarrow 5$	$5.4m$	10/10	9/10
$2 \rightarrow 6$	$6.0m$	5/10	5/10
$3 \rightarrow 6$	$4.0m$	4/10	5/10
$4 \rightarrow 5$	$6.3m$	9/10	10/10
$7 \rightarrow 5$	$6.2m$	9/10	10/10
Average	$5.48m$	78.5%	81.4%

problem is motivated by the need of investigating intuitive means to provide prior information of the environment, avoiding time-consuming operations such as algorithmic exploration or teleoperation. In our approach, we used an extension of Monte Carlo Localization algorithm to track the robot pose together with two scale factors that approximate the discrepancy between the sketch map and the real world. For this, we construct a theoretical framework where the sketch map is regarded as a two-dimensional metric manifold. The local metric can be thereby used as a conversion between the sketch map and the real world as well as the robot's sensors. In addition, we proposed a trajectory planner on the sketch map that couples a rough global planner that accounts for the topological properties of the map and a local planner that performs local obstacle avoidance according to the transformed sensor readings. We evaluated our approach by performing autonomous navigation tasks in two real world scenarios. Despite some failures in the experiments, mainly due to gross inconsistencies of the sketch map, we show that the robot is able to perform autonomous navigation tasks up to a success rate of 87% in a real cluttered environment.

ACKNOWLEDGEMENTS

The authors are thankful to Pratik Agarwal, for the helpful discussions and ideas. This work has partly been supported by the European Commission under contract numbers FP7–610532–SQUIRREL, FP7–610603–EUROPA2, Horizon 2020–645403–RobDREAM.

REFERENCES

[1] B. Behzadian, P. Agarwal, W. Burgard, and G. D. Tipaldi, "Monte Carlo localization in hand-drawn maps," 2015, arXiv:1504.00522v1.

[2] S. Koenig and R. G. Simmons, "Passive distance learning for robot navigation," in *ICML*, 1996, pp. 266–274.

[3] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3041–3047.

[4] V. Setalaphruk, A. Ueno, I. Kume, Y. Kono, and M. Kidode, "Robot navigation in corridor environments using a sketch floor map," in *In Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, vol. 2. IEEE, 2003, pp. 552–557.

[5] M. Skubic, P. Matsakis, B. Forrester, and G. Chronis, "Extracting navigation states from a hand-drawn map," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).*, vol. 1. IEEE, 2001, pp. 259–264.

[6] G. Chronis and M. Skubic, "Sketch-based navigation for mobile robots," in *Proc. of the The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03*, vol. 1. IEEE, 2003, pp. 284–289.

[7] M. Skubic, C. Bailey, and G. Chronis, "A Sketch interface for mobile robots," in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 1. IEEE, 2003, pp. 919–924.

[8] M. Skubic, S. Blisard, C. Bailey, J. A. Adams, and P. Matsakis, "Qualitative analysis of sketched route maps: Translating a sketch into linguistic descriptions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 2, pp. 1275–1282, 2004.

[9] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a hand-drawn sketch to control a team of robots," *Autonomous Robots*, vol. 22, pp. 399–410, 2007.

[10] D. C. Shah and M. E. Campbell, "A robust qualitative planner for mobile robot navigation using human-provided maps," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2580–2585.

[11] K. Kawamura, A. Koku, D. M. Wilkes, R. A. Peters, and A. Sekmen, "Toward egocentric navigation," *International Journal of Robotics and Automation*, vol. 17, no. 4, pp. 135–145, 2002.

[12] J. Yun and J. Miura, "A quantitative measure for the navigability of a mobile robot using rough maps," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2008, pp. 3458–3464.

[13] G. Parekh, M. Skubic, O. Sjahputera, and J. M. Keller, "Scene matching between a map and a hand drawn sketch using spatial relations," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 4007–4012.

[14] X. Li, X. Zhang, B. Zhu, and X. Dai, "A visual navigation method of mobile robot using a sketched semantic map," *International Journal of Advanced Robotic Systems*, vol. 9, no. 138, 2012.

[15] K. Matsuo and J. Miura, "Outdoor visual localization with a hand-drawn line drawing map using fastslam with pso-based mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 202–207.

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

[17] J. Roewekaemper, C. Sprunk, G. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Villamoura, Protugal, 2012, pp. 3158–3164. [Online]. Available: http://ais.informatik.uni-freiburg.de/publications/papers/roewekaemper12iros.pdf

[18] S. Thrun and D. Dolgov, "Detection of principal directions in unknown environments for autonomous navigation," 2009, pp. 73–79.

[19] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the Progressive Probabilistic Hough Transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[20] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[21] C. Stachniss. [Online]. Available: http://www2.informatik.uni-freiburg.de/ stachnis/datasets/datasets/fr079/fr079-complete.gfs.png