The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Temporary Maps for Robust Localization in Semi-static Environments

Daniel Meyer-Delius Jürgen Hess Giorgio Grisetti Wolfram Burgard

***Abstract*— Accurate and robust localization is essential for the successful navigation of autonomous mobile robots. The majority of existing localization approaches, however, is based on the assumption that the environment is static which does not hold for most practical application domains. In this paper, we present a localization framework that can robustly track a robot's pose even in non-static environments. Our approach keeps track of the observations caused by unexpected objects in the environment using temporary local maps. It relies both on these temporary local maps and on a reference map of the environment for estimating the pose of the robot. Experimental results demonstrate that by exploiting the observations caused by unexpected objects our approach outperforms standard localization methods for static environments.**

I. INTRODUCTION

Robot localization consists of estimating the pose of the robot relative to a given map of the environment. The majority of existing approaches, however, assume the environment to be static. A common technique for dealing with non-static environments is to simply ignore measurements that are not explained by the reference map. This assumes that objects are either static and represented in the map or dynamic and should be ignored for localization. Whereas this technique has been demonstrated to be robust in highly dynamic environments, it ignores valuable localization information when the changes in the environment occur slowly.

In this paper, we present a localization framework that exploits the measurements caused by semi-static objects to build local maps that temporarily extend the static map of the environment. Using these temporary maps allows the robot to reliably estimate its pose also in regions that are subject to persistent changes. The motivation behind our approach is that many objects change their locations with a relatively low frequency and therefore provide important localization information. We will refer to these objects as being semi-static. For example, consider a parking lot as the one depicted in Figure 1. In such an environment there are only a few static objects. Additionally, the parked cars occlude these objects most of the time. As a result almost no features remain that can be used for localization by standard approaches. The parked cars, however, provide on their own important features for localization. Taking advantage of the measurements caused by semi-static objects can improve the localization capabilities of a mobile robot.

In many practical mobile robot applications, a reference map representing the static parts of the environment is available beforehand. However, most environments are not static

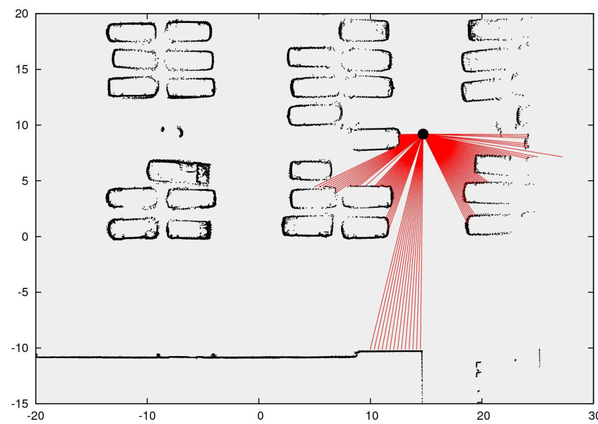

Fig. 1. In large open spaces like a parking lot, the semi-static objects (parked cars) provide an abundant source of important features for localization. The static objects (walls) are not only few but are also occluded by the non-static objects.

and mobile robots must be able to deal with changes in the environment. Our approach is an extension of the standard particle filter localization approach for static environments. We assume that a reference map representing the static objects in the environment is given. When the observations of the robot are consistent with this map, our approach corresponds exactly to the standard particle filter approach. However, we also use temporary maps to keep track of the inconsistent observations caused by semi-static objects. Whenever the robot enters an area for which a temporary map already exists we try to use this map to improve the localization. Taking advantage of the measurements caused by semi-static objects is particularly important in large open spaces like the parking lot in Figure 1 or warehouses, where the static parts of the environment are few and usually occluded, but many semi-static objects provide valuable localization information.

The contribution of this paper is a localization approach capable of dealing with semi-static environments. We provide a probabilistic formulation of the localization problem where the semi-static aspects of the environment are explicitly modeled. Our algorithm keeps track of the observations caused by the semi-static objects in the environment in the form of local maps that temporarily extend the static map of the environment. At its core, our localization framework relies on a particle filter for estimating the pose of the robot using the extended static map. Experimental results demonstrate that by exploiting the observations caused by semi-static objects our approach is capable of robustly and accurately estimating the pose of the robot where standard approaches fail.

All authors are with the University of Freiburg, Dept. of Computer Science, D-79110 Freiburg, Germany.
{meyerdel,hess,grisetti,burgard}@informatik.uni-freiburg.de

II. RELATED WORK

In the past, several authors have studied the problem of mobile robot localization in non-static environments. Fox *et al.* [1], for example, use an entropy gain filter to identify the measurements caused by dynamic objects. Burgard *et al.* [2] additionally use a distance filter which selects individual measurements based on the difference between their measured and their expected distance. Montemerlo *et al.* [3] propose a method for tracking people while simultaneously localizing the robot which increases the robustness of the robot pose estimation.

The problem of dealing with dynamic objects has also been investigated in the field of simultaneous localization and mapping (SLAM). Wang and Thorpe [4] employ a feature-based heuristic to identify dynamic objects in range measurements and use the filtered result for localizing the robot and building a map at the same time. Hähnel *et al.* [5] use a probabilistic method for tracking people and filter out the corresponding measurements to improve the map building process. Although these filtering approaches have proven to be robust in highly dynamic environments, they discard valuable localization information when the changes in the environment occur with a relatively low frequency.

Stachniss and Burgard [6] approach this problem by estimating typical configurations of dynamic areas in the environment. They show that the integration of this information into a particle filter framework improves the robot pose estimation. Anguelov *et al.* [7] deal with non-stationary objects representing them using learned geometric models. They apply a hierarchical EM algorithm based on occupancy grid mapping to learn a shape model of the objects and ultimately use this information to correct the mapping process. Andrade-Cetto and Sanfeliu [8] describe an approach where landmarks are introduced and removed depending on how often they had been observed. Biber and Duckett [9] propose a spatio-temporal map where the environment is represented at multiple time-scales simultaneously. In contrast to mapping approaches, we do not aim at generating a consistent representation of the environment. The local maps constructed by our approach are only temporary and are discarded as soon as inconsistencies are detected.

The idea of using sets of local maps as representation of the environment has been proposed by many authors in the past. Estrada *et al.* [10], for example, build a graph of local stochastically independent maps and correct the position of the local maps whenever the robot enters an area already visited. Williams *et al.* [11] create a local submap of the features around the robot and fuse the local maps regularly with a global map. Gutmann and Konolige [12] construct locally consistent maps and use them to determine when the robot is entering an area already visited in the global map. All these approaches assume the environment to be static and use the gathered information for constructing globally consistent maps. In contrast, our goal is to improve the localization capabilities of the robot by constructing local maps that temporarily extend the reference map of the environment.

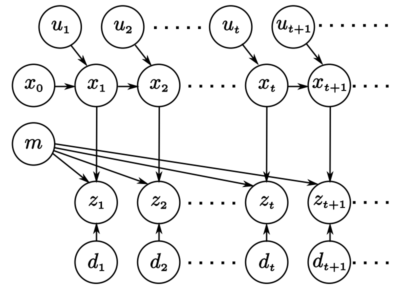

Fig. 2. Graphical model of the mobile robot localization process in semi-static environments. The observations z_t at time step t are explained by both the static map m and the corresponding semi-static map d_t.

III. LOCALIZATION IN SEMI-STATIC ENVIRONMENTS

Mobile robot localization consists of estimating the probability density $p(x_t \mid z_{1:t}, u_{1:t}, m)$ of the pose x_t of the robot in a known map m given a sequence of observations of the environment $z_{1:t} = \{z_1, \ldots, z_t\}$ and odometry measurements $u_{1:t} = \{u_1, \ldots, u_t\}$. Most existing solutions to the localization problem assume that the environment is static. In our work, we classify the objects in the environment according to their dynamics into three different classes:

- **Static objects:** like buildings, that do not change their location.
- **Semi-static objects:** like parked cars, that change their locations with a relatively low frequency. In particular, we assume that these objects do not change their location while the robot is observing them.
- **Dynamic objects:** like moving people, that frequently change their location. Unlike semi-static objects, dynamic objects do change their location while being observed by the robot.

We assume that dynamic objects are detected and filtered out. Accordingly, the map m represents the static objects in the environment whereas the map d_t represents the semi-static objects at time t. Figure 2 depicts the dynamic Bayesian network describing the localization process in a semi-static environment. The main difference to standard localization approaches in static environments is that we explicitly model the fact that the observation z_t at time step t is explained by both, the static map m and the semi-static map d_t. Each observation z_t is divided into two independent parts: z_t^d caused by semi-static objects and z_t^m caused by the static objects. Additionally, since semi-static objects change their location with a relatively low frequency, observations obtained at different time steps can be explained by a single semi-static map as illustrated in Figure 3. In principle, the semi-static map d_t at time t depends on the semi-static map d_{t-1} at the previous point in time. In our work, however, we do not reason about the temporal dependency between semi-static maps. For simplicity we assume that a semi-static map either consistently explains the measurements z_t^d at time t or it is not valid anymore and needs to be re-estimated. This corresponds to a uniform distribution about the evolution of the semi-static maps.

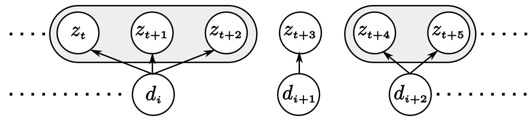

Fig. 3. Since semi-static objects change their locations with a relatively low frequency, one semi-static map d_i can explain observations at different time steps.

According to our formulation, robot localization in semi-static environments requires to jointly estimate the probability distribution $p(x_t, d_t \mid z_{1:t}, u_{1:t}, m)$ over robot locations and semi-static maps. To this end one could use one of the many SLAM algorithms available and initialize it with the known static map. However, the majority of SLAM approaches is based on the assumption that the environment is static and the presence of non-static objects can lead to serious errors in the resulting maps. Instead of estimating the full distribution over the potential semi-static maps d_t, we rely on a local maximum likelihood estimate d_t^* considering the trajectory of the robot while it moves in non-static areas. In the following section we present our approach for robot localization in semi-static environments in more detail.

IV. TEMPORARY MAPS

The key idea of our approach is to use the measurements caused by semi-static objects to improve the localization capabilities of the robot. With these measurements, we build local semi-static maps that temporarily extend the static map of the environment. Due to the temporal nature of these maps we refer to them throughout this paper both as semi-static or temporary maps.

A semi-static map represents the semi-static objects in the environment as observed by the robot while navigating through it. As illustrated in Figure 3, a semi-static map d_i is associated to a sequence of measurements $z_{n:m}$ with their corresponding poses $x_{n:m}$. Semi-static maps are created as the robot navigates through the environment using a maximum-likelihood approach.

When the observations obtained by the robot are consistent with the static map of the environment, we use this map as reference to estimate the pose of the robot. However, when the observations are inconsistent, we select the local map closest to the current pose of the robot. If a map is found, we try to use it as reference for localization instead of the static map. In the remainder of this section we describe the different components of our localization framework.

A. Particle Filter Localization

For estimating the pose of the robot our localization framework relies on a particle filter that uses the extended static map of the environment. In the context of robot localization, particle filters are used to estimate the posterior distribution $p(x_t \mid z_{1:t}, u_{1:t}, m)$ of the robot's pose x_t at time t conditioned on the observations $z_{1:t}$, odometry measurements $u_{1:t}$, and map of the environment m. The key idea of particle filters is to represent the posterior by a set of weighted samples or particles, where each particle corresponds to a potential pose of the robot. The particle filter algorithm computes the particle set at time t recursively from the particle set at time $t-1$ using the most recent observation z_t and odometry measurement u_t. The resulting set of particles represents a sample-based approximation of the continuous posterior distribution. To extract a continuous distribution out of the particle set, we use a Gaussian approximation of the weighted particle set.

In the sampling step of the algorithm we use a probabilistic model $p(x_t \mid x_{t-1}, u_t)$ of the robot's motion as proposal distribution. This motion model describes a posterior density over possible poses x_t given the previous pose x_{t-1} and most recent odometry measurement u_t. The weights of the particles are computed as $w_t = p(z_t \mid x_t, m) \cdot w_{t-1}$. The first term $p(z_t \mid x_t, m)$ corresponds to the observation model that represents the likelihood of the most recent observation z_t given the map of the environment m and the pose x_t. The second term w_{t-1} corresponds to the weight of the particle at the previous time step.

B. Perceptual Model

We assume that the observations are obtained from a range scanner and that each observation z_t consists of a set of range measurements. To evaluate the likelihood $p(z_t \mid x_t, m)$ of an observation z_t given the pose x_t of the robot and a reference map m, we use the likelihood fields model [13]. In this model, the individual range measurements of the observation z_t are assumed to be independent of each other and the likelihood of each one is computed based on the distance between the endpoint of the range measurement and its closest obstacle in the map m.

In our current implementation, we use this distance as a heuristic to decide whether a measurement is explained by the map m or not. Concretely, if the distance to the closest object in the map is larger than a given threshold, we assume that the measurement is inconsistent with the map and consider it an outlier. The outlier ratio for a given observation z_t corresponds to the fraction of range measurements that do not correspond to their expected values according to the map. Throughout this paper we use the weighted average outlier ratio of the particle set given a map as criterion of how well a given observation is explained by the map.

C. Constructing Temporary Maps

A temporary or semi-static map consists of a sequence of measurements $z_{n:m}$ with their associated poses $x_{n:m}$ that implicitly represent a part of the environment. To construct such a map we incrementally perform scan-matching on consecutive observations. The idea is to compute, for each new observation, the pose that best aligns the observation with respect to a reference map. This map is then extended by adding the aligned observation together with its corresponding pose.

The concrete scan matching technique used in our work is similar to the correlative scan matching approach proposed by Olson [14]. The idea is to evaluate the observation likelihood $p(z_t \mid x_t, m)$ using a previous scan z_{t-1} as reference map m over a discrete three-dimensional volume of potential poses x_t. The maximum-likelihood pose corresponds to the best alignment between the two scans. Whereas in Olson's approach only a single scan z_{t-1} is used as reference, we use a history of previously aligned scans $z_{t-k:t-1}$. In our experiments, we found that using a history of scans instead of a single one increases the accuracy and robustness of the scan matching technique.

One drawback of incrementally constructing maps using scan-matching is that the pose estimates are never corrected. To overcome this problem, we adjust the poses of the temporary map whenever the robot enters a known area in the static map or an area for which a temporary map already exists. This problem corresponds to an instance of the graph-based SLAM [15] problem, where the poses of the robot correspond to nodes in a graph. An edge between two nodes represents the relative movement between the corresponding poses as estimated by the scan matcher. In addition to these spatial constraints between consecutive poses we also consider the global constraint that results from the robot entering a known area. This global constraint corresponds to the relative movement of the robot between the first and last pose in the semi-static map, being the last pose the point where the robot reentered a known area.

To efficiently compute the maximum-likelihood semi-static map we use the approach described by Grisetti *et al.* [15]. Note that the optimization is only performed when reentering a know area. Furthermore, in contrast to the pure SLAM problem, we do not adjust the nodes in the graph that correspond to the robot being in a known area in the static map of the environment.

D. Extending the Static Map

In our work, we assume that a static map of the environment is given. When the observations of the robot are inconsistent with the static map, we select the semi-static map closest to the current pose of the robot and try to use it as reference for localization instead of the static map. Note that our approach is an extension of the standard particle filter localization for static environments. Whenever the observations of the robot are consistent with the static map of the environment, our approach corresponds exactly to the standard particle filter approach.

Semi-static maps are created whenever the following two conditions hold. First, the observations of the robot are inconsistent with the static map of the environment. And second, there is no other semi-static map close to the current pose of the robot that explains the observations and can be used as reference for localization instead of the static map. Whenever these two conditions hold over multiple consecutive time steps a new semi-static map is created as described in the previous section. To determine if an observation is inconsistent with a map or not we use the average outlier ratio of the particle set as described in Section IV-B.

We assume that a semi-static map either consistently explains the observations at a given time or it is not valid anymore. Accordingly, semi-static maps are discarded if the average outlier ratio of the particle set is too high over multiple consecutive time steps. Ideally, semi-static maps should only be eliminated if the environment has changed since the moment of its creation. To reduce the problem of incorrectly eliminating a map, the uncertainty in the pose estimate is also taken into account. Maps are discarded if the uncertainty of the particle set is below a given threshold.

E. Localization Using Temporary Maps

Whenever the static-map of the environment does not explain the observations of the robot, we search for a semi-static map near the current pose of the robot to use as reference for localization instead of the static map.

To choose the nearest semi-static map, we use the Mahalanobis distance as proximity measure between the pose of the robot and the poses in the local maps. In this way we can take the uncertainty in the pose estimate into account when selecting an adequate map. We use a kd-tree to store the poses of the local maps to make the search efficient.

Using temporary maps for localization, the weighs of the particles are computed as $w_t = p(z_t \mid x_t, m, d_t) \cdot w_{t-1}$, where d_t corresponds to the nearest semi-static map. The observation likelihood $p(z_t \mid x_t, m, d_t)$ is computed as

$$p(z_t \mid x_t, m, d_t) = p(z_t \mid x_t, m)^{I(z_t,m)} \cdot p(z_t \mid x_t, d_t)^{I(z_t,d_t)} . \quad (1)$$

In the equation above, $I(z_t, m')$ is an indicator function defined as

$$I(z_t, m') = \begin{cases} 1 & \text{if } e(z_t, m') < \epsilon \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $e(z_t, m')$ is the average outlier ratio for observation z_t and map m', and ϵ represents the threshold at which the observation z_t is considered inconsistent with the map m'. Semi-static maps are only created in areas where the observations are inconsistent with the static map of the environment. Thus, (1) states that the weights of the particles are computed according to either the static map of the environment m or the closest semi-static map d_t, provided that d_t is consistent with z_t. Note that when no map is consistent with the observations, all particles will be assigned the same weight, and the particle set will evolve exclusively according to the motion model of the robot.

Since our approach is based on a particle filter, the complexity of the algorithm depends mostly on the number of particles used. The construction, including optimization, of a semi-static map is approximately lineal in the number of poses in the map. Adding semi-static maps to the kd-tree and searching for the closest semi-static map is logarithmic in the number of poses of all semi-static maps. However, constructing and adding semi-static maps to the kd-tree is not a frequent operation and searching for the closest semi-static maps takes place only when the observations of the robot are inconsistent with the static map of the environment. More importantly, none of these operations depend on the number of particles. As a result, the overall complexity of the algorithm depends linearly on the number of particles.

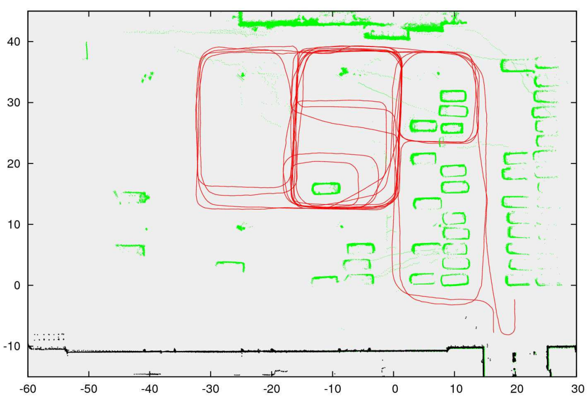

Fig. 4. Trajectory of the robot obtained during the experiments with our localization approach. The dark colored structures at the bottom correspond to the static parts of the environment used as reference map. Note that by limiting the line-of-sight of the robot to 20 meters the reference map could only be observed sporadically.

To summarize our approach, we assume that a static map of the environment is given. When the observations of the robot are inconsistent with this map we try to find a semi-static map to use as reference for localization instead. The semi-static map is selected based on the distance to the current pose of the robot. If a map is found, we try to localize the robot using the temporary map instead of the reference map of the environment. Temporary maps are eliminated if they are inconsistent with the robot's observation.

V. EXPERIMENTAL EVALUATION

We implemented and tested our approach using real data gathered with a MobileRobots Powerbot equipped with a SICK LMS laser range finder. The experiments show that by exploiting the observations caused by semi-static objects our method can robustly and accurately estimate the pose of the robot where standard approaches fail.

A. Localization in Large Open Spaces

To evaluate the robustness and accuracy of our approach we steered the robot through a parking lot and created a map containing only the static elements of the environment. Although there exists approaches for generating static maps in dynamic environments, this was not the aim of our work so we manually removed the dynamic elements. Furthermore, we removed some areas of the static map to better evaluate the behavior of our approach. Figure 4 shows the part of the environment that was used as static map for this experiment.

We evaluated our approach in the task of position tracking and did not consider the problem of global localization. Since our framework is an extension of a particle filter, global localization is possible as long as enough features of the reference map are observed. Figure 4 plots the average trajectory of the robot obtained using our localization approach over 10 repetitions of the experiment. In every repetition, 30 particles where used and the maximum range of the laser beams was set to 20 meters. In this way, we reduced the number of observations caused by the reference map. As pose estimate,

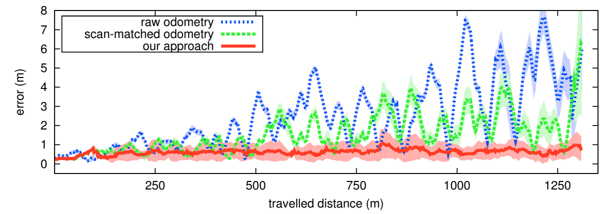

Fig. 5. Average error and standard deviation of the estimated pose obtained using our approach. We also compared the results against a standard particle filter using the raw odometry of the robot in one case and using an improved odometry based on scan-matching in the other.

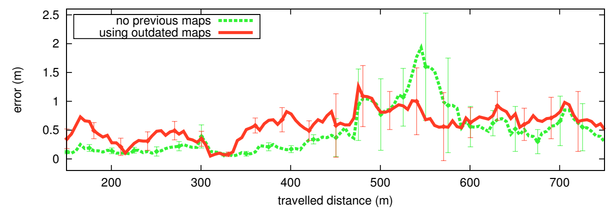

Fig. 6. Average localization error obtained when using outdated semi-static maps compared against the error when no maps were given beforehand.

we used the weighted mean of the particle set. The ground truth map, also shown in Figure 4, was computed using a static SLAM approach [15] considering the full 80 meter depth range of the laser scanner. Note that the reference map was only observed during short time intervals at the beginning and the end of the trajectory. Despite of this, the pose of the robot could be accurately estimated during the whole experiment.

To quantitatively measure the accuracy of our approach, Figure 5 plots the average error and standard deviation between the estimated poses and the ground truth. We also compared our results against a standard particle filter using the raw odometry of the robot in one case and using an improved odometry based on scan-matching in the other. As can be seen from the figure, our approach only produces a small and relatively constant error along the entire trajectory. The standard particle filter, in contrast, results in a substantially larger error, even when the robot utilized the improved odometry. Thus, the utilization of observations caused by semi-static objects substantially increases the localization capabilities of the robot.

B. Localization in Non-Static Environments

The goal of this second experiment is to evaluate how our approach handles large changes in the environment. We collected data on two different days on our parking lot so that the configuration of the parked cars would be considerably different. We ran our algorithm on the data of the first day and used the obtained temporary maps as the initial extended map for the data of the second day. The objective of the experiment was to analyze the effect of inconsistent temporary maps in the accuracy and robustness of the localization.

Figure 6 plots the average error and standard deviation of our approach when using the outdated semi-static maps compared against the error when no semi-static maps were

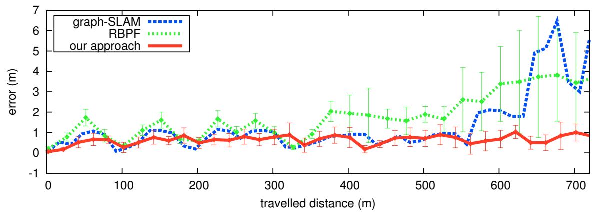

Fig. 7. Average error and standard deviation of the estimated pose obtained using our approach and two static SLAM techniques: Rao-Blackwellized Particle Filter (RBPF) and graph-based SLAM.

given beforehand. As can be seen, there are no significant differences between both errors. This shows that our approach can correctly identify when a temporary map is not valid anymore and discards it accordingly. As explained in Section IV-D, a temporary map is considered invalid if the average outlier ratio of the particle set for that map is above a given threshold. We determined this value empirically and set it to 0.8 in all our experiments. On the one hand, using smaller values sometimes caused maps to be deleted even when the environment had not changed. On the other hand, using larger values made the algorithm overconfident in the available maps which sometimes lead to less accurate results. This explains the slightly higher errors obtained when using the outdated semi-static maps in the first half of the trajectory shown in Figure 6.

C. Standard SLAM in Non-Static Environments

The goal of this experiment is to compare standard SLAM approaches with our approach. We created several artificial maps representing a parking lot in different configurations. Using a simulation environment we switched between the different maps while the robot moved to create the effect of a semi-static environment. For the comparison we considered two state-of-the-art static SLAM techniques: a Rao-Blackwellized Particle Filter (RBPF) [16] and a graph-based SLAM approach [15]. To measure the accuracy of the approaches we utilized the error metric described in [17]. The displacements from the initial pose where used to emphasize the overall geometry of the environment in the metric.

Figure 7 compares the translational error obtained using our approach and the static SLAM techniques. As the number of observations caused by non-static objects increases in the map, it becomes more difficult for the static SLAM approaches to distinguish between inconsistent observations caused by changes in the environment, sensor noise, and localization errors. In particular because changes in the environment are not explicitly considered. This is reflected by the growth in the error as the robot navigates the environment. By relying on an unmodifiable static map and discarding the semi-static maps as they become inconsistent, our approach is robust against changes in the environment as can be seen in the figure by the almost constant error.

VI. CONCLUSIONS

In this paper, we presented a localization framework that exploits the measurements caused by semi-static objects to improve the localization capabilities of a mobile robot in dynamic environments. Our approach constructs local maps using the measurements caused by semi-static objects. These maps temporarily extend the reference map of the environment and are used as fall-back map whenever the observations of the robot are inconsistent with the reference map. Our approach is an extension to standard particle filter localization that only employs a map of the static aspects of the environment to estimate the pose of the robot. We implemented our approach and tested it on data gathered with a real robot. Experimental results demonstrate that by exploiting the observations caused by semi-static objects our approach is capable of robustly and accurately estimating the pose of the robot even in situations in which state-of-the-art approaches fail.

REFERENCES

[1] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, 1999.

[2] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.

[3] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[4] C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[5] D. Hähnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[6] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[7] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun, "Learning hierarchical object maps of non-stationary environments with mobile robots," in *Proc. of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

[8] J. Andrade-Cetto and A. Sanfeliu, "Concurrent map building and localization on indoor dynamic environments," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 3, 2002.

[9] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[10] C. Estrada, J. Neira, and J. Tardós, "Hierarchical SLAM: Real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, 2005.

[11] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localisation and mapping problem," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[12] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. of the Conf. on Intelligent Robots and Applications (CIRA)*, 2000.

[13] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, 2001.

[14] E. Olson, "Real-time correlative scan matching," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[15] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2d and 3d mapping," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[16] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[17] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós, "A comparison of slam algorithms based on a graph of relations," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.