Accurate Localization with Respect to Moving Objects via Multiple-Body Registration

Jörg Röwekämper Benjamin Suger Wolfram Burgard Gian Diego Tipaldi

***Abstract*— Many mobile manipulation tasks require the robot to be accurately localized with respect to the object where the manipulation has to be executed. These tasks include autonomous docking and positioning as well as pick and place or logistics tasks. State-of-the-art approaches to the problem commonly assume that the environment is static and localize the robot with respect to predetermined locations. In this paper, we present an approach that relaxes the static assumption and enables a robot to accurately localize with respect to a reference object that could be moved in the environment. The core of the paper is an extension of the generalized ICP method to handle multiple rigid bodies that move independently to each others. Experiments in both simulated and real world scenarios show that our approach is able to localize the robot with respect to a moved object with an accuracy of less than one centimeter.**

I. Introduction

Localization is one of the most important problems in robotics and represent the enabling technology for many applications. In the last 20 years, many researcher addressed this problem and many significant contributions have been made. Probabilistic techniques have demonstrated the capability to robustly estimate the pose of robots in a large variety of application scenarios.

In many scenarios, and especially in industrial applications, highly accurate localization is a key requirement to perform tasks such as pick and place, logistics, and mobile manipulation. Typical solutions in factory floors rely on modifications of the environment such as wires embedded in the floor or magnetic tapes. Modifications of the environment, however, are costly and limit the flexibility of rearranging the factory floor. In our previous work [7], we demonstrated that accurate localization can be achieved at predetermined locations in static environments. We achieved those results without the need of artificial markers and solely relying on safety laser range finders.

In this paper, we extended the fine localization approach by relaxing the assumption of static environments and localizing the robot with respect to particular objects that can be moved around. This is a key technology for flexible factories, where the robot is required to pick parts from bins and boxes, or load pallets from a logistic area. In this circumstances, the bins, boxes, or pallets could move from one time frame to another, being moved by shop-floor workers or unloaded at slight different locations. Another case is where the shop-floor workers defined positioning tasks with respect to CAD models, where the exact location of the fixture may differ from the original plan. Fig. 1 illustrates an example scenario, where the robot is required to position itself in front of an object that could be moved around.

All authors are with the Department of Computer Science at the University of Freiburg, Germany. This work has partly been supported by the European Commission under FP7-610917-STAMINA, ERC-AG-PE7-267686-LifeNav and H2020-645403-ROBDREAM.

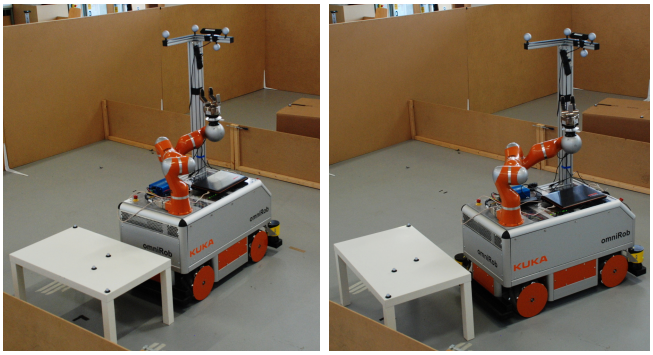

Fig. 1. Localization of the omniRob in front of moving object. The left figure illustrates the teaching phase, when we drove the robot to the reference location and build a model of the object (table). The right figure depicts the robot localizing and positioning with respect to the rotated object.

We propose an approach that solely relies on a laser range finder, a sensor that is commonly used in industrial applications for safety reasons. Instead of localizing only with respect to a pre-built grid-map of the environment or to predefined locations, we also use a sensor based representation of a reference object and estimate the robot pose with respect to the object directly. We formulate the problem in terms of sensor registration of multiple rigid bodies and provide means to compute an accurate reference model for the object as well as its relative position with respect to the current robot pose.

II. Related Work

The problem of accurate localization for industrial tasks has been addressed by several researchers. Saarinen et al. proposed to use the normal distribution transform in a Monte Carlo localization framework (NDT-MCL) [8]. The NDT-MCL uses a set of normal distributions to represent the environment in a piecewise continuous way, reducing the discretization effect of occupancy grid maps. The authors reported a positioning accuracy of $1.4\,\mathrm{cm}$ in a repeatably test and claim that the accuracy required for loading actions in industrial environment has to be $3\,\mathrm{cm}$. Röwekämper et al. showed that it is possible to achieve a localization and positioning accuracy of a few millimeters at taught-in reference locations, by leveraging Monte Carlo localization and scan alignment techniques [7]. Both works, however,

assume that the robot needs to be localized with respect to a static location in the environment. In contrast to them, our method achieves high localization accuracy even with respect to objects that can change their position over time.

Few works addressed the problem of how to determine possible docking locations from sensor data. Williamson et al. [14] present an approach based on mixtures of Gaussians and an Expectation Maximization (EM) algorithm for robot docking. Each mixture represent a possible object and the EM algorithm is used to assign points to objects. Jain and Argall [6] leverage point clouds from a Kinect sensor to detect edges of tables, bowls, or cups to find possible docking candidates. However, the focus of the work is on safety for humans on a wheelchair and the results show an accuracy too low for industrial settings.

Other authors addressed the problem of moving objects in the environment during localization and mapping. The approaches of Anguelov et al. [2] and Biswas et al. [3], for example, compute shape models of non-stationary objects. They create maps at different points in time and compare those maps using an EM-based algorithm to identify the parts of the environment that change over time. However, their focus was more on understanding which object moved and the matching accuracy is too low for the purpose of this work. Salas-Moreno et al. [9] estimated the location of objects in the context of simultaneous localization and mapping (SLAM). Although they are able to estimate the object location with high accuracy, the authors still rely on the assumption that the objects do not move in the environment. Ahmad et al. [1] extended the graph-based formulation of SLAM to also include moving objects. However, they focus on objects that move continuously in the environment and assume a known motion model. In this paper, we do not assume any knowledge about the motion and we considered changing environment, where objects are moved by the users and then keep their location fixed.

Some other authors instead focused on simultaneously performing scan matching and motion tracking from laser scanners. Tipaldi and Ramos [12] proposed CRF-Clustering, a conditional random field approach to perform motion segmentation and at the same time compute the number of objects and their displacement. Van de Ven et al. [13] extended it by integrating a data association step within the probabilistic framework. Yang et al. [15] introduced a multi-model extension of RANSAC to deal with environments with rapid changes. The same authors extended the work by proposing a multi-scale algorithm in order to be robust against segmentation errors [17] and exploit spatio-temporal information of both stationary and moving objects [16]. With respect to those work, we do not limit ourselves to just use two sequential scan measurements, but we accumulate points over time, while the robot is approaching the target, to improve the localization accuracy.

III. Accurate Localization to Moving Objects

In this paper, we address the problem of accurately localizing a robot with respect to objects that can be moved in the environment. It is important to accurately localize a mobile robot locally while still maintaining the knowledge of the robot position in a global reference frame. To achieve this, we decompose the localization problem in two main parts: global localization with respect to a map and relative localization with respect to an object. For the global localization, we employ the Monte-Carlo localization (MCL) approach proposed by [4], which recursively estimates the posterior about the robot's pose given the motion command and the sensor observations. MCL approximates the posterior distribution with a set of weighted samples, called particles. When a new odometry measurement is available, the algorithm uses the motion model to draw a new set of samples. Those samples are then weighted according to the observation model whenever a new sensor measurement is available. We compute the number of samples to draw at each prediction step according to the KLD sampling [5] algorithm.

For the relative localization, we build upon the accurate localization framework we developed [7] and extend it in order to deal with objects that can be moved around. In our previous work, we drove the robot to the location where high accuracy was needed and stored local sensor measurements as reference observations in the map. Those measurements were recorded while the robot was standing still and then used for scan matching during runtime. In this work, we augment the reference measurements with semantic labels to indicate the object of reference. Without loss of generality, we assume that the labels are assigned by the user using a graphical interface. Fig. 5 shows an example of a reference model, where the object is depicted in red and the rest of the environment in blue.

A. Generalized ICP for multiple rigid bodies

We formulate the relative localization problem in terms of multi-body sensor registration, where we seek to align multiple rigid bodies that undergo different rigid body motion. We extended the generalized iterative closest point (ICP) [10] algorithm to handle multiple rigid bodies. Given a reference point cloud $\mathcal{M}_p = \{\mathbf{p}_{p,i}\}$ and a local point cloud $\mathcal{M}_r = \{\mathbf{p}_{r,j}\}$, where $\mathbf{p}_{q,k} = (x, y)$ is the k-th point expressed in the local reference frame $\mathbf{x}_q$ of the point cloud q, generalized ICP estimates the transformation $\mathbf{T} = (\mathbf{x}_p \ominus \mathbf{x}_r)$ that best aligns the two point clouds. Here, $\ominus$ is the inverse of the compound operator $\oplus$ as introduced by Smith et al. [11]. Given an initial guess for the transformation, the algorithm proceeds iteratively to first compute correspondences between the points in the two clouds and then to minimize their reprojection error. The correspondences are computed according to their Euclidean distance after applying the transformation

$$j^\star = \underset{j}{\operatorname{argmin}} \|\mathbf{p}_{r,j} - \mathbf{T} \oplus \mathbf{p}_{p,l}\|_2. \tag{1}$$

The point correspondence $\langle \mathbf{p}_{r,j^\star}, \mathbf{p}_{p,l} \rangle$ is computed by assuming the existence of a landmark l in the environment, which is observed in both point clouds. Let the reprojection

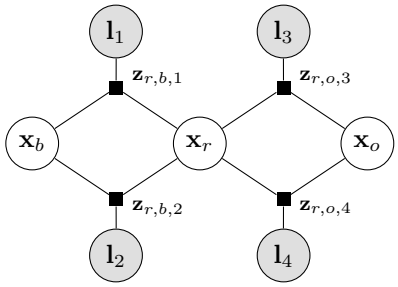

Fig. 2. Example factor graph with two rigid bodies. The positions of the landmarks $\mathbf{l}_k$ (shaded nodes) implicitly provide measurements $\mathbf{z}_{ij}$ (black squares) between the current robot pose $\mathbf{x}_r$ and the reference pose of the objects $\mathbf{x}_b$ and $\mathbf{x}_o$.

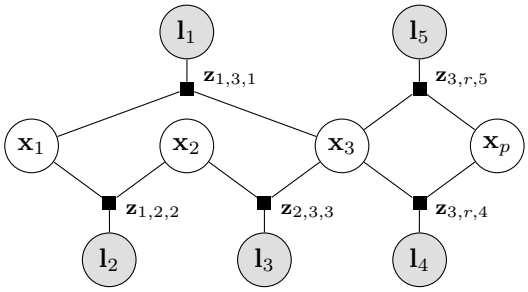

Fig. 3. Example factor graph for computing the reference point cloud. The positions of the landmarks $\mathbf{l}_k$ (shaded nodes) implicitly provide measurements $\mathbf{z}_{ij}$ (black squares) between the reference pose $\mathbf{x}_p$ and the laser scans $\mathbf{x}_i$ (white nodes).

error associated with the landmark l be

$$\mathbf{z}_{r,p,l} = \mathbf{p}_{r,l} - \mathbf{T} \oplus \mathbf{p}_{p,l}. \tag{2}$$

To simplify notation, we assume that the points are reordered according to the landmark they belong to and corresponding points have the same index l. The transformation is computed by solving the least squares problem

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_l \|\mathbf{z}_{r,p,l}\|^2_{\Omega_{r,p,l}} \tag{3}$$

where $\Omega_{r,p,l}$ is the information matrix associated with the error term $\mathbf{z}_{r,p,l}$. Generalized ICP uses the information matrix $\Omega_{r,p,l}$ to define different set of error functions, from point-to-point to point-to-plane and plane-to-plane. Since no closed form solution for the this problem exists, minimization is performed using Levenberg-Marquardt.

To handle multiple bodies, we extended the framework to estimate multiple transformations at the same time. Let consider the two point clouds $\mathcal{M}_p$ and $\mathcal{M}_r$ defined above and let assume the reference point cloud $\mathcal{M}_p$ to be partitioned in disjoint sets, each containing the measurements belonging to one of the rigid body. Let consider the augmented point cloud $\widehat{\mathcal{M}}_p = \{\mathbf{p}_{p,o,i}\}$, where $\mathbf{p}_{p,o,i} = (x, y)$ is the i-th point belonging to object o and expressed in the object reference frame $\mathbf{x}_o$. To avoid the combinatorial complexity of considering all possible transformations between the objects, we estimate the reference $\mathbf{x}_o$ of all the objects with respect to the robot reference frame $\mathbf{x}_r$. This is similar in spirit of graph-based SLAM approaches and bundle adjustment. The main difference, is that we iteratively re-estimate the point correspondences and the object labels.

Given an estimate for the object poses $\mathbf{x}_o$, our multiple-body extension begins by finding the point correspondences and the semantic labeling according to the Euclidean distance between the points of the to the o-th rigid body and the points in the second point cloud $\mathcal{M}_r = \{\mathbf{p}_{q,j}\}$, expressed in the robot reference frame $\mathbf{x}_r$

$$(j, o)^\star = \underset{j,o}{\operatorname{argmin}} \|\mathbf{p}_{r,j} - (\mathbf{x}_o \ominus \mathbf{x}_r) \oplus \mathbf{p}_{p,o,l}\|_2. \tag{4}$$

This results in the augmented point cloud $\widehat{\mathcal{M}}_r = \{\mathbf{p}_{r,o,j}\}$. For each correspondence, we define the reprojection error as

$$\mathbf{z}_{r,o,l} = \mathbf{p}_{r,o,l} - (\mathbf{x}_o \ominus \mathbf{x}_r) \oplus \mathbf{p}_{q,o,l}, \tag{5}$$

where we assume again that the points have been reordered such that points belonging to the same object o and landmark l have the same index. The poses of all the objects are then estimated by solving the following least squares problem

$$(\mathbf{x}_1, \ldots, \mathbf{x}_O)^\star = \underset{\mathbf{x}_1,\ldots,\mathbf{x}_O}{\operatorname{argmin}} \sum_{l,o} \|\mathbf{z}_{r,o,l}\|^2_{\Omega_{r,o,l}}, \tag{6}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_O$ are the poses of the objects in the scene. In this work, we employ the information matrix related to the point-to-plane error function. We first compute the normal vector at each point of the reference point cloud by estimating the principal axis of the point set in its local neighborhood. The corresponding information matrix is then set to high informative content along the normal and low informative content on the other directions. For more details on that, please refer to the generalized ICP paper [10].

Once new poses have been obtained, the algorithm re-computes the correspondences and the semantic labels and re-iterates until no new correspondences can be found or the reprojection error is below a threshold. Fig. 2 shows the factor graph underlying this minimization problem in the case where we only consider two rigid bodies: one moving object o and the static background b.

IV. Point Cloud Generation

In the previous section we described how we registered two point clouds in the presence of multiple rigid bodies. In this section, we will describe how those point clouds are generated during both the modeling and localization phase. In principle, one could use a single sensor measurement from a laser range scanner or time-of-flight camera and achieve millimeter accuracy in a static environment [7]. However the amount of measurements obtained from the object in a single step is limited and may not be enough to accurately localize with respect to it. To overcome this problem, we propose to use multiple measurements to build a local model of the environment, taken at different locations.

A. Generating the reference point cloud

To generate a reference point cloud, we first drive the robot close to the reference object with respect to which we want to be localized. We then move the robot around the reference location to collect multiple measurements of the scene and the reference object. We build the reference model by registering all the collected measurements using the same multi-body variant of generalized ICP as we did in the previous section. In this case, each rigid body is one particular scan and its reference pose is the pose where the robot was standing while recording the scan. To initialize the reference frames, we use the odometry readings from the wheel encoder. Fig. 3 depicts the factor graph corresponding to the registration step. There, $\mathbf{x}_p$ denotes the initial pose of the robot when stopped and $\mathbf{x}_i,\ i = 1, \ldots, 3$ the poses where the robot collected the additional measurements.

Once the ICP algorithm converged, we collect all the points from the multiple scans and merge them together in a single point cloud. We use $\mathbf{x}_r$ as the reference pose for the whole point cloud and transform all the points accordingly. Once the point cloud has been assembled, we manually label the points belonging to the object of interest and initially set their reference pose $\mathbf{x}_o$ to be equal to the reference pose $\mathbf{x}_r$.

B. Generating the local point cloud

Given the real-time constraints from the robot control loop, we cannot employ the multi-body registration technique for generating a local point cloud during localization. Instead, we generate it incrementally from the sensor readings in the following way. We start to generate the local point cloud when the robot reaches the location during the MCL localization part. At this point, we first run a full multi-body registration step from the last N sensor measurements, as we did for the reference point cloud. This results in N estimates of the robot poses $\mathbf{x}_i,\ i = 1, \ldots, N$, where we fixed the first pose to the one returned by MCL to remove the problem of gauge freedom. After that, we integrate each new sensor measurement by only considering the last N scans. We use the odometry from the wheel encoders to compute an initial guess of the current robot location and compute the point correspondences from the current sensor measurements and the previous N scans. The main difference with the multi-body registration is that we do not optimize with respect to all the scan poses, but only with respect to the current one and we assume the previous poses to be observed. After the current robot position is estimated, we recompute the point correspondences and re-iterate until convergence. Fig. 4 shows the underlying factor graph. There, $\mathbf{x}_r$ denotes the current pose of the robot and $\mathbf{x}_i,\ i = 1, \ldots, 3$ the last N poses that have been optimized in the previous steps. Note that we do not have any factor between the previous poses and that we consider them as observed.

Once the joint ICP procedure converged, we collect all the points from the current scan and the N previous ones and merge them together in a single point cloud. We consider the newly optimized robot pose $\mathbf{x}_r$ as the reference pose for the local point cloud and transform all the points accordingly.

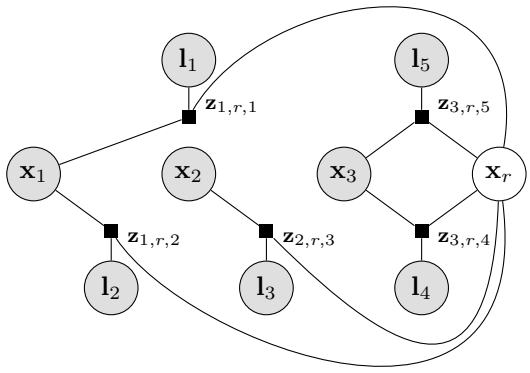

Fig. 4. Example factor graph for computing the local point cloud. The positions of the landmarks $\mathbf{l}_k$ (shaded nodes) implicitly provide measurements $\mathbf{z}_{ij}$ (black squares) between the current robot pose $\mathbf{x}_r$ (white node) and the previous laser scans $\mathbf{x}_i$ (shaded nodes). Note that in this case, we assume the poses of the laser scans to be observed.

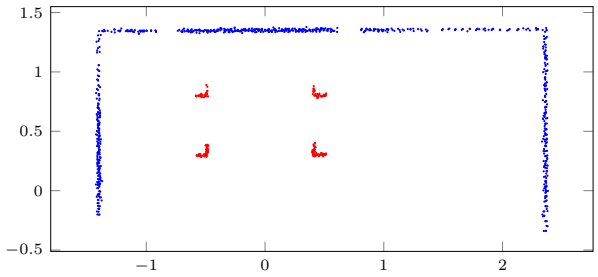

Fig. 5. Labeled model of the table for the simulation environment. The table is depicted in red and the background in blue.

V. Experiments

We performed a set of experiments to evaluate the performance of the proposed approach in both simulated environments and with a real robot. In both cases, we considered an environment composed of three different objects: a table, a box and a shelf. We considered these three objects for their peculiar shape characteristics when observed from a laser range finder. The box has a convex shape, which is easily identifiable from different view points. The shelf has a concave U-shaped footprint, which can create self-occlusions if observed from different angles. Finally, the table has a very hard shape to align with, since only the four small legs are visible from the laser range finder.

For each object, we manually drove the robot in its neighborhood to start the procedure to compute the reference model and we manually labeled the object in the reference point cloud. To remove the possible bias in the viewpoint that could be introduced by running an autonomous navigation system, we randomly drove the robot around the environment visiting each location in a random order. During the evaluation, we constantly moved each object in a different location each time. To further characterize the performances with respect to the object displacement, we translated the objects up to $10\,\mathrm{cm}$ and rotated them up to $10\,\mathrm{deg}$ in different settings. To measure the localization performance, we computed the relative transformation between the robot

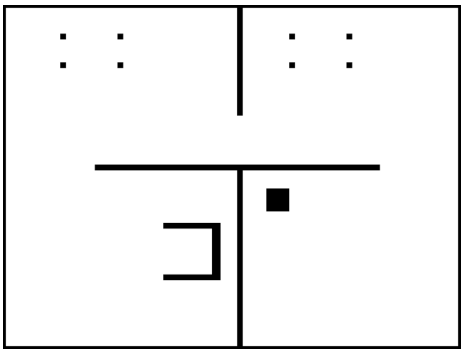

Fig. 6. Map of the simulated environment. The size of the in environment is $8\,\mathrm{m}$ by $6\,\mathrm{m}$. The table on the upper right, the box on the lower right, and the u-shape shelf on the lower left are used in the experiments.

pose and the reference pose of the object after being moved.

A. Simulation experiments

For the experiments in the simulation environment, we considered a differential drive robot equipped with a laser range finder. We simulated the measurements from the wheel encoders according to a velocity motion model, where the translational velocity was affected by a zero-mean Gaussian noise with a standard deviation of $0.1\,\mathrm{m/s}$ and the rotational velocity by a zero-mean Gaussian noise with standard deviation of $0.1\,\mathrm{rad/s}$. We simulated a laser range finder with a field of view of $180\,\mathrm{deg}$ and an angular resolution of $0.5\,\mathrm{deg}$. The simulated range reading were affected by a zero-mean Gaussian noise with a standard deviation of $0.01\,\mathrm{m}$. The map of the simulated environment is pictured in Fig. 6.

We perform six different sets of experiments. We first only moved the object around by $0.05\,\mathrm{m}$ (Set 1) and $0.1\,\mathrm{m}$ (Set 2), without rotating it. We then only perform a rotation of $5\,\mathrm{deg}$ (Set 3) and $10\,\mathrm{deg}$ (Set 4). Finally, we combined a translation of $0.05\,\mathrm{m}$ and a rotation $5\,\mathrm{deg}$ (Set 5), followed by a combined translation of $0.1\,\mathrm{m}$ and a rotation of $10\,\mathrm{deg}$ (Set 6). For each set, we performed 10 localization runs.

Fig. 8 illustrates the results of the simulated experiment. The mean translation and rotation error is roughly the same in each set, showing that the performance of our method is not affected by mild perturbations in the object position. The mean translation error is less than $2.66\,\mathrm{mm}$ for all settings and for all the objects and the maximum error is about $5\,\mathrm{mm}$. With respect to rotation, the mean error is less than $0.23\,\mathrm{deg}$ and the maximum error is about $0.85\,\mathrm{deg}$ in all the settings and for all the objects. The standard deviation, depicted with error bars in the figure, is always less than $1.4\,\mathrm{mm}$ in translation and $0.25\,\mathrm{deg}$ in rotation.

Our approach share the same limitations of generalized ICP. For example, if we have a box as reference object and we rotate it of about $45°$, the approach will suffers from the symmetry of the object. Similarly, the semantic labeling process will fail if the object is shifted too far away. In this situation, the measurements from the object are closer to the background than to the original observations.

Fig. 7. Environment used for the real world experiments with three locations and objects: a table (bottom left), a shelf (top right), and a card box (top left).

B. Real-World experiments

For the real world experiments we used the KUKA omniRob (Fig. 1) equipped with a SICK-S300 Professional laser scanner. For the ground truth, we employed the optical motion capture system from Motion Analysis Digital. The system is composed of ten high speed Raptor-E cameras that use infrared light to detect and track reflective markers. We equipped each object and the robot with different constellations of markers, such that both the pose and the identity of the object could be retrieved by the system. We conducted the experiment in an artificial environment consisting of three rooms, containing the three objects (see Fig. 7 for a picture of the experimental environment).

In this experiment, we considered three different sets for the object displacement. We first shifted the objects by $0.1\,\mathrm{m}$, without rotation (Set 1). We then rotated the objects by $10\,\mathrm{deg}$, without translation (Set 2). Finally, we combined a translation of $0.1\,\mathrm{m}$ and a rotation of $10\,\mathrm{deg}$ (Set 3). For each set, we performed 10 localization runs.

Fig. 9 illustrates the results of the experiment. When the objects are only shifted by $10cm$ (Set 1) the mean translation error for all objects is around $5\,\mathrm{mm}$ and the maximum error was measured at about $9\,\mathrm{mm}$. When to objects are rotated by $10\,\mathrm{deg}$ (Set 2) the translation errors are more spread, from $4\,\mathrm{mm}$ for the u-shape up to $11\,\mathrm{mm}$ for the table. If the objects are shifted and rotated (Set 3) the mean translation error for the table grows up to $13\,\mathrm{mm}$ with a maximum error of $19\,\mathrm{mm}$, while the errors for the box and u-shaped object are in the same range of Set 2.

Note that the increase in error for Set 2 and Set 3 is partially due to the distance between the robot and the objects. Given the lack of a correct ground-truth location of the robot after the object has been moved, we compute the reference location by applying to the motion capture estimate of the object the rigid transformation that was present during the registration of the reference model. The distance between the objects and the robot was about one and half meter, which would results in about $6\,\mathrm{mm}$ translation error if the motion capture has an angular error of a quarter of a degree. Those results are not directly comparable with our previous evaluation. In our previous work, we were able to directly measure the location of the robot in the environment, while here, we need to measure the relative

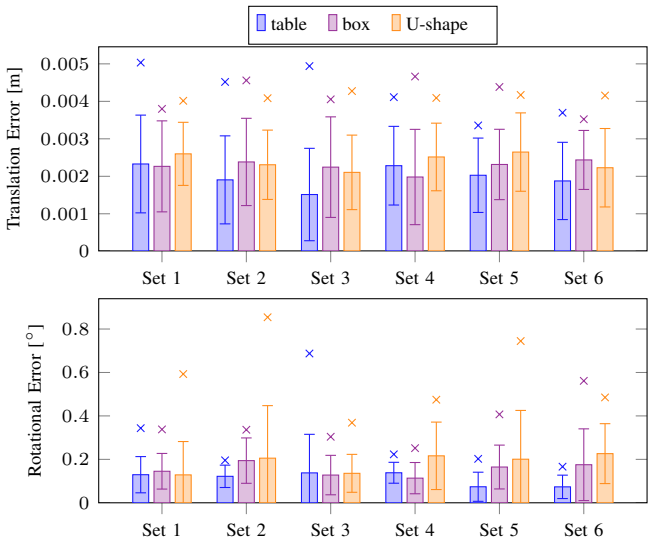

Fig. 8. Results of the simulation experiments with the object shifted by 0.05 m (Set 1), 0.1 m (Set 2), rotated by 5° (Set 3), 10° (Set 4), simultaneously translated by 0.05 m and rotated by 5° (Set 5), and simultaneously translated by 0.1 m and rotated by 10° (Set 6). The boxes indicate the mean error over all runs and the bars its standard deviation. The crosses represent the maximum errors.

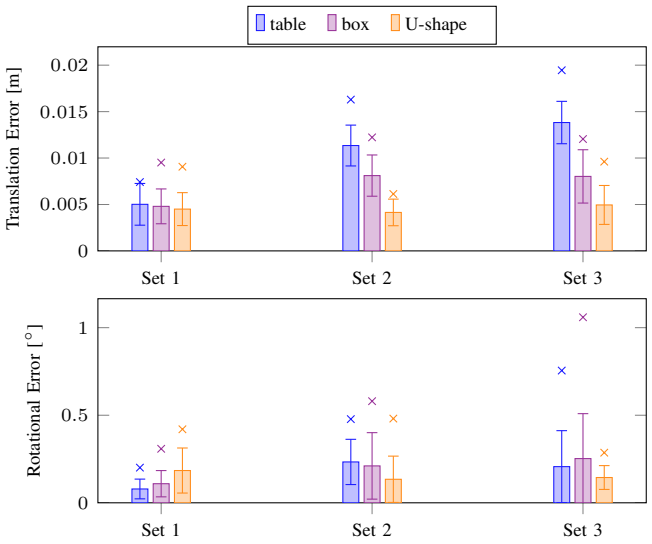

Fig. 9. Results of the real world experiments with the object shifted by 0.1 m (Set 1), rotated by 10° (Set 2), and simultaneously rotated and translated (Set 3). The boxes indicate the mean error over all runs and the bars its standard deviation. The crosses represent the maximum errors.

displacement between the object and the robot. The resulting measurements are then affected twice by the motion capture noise and position estimates are amplified by small angular errors, since the markers could not be mounted on the edges. As a consequence, we take the results in the real world scenario as lower bounds on the accuracy of the approach. Nevertheless, all measured errors are sufficient for loading actions in a warehouse, as claimed by Saarinen et al. [8].

VI. Conclusion

In this paper we presented a novel approach for localizing a mobile robot with respect to an object that can be moved in the environment. We decompose the localization problem in two main parts: global localization with respect to a map and relative localization with respect to an object. For the first part, we employ a state-of-the-art Monte Carlo localization algorithm [7]. For the second part, we propose a novel extension of the generalized ICP algorithm to handle multiple rigid bodies that undergo a set of independent rigid motions. To improve accuracy, we further propose a method to compute accurate reference models for the object during both training and localization phase. We analyzed our approach with simulated and real world experiments. The results show that our method achieves a localization accuracy of less than a centimeter, which is sufficient for typical industrial tasks, even when objects are moved around.

References

[1] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard. Cooperative Robot Localization and Target Tracking based on Least Square Minimization. In *IEEE Int. Conf. on Rob. & Aut.*, 2013.

[2] 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 Conference on Uncertainty in AI (UAI)*, 2002.

[3] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun. Towards object mapping in non-stationary environments with mobile robots. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2002.

[4] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE Int. Conf. on Rob. & Aut.*, 1999.

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

[6] S. Jain and B. Argall. Automated perception of safe docking locations with alignment information for assistive wheelchairs. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2014.

[7] J. Röwekämper, C. Sprunk, G. D. 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 *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2012.

[8] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal. Normal distributions transform monte-carlo localization (ndt-mcl). In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2013.

[9] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2013.

[10] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems*, volume 2, 2009.

[11] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[12] G. D. Tipaldi and F. Ramos. Motion Clustering and Estimation with Conditional Random Fields. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2009.

[13] J. van de Ven, F. Ramos, and G. D. Tipaldi. An Integrated Probabilistic Model for Scan-Matching, Moving Object Detection and Motion Estimation. In *IEEE Int. Conf. on Rob. & Aut.*, 2010.

[14] M. Williamson, R. Murray-Smith, and V. Hansen. Robot docking using mixtures of gaussians. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, 1999.

[15] S.-W. Yang and C.-C. Wang. Multiple-model ransac for ego-motion estimation in highly dynamic environments. In *IEEE Int. Conf. on Rob. & Aut.*, 2009.

[16] S.-W. Yang and C.-C. Wang. Simultaneous egomotion estimation, segmentation, and moving object detection. *Journal of Field Robotics*, 28(4):565–588, 2011.

[17] S.-W. Yang, C.-C. Wang, and C.-H. Chang. Ransac matching: Simultaneous registration and segmentation. In *IEEE Int. Conf. on Rob. & Aut.*, 2010.