Global Localization by Soft Object Recognition from 3D Partial Views

Fernando Ribeiro[1] and Susana Brandão[2] and João P. Costeira[3] and Manuela Veloso[4]

***Abstract*— Global localization is a widely studied problem, and in essence corresponds to the online robot pose estimation based on a given map with landmarks, an odometry model, and real robot sensory observations and motion. In most approaches, the map provides the position of visible objects, which are then recognized to provide the robot pose estimation. Such object recognition with noisy sensory data is challenging. In this paper, we present an effective global localization technique using *soft 3D object recognition* to estimate the pose with respect to the landmarks in the given map. A depth sensor acquires a partial view for each observed object, from which our algorithm extracts the robot pose relative to the objects, based on a library of 3D *Partial View Heat Kernel* descriptors. Our approach departs from methods that require classification and registration against complete 3D models, which are prone to errors due to noisy sensory data and object misclassifications in the recognition stage. We experimentally validate our method in different robot paths with different common 3D environment objects. We also show the improvement of our method compared to when the partial view information is not used.**

I. INTRODUCTION

For robots to interact with humans on a daily basis and safely operate in environments common to both, they must be able to navigate using natural landmarks, which are complex and difficult to identify. The most interesting landmarks in human environments are often medium sized objects, such as sofas, whose shape and texture yields them unique, hence valuable for localization in large environments. However, localizing the robot based on such objects is not a trivial matter as it implies recognizing both the object and its pose. Furthermore, errors in either of these tasks are common and can strongly affect localization results.

In this work, we contribute by developing a method that provides a coarse pose estimation of the robot without performing classification nor registration. This method is used in the core of our algorithm, Global Localization by Soft 3D Object Recognition (GLSOR-3D), which estimates a robot position and orientation in a global coordinate system using as landmarks multiple medium size objects.

*This research is partially supported by the NSF under award NSF IIS-1012733, a student fellowship from the FCT within the CMU-Portugal dual degree program and from the FCT under strategy grant FCT [UID/EEA/50009/2013]. The views and conclusions contained herein are those of the authors only.

[1]Fernando Ribeiro is with ECE at IST, Portugal. fernandolribeiro@ist.utl.pt

[2]Susana Brandão is with ECE at CMU, USA and IST, Portugal. sbrandao@ece.cmu.edu

[3]João P. Costeira is with Faculty of ECE at IST, Portugal. jpc@isr.ist.utl.pt

[4]Manuela Veloso is with Faculty of CS, CMU USA. mmv@cs.cmu.edu

GLSOR-3D assumes that some prior knowledge is available concerning both the landmarks distribution and orientation in a global coordinate system, as well as their appearance models. While the robot navigates in an environment with multiple objects, as illustrated in Fig. 1, GLSOR-3D provides an estimation of the robot global pose from data collected by a depth sensor, namely a Kinect Camera. In particular, the sensor provides the relative object position, and object observations in the form of a partial view, corresponding to the visible 3D surface of objects as seen from the sensor. By comparing partial views with the information in the object appearance models, GLSOR-3D extracts information on the object class and pose.

Fig. 1. Robot with a map, navigating a scenario with multiple objects.

However, the information on the relative pose and object identity is coarse as it is subject to ubiquitous sensor noise, symmetric objects, similarity between different objects and view angle discretization in the dataset. GLSOR-3D addresses this problem by:

1) using a state-of-the art partial view descriptor, the Partial View Heat Kernel (PVHK) [1], which allows for easy comparison between partial views without requiring registration against complete 3D models;
2) accumulating several observations from different positions, while estimating the displacement between observations through odometry; and
3) performing soft object detection, i.e., by computing the probability associated with each object class instead of performing hard classification.

Finally, to integrate the information from the sequence of observations and odometry readings with the environment map, GLSOR-3D uses a particle filter framework [2]. Particle filters are specially suitable when the observations are ambiguous and are not linearly related with the dynamics, as is the case for the PVHK descriptor.

We validate GLSOR-3D by performing an extensive set of experiments with different number of objects, displayed in different layouts and considering different paths. Furthermore, we assess the impact of using the coarse pose

estimation using the object models by solving the same problems with and without partial view information.

II. RELATED WORK

We present related work in indoor localization and an overview of the particle filter algorithm for localization.

A. Indoor localization

In recent years we have seen robots perform increasingly complicated tasks in human environments, for which they rely on navigation in global coordinate systems. The resilience of current localization methods, such as [3], which relies on 3D information as GLSOR-3D , ensures that robots can follow a map through numerous tasks. However, most algorithms rely only on static landmarks such as walls and corners. The ambiguity of such landmarks means that algorithms rely on external initialization. GLSOR-3D , by making use of more singular landmarks, performs global localization without requiring any other sources of information.

Also using 3D active cameras, the state-of-the-art Simultaneous Localization and Mapping++ algorithm (SLAM++) [4], provides impressive results for localization. SLAM++ tracks the camera pose while creating an object pose map of the unknown environment. On the other hand, GLSOR-3D has access to the environment map, but needs to find its own position in global coordinates by also tracking object positions. While we could use similar tracking methods using registration against dense and complete 3D models, all these steps are computationally very demanding, requiring highly parallel GPU implementations to obtain real time performance. Furthermore, this method requires classification upon object recognition, risking an erroneous decision that might affect the rest of the process. The same risk is also present in the relocation from an existing map which is addressed in [4] by choosing a highest voted pose. GLSOR-3D avoids the risk or propagating an initial erroneous estimation, by using probabilistic approaches to classification and pose estimation.

The use of joint localization and object recognition has also been addressed previously either to solve a localization [5], [6] or detection and mapping [7] problem, or for the purpose of object recognition [8], [9]. In [5], authors introduce the concept of hard and soft object detection, whether explicit classification is made or not. The devised approach performs soft recognition in a omni-directional image. It computes a per-pixel vector of detection scores, using local image features, for each object class present in the dataset. These results in a heatmap, representing the probability that a given object appears in a particular position of the image. The localization is then estimated by integrating the observations with a map annotated with the position and label of the objects using a particle filter framework [10]. GLSOR-3D has a very similar approach, however we make full use of a 3D sensor, a Kinect camera, to both estimate the probability of each object class, and to retrieve the relative position between robot and objects. The 3D information allows for easy object segmentation, and retrieval of their partial views. Furthermore, we off-line learn the appearance of several partial views per object, accounting for different view angles and thus allowing GLSOR-3D to estimate the object orientation with respect to the sensor.

Finally, GLSOR-3D uses the Partial View Heat Kernel [1], which represents partial views in a holistic form, contrary to other approaches that rely on feature matching, e.g., the approach proposed in [7]. The choice results in a less complex recognition step and a more stable representation with respect to sensor noise. However, the resulting set of observations changes in a highly non-linear, non-analytical form with respect to the robot position or dynamics. Thus, following [5], we chose particle filters above other methods such as Extended Kalman Filters also common in several localization approaches such as [11].

B. Particle filter for localization

The objective of robot localization is to estimate the robot state $s = ([x_s, y_s], \theta_s\}$, as defined by its position and orientation in a global coordinate system, at instant t, by integrating all previous observations $Z_{1:t}$, all previous actions $U_{1:t}$, and previously known information from the environment, m.

$$\hat{s_t} = \arg\max_s \{p(s|Z_{1:t}, U_{1:t}, m)\}. \quad (1)$$

In a Markovian setting, the posterior probability distribution in (1) can be defined recursively as:

$$p_{\text{target}}(s_t) = p(s_t|Z_t, U_t, m) = \mu\, p(Z_t|s_t, m) q_{\text{proposal}}(s_t) \quad (2)$$

where $\mu \in \mathbb{R}$ is a normalization constant and $q_{\text{proposal}}(s_t) = \int p(s_t|s_{t-1}, U_t, m) p(s_{t-1}|Z_{t-1}, U_{t-1}, m) \mathrm{d}s_{t-1}$, is the proposal probability distribution. A particle filter approximates the target probability in Eq. 2 at any time instant t by a set of weighted particles $S_t = \{s_t^{[j]}, w_t^{[j]}\}_{j=1,\ldots,J}$. The recursive estimation can be described in three steps:

1) **predicting** the position of a set of particles $\overline{S}_t$ from S_{t-1} according to a motion model $p(s_t|s_{t-1}, U(t,m))$;
2) **updating** the weights with the new observation:

$$w_t^{[j]} \propto p(Z_t|s_t^{[j]}, m); \quad (3)$$

3) **resampling** a new set of particles from S_t, to focus the computational effort on regions with higher probability.

Different approaches have been proposed for each step. And while there are widely accepted motion models, e.g., we use the one defined in [12], different types of observations have led to different approaches to estimate $p(Z_t|s_t^{[j]}, m)$. For example, Z_t can be either a building 3D map [3], wifi signal [13], or object position [5], [6]. Also resampling can differ on the definition of regions of high probability [8].

This work contributes to the literature on localization by focusing on the update step. In particular, we use as observations both the position of a set of landmarks and a descriptor which depends on the relative pose between sensor and landmarks, i.e., while we use information from class and relative object pose, we do not explicitly use neither.

III. UPDATE STEP FOR LOCALIZATION

GLSOR-3D takes as observations object positions and partial view descriptors. We highlight our contributions by contrasting the impact on GLSOR-3D of using such observations, with other more traditional approaches using just object positions or object position and class.

A. Update from landmark positions

When most objects or relevant features in the environment are ambiguous, robots have to localize themselves using just the landmark position, as proposed by the RoboCup SPL.

In this case, the environment map would be defined as a tuple $m = (\mathscr{S}, O)$ where: *i)* $\mathscr{S}$ represents the state space, i.e., the set of states available to the robot; *ii)* $O = \{o_1, \ldots, o_K\}$ is the set of K objects present in the scenario, and where each object is represented by a vector $o_k = [x_{o_k}, y_{o_k}]$, with the position in the global coordinate system.

At each time step, the robot collects a set of observations $Z_t = \{z_{1,t}, \ldots, z_{N_t,t}\}$, from each of the $N_t \leq K$ visible objects at time instant t. In this model, observations correspond to object positions in the robot coordinate system, $z = \tilde{T} \in \mathbb{R}^2$. For $N_t = 1$, we compute $p(Z_t = \{\tilde{T}_{1,t}\}|s,m)$ by:

1) Estimating the position vector $T_{1,t} = R_s \tilde{T}_{1,t}$ in global coordinates, where R_s is the rotation matrix computed from the robot orientation $\theta_s \in s$.
2) Computing $p(T_{1,t}|s,o_k,m)$ from a Gaussian distribution centered in the object o position in the environment $[x_{o_k}, y_{o_k}]$, and with variance σ_T^2:

$$p(T_{1,t}|s,o_k,m) \propto \exp\left\{\frac{-\|(x_s,y_s)+T_{1,t}-(x_{o_k},y_{o_k})\|^2}{(2\sigma_T^2)}\right\} \quad (4)$$

3) Computing $p(Z_t|s,m) = \sum_{o_k \in O} p(T_{1,t}|s,o_k,m)$, by marginalizing over all objects in the map.

When $N_t > 1$, the observations are mutually exclusive, i.e., $T_{1,t}$ and $T_{2,t}$ cannot be both from o_1. Thus, we account for all ordered sets of N_t objects from the K objects in the map, $\Gamma(N_t)$. Each set of ordered objects, $\gamma = \{o_1, \ldots, o_{N_t}\} \in \Gamma(N_t)$, corresponds to an assignment of observations to objects, i.e., z_n is attributed to the object in γ_n. Thus, GLSOR-3D marginalizes over all $\gamma \in \Gamma(N_t)$, and the third step becomes:

3) Computing $p(Z_t|s,m) = \sum_{\gamma \in \Gamma(N_t)} \sum_{n=1}^{N} p(\tilde{T}_{n,t}|s,\gamma_n,m)$.

B. Update from landmark position and class

Most algorithms use as observations the class and position of either objects or features, [5], [7], [6]. Thus, observations must also contain information on the object class in the form of some descriptor, d_n, as illustrated in Fig. 2. In this case each observation z_n becomes a tuple: $z_n = (\tilde{T}_n, d_n)$.

To accommodate the class estimation, the knowledge map must be extended to contain object models, $m = (\mathscr{S}, O, \mathscr{D} = \{D_1, \ldots, D_K\})$, that allow a soft classifier to compute $p(d|D_c)$, for each object class c. Thus, each element $o \in O$ must also contain the object class, and becomes a tuple defined as $o_k = (c_k, [x_k, y_k])$.

Assuming that when the robot state, s, is known, $\tilde{T}_{n,t}$ is independent of the descriptor $d_{n,t}$ and that $d_{n,t}$ only depends on the object class, we estimate the update probability by:

1-2) as in Section III-A;
3) estimate $p(d_{n,t}|D_{c_k})$ for each object class;
4) marginalize over all $\gamma \in \Gamma_{N_t}$;

$$p(Z_t|s,m) = \sum_{\gamma \in \Gamma(N_t)} \sum_{n=1}^{N_t} p(\tilde{T}_{n,t}|s,\gamma_n,m) p(d_n|D_{c \in \gamma_n}). \quad (5)$$

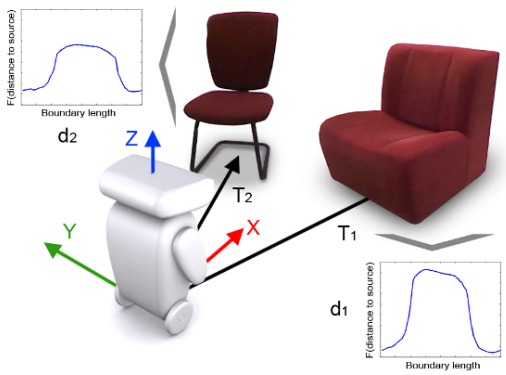


Fig. 2. Data from each observation Z, namely a descriptor d_n and a translation vector T_n for each visible object.

C. GLSOR-3D update from landmark position, pose and class

GLSOR-3D uses information on the pose and object class provided by a descriptor vector $d_n \in \mathrm{R}^L$. Each object model $D_{c_k} \in \mathscr{D}$ is a collection of descriptors indexed by the view angles v_{o_k} in the object intrinsic coordinate system. Thus, when given a state in global coordinates, s, GLSOR-3D first computes the equivalent in the object coordinate system. The conversion between the two systems is defined by the object orientation θ_o, which we include in each $o_k = (c_k, [\tilde{x}_k, \tilde{y}_k], \theta_k) \in O$.

Finally, given a soft classifier that estimates the probability that a given descriptor d was generated from observing the object o from the view angle v, $p(d|D_{c \in o,v})$, GLSOR-3D estimates $p(z_t|s,o_k,m)$ by:

1-2) as in Section III-A;
3) estimate the view angle v associated with position $(x,y)_s$ and the object orientation θ_k:

$$v = \arctan\{(y_s - y_k)/(x_s - x_k)\} - \theta_k; \quad (6)$$

4) estimate $p(d|D_{c_k,v})$ with the classifier;
5) marginalize over all the sets $\gamma \in \Gamma_{N_t}$, using Eq.5.

Thus, GLSOR-3D needs to estimate probability $p(d|D_{c_k,v})$ using a partial view representation that depends not only on the object class, but also on the view angle.

IV. PARTIAL VIEW REPRESENTATION

The PVHK descriptor, introduced in [1], is a vector $d \in \mathbb{R}^L$ that represents any object partial view in an informative way, and that depends on the observer view angle. Here we briefly describe the descriptor, introduce our object models D and our soft classification approach to estimate $p(d|D)$.

A. PVHK descriptor

The gist of PVHK is to represent partial views by the geodesic distance between a reference point in the object surface and the ordered set of points in the partial view boundary. The reference point is chosen systematically and depends on the relative position between sensor and object. As the distance between a point and the boundary uniquely defines the surface, apart from isometric transformations, the PVHK is unique for any given pair of object and view angle. To handle the impact of noisy 3D data on the geodesic distances, PVHK uses diffusive geometry concepts, which are considerably more robust [14]. In particular, it relies on the heat propagation, to obtain proxies for distances. The propagation, illustrated in Fig. 3, considers an instantaneous heat source at the reference point. By stopping the propagation when temperature on all points is above a given threshold, PVHK obtains a temperature profile correlated with the desired geodesic distance, but also resilient to noise.

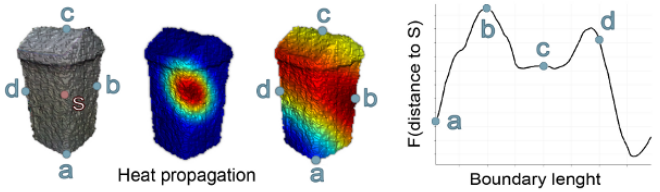


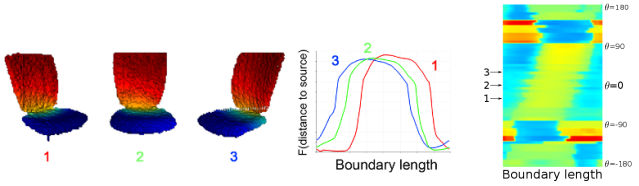
Fig. 3. Representation of an object with the PVHK descriptor. Heat propagates from a source s and is measured at boundary points $a,\ldots,d$.

Thus, the PVHK descriptor is a vector $d \in \mathbb{R}^L$, whose entry d_i is equivalent to the temperature on the boundary point i. The boundary points are equally spaced in terms of length over the boundary, e.g., for a square with side l, they would be spaced so that the distance between consecutive points would be $4l/L$. We here use $L = 80$. Furthermore, the reference point for the source is selected by first projecting the partial view surface into the camera plane and then finding the point closest to the center of the 2D projection.

The PVHK changes smoothly with the view angle, as illustrated in Fig. 4. On Fig. 4(a), we illustrate the temperature profile over the chair when the view angle slightly changes. In the three images, red regions have higher temperatures than the blue ones. The descriptors of the three partial views are represented in the middle graph, and we can see that while the shape of each curve is the same, there is a translation associated with each descriptor. Finally on the right, we represent, as rows in a matrix, the collection of descriptors for the chair. Again, red corresponds to higher values of the descriptor, and blue to lower values.

B. Soft Classification of pose and object class

The PVHK was first used in the context of recognition from a sequence of view angles [8], where a particle filter is also used. However, no localization is performed and only a single object was considered at all times. We here use the same approach to compare descriptors and estimate probabilities. Namely, we establish the probability that a



(a) Descriptors from a sequence of view angles. (b) Chair dataset.

Fig. 4. The PVHK changes smoothly with the view angle. Fig. 4(a) shows the heat profile changing smoothly between three consecutive view angles. Fig. 4(b) represents a collection of descriptors, one per row, which are ordered by view angles. In both figures, red correspond to higher temperatures, and blue to colder ones.

descriptor d corresponds to an object class c and view angle v, by computing the distance, $\rho : \mathbb{R}^L \times \mathbb{R}^L \rightarrow \mathbb{R}$, between d and the reference descriptor in the object model $D_{c,v}$. Then, as in [8], we assume an exponential distribution on the distances, as it has a smoother cut off distance than, e.g., the Gaussian distribution.

$$p(d|c,v) = \exp\{(-\rho(d,d^{c,v})/\lambda)/\lambda\}. \quad (7)$$

The normalization constant, λ, represents the average inner distance between descriptors in the dataset.

We also use the distance function ρ proposed in [8], i.e., we compare descriptors by comparing the shape of the curves they define in graphics such as those in Fig. 4, where temperature is plotted as a function of boundary length. Thus, we first convert each vector d to a curve η, defined as the set of points $\eta = \{[1/L,d_1],[2/L,d_2],...,[1,d_L]\}$. Then we use the Modified Hausdorff Distance ($\mathscr{H}$) to establish the distance ρ between sets: $\rho(d,d') = \mathscr{H}(\eta,\eta')$, where $\mathscr{H}(\eta,\eta') = \min\left\{\sum_{x\in\eta}\inf_{y\in\eta'}\|x-y\|^2, \sum_{x\in\eta'}\inf_{y\in\eta}\|x-y\|^2\right\}$.

V. EXPERIMENTAL SETUP

We validate our localization algorithm in a diversified set of experiments, taking place in our office, using every day objects such as sofas and chairs. We designed several experiments in order to: *a*) show that GLSOR-3D effectively estimates a correct final position by reducing errors with new observations; *b*) show how the inclusion of multiple objects leads to better estimations from the observations. *c*) show the impact of the soft object recognition versus the use of landmark position alone;

A. Data collection

To construct both object models and the environment map, we use augmented reality markers [15]. These markers, resembling QR codes, are easy to identify in a RGB image and allow to estimate both the 3D position and orientation of the observer with respect to the marker. The markers can then be combined to define an exterior coordinate system, where the observer can localize himself.

To create each object model, D_c, we placed a set of markers attached to each object, as illustrated in the photographs in Fig. 5. We then acquired several partial views by moving around the object, and computed the descriptors for each. The matrices at the bottom of Fig. 5 represent the object datasets,

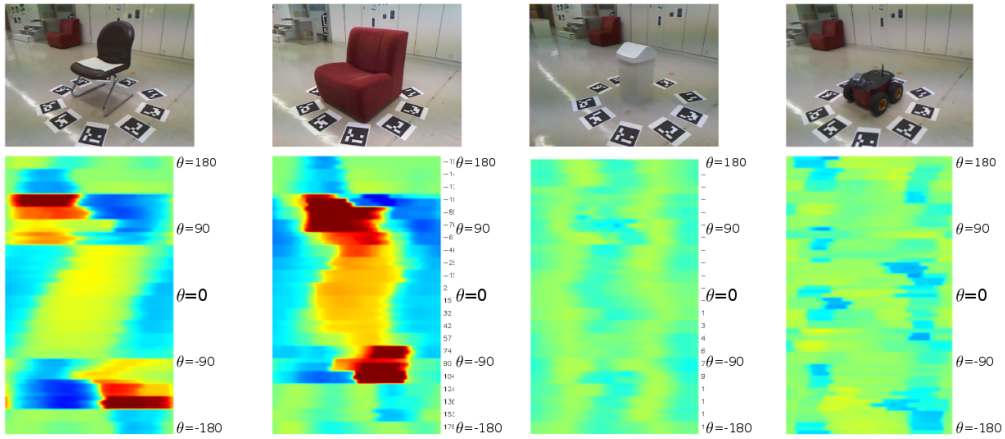


Fig. 5. Objects used for localization, in addition to the chair presented in Fig. 4. In the top row, images of the objects are visible while in the bottom row the acquired descriptors are represented, ordered by θ. From left to right the objects are: brown chair, sofa, Trash can and robot.

with a descriptor per row and respective view angle, as computed from the markers. In total we constructed 5 object models: the four in Fig. 5 corresponding to a brown chair, a sofa, a trash can and a robot; and the chair in Fig. 4(b). These objects present strong similarities and symmetric shapes to assess the GLSOR-3D robustness to ambiguous observations.

We used a second set of markers to construct a global coordinate system. The markers attached to each object provide the respective pose in global coordinates, and allow the easy creation of different environment maps, with different object layouts. Namely, we created a total of 9 maps, from where we gathered information over 22 different paths. Each map differs with respect to the number of objects and their distribution in the map: *a*) a single object - total of seven paths, one per object, except the sofa with 3; *b*) pairs of objects - total of six paths, two per each of three different layouts; and *c*) all objects - total of seven paths.

The online data for the experiments was collected using a hand-held Kinect camera. We estimated the odometry data from changes in position assessed using markers, which was then corrupted with Gaussian noise. Furthermore, observational evidence showed that the sensor had an error in the object position vector of 15 cm, so we used $\sigma_T = 15$cm. To estimative the robot state from the set of particles, GLSOR-3D randomly chooses one percent of all the particles as anchors. For each anchor, it computes the number of particles inside a neighborhood of radius $\tau = 30$cm and the one with most neighbors is the expected state. Finally, each experiment used $J = 2000$ particles.

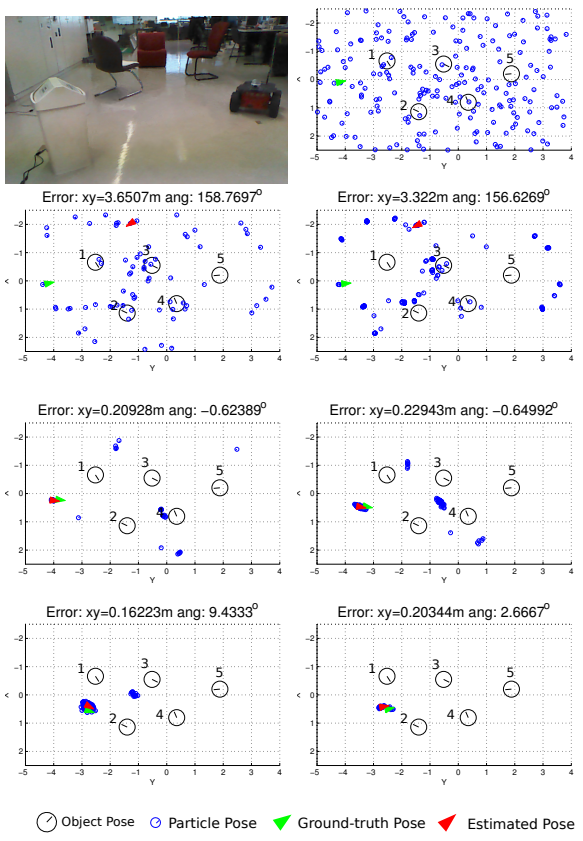


Fig. 6. Execution of the implemented algorithm. Only a sample of the total particles are presented. Objects are numbered as: 1-trash can; 2-robot; 3-brown chair; 4-red chair; 5- sofa.

B. Results

In Fig. 6 we represent a sequence of steps in the execution of GLSOR-3D . Steps 2-3, show that initially particles are scattered around two objects: 1 and 5, but as the robot moves and more data is collected, the particles get centered around the ground truth, as can be seen in the steps 4-7. The same behavior is also noticeable in Fig. 7, where we can see the error on both position and orientation decreasing with the first observations, but then remaining constant.

In Table I we compare the estimation errors obtained with GLSOR-3D for different sets of objects. The results show that for single objects, the descriptor does not always identify the view angle due to geometric symmetries. This is specially noticeable for the trash can and the robot examples, which have roughly a square symmetry and thus ambiguous observations. With object pairs, GLSOR-3D disambiguates

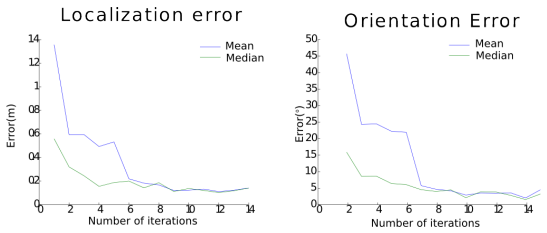


Fig. 7. Error on the estimation of both position and orientation. The mean and the median are taken over all the experiments, including all the maps layout and paths.

symmetries by leveraging on the view angle estimation from both objects. In sets with all objects, observations either capture just one or two objects at the same time. Therefore the resulting error falls between the two previous.

TABLE I

FINAL LOCALIZATION ERRORS

	Position (m)		Angle (°)	
Objects	Mean	Median	Mean	Median
Single	0.2279	0.2057	7.4610	6.6865
Pairs	0.1271	0.1243	2.9535	2.5121
5 Obj.	0.1909	0.1949	4.3336	4.3128
All	0.1847	0.1586	5.0142	4.8695

To assess the impact of soft recognition, we ran the same experiments using the algorithm proposed in SectionIII-A. However, we use only the sets with multiple objects, as we cannot estimate the position from the relative position to a single object. As can be seen in Table II, errors greatly increase when compared to Table I, as observations are still fairly ambiguous. E.g., Fig. 8a) shows the two symmetric, high probability regions resulting from the use of just two landmarks. In experiments with more objects, the difficulty in removing the initial ambiguity may lead to depletion of particles in the correct position, resulting in a final erroneous estimation (Fig. 8b). Finally, using only landmark positions for localization, lead to a slower convergence of the particles, (Fig. 9), and often results in higher estimation errors.

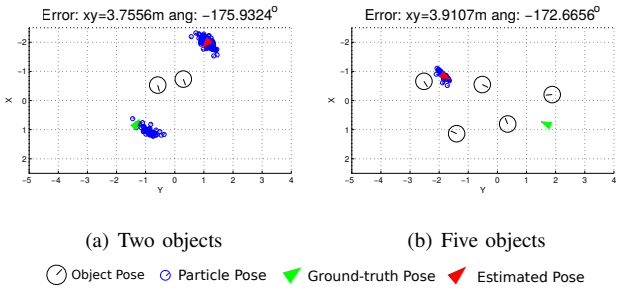


Fig. 8. Examples of errors encountered when the descriptors were not used for pose estimations.

VI. CONCLUSIONS

In this paper we presented a method for mobile robot localization using multiple objects as landmarks while avoiding explicit classification and avoiding registration against

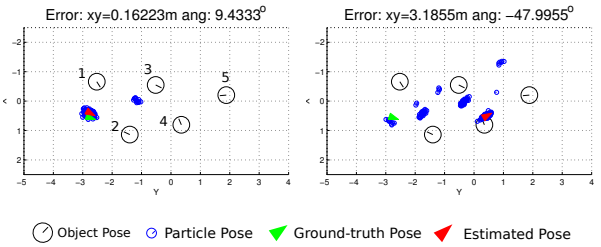


Fig. 9. Comparison between the particles distribution after 5 iterations of GLSOR-3D (left) and the particles obtained when only the position is used(right), highlighting the higher convergence rate, of GLSOR-3D .

TABLE II

FINAL LOCALIZATION ERRORS W/O DESCRIPTORS

	Position (m)		Angle (°)	
Objects	Mean	Median	Mean	Median
Single	-	-	-	-
Pairs	2.5232	3.1588	118.3304	174.0345
5 Obj.	1.1274	0.2332	26.9688	7.3373
All	1.7716	0.2894	69.1357	7.3522

complete 3D models for pose estimation. GLSOR-3D achieves this by exploiting the PVHK descriptor and the Modified Hausdorff distance as tools to recognize and compare the similarity between objects represented by their partial views.The results presented empirically show that indeed GLSOR-3D performs localization using multiple objects as landmarks and that soft recognition of objects and pose improve the localization considerably.

REFERENCES

[1] S. Brandão, J. P. Costeira, and M. Veloso, "The partial view heat kernel descriptor for 3d object representation," in *ICRA*, 2014.
[2] M. Arulampalam, N. G. S. Maskell, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *In IEEE Trans. on Sig. Processing*, 2002.
[3] J. Biswas and M. M. Veloso, "Localization and navigation of the cobots over long-term deployments," *I. J. Robotic Res.*, 2013.
[4] 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 *CVPR*, 2013.
[5] R. Anati, D. Scaramuzza, K. Derpanis, and K. Daniilidis, "Robot localization using soft object detection," in *ICRA*, 2012.
[6] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas, "Semantic Localization Via the Matrix Permanent," in *Robotics: Science and Systems (RSS)*, 2014.
[7] N. Fioraio and L. Di Stefano, "Joint detection, tracking and mapping by semantic bundle adjustment," in *CVPR*, 2013.
[8] S. Brandão, M. Veloso, and J. P. Costeira, "Multiple hypothesis for object class disambiguation from multiple observations," in *3DV*, 2014.
[9] F. von Hundelshausen and M. Veloso, "Active monte carlo recognition," in *KI-2006*, 2006.
[10] S. Thrun, D. Fox, W. Burgard, and D. F., "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, 2001.
[11] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *ICCV*, 2003.
[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
[13] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *ICRA*, 2010.
[14] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably infor. multi-scale signat. based on heat diffusion," in *SGP*, 2009.
[15] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, 2014.