Merging maps of multiple robots*

Nagesh Adluru, Longin Jan Latecki, Marc Sobel, Rolf Lakaemper
{*nagesh,latecki,marc.sobel,lakamper*}@*temple.edu*

Abstract

Merging local maps, acquired by multiple robots, into a global map, (also known as map merging) is one of the important issues faced by virtually all cooperative exploration techniques. We present a novel and simple solution to the problem of map merging by reducing it to the problem of SLAM of a single *"virtual" robot. The individual local maps and their shape information constitute the sensor information for the virtual robot. This approach allows us to adapt the framework of Rao-Blackwellized particle filtering used in SLAM of a single robot for the problem of map merging.*

1 Introduction

One of the key challenges for insuring the autonomy in multi-robot systems is to *cooperatively* explore and build the world model of their environment. In a typical multi-robot system, the individual robots build maps in their local coordinate frames. These local maps have to be transformed into a common (global) coordinate frame. The problem of estimating these transformations is known as the map-merging problem. In virtually all existing work the problem is treated as a *search* problem by iteratively proposing candidate transformations and verifying the quality of the proposal. The differences are characterized by factors guiding their proposal and verification processes. A general heuristic guiding the techniques is the solution must produce "perceptually good and consistent maps".

For example, [4] designed an adaptive random walk based motion planning which they use for proposal. They use an image dissimilarity metric based on [2] for verifying the transformations and adapting the sampling distribution for their random walk. Their approach is superior to most of the existing approaches, but their main drawback is the fact that their random walks need to be sampled from Gaussian distributions for convergence. If the initial set of transformations places local maps too far apart in the global frame, their metric cannot properly modify the distribution appropriately and puts convergence at risk. In the proposed approach we employ Sequential Monte Carlo estimation of transformations in the merging process by tracking multiple hypotheses. The estimation process is guided by shape information in the local maps. The main difference between straightforward multi-robot SLAM as in [10, 6] and our ViRtual SLAM (VR-SLAM) is that we restrict the state space to that of a trajectory of the single (virtual) robot. As a consequence we do not estimate the joint trajectories of individual robots. This makes our technique more scalable in the number of local maps, and more robust when overlap among local maps is minimal (at least one common structure). This can be observed in our results where maps built by up to 10 individual robots are merged. This can be compared to results in [4] where they merge up to 6 maps.

2 Overview of VR-SLAM

In a single robot SLAM a joint posterior over trajectories, $x_{1:t}$, and maps, m_t, is maximized constrained by a sequence of range measurements, $z_{1:t}$ and odometry readings, $u_{1:t}$. The goal is to find $\underset{x_{1:t},m_t}{\operatorname{argmax}}\ p(x_{1:t}, m_t | z_{1:t}, u_{1:t})$. For more details see [11]. A very successful framework for such optimization is Rao-Blackwellized particle filtering as presented in [7]. In this framework the posterior is represented using a set of particles. Each particle represents a trajectory and an associated map. At every time step t, the most likely particle is used by the robot for navigational purposes. A range measurement z_t at time t captures a small part of the environment as a local scan. An odometry reading u_t provides the update information for the robot's pose. Hence this optimization can be viewed as the process of consistently merging a sequence of local scans into a global map by tracking multiple hypotheses.

In the proposed VR-SLAM a similar optimization is performed. For ease of understanding the optimization process we imagine a *virtual* robot trying to navigate using the individual robots as its sensors. The local maps built by the individual robots replace the range measurements (local scans). The odometry readings are derived from registration of similar structures

*This work was support in part by the NSF Grant No. IIS-0534929 and by the DOE Grant No. DE-FG52-06NA27508.

in local maps. The main differences between a single robot SLAM and our VR-SLAM are due to the differences in motion model and perception model of the virtual robot. This leads to differences in designing the *proposal distribution* and computing the *importance weights* of the particles when using the particle filtering framework. The design of the proposal is based on the motion model of the robot while computing weights is based on the perception model. Thus, once we formulate the navigational behavior of our virtual robot, the framework used in [7] can be adapted for maximizing the joint posterior of "virtual" trajectories and maps built by the virtual robot. This results in a solution to the map-merging problem. The motion model and proposal are described in Section 3. The perception model and importance weights are described in Section 4. Henceforth we use the following notation: (i) Z to denote the set of local maps to be merged, and (ii) φ_t to denote the index of the local map that is merged at time step t. Usually the individual robots can be ordered based on their proximities; this gives us the *sequence* of local maps i.e. $Z[\varphi_{1:t}]$. Even if such an ordering is not available, the framework remains similar except that the motion model distribution has a larger number of modes because at each time step structure registration is performed between merged and *all* unmerged local maps instead of just one unmerged map.

3 Motion from structure registration

We use the optimal proposal with ability to handle multiple modes as described in [5] which is an improvement over [7]. There are two key differences between implementing the proposal for regular SLAM and that for VR-SLAM: (1) Drawing the follower poses from the distribution modeling the motion of the robot i.e. $p(x_t|x_{t-1}, u_t)$, which is explained in this section. (2) The computation of the likelihood of poses i.e. $p(z_t|x_j, m_{t-1})$ (Section 4). The rest of the procedure remains same as described in [5]. Simulation from the above distribution is based on the motion model of the robot. Now we explain the structure registration based motion model for our virtual robot. The virtual robot always moves in the global frame. Its pose is initialized at the origin of one of the local maps, which becomes the reference coordinate frame (global frame). Its motion update is then guided by "structure registration" among local maps. Its position in a local map is always at the local origin. Similar structures among local maps are extracted and registered using closed form solutions like in [9, 8, 1]. The correspondences are obtained by shape matching between similar structures. u_t encapsulates the motion updates by registering similar structures between local maps $Z[\varphi_{t-1}]$ and $Z[\varphi_t]$. There could be several similar structures between $Z[\varphi_{t-1}]$ and $Z[\varphi_t]$. As a consequence, our proposal distribution is multi-modal with peaks around the poses predicted using structure registration. Distribution of typical, odometry based motion model is shown in Fig. 1 (a) while that in our case is shown in Fig. 1 (b). Fig. 2 shows a sample motion update process with two modes. We note that each possible pose update is actually a transformation of the local map into the global frame of m_{t-1}.

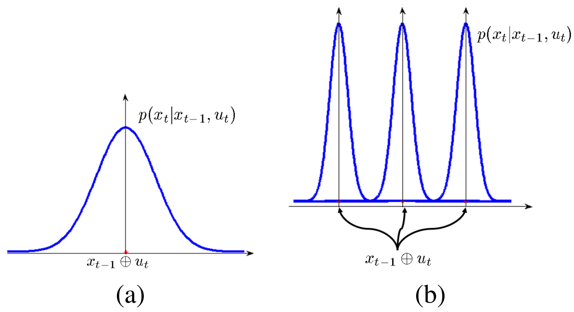

Figure 1. (a) Typical distribution of odometry based motion model in a single robot SLAM. (b) In our case the distribution is multi-modal with number of modes being equal to the number of pairs of similar structures.

4 Perception model using image similarity

The main component in recursively estimating importance weights as described in [7] is $p(z_t|x_j, m_{t-1})$, which is also used in implementing the optimal proposal [7]. The computation of the above likelihood is based on the perception model of the robot. As mentioned earlier the local maps built by the individual robots form the sensor readings of the virtual robot. Hence z_t is characterized by the occupancy grid of the local map $Z[\varphi_t]$. Our approach is motivated by correlation based models used in regular SLAM [11]. The standard correlation metric based on the normalized quadratic distance does not address the "see through walls" problem [11]. A correlation metric Δ based on image distance ψ (introduced in [2]) and a heuristic to address the inconsistent merge issue was introduced in [3]. The heuristic is based on the idea of "locking" the merge process to avoid slipping into inconsistent merges with small ψ. This heuristic although *preventing* slipping into inconsistent merges, fails to *guide* the process. This is because their random walks are guided by the gradients computed on ψ.

We present a correlation metric that addresses the "see through walls" issue better than heretofore. The

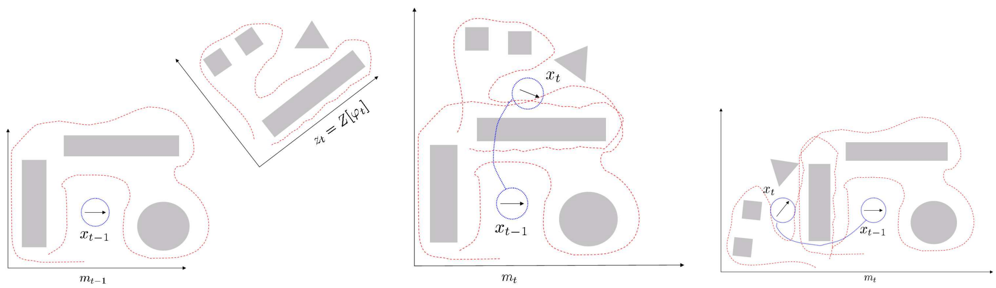

Figure 2. Left: The virtual robot's pose in the global frame is shown with an arrow inside the circle. The new local map, $Z[\varphi_t]$ is shown in its local frame. The dotted red lines are the trajectories of the individual robots. Middle: One possible structure registration updates the pose of the virtual robot. The virtual robot's jump is shown in dotted blue line. Right: Another possible structure registration leads to a different pose update for the virtual robot.

local maps are converted into digital images by discretizing cell probability values into a set $C = \{free, occupied, unknown\}$. There are two main components of the proposed measure viz. Ψ and γ. Ψ takes account of the "overlap" between two maps, m_1, m_2 in a common coordinate frame and γ takes account of the "see through walls" issue.

Ψ is computed in a similar fashion to that of ψ presented in [3]. The main difference is that we compute *similarity* between images in a way similar to that used to compute likelihood fields of maps [11]. The basic idea is to treat the images as arrays of pixels and reward matching pixels based on their values and relative positions. The arrays are scanned for each class of pixels. For each pixel, the distance to its closest similarly valued pixel in the other image is computed and its score is taken to be proportional to a Gaussian transform of this distance. Fortunately the distances can be computed in linear time in the number of pixels in the images using "distance-maps" [3].

The inconsistency of a merge is defined as the mismatch in the perception of the robot between two differently calculated positions. γ is designed in such a way that its magnitude is made proportional to this merge inconsistency. The mismatch in robot perception is based on the number of disagreeing pixels in the two images. Thus its computation involves similar steps as those for Ψ. The larger the number of matching pixels ($c_1 = c_2$), the larger Ψ is. In contrast to this, the larger the number of mismatched pixels ($c_1 \neq c_2$), the larger $gamma$ is. Each pose update of the virtual robot represents a transformation of the local map into the global frame (cf., 3). The likelihood scores for the poses are computed by normalizing Ψ and γ over all the set of poses, $\{x_j\}_{j=1}^n$ at time t as:

$$\widetilde{\Psi}_j = \frac{\Psi_j}{\sum_{j=1}^n \Psi_j}, \text{where } \Psi_j = \Psi(m_{t-1}, T_{x_j}(z_t))$$

$$\widetilde{\gamma}_j = \frac{\gamma_j - min(\gamma_j)}{\sum_{j=1}^n (\gamma_j - min(\gamma_j))}, \text{where } \gamma_i = \gamma(m_{t-1}, T_{x_j}(z_t))$$

T_{x_j} is the homogenous transformation matrix obtained using the pose x_j. The $\{\widetilde{\Psi}_j\}_{j=1}^n$ and $\{\widetilde{\gamma}_j\}_{j=1}^n$ form discrete *independent* probability distributions. The likelihood score $p(z_t|m_{t-1}, x_j)$ is given by the product $\widetilde{\Psi}_j.\widetilde{\gamma}_j$. The likelihood scores of a typical set of poses is shown in Fig. 3. The figure also shows how neither $\widetilde{\Psi}$ nor $\widetilde{\gamma}$ by itself is sufficient but their combination helps improve the likelihood scores.

5 Experimental results

We applied our technique to the data collected by National Institute of Standards and Technology (NIST) in a "maze" type of environment[1]. We are provided only with the local maps without any initial estimate of the relative poses. For feature extraction we fit lines to the data points and determine corners. For typical indoor local maps, corners are sufficiently distinct feature descriptors to represent the environment structure. Sample corners extracted from two partial maps can be seen in Fig. 3. On average there were about 10 corners in a local map and there were about 5 matching corner pairs between two local maps. The results are shown for merging maps from $3, 5, 6$ and 10 robots in Fig. 4. We used an average of 50 to 100 particles. As can be seen we could successfully merge maps of 10

[1]We would like to thank Raj Madhavan (NIST) for providing the dataset.

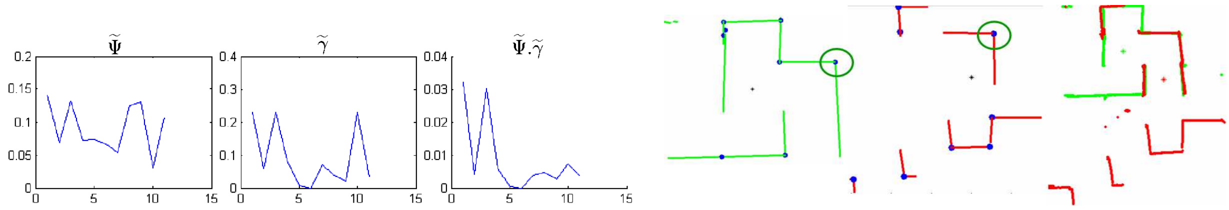

Figure 3. The scores for a set of 11 poses are shown. The $8^{th}, 9^{th}$ and 11^{th} poses have high $\tilde{\Psi}$ indicating good overlap but low consistency scores as seen by $\tilde{\gamma}$. The final likelihood scores obtained by their product reflect scores in which both overlap and consistency are satisfied. Corners extracted from two local maps (green and red) are shown above in blue. A pair of similar corners are marked with green circles. The merged map obtained by registering the two similar corners is shown in the right.

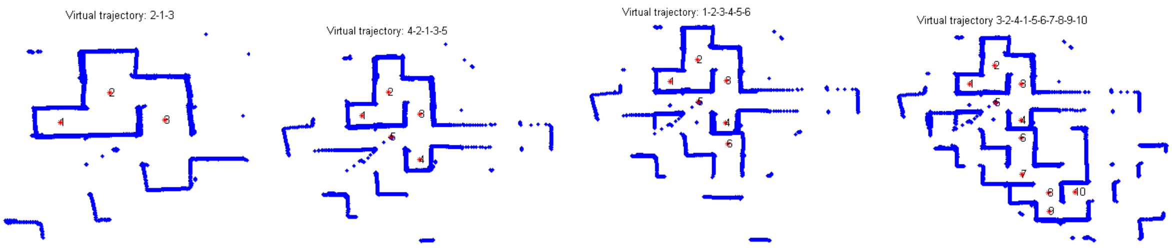

Figure 4. Merged maps of $3, 5, 6$ and 10 robots. The map indices are placed at the local origins of the individual maps. The sequence of the maps merged is shown in the respective titles.

robots. Since we did not assume the sequence of local maps our process estimates the sequence also automatically as mentioned in Section 2. For e.g. the optimal sequence of maps for 10 robots (Fig. 4) is $(3, 2, 4, 1, 5, 6, 7, 8, 9, 10)$. The time complexity of our algorithm depends on the number of local maps to be merged (N), the average number of corners in each map (c) and the number of particles (N_p). If the sequence of local maps to be merged ($Z[\varphi_{1:t}]$) is known then the complexity is $O(N_p c^2 N^2)$ otherwise it is $O(N_p c^2 N^3)$.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[2] A. Birk. Learning geometric concepts with an evolutionary algorithm. In *Proc. of The Fifth Annual Conference on Evolutionary Programming*. The MIT Press, Cambridge, 1996.

[3] A. Birk and S. Carpin. Merging occupancy grid maps from multiple robots. In *Proceedings of the IEEE*, volume 94, pages 1384 – 1397, July 2006.

[4] S. Carpin, A. Birk, and V. Jucikas. On map merging. *Robotics and Autonomous Systems*, 53(1):1–14, 2005.

[5] W. B. Cyrill Stachniss, Grisetti Giorgio and N. Roy. Analyzing gaussian proposal distributions for mapping with rao-blackwellized particle filters. In *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (To appear)*, 2007.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Efficient multi-robot localization based on monte carlo approximation, 1999.

[7] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. on Robotics*, 23:34–46, 2007.

[8] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[9] B. K. P. Horn, H. M. Hilden, and S. Negahdaripourt. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[10] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

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