Towards Object Mapping in Non-Stationary Environments With Mobile Robots

Rahul Biswas[1], Benson Limketkai[1], Scott Sanner[1], Sebastian Thrun[2]

[1]*Stanford University, Stanford, CA, USA, {rahul, bensonl, ssanner}@cs.stanford.edu*
[2]*Carnegie Mellon University, Pittsburgh, PA, USA, thrun@cs.cmu.edu*

Abstract

We propose an occupancy grid mapping algorithm for mobile robots operating in environments where objects change their locations over time. Virtually all existing environment mapping algorithms rely on a static world assumption, rendering them inapplicable to environments where things (chairs, desks, ...) move. A natural goal of robotics research, thus, is to learn models of non-stationary objects, and determine where they are at any point in time. This paper proposes an extension to the well-known occupancy grid mapping technique. Our approach uses a straightforward map differencing technique to detect changes in an environment over time. It employs the expectation maximization algorithm to learn models of non-stationary objects, and to determine the location of such objects in individual occupancy grid maps built at different points in time. By combining data from multiple maps when learning object models, the resulting models have higher fidelity than could be obtained from any single map. A Bayesian complexity measure is applied to determine the number of different objects in the model, making it possible to apply the approach to situations where not all objects are present at all times in the map.

1 Introduction

The field of robotic mapping is among the most active in mobile robotics research [7, 16]. Mapping addresses the problem of acquiring an environment model with a mobile robot, suitable for navigation and visualization. Recent innovations include scalable online techniques for concurrent mapping and localization [5, 8], algorithms for generating compact three-dimensional maps [6], and autonomous exploration techniques for controlling robots during mapping [14].

However, most existing robotic mapping algorithms possess one important deficiency – they all assume that the world is static. Thus, things may not move when acquiring a map. Dynamic effects, such as people that may briefly obstruct the robot's sensors, are filtered away at best, and lead to mapping failure at worst. The static world assumption in robotic mapping is motivated by the fact that even for static worlds, the mapping problem is very hard [15]. Efforts have been made to learn certain types of dynamic effects, e.g. the presence of doors [13], but have limited applicability due to their specificity. However, most natural environments are not stationary. For example, office environments contain objects such as chairs, desks, and people, which frequently change their location. The goal of this research, thus, is to devise methods that can identify such non-stationary objects and model their time-varying locations.

This paper proposes an occupancy grid mapping algorithm—called *robot object mapping algorithm* or *ROMA*—capable of modeling non-stationary environments. Our approach assumes that objects in the environment move sufficiently slowly that they can safely be assumed to be static for the time it takes to build an occupancy grid map. However, their locations may change over longer time periods (e.g., from one day to another). An example of such a situation is an office delivery robot, which may enter offices in regular time intervals. From one visit to another, the configuration of the environment may have changed in unpredictable ways (e.g., chairs moved around and in or out of a room). Since the robot may not witness the motion directly, conventional tracking techniques [2, 9] are inapplicable. The algorithm described in this paper is capable of identifying such moving objects, learning models of them, and determining their locations at any point in time. It also estimates the total number of different objects in the environment, making the approach applicable to situations where not all non-stationary objects are visible at all times.

ROMA builds on the well-known occupancy grid mapping paradigm [11]. In regular time intervals, the robot acquires a static occupancy grid map [17]. Each map captures a "snapshot" of the environment at a specific point in time. Changes in the environment are detected using a straightforward map differencing technique. Our approach learns models of these objects using a modified version of the expectation maximization (EM) algorithm [4, 10], in a way similar to techniques previously developed for traffic surveillance [12]. The E-step of ROMA's EM establishes correspondence between different object sightings at different points in time. The M-step uses these probabilistic correspondences to generate refined object models, represented by occupancy grid maps. By iterating both steps, high fidelity object models are learned from multiple sightings, and the location of each individual object in each map is also determined. Since the total number of non-stationary objects may be unknown, our approach employs a model selection technique for determining the most plausible number of objects, under an exponential prior.

In our empirical evaluation, we found the ROMA algorithm to be highly reliable in identifying and localizing objects, and learning high fidelity models of them. The paper provides experimental results for two room-style environments, where a collection of natural objects is moved over time.

2 The ROMA Algorithm

2.1 Static Mapping and Map Segmentation

ROMA identifies objects that move by comparing multiple grid maps of the same environment, recorded at different points in time. At each point in time t, the robot builds a (static) occupancy grid map of its environment, denoted m_t. In a nutshell, occupancy grid maps represent robot environments by a fine-grained grid, where each grid cell carries a probability of occupancy [11]. Our implementation is based on a technique described in [17], which simultaneously localizes one or more robots during mapping.

In a preprocessing step, the ROMA algorithm decomposes the environmental model into a static occupancy grid map, and a collection of smaller occupancy grid maps, one for each non-stationary object. Non-stationary objects are identified by a map differencing technique, which builds on well-known algorithms in the field of computer vision. Our approach identifies objects by finding regions that in some of the maps are occupied, and free in others. If the occupancy of a grid cell is the same in all maps, it does not belong to a non-stationary object; instead, it is either part of a permanent free region or part of a static object such as a wall. If the occupancy varies across maps, it is potentially part of a non-stationary object in those maps where the grid cell is occupied. This map differencing technique yields a set of candidate objects. A standard low-pass computer vision filter [18] is then employed to remove noise, which is usually found on the border of free and occupied space. The result is a list of "snapshots" of non-stationary objects, each represented by a local occupancy grid map.

Let us denote the number of non-stationary objects (snapshots) found in the t-th map by K_t, and the individual objects by

$$\mu_t \quad = \quad \{\mu_{1,t}, \mu_{2,t}, \ldots, \mu_{K_t,t}\} \tag{1}$$

Here $\mu_{k,t}$ is the k-th snapshot extracted from t-th map m_t, where extracted objects are arranged in no specific order. Each snapshot $\mu_{k,t}$ is a local occupancy grid map extracted from a single occupancy grid map m_t. The set of all sets of object snapshots μ_t will be denoted

$$\mu \quad = \quad \{\mu_1, \mu_2, \ldots, \mu_T\}, \tag{2}$$

where T is the total number of available maps. The set μ is the input to the ROMA algorithm.

2.2 Models of Moving Objects

From these object snapshots, the ROMA algorithm constructs models of the non-stationary objects. Let the total number of non-stationary objects be N. The *non-stationary object model*, which refers to the set of all non-stationary objects, will be denoted

$$\theta \quad = \quad \{\theta_1, \ldots, \theta_N\}. \tag{3}$$

Each θ_n is a model of an individual non-stationary object, represented by a small occupancy grid map.

To learn θ from the snapshots μ, ROMA uses the following probabilistic model. Notice that both the models θ_n and the snapshots $\mu_{k,t}$ are represented by grid cells. Each grid cell $\theta_n[j]$ in θ_n is a real number in the interval $[0, 1]$. We interpret each occupancy value as a probability of occupancy. Since the robot scans each grid cell multiple times during mapping, we use a Gaussian distribution representing a single real-valued observation. This yields the following probability of observing $\mu_{k,t}$ given that the true underlying object is θ_n:

$$p(\mu_{k,t} \mid \theta_n, \delta_{k,t}) \;\propto\; e^{-\frac{1}{2\sigma^2}\sum_j (f(\mu_{k,t},\delta_{k,t})[j]-\theta_n[j])^2} \tag{4}$$

The function $f(\mu_{k,t}, \delta_{k,t})$ denotes the snapshot $\mu_{k,t}$ at its optimal alignment, and $f(\mu_{k,t}, \delta_{k,t})[j]$ denotes its j-th grid cell. The rotation and translation parameters of the alignment are specified by the $\delta_{k,t}$. This alignment is easily determined by search in the space of all possible alignments. The parameter σ^2 is the variance of the noise.

2.3 Expected Log Likelihood of the Data

The measurement probability $p(\mu_{k,t}|\theta_n)$ enables us to calculate the likelihood of the snapshots μ given the models θ—a necessary step for defining our maximum likelihood algorithm for finding new models θ. To do so, it will be convenient to define so-called *correspondence variables*: α_t. Each α_t specifies the correspondence between the set of snapshots μ_t, and the set of models θ. Thus,

$$\alpha_t \quad = \quad \{\alpha_{1,t}, \ldots, \alpha_{K_t,t}\} \tag{5}$$

where each correspondence variable $\alpha_{k,t}$ assigns to the k-th observed object in μ_t the index of the corresponding model θ_n. Thus,

$$\alpha_{k,t} \quad \in \quad \{1, \ldots, N\} \tag{6}$$

Of great importance is a *mutual exclusion constraint* [3, 9, 12] which specifies that the same model θ_n cannot be observed at two different locations in any of the maps m_t. This implies that for any two different snapshots k and k' we have that the correspondence variables point to different models in θ:

$$k \neq k' \implies \alpha_{k,t} \neq \alpha_{k',t} \tag{7}$$

Clearly, the correspondences α_t are latent variables, that is, they cannot be observed. Thus, the problem of identifying the maximum likelihood models θ is an optimization problem with latent variables.

We will now derive the exact likelihood function, used to maximize the joint probability over the snapshots μ, the learned occupancy grids θ and the alignment parameters δ:

$$\operatorname*{argmax}_{\theta,\delta} p(\theta, \delta, \mu) \tag{8}$$

EM starts with a random initial set of correspondences and generate a sequence of models $\theta^{[1]}, \theta^{[2]}, \ldots$ and alignment parameters $\delta^{[1]}, \delta^{[2]}, \ldots$ with non-decreasing likelihood. Let $\langle\theta^{[i]}, \delta^{[i]}\rangle$ be the i-th such set of parameters. EM find an $(i+1)$th model $\langle\theta^{[i+1]}, \delta^{[i+1]}\rangle$ for which

$$p(\theta^{[i+1]}, \delta^{[i+1]}, \mu) \quad \geq \quad p(\theta^{[i]}, \delta^{[i]}, \mu) \tag{9}$$

We achieve this goal by maximizing the expected log likelihood [10]

$$\begin{aligned} &\langle\theta^{[i+1]}, \delta^{[i+1]}\rangle \\ &= \operatorname*{argmax}_{\theta,\delta} E_\alpha \left[\log p(\alpha, \theta, \delta, \mu) \;\middle|\; \theta^{[i]}, \delta^{[i]}, \mu\right] \end{aligned} \tag{10}$$

Here E_α is the mathematical expectation over the latent correspondence variables α, relative to the distribution $p(\alpha \mid \theta^{[i]}, \delta^{[i]}, \mu)$. The probability inside the logarithm in (10) factors as follows, exploiting natural independences and assuming uniform priors over correspondences α:

$$\begin{aligned} p(\alpha, \theta, \delta, \mu) &= p(\alpha)\; p(\delta)\; p(\theta)\; p(\mu \mid \delta, \alpha, \theta) \\ &\propto p(\mu \mid \delta, \alpha, \theta) \end{aligned} \tag{11}$$

The probability $p(\mu \mid \delta, \alpha, \theta)$ of the snapshots μ given the object models θ and the correspondences α is essentially defined via (4). Here we recast it using a notation that makes the conditioning on α explicit:

$$\begin{aligned} &p(\mu \mid \delta, \alpha, \theta) \quad \propto \\ &\prod_{t=1}^{T} \prod_{k=1}^{K_t} e^{-\frac{1}{2\sigma^2} \sum_{n=1}^{N} I(\alpha_t(k)=n) \sum_j (f(\mu_{k,t}, \delta_{k,t})[j] - \theta_n[j])^2} \end{aligned} \tag{12}$$

where $I(\;)$ is an indicator function which is 1 if its argument is true, and 0 otherwise. Substituting the product (11) with (12) into the expected log likelihood (10) gives us:

$$\begin{aligned} &\langle\theta^{[i+1]}, \delta^{[i+1]}\rangle = \operatorname*{argmax}_{\theta,\delta} \\ &-\sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K_t} \frac{p(\alpha_t(k)=n \mid \Psi^{[i]}, \mu)}{\sigma^2} \sum_j (f(\mu_{k,t}, \delta_{k,t})[j] - \theta_n[j])^2 \end{aligned}$$

In deriving this expression, we exploit the linearity of the expectation, which allows us to replace the indicator variables with probabilities (expectations).

That defines the E-step of the EM algorithm. The next step is the M-step through which we generate a new set of models. The M-step requires the calculation of the most likely object models θ_n given the snapshots μ and correspondences α. Assuming constant alignment, this calculation can be carried out separately for each grid cell, exploiting the additive nature of (4). The occupancy value of model grid cell $\theta_n^{[i]}[j]$ is set to the weighted sum of the corresponding snapshot grid cells:

$$\frac{\displaystyle\sum_{t=1}^{T} \sum_{\alpha_t} p(\alpha_t | \theta^{[i-1]}, \mu) \sum_{k=1}^{K_t} d_{k,t}[j]}{\displaystyle\sum_{t=1}^{T} \sum_{\alpha_t} p(\alpha_t | \theta^{[i-1]}, \mu) K_t} \tag{13}$$

After calculating a new set of models $\theta^{[i]}$, the alignments between the models θ_n and the individual snapshots $\mu_{k,t}$ are recomputed.

One disadvantage of the formulation above is that the sum over all α_t in (13) is exponential in the number of map objects K_t. In our test environments, K_t was generally small (e.g., less than 4), in which case the full sum could easily be computed. In cases where this exponential complexity poses a serious computational burden, however, MCMC sampling techniques such as the chain flipping algorithm in [3, 12] can be adopted to lead to provably polynomial approximations of the true expectation.

2.4 Determining the Number of Objects

The ROMA algorithm outlined so far assumes knowledge of the total number of objects N. In practice, N is unknown. Bounds on N can easily be extracted from the data. In particular, N is bounded below by the maximum number of objects identified in a single map K_t, and bounded above by the total number of object snapshots:

$$\max_{t=1\ldots T} K_t \quad \leq \quad N \quad \leq \quad \sum_{t=1\ldots T} K_t \tag{14}$$

From an estimation standpoint, increasing the model capacity N increases the likelihood. Thus, maximum likelihood estimation would fail to estimate the number of objects N in any reasonable way. Our approach follows

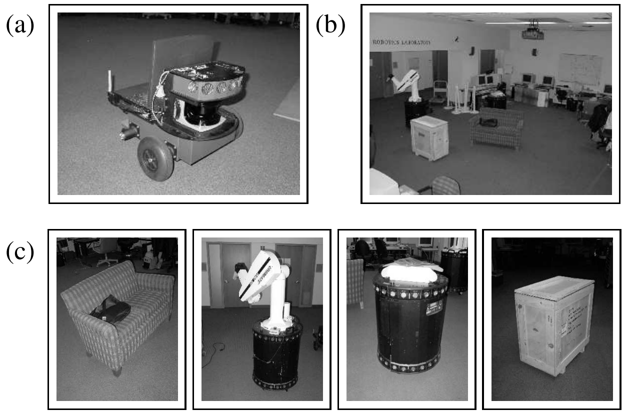

Figure 1: *(a) The Pioneer robot used to collect laser range data. (b) The robotics lab where the second data set was collected. (c) Actual images of non-stationary objects used in the second data set.*

common statistical methodology by assigning an exponential prior over N. That is, a priori we assume that large values of N are exponentially less likely:

$$p(N) \quad = \quad \text{const} \cdot e^{-pN} \tag{15}$$

where $p > 0$ is a penalty factor. The robot object mapping algorithm optimized the Bayesian *posterior*, given (in logarithmic form) by:

$$\begin{aligned} \log p(N, \theta | \mu) &= \text{const} + \log p(\mu | N, \theta) + \log p(N) \\ &= \text{const} + \log p(\mu | N, \theta) - pN \end{aligned} \tag{16}$$

where $\log p(\mu | N, \theta)$ is approximated by the expected log-likelihood (12) defined in the previous section. Put differently, our approach maximizes the expected log likelihood while simultaneously minimizing a complexity penalty term. Since N is usually small, our approach does this by running EM with fixed values of N, starting with the lower bound established in (14). When the log posterior goes down, the search is terminated, and the value of N that maximizes the log posterior is assumed to reflect the correct number of objects in the map.

3 Experimental Results

The ROMA algorithm was extensively tested in both simulated and physical environments. For brevity, we omit any simulation results and only provide real robot results. We consistently found that ROMA is able to infer the correct number of objects, and to learn models that are more accurate than the snapshots extracted from a single occupancy grid map—as long as the objects were sufficiently apart from each other that they were segmented correctly in the preprocessing stage. The correspondence estimates were accurate when all objects looked different. When multiple objects of the same shape were present, the correspondence estimates were split accordingly.

In the following sections, we cover our results for data collected from two real-world room-style environments.

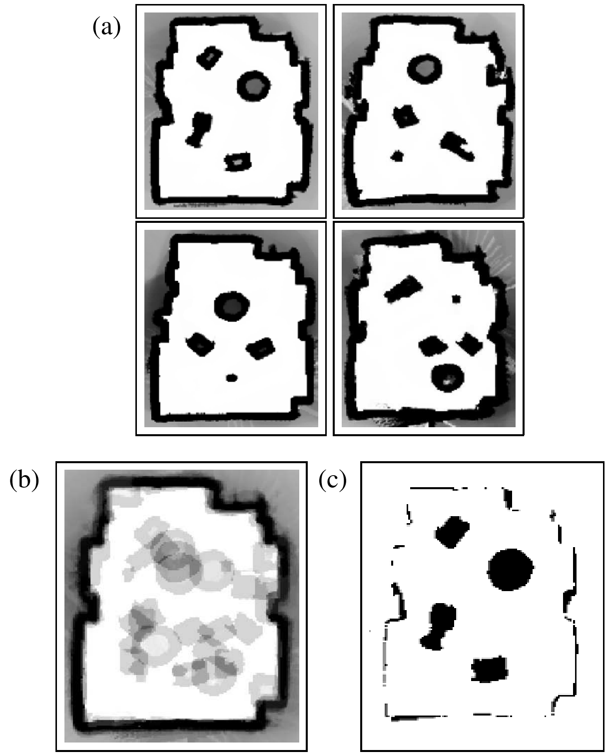

Figure 2: *(a) Four maps used for learning models of non-stationary objects using a* fixed *number of objects per map. (b) Overlay of optimally aligned maps. (c) Difference map* before *low-pass filtering.*

The laser range data used for mapping was collected with the Pioneer robot shown in Figure 1a. In the first data set, we collected maps with a fixed number of objects per map which are shown in Figure 2a. In the second data set we collected maps from the robotics lab shown in Figure 1b. These maps used a variable number of non-stationary objects per map; actual photos of the four objects used in these maps are shown in Figure 1c. The collected maps for this data set are shown in Figure 3a.

3.1 Map Segmentation and Object Extraction

The object snapshot extraction worked very reliably. Figures 2a and 3a show the maps used for learning in the two data sets. An overlay of these maps for each of the respective data sets is shown in Figures 2b and 3b. Results from image differencing with the overlay are shown for the respective data sets in Figures 2c and 3c. Once the differenced maps are produced, they are run through a low-pass noise filter [18]. After filtering, each object of sufficient size is extracted into its own occupancy grid map. For the given data sets, this final step worked flawlessly, extracting exactly the number of expected non-stationary objects for each of the respective static maps.

3.2 ROMA Applied to a Fixed Number of Objects

The first set of results that we provide assumes a fixed number of objects and uses the map data shown in Figure 2a. Figure 4a shows successive EM iterations of the

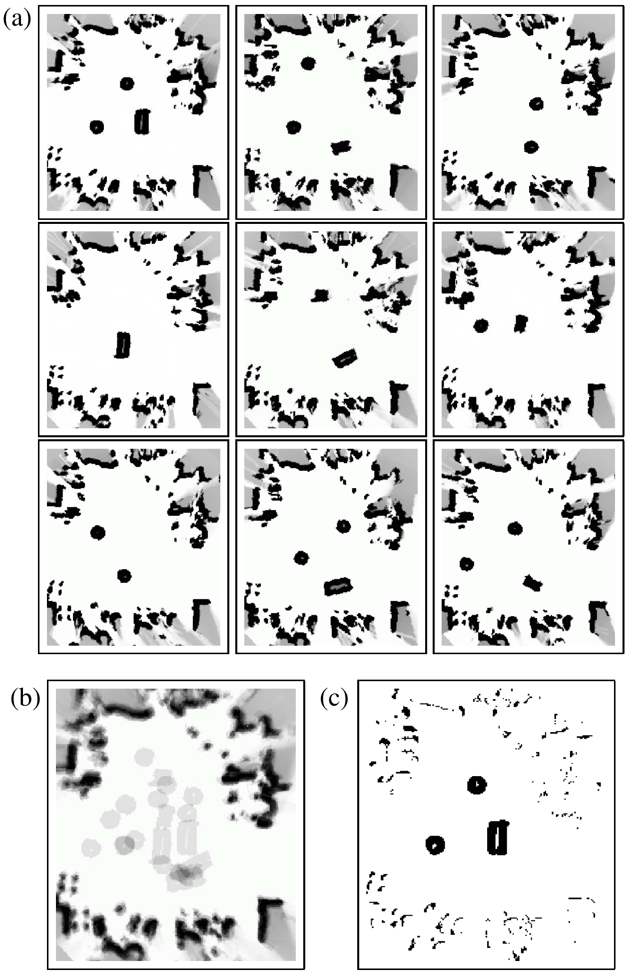

Figure 3: *(a) Nine maps used for learning models of non-stationary objects using a* variable *number of objects per map. (b) Overlay of optimally aligned maps. (c) Difference map* before *low-pass filtering. The objects are clearly identifiable.*

ROMA algorithm starting from an initial random models (unshown). On each successive iteration of the EM algorithm we note that the models resemble the objects in the original maps with increasingly higher fidelity and that the final set of objects clearly represents a fairly accurate representation of the four objects in the original maps. Furthermore, the final maximum likelihood correspondences perfectly match the objects in the original maps with the objects in the final iteration models.

3.3 ROMA Applied to a Variable Number of Objects

The second set of results that we provide allows a variable number of objects per map and uses the map data shown in Figure 3a. This algorithm uses the extension previously described for determining the number of objects in the model (Equations 14-16). Since the entire ROMA algorithm has to be run once for each hypothesized number of objects, we can compute the final iteration model score (i.e. Bayesian *posterior*) of each algorithm run. This score is the log of the model likelihood minus the complexity penalty as given in (16). Figure 5 shows the model score for a varying number of model objects for the current data set. Note that for a complexity penalty coefficient of $p = 120.0$ this graph peaks for $N = 4$ objects which is in fact the actual number of different objects in the original set of maps.

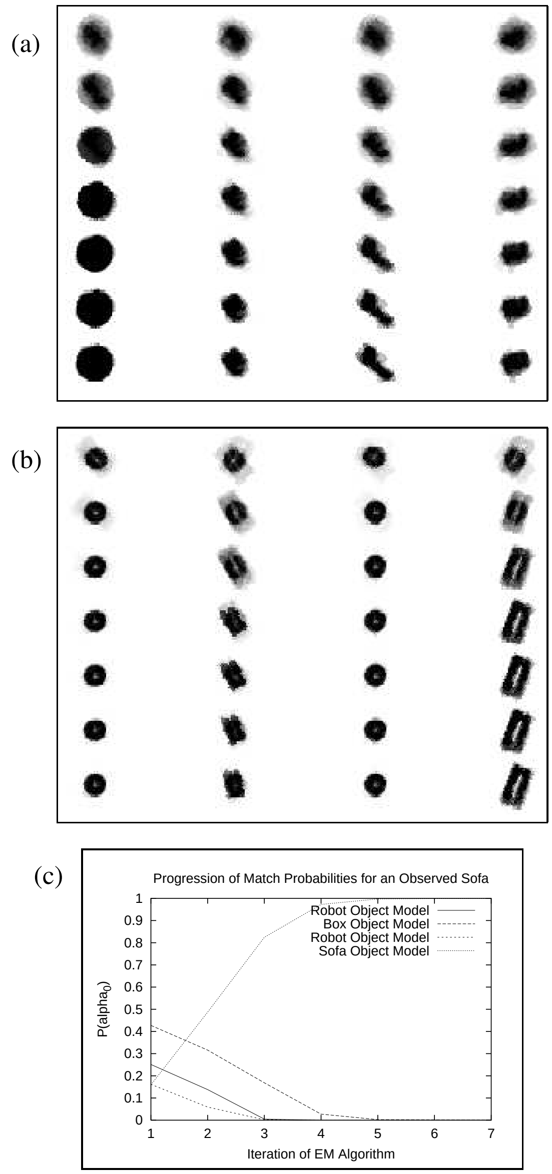

Figure 4: *(a) Seven iterations of EM for the data set containing a fixed number of objects per map. (b) Seven iterations of EM for the data set containing a variable number of objects per map. (c) Correspondence probabilities between an observed object and different object models.*

Figure 4b shows successive EM iterations for the data set in Figure 3 under the maximal Bayesian posterior estimate of $N = 4$ objects. The correspondences between a sample observed object and the different models is shown in Figure 4c. While the correspondences are initially randomly distributed, the observed object quickly establishes

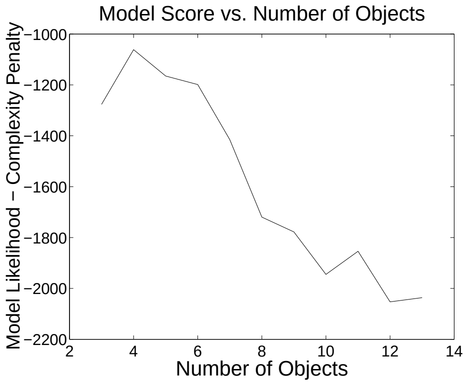

Figure 5: *Graph of model score vs. number of objects.*

a strong correspondence to the correct model as EM progresses. Moreover, on each successive iteration, it is clear that the object models more closely reflect the objects in the original maps. Additionally, under the maximal model score hypothesis of $N = 4$ objects, the final maximum likelihood correspondences perfectly match the objects in the original maps with the objects in the final iteration models.

4 Conclusion

The paper proposed an occupancy grid mapping algorithm for non-stationary environments, where objects may change their locations over time. In a preprocessing stage, the algorithm extracts sets of non-stationary object "snapshots" from a collection of occupancy grid maps, recorded at different points in time. The EM algorithm is applied to learn object models of the individual non-stationary objects in the world, represented as local occupancy grid maps. The number of objects is estimated as well. Experimental results presented in this paper demonstrate the robustness of the approach. In simulated and real-world setting, we consistently found that high-fidelity object models were learned from multiple sightings of the same object at different locations.

In its present state, the ROMA algorithm possesses a range of limitations which warrant future research. First, objects have to move slowly enough that they are captured as static objects in each occupancy grid map. This precludes the inclusion of fast-moving people in the map. Second, it would be desirable to develop a hierarchy of objects, paying tribute to the fact that many objects may look alike (e.g., chairs; see [1]). Finally, we believe that the same techniques can be applied to more advanced representation than occupancy grid maps (e.g. integrating multi-modal sensor input from camera images, etc…). However, such an extension is subject to future research.

Acknowledgments

This work was supported by DARPA's MARS Program (Contract number N66001-01-C-6018) and the National Science Foundation (CAREER grant number IIS-9876136 and regular grant number IIS-9877033). It was completed while Sebastian Thrun was visiting Stanford University.

References

[1] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. *Proc. UAI-2002*.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.

[3] F. Dellaert, S.M. Seitz, C. Thorpe, and S. Thrun. "EM, MCMC, and chain flipping for structure from motion with unknown correspondence". *Machine Learning*, forthcoming.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[5] J. Guivant and E. Nebot. "Optimization of the simultaneous localization and map building algorithm for real time implementation". *IEEE Transaction of Robotic and Automation*, May 2001.

[6] L. Iocchi, K. Konolige, and M. Bajracharya. "Visually realistic mapping of a planar environment with stereo". *Proc. ISER-2000*.

[7] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *Artificial Intelligence and Mobile Robots: Case Studies of successful robot systems*. MIT Press, 1998.

[8] J.J. Leonard and H.J.S. Feder. "A computationally efficient method for large-scale concurrent mapping and localization". *Proc. ISRR-1999*.

[9] J. MacCormick and A. Blake. "A probabilistic exclusion principle for tracking multiple objects". *Proc. ICCV-1999*.

[10] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.

[11] H. P. Moravec. "Sensor fusion in certainty grids for mobile robots". *AI Magazine*, 9(2):61–74, 1988.

[12] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. "Tracking many objects with many sensors". *Proc. IJCAI-1999*.

[13] F. E. Schneider. "Sensorinterpretation und Kartenerstellung für mobile Roboter". Master's Thesis. Dept. of Computer Science III, University of Bonn, 1994. In German.

[14] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. "Coordination for multi-robot exploration and mapping". *Proc. AAAI-2000*.

[15] R. Smith, M. Self, and P. Cheeseman. "Estimating uncertain spatial relationships in robotics". In *Autonomous Robot Vehicles*, 167–193. Springer, 1990.

[16] C. Thorpe and H. Durrant-Whyte. "Field robots". *Proc. ISRR-2001*.

[17] S. Thrun. "A probabilistic online mapping algorithm for teams of mobile robots". *International Journal of Robotics Research*, 20(5):335–363, 2001.

[18] S.W. Zucker. "Region growing: Childhood and adolescence". *Comput. Graphics Image Processing*, 5:382–399, 1976.