Simultaneous Localization, Mapping and Moving Object Tracking

Chieh-Chih Wang
Department of Computer Science and Information Engineering and
Graduate Institute of Networking and Multimedia
National Taiwan University
Taipei 106, Taiwan
bobwang@ntu.edu.tw

Charles Thorpe, Martial Hebert
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
cet@ri.cmu.edu, hebert@ri.cmu.edu

Sebastian Thrun
The AI group
Stanford University
Stanford, CA 94305, USA
thrun@stanford.edu

Hugh Durrant-Whyte
The ARC Centre of Excellence for Autonomous Systems
The University of Sydney
NSW 2006, Australia
hugh@acfr.usyd.edu.au

Abstract

Simultaneous localization, mapping and moving object tracking (SLAMMOT) involves both simultaneous localization and mapping (SLAM) in dynamic environments and detecting and tracking these dynamic objects. In this paper, we establish a mathematical framework to integrate SLAM and moving object tracking. We describe two solutions: SLAM with generalized objects, and SLAM with detection and tracking of moving objects (DATMO). SLAM with generalized objects calculates a joint posterior over all generalized objects and the robot. Such an approach is similar to existing SLAM algorithms, but with additional structure to allow for motion modeling of generalized objects. Unfortunately, it is computationally demanding and generally infeasible. SLAM with DATMO decomposes the estimation problem into two separate estimators. By maintaining separate posteriors for stationary objects and moving objects, the resulting estimation problems are much lower dimensional then SLAM with generalized objects. Both SLAM and moving object tracking from a moving vehicle in crowded urban areas are daunting tasks. Based on the SLAM with DATMO framework, we propose practical algorithms which deal with issues of perception modeling, data association, and moving object

detection. The implementation of SLAM with DATMO was demonstrated using data collected from the CMU Navlab11 vehicle at high speeds in crowded urban environments. Ample experimental results shows the feasibility of the proposed theory and algorithms.

1 Introduction

Establishing the spatial and temporal relationships among a robot, stationary objects and moving objects in a scene serves as a basis for scene understanding. *Localization* is the process of establishing the spatial relationships between the robot and stationary objects, *mapping* is the process of establishing the spatial relationships among stationary objects, and *moving object tracking* is the process of establishing the spatial and temporal relationships between moving objects and the robot or between moving and stationary objects. Localization, mapping and moving object tracking are difficult because of *uncertainty* and *unobservable states* in the real world. Perception sensors such as cameras, radar and laser range finders, and motion sensors such as odometry and inertial measurement units are noisy. The intentions, or control inputs, of the moving objects are unobservable without using extra sensors mounted on the moving objects.

Over the last decade, the simultaneous localization and mapping (SLAM) problem has attracted immense attention in the mobile robotics and artificial intelligence literature (Smith and Cheeseman, 1986; Thrun, 2002). SLAM involves simultaneously estimating locations of newly perceived landmarks and the location of the robot itself while incrementally building a map. The moving object tracking problem has also been extensively studied for several decades (Bar-Shalom and Li, 1988; Blackman and Popoli, 1999). Moving object tracking involves both state inference and motion model learning. In most applications, SLAM and moving object tracking are considered in isolation. In the SLAM problem, information associated with stationary objects are positive; moving objects are negative, which degrades the performance. Conversely, measurements belonging to moving objects are positive in the moving object tracking problem; stationary objects are considered background and filtered out. In (Wang and Thorpe, 2002), we pointed out that SLAM and moving object tracking are mutually beneficial. Both stationary objects and moving objects are positive information to scene understanding. In (Wang et al., 2003b), we established a mathematical framework to integrate SLAM and moving object tracking, which provides a solid basis for understanding and solving the whole problem, simultaneous localization, mapping and moving object tracking, or SLAMMOT.

It is believed by many that a solution to the SLAM problem will open up a vast range of potential applications for autonomous robots (Thorpe and Durrant-Whyte, 2001; Christensen, 2002). We believe that a solution to the SLAMMOT problem will expand the potential for robotic applications still further, especially in applications which are in close proximity to human beings. Robots will be able to work not only *for* people but also *with* people. Figure 1 illustrates a commercial application, safe driving, which motivates the work in this paper.

To improve driving safety and prevent traffic injuries caused by human factors such as

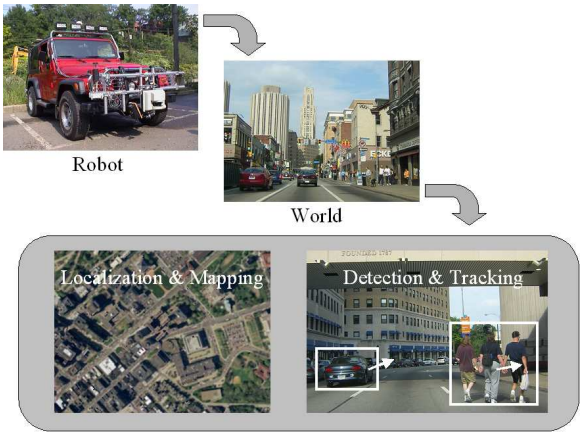


Figure 1: Robotics for safe driving. Localization, mapping, and moving object tracking are critical to driving assistance and autonomous driving.

speeding and distraction, methods for understanding the surroundings of the vehicle are critical. We believe that being able to detect and track every stationary object and every moving object, to reason about the dynamic traffic scene, to detect and predict critical situations, and to warn and assist drivers in advance, is essential to prevent these kinds of accidents.

To detect and track moving objects using sensors mounted on a moving ground vehicle at high speeds, a precise localization system is essential. It is known that GPS and DGPS often fails in urban areas because of "urban canyon" effects, and good inertial measurement units (IMU) are very expensive.

If we have a stationary object map in advance, map-based localization techniques (Olson, 2000)(Fox et al., 1999)(Dellaert et al., 1999) can be used to increase the accuracy of the pose estimate. Unfortunately, it is difficult to build a usable stationary object map because of temporary stationary objects such as parked cars. Stationary object maps of the same scene built at different times could still be different, which means that online map building is required to update the current stationary object map.

SLAM allows robots to operate in an unknown environment, to incrementally build a map of this environment and to simultaneously use this map to localize the robots themselves. However, we have observed (Wang and Thorpe, 2002) that SLAM can perform badly in crowded urban environments because the static environment assumption may be violated; moving objects have to be detected and filtered out.

Even with precise localization, it is not easy to solve the moving object tracking problem in crowded urban environments because of the wide variety of targets (Wang et al., 2003a). When cameras are used to detect moving objects, appearance-based approaches are widely used and moving objects should be detected no matter whether they are moving or not. If laser scanners are used, feature-based approaches are usually the preferred solution. Both appearance-based and feature-based methods rely on prior knowledge of the targets. In urban areas, there are many kinds of moving objects such as pedestrians, bicycles, motorcycles,

cars, buses, trucks and trailers. Velocities range from under 5 mph (such as a pedestrian's movement) to 50 mph. When using laser scanners, the features of moving objects can change significantly from scan to scan. As a result, it is often difficult to define features or appearances for detecting specific objects.

Both SLAM and moving object tracking have been solved and implemented successfully in isolation. However, when driving in crowded urban environments composed of stationary and moving objects, neither of them is sufficient in isolation. The SLAMMOT problem aims to tackle the SLAM problem and the moving object tracking problem concurrently. SLAM provides more accurate pose estimates together with a surrounding map. Moving objects can be detected using the surrounding map without recourse to predefined features or appearances. Tracking may then be performed reliably with accurate robot pose estimates. SLAM can be more accurate because moving objects are filtered out of the SLAM process thanks to the moving object location prediction. SLAM and moving object tracking are mutually beneficial. Integrating SLAM and moving object tracking would satisfy both the *safety* and *navigation* demands of safe driving. It would provide a better estimate of the robot's location and information of the dynamic environments, which are critical to driving assistance and autonomous driving.

In this paper we first establish the mathematical framework for performing SLAMMOT. We will describe two algorithms, SLAM with generalized objects and SLAM with detection and tracking of moving object (DATMO). SLAM with DATMO decomposes the estimation problem of SLAMMOT into two separate estimators. By maintaining separate posteriors for stationary objects and moving objects, the resulting estimation problems are much lower dimensional than SLAM with generalized objects. This makes it feasible to update both filters in real-time. In this paper, SLAM with DATMO is applied and implemented.

There are significant practical issues to be considered in bridging the gap between the presented theory and its applications to real problems such as driving safely at high speeds in crowded urban areas. These issues arise from a number of implicit assumptions in perception modeling and data association. When using more accurate sensors, these problem are easier to solve, and inference and learning of the SLAM with DATMO problem become more practical and tractable. Therefore, we mainly focus on issues of using active ranging sensors. SICK laser scanners (see Figure 2) are used and studied in this work. Data sets (Wang et al., 2004) collected from the Navlab11 testbed are used to verify the theory and algorithms. Visual images from an omni-directional camera and a tri-camera system are only used for visualization. Sensors carrying global localization information such as GPS and DGPS are *not* used.

The remainder of this paper is organized as follows. In Section 2, related research is addressed. The formulations and algorithms of SLAM and moving object tracking are briefly reviewed in Section 3. In Section 4, we introduce SLAM with generalized object and SLAM with DATMO and discuss some of important issues such as motion mode learning, computational complexity and interaction. The proposed algorithms which deal with issues of perception modeling, data association and moving object detection are described in Section 5, Section 6 and Section 7, respectively. Experimental results which demonstrated the feasibility of the proposed theory and algorithms are in Section 8. Finally, the conclusion and

Figure 2: Right: the Navlab11 testbed. Left: SICK LMS221, SICK LMS291 and the tri-camera system.

future work are in Section 9.

2 Related Work

The SLAMMOT problem is directly related to a rich body of the literature on SLAM and tracking.

Dirk Hähnel et al. (Hähnel et al., 2002) presented an online algorithm to incorporate people tracking into the mapping process with a mobile robot in populated environments. The local minima in the scans are used as the features for people detection. The sampling-based joint probabilistic data association filters are used for tracking people. A hill climbing strategy is used for scan alignment. Their work performs well in indoor environments populated with moving people. However, feature-based detection may fail in environments where a wide variety of moving objects exist. In (Hähnel et al., 2003), the EM algorithm is used for segmenting stationary and moving object without defining features. The technique was tested with data collected in indoor and outdoor environments. However this is an off-line algorithm which is not suitable for real-time applications.

The approach in (Biswas et al., 2002)(Anguelov et al., 2002) uses simple differencing to segment temporary-stationary objects, and then learn their shape models and identify classes of these objects using a modified version of the EM algorithm. In (Anguelov et al., 2004), a predefined probabilistic model consisting of visual features (shape and color) and behavioral features (its motion model) is learned using the EM algorithm and is used for recognizing door objects in corridor environments. Although recognition could improve the performance of SLAMMOT and provide higher level scene understanding, these off-line algorithms are not feasible for real-time applications and urban areas contain richer and more complicated objects in which recognition is still a hard problem both theoretically and practically.

Wolf and Sukhatme (Wolf and Sukhatme, 2005) proposed to use two modified grid occupancy maps to classify static and moving objects, which is similar to our consistency-based moving object detection algorithm (Wang and Thorpe, 2002). The third map containing static corner features is used for localization. Without dealing with the moving object motion modeling issues and without moving object pose prediction capability, their approach would be less

robust than the proposed approach in this paper. Montesano et al. (Montesano et al., 2005) integrated the SLAMMOT and planning processes to improve robot navigation in dynamic environments as well as extended our SLAM with DATMO algorithm to jointly solve the moving and stationary object classification problem in an indoor environment. In this paper, the proposed theory of SLAM with generalized objects and SLAM with DATMO address more related issues such as interaction, and the proposed algorithms are demonstrated from a ground vehicle at high speeds in crowded urban environments.

The SLAMMOT problem is also closely related to the computer vision literature. Demirdjian and Horaud (Demirdjian and Horaud, 2000) addressed the problem of segmenting the observed scene into static and moving objects from a moving stereo rig. They apply a robust method, random sample consensus (RANSAC), for filtering out the moving objects and outliers. Although the RANSAC method can tolerate up to 50% outliers, the percentage of moving objects is often more than stationary objects and degeneracies exist in our applications. With measurements from motion sensors, stationary and moving object maps, and precise localization, our moving object detectors perform reliably in real-time.

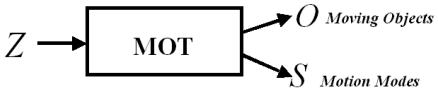
Recently, the problem of recovery non-rigid shape and motion of dynamic scenes from a moving camera has attracted immense attention in the computer vision literature. The ideas based on the factorization techniques and the shape basis representation are presented in (Bregler et al., 2000)(Brand, 2001)(Torresani et al., 2001)(Xiao et al., 2004) where different constraints are used for finding the solution. Theoretically, all of these batch approaches are inappropriate to use in real-time. Practically, these methods are computational demanding and many difficulties such as occlusions, motion blur and lighting conditions remain to be developed.

3 Foundations

SLAM assumes that the surrounding environment is static, containing only stationary objects. The inputs of the SLAM process are measurements from *perception sensors* such as laser scanners and cameras, and measurements from *motion sensors* such as odometry and inertial measurement units. The outputs of the SLAM process are the robot pose and a stationary object map (see Figure 3.a). Given that the sensor platform is stationary or that a precise pose estimate is available, the inputs of the moving object tracking problem are perception measurements and the outputs are locations of moving objects and their motion modes (see Figure 3.b). The SLAMMOT problem can also be treated as a process *without* the static environment assumption. The inputs of this process are the same as for the SLAM process, but the outputs are both robot pose and map, together with the locations and motion modes of the moving objects (see Figure 3.c).

Leaving aside perception and data association, the key issue in SLAM is the computational complexity of updating and maintaining the map, and the key issue of the moving object tracking problem is the computational complexity of motion modelling. As SLAMMOT inherits the complexity issue from SLAM and the motion modelling issue from moving object tracking, the SLAMMOT problem is both an *inference* problem and a *learning* problem.

(a) the simultaneous localization and mapping (SLAM) process



(b) the moving object tracking (MOT) process

(c) the simultaneous localization, mapping and moving object tracking (SLAMMOT) process

Figure 3: The SLAM process, the MOT process and the SLAMMOT process. Z denotes the perception measurements, U denotes the motion measurements, x is the true robot state, M denotes the locations of the stationary objects, O denotes the states of the moving objects and S denotes the motion modes of the moving objects.

In this section we briefly introduce SLAM and moving object tracking.

3.1 Notation

Let k denote the discrete time index, u_k the vector describing a motion measurement from time $k-1$ to time k, z_k a measurement from perception sensors such as laser scanners at time k, x_k the state vector describing the true pose of the robot at time k, and M_k the map contain l landmarks, $m^1, m^2, \ldots, m^l$, at time k. In addition, we define the following sets:

$$X_k \triangleq \{x_0, x_1, \ldots, x_k\} \tag{1}$$

$$Z_k \triangleq \{z_0, z_1, \ldots, z_k\} \tag{2}$$

$$U_k \triangleq \{u_1, u_2, \ldots, u_k\} \tag{3}$$

3.2 Simultaneous Localization and Mapping

The SLAM problem is to determine the robot poses x_k and the stationary object map M_k given perception measurements Z_k and motion measurement U_k.

The formula for sequential SLAM can be expressed as

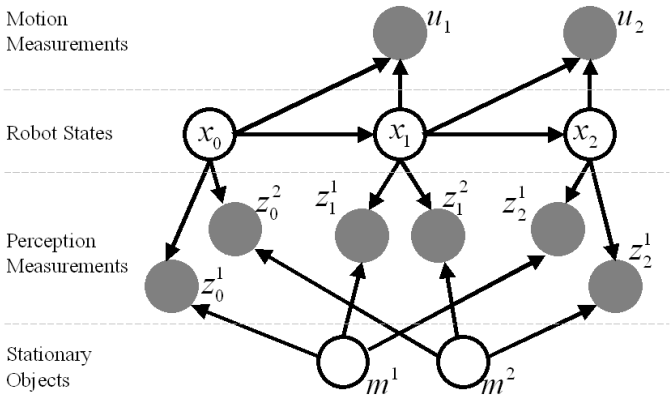


Figure 4: A Dynamic Bayesian Network (DBN) of the SLAM problem of duration three. It shows the dependencies among the motion measurements, the robot, the perception measurements and the stationary objects. In this example, there are two stationary objects, m^1 and m^2. Clear circles denote hidden continuous nodes and shaded circles denote observed continuous nodes. The edges from stationary objects to measurements are determined by data association.

$$p(x_k, M_k \mid U_k, Z_k) \cdot \tag{4}$$

Using Bayes' rule and assumptions that the vehicle motion model is Markov and the environment is static, the general recursive Bayesian formula for SLAM can be derived and expressed as (See (Thrun, 2002; Durrant-Whyte et al., 2003) for more details.)

$$\underbrace{p(x_k, M_k \mid Z_k, U_k)}_{\text{Posterior at } k} \propto \underbrace{p(z_k \mid x_k, M_k)}_{\text{Perception model}} \cdot \underbrace{\int p(x_k \mid x_{k-1}, u_k) \underbrace{p(x_{k-1}, M_{k-1} \mid Z_{k-1}, U_{k-1})}_{\text{Posterior at } k-1} dx_{k-1}}_{\text{Prediction}} \tag{5}$$

where $p(x_{k-1}, M_{k-1} \mid Z_{k-1}, U_{k-1})$ is the posterior probability at time $k-1$, $p(x_k, M_k \mid Z_k, U_k)$ is the posterior probability at time k, $p(x_k \mid x_{k-1}, u_k)$ is the motion model, and $p(z_k \mid x_k, M)$ is the stage describing the perception model. The motion model is calculated according to robot kinematics/dynamics. The perception model can be represented using different ways such as features/landmarks and occupancy-grids.

Equation 5 explains the computation procedures in each time step. Figure 4 shows a Dynamic Bayesian Network (DBN) for SLAM over three time-steps, which can be used to visualize the dependencies between the robot and stationary objects (Paskin, 2003).

The extended Kalman filter (EKF)-based solution (Smith and Cheeseman, 1986; Smith et al., 1990; Leonard and Durrant-Whyte, 1991) to the SLAM problem is elegant, but is

computational complex. Approaches using approximate inference, using exact inference on tractable approximations of the true model, and using approximate inference on an approximate model have been proposed (Paskin, 2003; Thrun et al., 2002; Bosse et al., 2003; Guivant and Nebot, 2001; Leonard and Feder, 1999; Montemerlo, 2003). Paskin (Paskin, 2003) included an excellent comparison of these techniques.

3.3 Moving Object Tracking

Just as with SLAM, moving object tracking can be formulated with Bayesian approaches such as Kalman filtering. Moving object tracking is generally easier than SLAM since only the moving object pose is maintained and updated. However, as motion models of moving objects are often time-varying and not known with accuracy, moving object tracking is more difficult than SLAM in terms of online motion model learning.

The general recursive probabilistic formula for moving object tracking can be expressed as

$$p(o_k, s_k \mid Z_k) \tag{6}$$

where o_k is the true state of a moving object at time k, and s_k is the *true motion mode* of the moving object at time k, and Z_k is the perception measurement set leading up to time k. The robot (sensor platform) is assumed to be stationary for the sake of simplicity.

Using Bayes' rule, Equation 6 can be rewritten as

$$p(o_k, s_k \mid Z_k) = \underbrace{p(o_k \mid s_k, Z_k)}_{\text{State inference}} \cdot \underbrace{p(s_k \mid Z_k)}_{\text{Mode learning}} \tag{7}$$

which indicates that the moving object tracking problem can be solved in two stages: the first stage is the *mode learning* stage $p(s_k \mid Z_k)$, and the second stage is the *state inference* stage $p(o_k \mid s_k, Z_k)$.

Without *a priori* information, online mode learning of time-series data is a daunting task. In practice, the motion mode of moving objects can be approximately composed of several motion models such as the constant velocity model, the constant acceleration model and the turning model. Therefore the mode learning problem can be simplified to a *model selection* problem. Figure 5 shows a DBN for multiple model-based moving object tracking.

Multiple model-based moving object tracking is still difficult though because the motion mode of moving objects can be time-varying. The only way to avoid the exponentially increasing number of histories is to use approximate and suboptimal approaches which merge or reduce the number of the mode history hypotheses in order to make computation tractable.

Generalized Pseudo-Bayesian (GPB) approaches (Tugnait, 1982) apply a simple suboptimal technique which keeps the histories of the target mode with the largest probabilities, discards the rest, and renormalizes the probabilities. In the GPB approaches of first order (GPB1), the state estimate at time k is computed under each possible *current* model. At the end of each cycle, the r motion mode hypotheses are merged into a single hypothesis. The GPB1 approach uses r filters to produce one state estimate. In the GPB approaches of second order (GPB2), the state estimate is computed under each possible model at *current* time

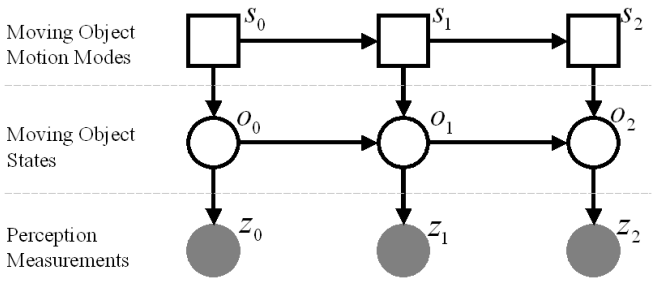


Figure 5: A DBN for multiple model based moving object tracking. Clear circles denote hidden continuous nodes, clear squares denotes hidden discrete nodes and shaded circles denotes continuous nodes.

k and *previous* time $k-1$. There are r estimates and covariances at time $k-1$. Each is predicted to time k and updated at time k under r hypotheses. After the update stage, the r^2 hypotheses are merged into r at the end of each estimation cycle. The GPB2 approach uses r^2 filters to produce r state estimates.

In the interacting multiple model (IMM) approach (Blom and Bar-Shalom, 1988), the state estimate at time k is computed under each possible current model using r filters and each filter uses a suitable mixing of the previous model-conditioned estimate as the initial condition. It has been shown that the IMM approach performs significantly better than the GPB1 algorithm and almost as well as the GPB2 algorithm in practice. Instead of using r^2 filters to produce r state estimates in GPB2, the IMM uses only r filters to produce r state estimates.

In both GPB and IMM approaches, it is assumed that a model set is given or selected in advance, and tracking is performed based on model averaging of this model set. The performance of moving object tracking strongly depends on the selected motion models. Given the same data set, the tracking results differ according to the selected motion models.

4 SLAMMOT

In the previous section, we have briefly described the SLAM and moving object tracking problems. In this section, we address the approaches to concurrently solve the SLAM and moving object tracking problems, SLAM with generalized objects and SLAM with DATMO.

4.1 SLAM with Generalized Objects

Without making any hard decisions about whether an object is stationary or moving, the SLAMMOT problem can be handled by calculating a joint posterior over all objects (robot pose, stationary objects, moving objects). Such an approach would be similar to existing SLAM algorithms, but with additional structure to allow for motion mode learning of the generalized objects.

4.1.1 Bayesian Formulation

The formalization of SLAM with generalized objects is straightforward. First we define that the generalized object is a hybrid state consisting of the state and the motion mode.

$$\mathbf{y}_k^i \triangleq \{y_k^i, s_k^i\} \qquad and \qquad \mathbf{Y}_k \triangleq \{\mathbf{y}_k^1, \mathbf{y}_k^2, \ldots, \mathbf{y}_k^l\} \tag{8}$$

where y_k is the true state of the generalized object, s_k is the true motion mode of the generalized object and l is the number of generalized objects. Note that generalized objects can be moving, stationary, or move-stop-move entities. We then use this hybrid variable $\mathbf{Y}$ to replace the variable M in Equation 5 and the general recursive probabilistic formula of SLAM with generalized objects can be expressed as:

$$p(x_k, \mathbf{Y}_k \mid U_k, Z_k) \tag{9}$$

Using Bayes' rules and assumptions that the motion models of the robot and generalized objects are Markov and there is *no interaction* among the robot and generalized objects, the general recursive Bayesian formula for SLAM with generalized objects can be derived and expresses as: (See Appendix A for derivation.)

$$\begin{aligned} &\underbrace{p(x_k, \mathbf{Y}_k \mid U_k, Z_k)}_{\text{Posterior at } k} \\ &\propto \underbrace{p(z_k \mid x_k, \mathbf{Y}_k)}_{\text{Update}} \int\int \underbrace{p(x_k \mid x_{k-1}, u_k)}_{\text{Robot predict}} \underbrace{p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1})}_{\text{generalized objs}} \\ &\quad \cdot \underbrace{p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1})}_{\text{Posterior at } k-1} dx_{k-1}\, d\mathbf{Y}_{k-1} \end{aligned} \tag{10}$$

where $p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1})$ is the posterior probability at time $k-1$, $p(x_k, \mathbf{Y}_k \mid U_k, Z_k)$ is the posterior probability at time k. In the prediction stage $p(x_k \mid x_{k-1}, u_k)p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1})$, the states of the robot state and generalized objects are predicted independently with the no interaction assumption. In the update stage $p(z_k \mid x_k, \mathbf{Y}_k)$, the states of the robot and generalized objects as well as their *motion models* are updated concurrently. In the cases that interactions among the robot and generalized objects exist, the formula for SLAM with generalized objects is also shown in Appendix A.

Figure 6 shows a DBN representing the SLAM with generalized objects of duration three with two generalized objects, which integrates the DBNs of the SLAM problem and the moving object tracking problem.

4.1.2 Motion Modeling/Motion Mode Learning

Motion modeling of generalized objects is critical for SLAM with generalized objects. A general mechanism for solving motion modeling of stationary, moving objects and objects between stationary and moving has to be developed.

The IMM algorithm and its variants (Mazor et al., 1998) have been successfully implemented in many tracking applications for dealing with the moving object motion modeling problem because of their low computational cost and satisfactory performance.

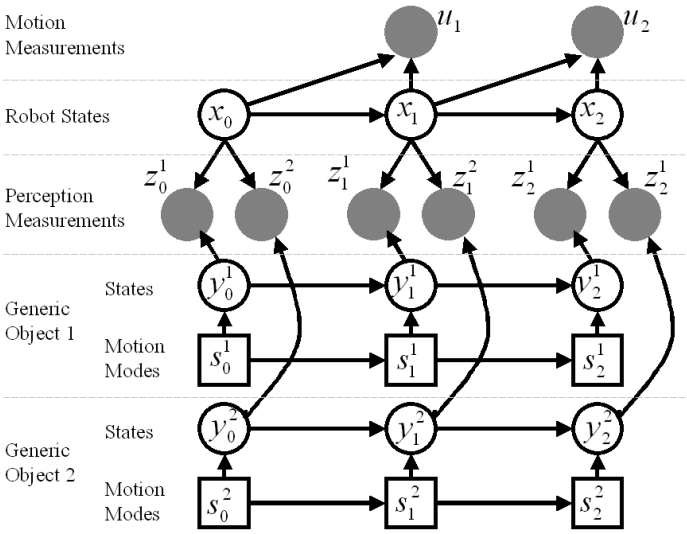


Figure 6: A DBN for SLAM with Generalized Objects. It is an integration of the DBN of the SLAM problem (Figure 4) and the DBN of the moving object tracking problem (Figure 5).

Adding a stationary motion model to the motion model set with the same IMM algorithm for dealing with move-stop-move maneuvers was suggested in (Kirubarajan and Bar-Shalom, 2000; Coraluppi et al., 2000; Coraluppi and Carthel, 2001). However, as observed in (Shea et al., 2000; Coraluppi and Carthel, 2001), all of the estimates tend to *degrade* when the stop (stationary motion) model is added to the model set and mixed with other moving motion models. This topic is beyond the scope intended by this paper. We provide a theoretical explanation of this phenomenon and a practical solution, move-stop hypothesis tracking, in Chapter 4 of (Wang, 2004).

4.1.3 Highly Maneuverable Objects

The framework of SLAM with generalized objects indicates that measurements belonging to *moving* objects contribute to localization and mapping as well as stationary objects. Nevertheless, highly maneuverable objects are difficult to track and often unpredictable in practice. Including them in localization and mapping would have a minimal effect on localization accuracy.

4.1.4 Computational Complexity

In the SLAM literature, it is known that a key bottleneck of the Kalman filter solution is its computational complexity. Because it explicitly represents correlations of all pairs among the robot and stationary objects, both the computation time and memory requirement scale quadratically with the number of stationary objects in the map. This computational burden restricts applications to those in which the map can have no more than a few hundred stationary objects. Recently, this problem has been subject to intense research.

In the framework of SLAM with generalized objects, the robot, stationary and moving objects are generally correlated through the convolution process in the prediction and update stages. Although the formulation of SLAM with generalized objects is elegant, it is clear that SLAM with generalized objects is much more computationally demanding than SLAM due to the required motion modeling of all generalized objects at all time steps. Given that real-time motion modeling of generalized objects and interaction among moving and stationary objects are still open questions, the computational complexity of SLAM with generalized objects is not further analyzed.

4.2 SLAM with DATMO

SLAM with generalized objects is similar to existing SLAM algorithms, but with additional structure to allow for motion modelling of generalized objects. Unfortunately, it is computationally demanding and generally infeasible. Consequently, in this section we provide the second approach, SLAM with DATMO, in which the estimation problem is decomposed into two separate estimators. By maintaining separate posteriors for stationary objects and moving objects, the resulting estimation problem is of lower dimension than SLAM with generalized objects, making it feasible to update both filters in real time.

4.2.1 Bayesian Formulation

Let $\mathbf{o}_k$ denote the true hybrid state of the moving object at time k.

$$\mathbf{o}_k^i \triangleq \{o_k^i, s_k^i\} \tag{11}$$

where o_k is the true state of the moving object, s_k is the true motion mode of the moving object.

In SLAM with DATMO, three assumption are made to simply the computation of SLAM with generalized objects. One of the key assumptions is that measurements can be decomposed into measurement of stationary and moving objects. This implies that objects can be classified as stationary or moving in which the general SLAM with DATMO problem can be posed as computing the posterior

$$p(x_k, \mathbf{O}_k, M_k \mid Z_k, U_k) \tag{12}$$

where the variable $\mathbf{O}_k = \{\mathbf{o}_k^1, \mathbf{o}_k^2, \ldots, \mathbf{o}_k^n\}$ denotes the true hybrid states of the moving objects, of which there are n in the world at time k, and the variable $M_k = \{m_k^1, m_k^2, \ldots, m_k^q\}$ denotes the true locations of the stationary objects, of which there are q in the world at time k. The second assumption is that when estimating the posterior over the stationary object map and the robot pose, the measurements of moving objects carry no information about stationary landmarks and the robot pose, neither do their hybrid states O_k. The third assumption is that there is no interaction among the robot and the moving objects. The robot and moving objects move independently of each other.

Using Bayes' rules and the assumptions addressed in Appendix B, the general recursive Bayesian formula for SLAM with DATMO can be derived and expresses as: (See Appendix

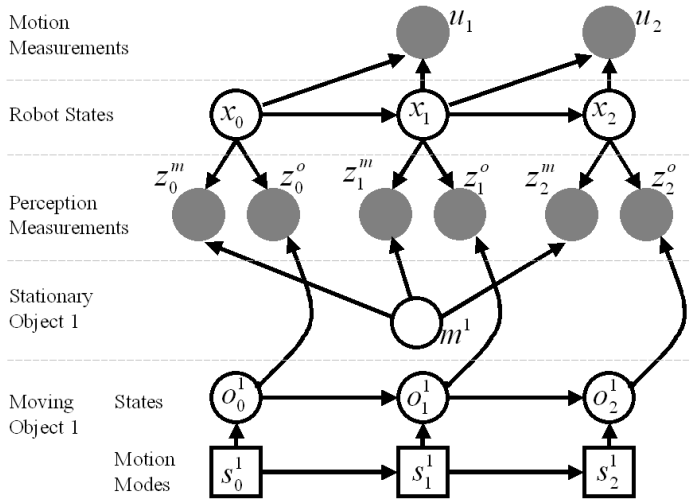


Figure 7: A Dynamic Bayesian Network of the SLAM with DATMO problem of duration three with one moving object and one stationary object.

B for derivation.)

$$
\begin{aligned}
& p(x_k, \mathbf{O}_k, M_k \mid Z_k, U_k) \qquad (13)\\
& \propto \underbrace{p(z_k^o \mid x_k, \mathbf{O}_k)\, p(\mathbf{O}_k \mid Z_{k-1}^o, U_k)}_{\text{DATMO}} \\
& \quad \cdot \underbrace{p(z_k^m \mid x_k, M_k)\, p(x_k, M_k \mid Z_{k-1}^m, U_k)}_{\text{SLAM}} \\
& = \quad \underbrace{p(z_k^o \mid \mathbf{O}_k, x_k)}_{\text{DATMO Update}} \\
& \cdot \quad \underbrace{\int p(\mathbf{O}_k \mid \mathbf{O}_{k-1})\, p(\mathbf{O}_{k-1} \mid Z_{k-1}^o, U_{k-1})\, d\mathbf{O}_{k-1}}_{\text{DATMO Prediction}} \\
& \cdot \quad \underbrace{p(z_k^m \mid M_k, x_k)}_{\text{SLAM Update}} \\
& \cdot \quad \underbrace{\int p(x_k \mid u_k, x_{k-1})\, p(x_{k-1}, M_{k-1} \mid Z_{k-1}^m, U_{k-1})\, dx_{k-1}}_{\text{SLAM Prediction}}
\end{aligned}
$$

where z_k^m and z_k^o denote measurements of stationary and moving objects, respectively. Equation 13 shows how the SLAMMOT problem is decomposed into separate posteriors for moving and stationary objects. It also indicates that DATMO should take account of the uncertainty in the pose estimate of the robot because perception measurements are directly from the robot.

Figure 7 shows a DBN representing three time steps of an example SLAM with DATMO problem with one moving object and one stationary object.

4.2.2 Detection/Classification

Correctly detecting or classifying moving and stationary objects is essential for successfully implementing SLAM with DATMO. In the tracking literature, a number of approaches have been proposed for detecting moving objects, which can be classified into two categories: *with* and *without* the use of *thresholding*. Gish and Mucci (Gish and Mucci, 1987) proposed an approach that detection and tracking occur simultaneously without using a threshold. This approach is called *track before detect* (TBD), although detection and tracking are performed simultaneously. However, the high computational requirements of this approach make the implementation infeasible. Arnold et al. (Arnold et al., 1993) showed that integrating TBD with a dynamic programming algorithm provides an efficient solution for detection without thresholding, which could be a solution for implementing SLAM with generalized objects practically. In Section 7, we will present two reliable approaches for detecting or classifying moving and stationary objects from laser scanners.

4.3 Interaction

Thus far, we have described the SLAMMOT problem which involves both SLAM in dynamic environments and detection and tracking of these dynamic objects. We presented two solutions, SLAM with generalized objects and SLAM with DATMO. In this section, we discuss one possible extension, taking interaction into account, for improving the algorithms.

The multiple moving object tracking problem can be decoupled and treated as the single moving object tracking problem if the objects are moving independently. However, in many tracking applications, objects move dependently such as sea vessels or air fighters moving in formation. In urban and suburban areas, cars or pedestrians often move in formation as well because of specific traffic conditions. Although the locations of these objects are different, velocity and acceleration may be nearly the same in which these moving objects tend to have highly correlated motions. Similar to the SLAM problem, the states of these moving objects can be augmented to a system state and then be tracked simultaneously. Rogers (Rogers, 1988) proposed an augmented state vector approach which is identical to the SLAM problem in the way of dealing with the correlation problem from sensor measurement errors.

In Appendix A, we provided the formula of SLAM with generalized objects in the cases that interactions among the robot and generalized objects exist. Integrating behavior and interaction learning and inference would improve the performance of SLAM with generalized objects and lead to a higher level scene understanding.

Following the proposed framework of SLAM with generalized objects, Wang et al. (Wang et al., 2007) introduced a scene interaction model and a neighboring object interaction model to take long-term and short-term interactions between the tracked objects and its surroundings into account, respectively. With the use of the interaction models, they demonstrated that anomalous activity recognition is accomplished easily in crowded urban areas. Interacting pedestrians, bicycles, motorcycles, cars and trucks are successfully tracked in difficult situations with occlusion.

In the rest of the paper, we will demonstrate the feasibility of SLAMMOT from a ground vehicle at high speeds in crowded urban areas. We will describe practical SLAM with DATMO algorithms which deal with issues of perception modeling, data association and classifying moving and stationary objects. Ample experimental results will be shown for verifying the proposed theory and algorithms.

5 Perception Modeling

Perception modeling, or *representation*, provides a bridge between *perception measurements* and *theory*; different representation methods lead to different means to calculate the theoretical formulas. Representation should allow information from different sensors, from different locations and from different time frames to be fused.

In the tracking literature, targets are usually represented by *point-features* (Blackman and Popoli, 1999). In most air and sea vehicle tracking applications, the geometrical information of the targets is not included because of the limited resolution of perception sensors such as radar and sonar. However, the signal-related data such as the amplitude of the radar signal can be included to aid data association and classification. On the other hand, research on mobile robot navigation has produced four major paradigms for environment representation: feature-based approaches (Leonard and Durrant-Whyte, 1991), grid-based approaches (Elfes, 1988; Thrun et al., 1998), direct approaches (Lu and Milios, 1994; Lu and Milios, 1997), and topological approaches (Choset and Nagatani, 2001).

Since feature-based approaches are used in both MOT and SLAM, it should be straightforward to use feature-based approaches for accomplish SLAMMOT. Unfortunately, it is extremely difficult to define and extract features reliably and robustly in outdoor environments according to our experiments. In this section, we present a hierarchical free-form object representation to integrate direct methods, grid-based approaches and feature-based approaches for overcoming these difficulties.

5.1 Hierarchical Free-Form Object Based Representation

In outdoor or urban environments, features are extremely difficult to define and extract as both stationary and moving objects do not have specific sizes and shapes. Therefore, instead of using ad hoc approaches to define features in specific environments or for specific objects, *free-form objects* are used.

At the preprocessing stage, scan points are grouped or segmented into *segments*. Hoover et al. (Hoover et al., 1996) proposed a methodology for evaluating range image segmentation algorithms, which are mainly for segmenting a range image into planar or quadric patches. Unfortunately, these methods are infeasible for our applications. Here we use a simple distance criterion, namely the distance between points in two segments must be longer than 1 meter. Although this simple criterion can not produce perfect segmentation, more precise segmentation will be accomplished by localization, mapping and tracking using spatial and temporal information over several time frames. An example of scan segmentation is shown

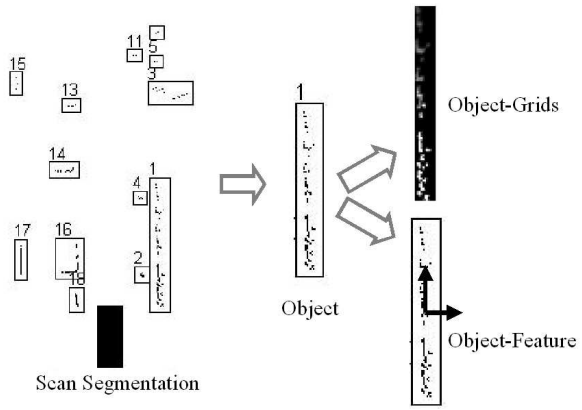


Figure 8: An example of scan segmentation. The black solid box denotes the robot (2mx5m). Each object has its own grid-map and coordinate system.

in Figure 8.

In this framework, the scan segments over different time frames are integrated into *free-form objects* after localization, mapping and tracking processes. This approach is hierarchical since these three main representation paradigms are used on different levels. Local localization is accomplished using direct methods, local mapping is accomplished using grid-based approaches and global SLAM is accomplished using feature-based approaches. This representation is also suitable for moving object tracking. Feature-based approaches such as Kalman filtering can be used for manage tracking uncertainty and the shape information of moving object is also maintained using grid-maps. Note that an *free-form object* can be as small as a pedestrian or as big as several street blocks.

5.2 Local Localization

Registration or localization of scan segments over different time frames can be done using the *direct* methods, namely the iterative closest point (ICP) algorithm (Rusinkiewicz and Levoy, 2001). As range images are sparser and more uncertain in outdoor applications than indoor applications, the pose estimation and the corresponding distribution from the ICP algorithm may not be reliable. Sparse data causes problems of *correspondence finding*, which directly affect the accuracy of direct methods. If a point-point metric is used in the ICP algorithm, one-to-one correspondence will not be guaranteed with sparse data, which will result in decreasing the accuracy of transformation estimation and slower convergence. Research on the ICP algorithms suggests that minimizing distances between points and tangent planes can converge faster. But because of sparse data and irregular surfaces in outdoor environments, the secondary information derived from raw data such as surface normal can be unreliable and too sensitive. The other issue is featureless data, which causes correspondence ambiguity as well.

In (Wang and Thorpe, 2004), we presented a sampling- and correlation-based range image matching (SCRIM) algorithm for taking correspondence errors and measurement noise into account. For dealing with the sparse data issues, a sampling-based approach is used to

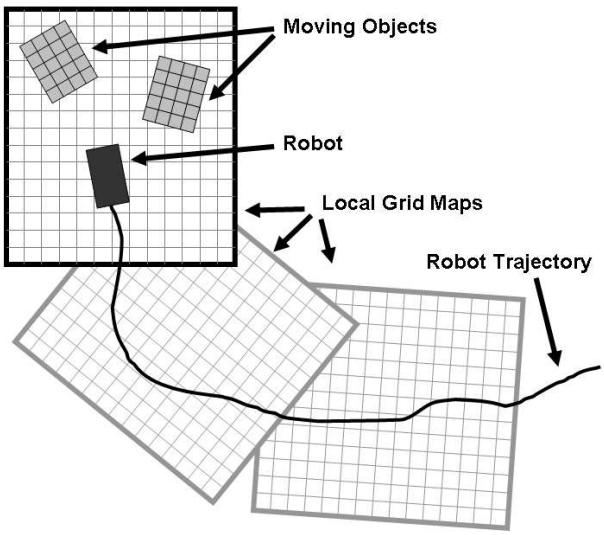


Figure 9: Hierarchical free-from object based representation.

estimate the uncertainty from correspondence errors. Instead of using only one initial relative transformation guess, the registration process is run 100 times with randomly generated initial relative transformations. For dealing with the uncertain data issues, a correlation-based approach is used with the *grid-based* method for estimating the uncertainty from measurement noise along with the sampling-based approach. Measurement points and their corresponding distributions are transformed into occupancy grids using our proposed SICK laser scanner noise model. After the grid maps are built, the correlation of the grid maps is used to evaluate how strong the grid-maps are related. Now the samples are weighted with their normalized correlation responses. We have shown that the covariance estimates from the SCRIM algorithm describe the estimate distribution correctly. See (Wang and Thorpe, 2004) for more detailed information of the SCRIM algorithm.

5.3 Local Mapping

The results of local localization or registration are integrated into grid-maps via grid-based approaches. Measurements belonging to stationary objects are integrated/updated into the local grid map. Each moving object has its own grid-map, which contains the shape (geometrical information) of this moving object and has its own coordinate systems. See Figure 9 for an illustration.

After locally localizing the robot using the SCRIM algorithm, the new measurement is integrated into the local grid map. The Bayesian recursive formula for updating the local grid map is computed by: (See (Elfes, 1988; Elfes, 1990) for a derivation.)

$$\begin{aligned} l_k^{xy} &= \log \frac{p(g^{xy} \mid Z_{k-1}^m, z_k)}{1 - p(g^{xy} \mid Z_{k-1}^m, z_k^m)} \\ &= \log \frac{p(g^{xy} \mid z_k^m)}{1 - p(g^{xy} \mid z_k^m)} + l_{k-1}^{xy} + l_0^{xy} \end{aligned} \tag{14}$$

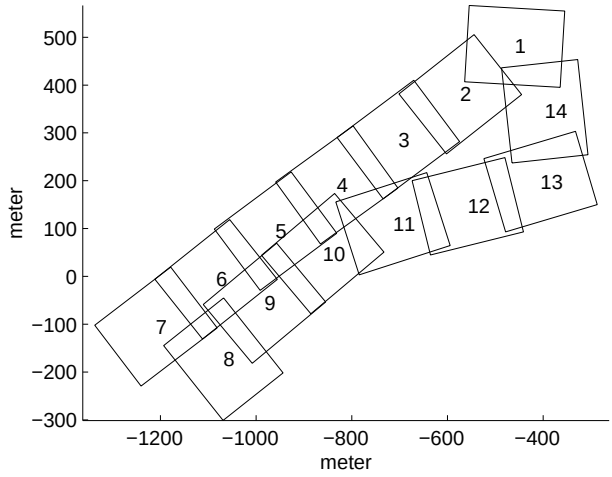


Figure 10: Generated grid maps along the trajectory. The boxes indicate the boundaries of the grid maps.

where g is the grid map, g^{xy} be the occupancy value of a grid cell at $\langle x, y \rangle$, l is the log-odds ratio, and

$$l_0^{xy} = \log \frac{p(g^{xy})}{1 - p(g^{xy})} \tag{15}$$

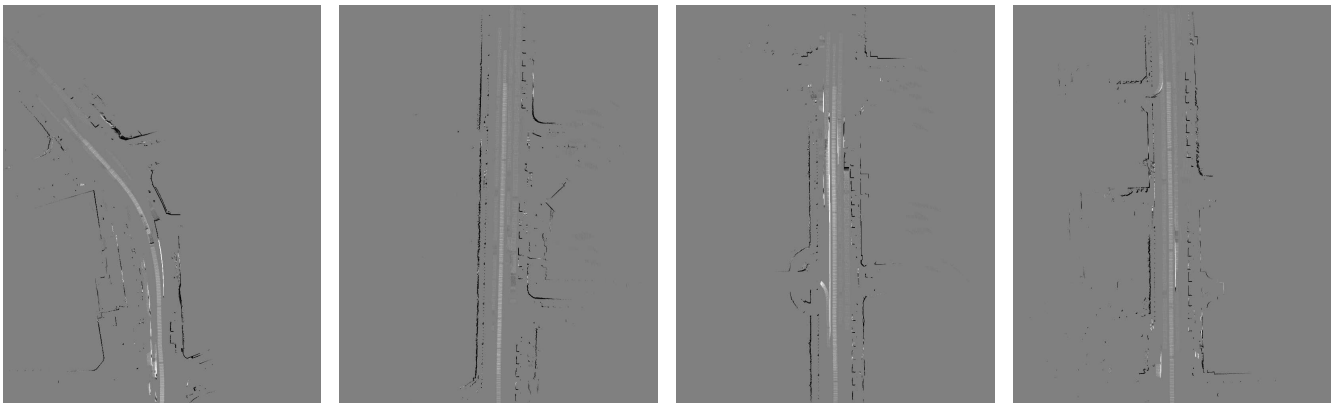
Theoretically, there are two important requirements to select the size and resolution of the local grid maps for accomplishing hierarchical free-form object based SLAM: one is that the local grid maps should not contain loops, and the other is that the quality of the grid map should be maintained at a reasonable level. To satisfy these requirements, the width, length and resolution of the local grid maps can be *adjusted* on-line in practice.

For the experiments addressed in this paper, the width and length of the grid maps are set as 160 meters and 200 meters respectively, and the resolution of the grid map is set at 0.2 meter. When the robot arrives at the 40 meter boundary of the grid map, a new grid map is initialized. The global pose of a local map and its corresponding distribution is computed according to the robot's global pose and the distribution. Figure 10 shows the local grid maps generated along the trajectory using the described parameters. Figure 11 shows the details of the grid maps, which contain information from both stationary objects and moving objects.

5.4 Global SLAM

As grid-based approaches need extra computation for loop-closing and all raw scans have to be used to generate a new global consistent map, the grid-map is only built locally. For accomplishing global SLAM in very large environments, each local grid map is treated as a three-degree-freedom feature as shown in Figure 9 and loop closing is done with the mechanism of the *feature-based* approaches.

Figure 12 shows the result without loop-closing and Figure 13 shows the result using the



(a) Grid Map 1. (b) Grid Map2. (c) Grid Map 3. (d) Grid Map 4.

Figure 11: Details of the grid maps. *Gray* denotes areas which are not occupied by both moving objects and stationary objects, *whiter than gray* denotes the areas which are likely to be occupied by moving objects, and *darker than gray* denotes the areas which are likely to be occupied by stationary objects.

feature based EKF algorithm for loop-closing with correct loop detection. Extension 2 provides a full reply of this loop closing processing. Information from moving objects is filtered out in both figures. The covariance matrix for closing this loop contains only 14 three degree-of-freedom features.

Since we set the whole local grid maps as features in the feature-based approaches for loop-closing, the uncertainty *inside* the local grid maps is not updated with the constraints from loop detection. Although Figure 13 shows a satisfactory result, the coherence of the overlay between grid maps is not guaranteed. Practically, the inconsistency between the grid-maps will not effect the robot's ability to perform tasks. Local navigation can be done with the current built grid map which contains the most recent information about the surrounding environment. Global path planning can be done with the global consistent map from feature-based approaches in a topological sense. In addition, the quality of the global map can be improved by adjusting sizes and resolutions of the local grid maps to smooth out the inconsistency between grid maps. At the same time, the grid-maps should be big enough to have high object saliency scores in order to reliably solve the revisiting problem.

5.5 Local Moving Object Grid Map

There is a wide variety of moving objects in urban and suburban environments such as pedestrians, animals, bicycles, motorcycles, cars, trucks, buses and trailers. The critical requirement for safe driving is that all such moving objects be detected and tracked correctly. Figure 14 shows an example of different kinds of moving objects in an urban area where the hierarchical free-form object representation is suitable and applicable because *free-form* objects are used without predefining features or appearances.

As the number of measurement points belonging to small moving objects such as pedestrians is often less than four, the centroid of the measurement points is used as the state vector of

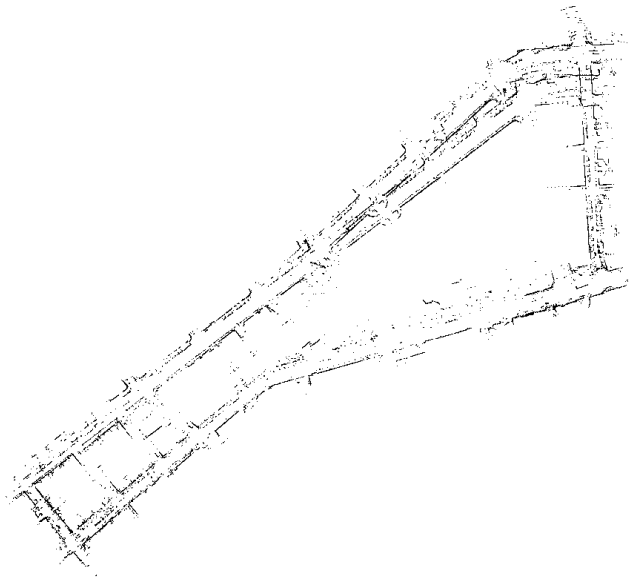


Figure 12: The result without loop-closing. Information from moving object is filtered out.

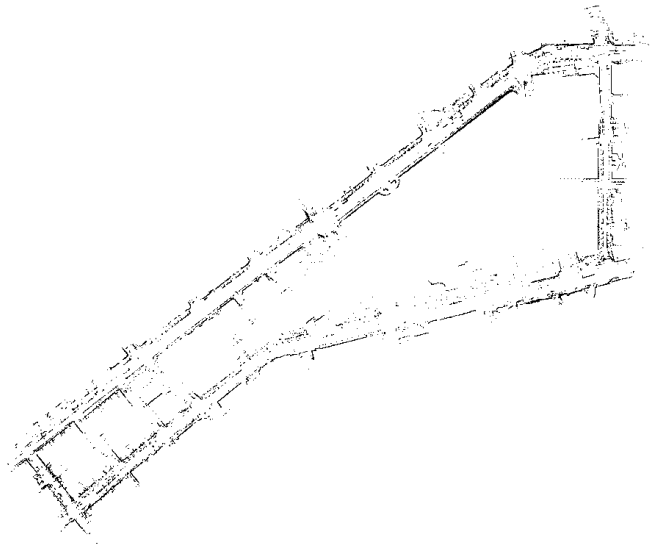


Figure 13: The result with loop-closing. Information from moving object is filtered out.

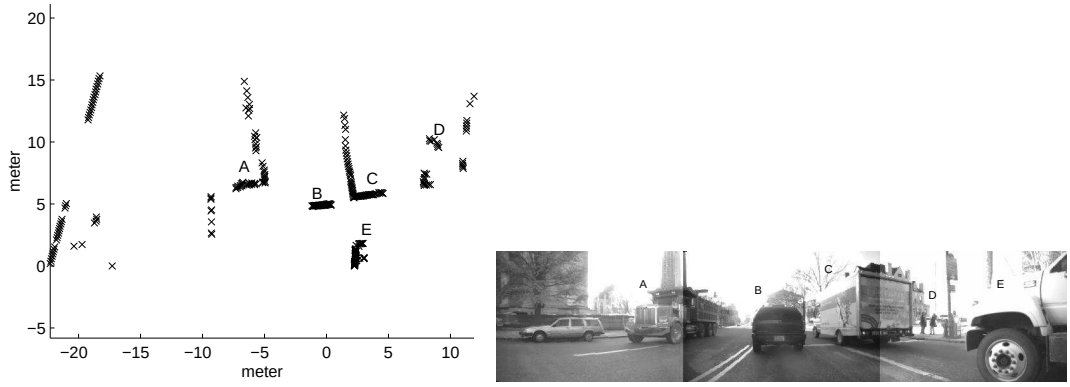


Figure 14: A wide variety of moving objects in urban areas. A: a dump truck, B: a car, C: a truck, D: two pedestrians, E: a truck.

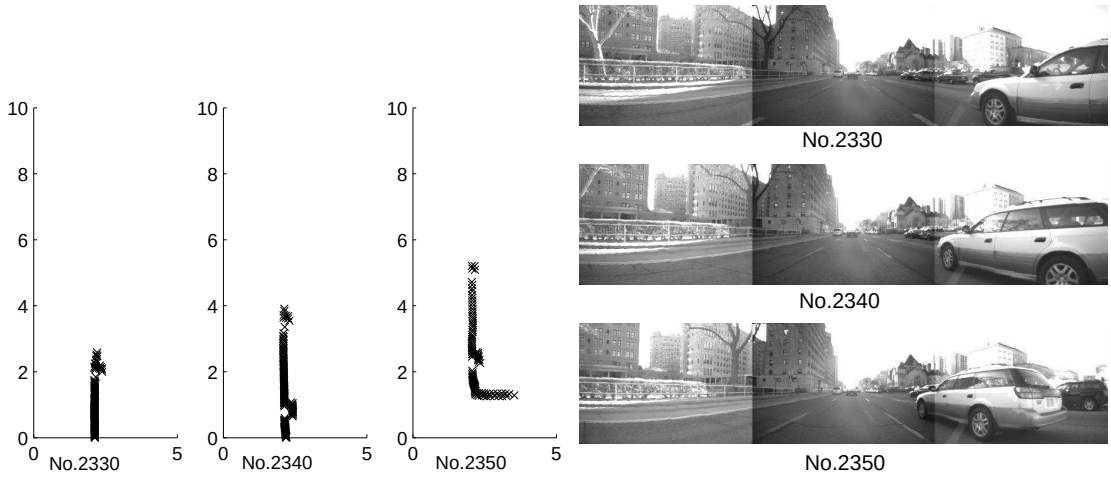


Figure 15: Different portions of a moving car.

the moving object. The state vector, or *object-feature* of a small moving object contains only location without orientation because the geometrical information is insufficient to correctly determine orientation.

However, when tracking large moving objects, using the centroid of the measurements is imprecise. Different portions of moving objects are observed over different time frames because of motion and occlusion. This means that the centroids of the measurements over different time frames do not present the same physical point. Figure 15 shows the different portions of a moving car observed over different time frames.

Therefore, the SCRIM algorithm is used to estimate the relative transformation between the new measurement and the *object-grids* and its corresponding distribution. As the online learned motion models of moving objects may not be reliable at the early stage of tracking, the predicted location of the moving object may not good enough to avoid the local minima problem of the ICP algorithm. Applying the SCRIM algorithm to correctly describe the uncertainty of the pose estimate is especially important.

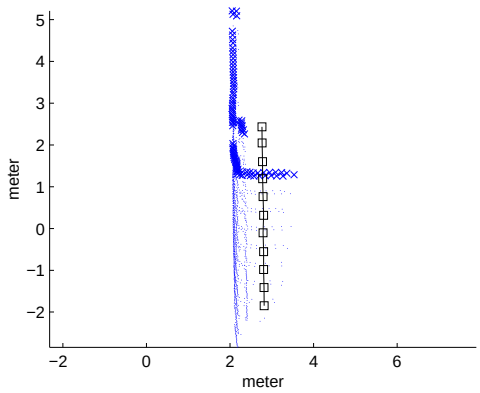


Figure 16: Registration results of the example in Figure 15 using the SCRIM algorithm. The states are indicated by Box, and the final scan points are indicated by $\times$.

Since the big object orientation can be determined reliably, the state vector, or object-feature, can consist of both location and orientation. In addition, the geometrical information is accumulated and integrated into the object-grids. As a result, not only are motions of moving objects learned and tracked, but their contours are also built. Figure 16 shows the registration results using the SCRIM algorithm.

The moving objects' *own* grid maps only maintain their shape information but not their trajectories. Although the trajectories of moving objects can be stored and maintained with *lists*, it is difficult to retrieve information from the lists of multiple moving objects. Therefore, local moving object grid maps are created to store trajectory information from moving objects using the same mechanism of maintaining local stationary object grid maps. Figure 11 shows examples of the local stationary and moving object grid maps.

By integrating trajectory information from moving cars and pedestrians, lanes and sidewalks can be recognized. This kind of information is extremely important to robots operating in environments occupied by human beings. In the applications of exploration, robots can go wherever there is no obstacle. However, for tasks in environments shared with human beings, robots at least have to follow the same rules that people obey. For example, a robot car should be kept in the lane and should not go onto the unoccupied sidewalks. Both the stationary object map and the moving object map provide essential and critical information to accomplish these tasks.

6 Data Association

In this section, we present simple yet effective solutions for solving data association issues in SLAMMOT.

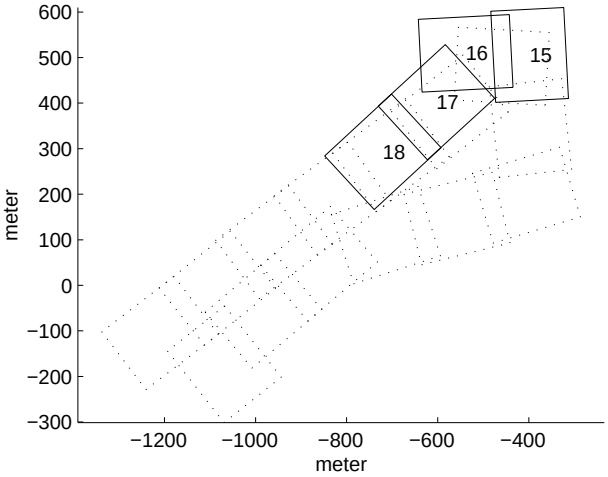


Figure 17: The revisiting problem. Because of the accumulated pose error, the current grid map is not consistent with the pre-built map

6.1 Revisiting in SLAM

One of the most important step to solve the global SLAM problem is to robustly *detect* loops or *recognize* the pre-visited areas. It is called the *revisiting* problem (Stewart et al., 2003; Thrun and Liu, 2003; Hähnel et al., 2003). Figure 17 shows that the robot entered the explored area and the current grid map is not consistent with the pre-built map. The revisiting problem is difficult because of accumulated pose estimate errors, unmodelled uncertainty sources, temporary stationary objects and occlusion. Here we describe one approach, *information exploiting*, for dealing with these issues.

For loop closing, not only *recognizing* but also *localizing* the current measurement within the global map has to be accomplished. Unfortunately, because of *temporary stationary objects*, *occlusion*, and *featureless areas* (Wang, 2004), recognizing and localizing places are difficult even with the proper information about which portions of the built map are more likely. For instance, Figure 18 shows that the currently built stationary object maps may be very different from the global stationary object map because of temporary stationary objects such as ground vehicles stopped by traffic lights and parked cars. Since the environments are dynamic, stationary objects may be occluded when the robot is surrounded by big moving objects such as buses and trucks.

In order to deal with the addressed situations, big regions are used for loop-detection instead of using raw scans. In large scale regions, large and stable objects such as buildings and street blocks are the dominating factors in the recognition and localization processes, and the effects of temporary stationary objects such as parked cars is minimized. It is also more likely to have more salient areas when the size of the regions is larger. In other words, the ambiguity of recognition and localization can be removed more easily and robustly. As the measurements at different locations over different times are accumulated and integrated into the local grid maps, the occlusion of stationary objects is reduced as well. Figure 19 shows a grid-map pair of the same regions built at different times. Although the details of local

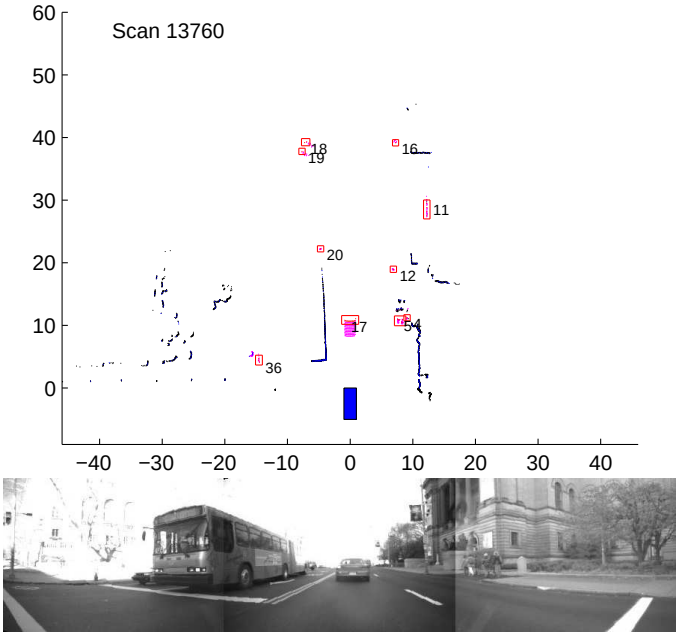


Figure 18: A temporary stationary bus. Rectangles denote the detected and tracked moving objects. The segment numbers of these moving objects are shown.

grid-maps are not the same in the same region because of the described reasons, full grid maps contain enough information for place recognition and localization.

As local grid maps are used, visual image registration algorithms from the computer vision literature can be used for recognition and localization. Following the SCRIM algorithm, we use the correlation between local grid maps to verify the recognition (searching) results, and we perform recognition (searching) between two grid maps according to the covariance matrix from the feature-based SLAM process instead of sampling. The search stage is speeded up using multi-scale pyramids. Figure 20 shows the recognition and localization results of the examples in Figure 19 using different scales.

6.2 Data Association in MOT

Once a new moving object is detected, our algorithm initializes a new track for this object, such as assigning an initial state and motion models to this new moving object. By using laser scanners, we can only get the position but not the velocity and orientation, therefore our algorithm uses the data from different times and then accomplishes data association in order to initialize a new track.

Data association and tracking problems have been extensively studied and a number of statistical data association techniques have been developed, such as the Joint Probabilistic Data Association Filter (JPDAF) (Fortmann et al., 1983) and the Multiple Hypothesis Tracking (MHT) (Reid, 1979)(Cox and Hingorani, 1996). Our system applies the MHT method, which maintains a hypothesis tree and can revise its decisions while getting new

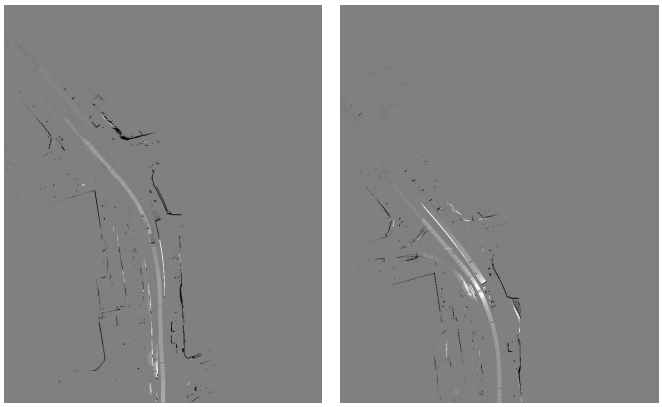


Figure 19: The grid-map pair of the same region built at different times: Grid-map 1 and Grid map 16. Different moving object activities at different times, occlusion and temporary stationary objects are shown.

Figure 20: Recognition and localization results using different scales of grid map 1 and grid map 16. From left to right: 1/8 scale, 1/4 scale and 1/2 scale. Two grid maps are shown with respect to the same coordinate system.

information. This delayed decision approach is more robust than other approaches. The main disadvantage of the MHT method is its exponential complexity. If the hypothesis tree is too big, it will not be feasible to search the whole hypotheses to get the most likely set of matching. Fortunately, the number of moving objects in our application is usually less than twenty and most of the moving objects only appear for a short period of time. Also, useful information about moving objects from laser scanners, such as location, size, shape, and velocity, is used for updating the confidence for pruning and merging hypotheses. In practice, the hypothesis tree is always managed in a reasonable size.

7 Moving Object Detection

Recall that SLAM with DATMO makes the assumption that the measurements can be decomposed into measurements of stationary and moving objects. This means that correctly detecting moving object is essential for successfully implementing SLAM with DATMO.

In this section, we describe two approaches for detecting moving objects: a consistency based approach and a motion object map based approach. Although these two approaches work with the use of thresholding, the experimental results using laser scanners are satisfactory.

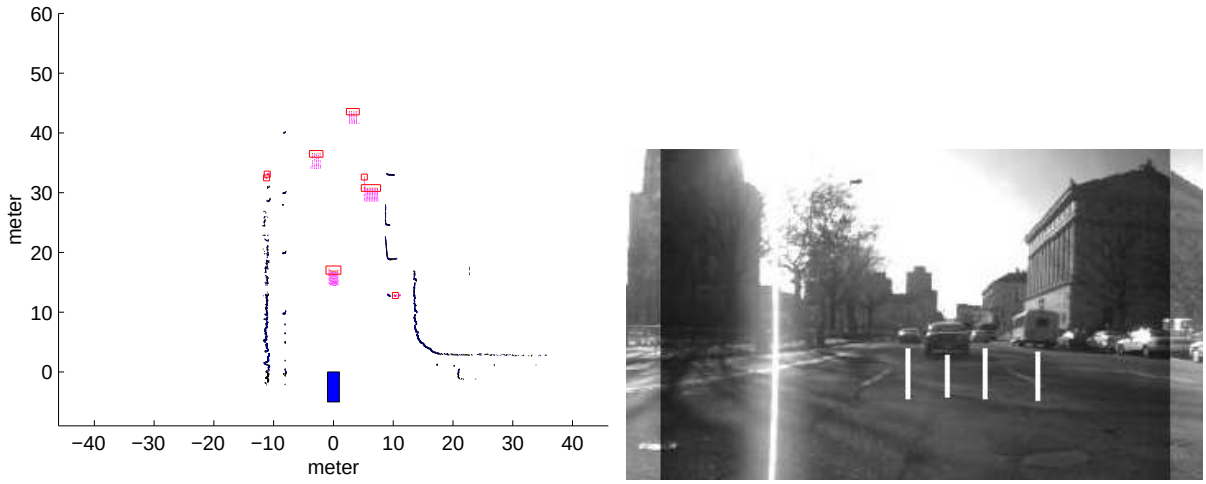


Figure 21: Multiple car detection and data association. Top: the solid box denotes the Navlab11 testbed and rectangles denote the detected moving objects. Bottom: the partial image from the tri-camera system. Four lines indicate the detected ground vehicles.

7.1 Consistency-based Detection

The consistency-based moving object detection algorithm consists of two parts: the first is the detection of moving points; the second is the combination of the results from segmentation and moving point detection for deciding which segments are potential moving objects. The details are as follows: given a new scan, the local surrounding map, and the relative pose estimate from direct methods, we first transform the local surrounding map to the coordinate frame of the current laser scanner, and then convert the map from a rectangular coordinate system to a polar coordinate system. Now it is easy to detect moving points by comparing values along the range axis of the polar coordinate system.

A segment is identified as a potential moving object if the ratio of the number of moving points to the number of total points is greater than 0.5. Note that the consistency-based detector is a motion-based detector in which temporary stationary objects can not be detected. If the time period between consecutive measurements is very short, the motions of moving objects will be too small to detect. Therefore, in practice an adequate time period should be chosen for maximizing the correctness of the consistency-based detection approach.

Figure 21 shows a result of the detection and data association algorithms, and the partial image from the tri-camera system.

7.2 Moving Object Map based Detection

Detection of pedestrians at very low speeds is difficult but possible by including information from the moving object map. From our experimental data, we found that the data associated with a pedestrian is very small, generally 1-4 points. Also, the motion of a pedestrian can be too slow to be detected by the consistency-based detector. As the moving object map

contains information from previous moving objects, we can say that if a blob is in an area that was previously occupied by moving objects, this object can be recognized as a potential moving object.

8 Experimental results

So far we have shown the procedures in detail for accomplishing ladar-based SLAM with DATMO from a ground vehicle at high speeds in crowded urban areas. To summarize, Figure 22 shows the flow diagram of SLAM with DATMO, and the steps are briefly described below:

8.0.1 Data collecting and preprocessing

Measurements from motion sensors such as odometry and measurements from perception sensors such as laser scanners are collected. In our applications, laser scans are segmented. The robot pose is predicted using the motion measurement and the robot motion model.

8.0.2 Tracked moving object association

The scan segments are associated with the tracked moving objects with the MHT algorithm. In this stage, the predicted robot pose estimate is used.

8.0.3 Moving object detection and robot pose estimation

Only scan segments not associated with the tracked moving objects are used in this stage. Two algorithms, the consistency-based and the moving object map-based detectors, are used to detect moving objects. At the same time, the robot pose estimate is improved with the use of the SCRIM algorithm.

8.0.4 Update of stationary and moving objects

With the updated robot pose estimate, the tracked moving objects are updated and the new moving objects are initialized via the IMM algorithm. If the robot arrives the boundary of the local grid map, a new stationary object grid map and a new moving object grid map are initialized and the global SLAM using feature-based approaches stages is activated. Otherwise, the stationary object grid map and the moving object grid map are updated with the new stationary scan segments and the new moving scan segments, respectively. Now we complete one cycle of local localization, mapping and moving object tracking.

8.0.5 Global SLAM

The revisiting problem is solved. The local grid maps are treated as three degree-of-freedom features and the global SLAM problem is solved via extended Kalman filtering.

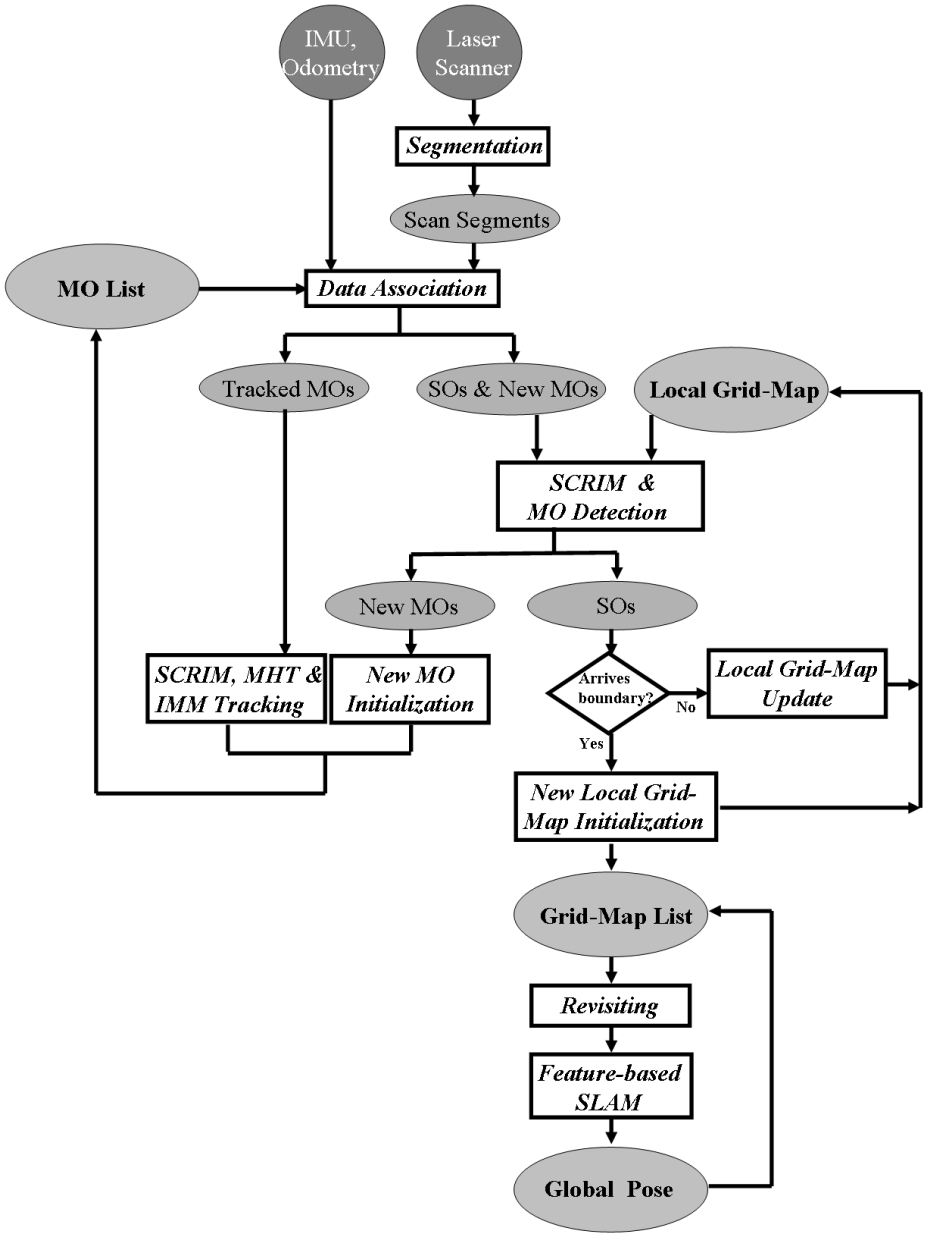


Figure 22: The flowchart of the SLAM with DATMO algorithm. Dark circles are data, rectangles are processes and grey ovals are inferred or learned variables.

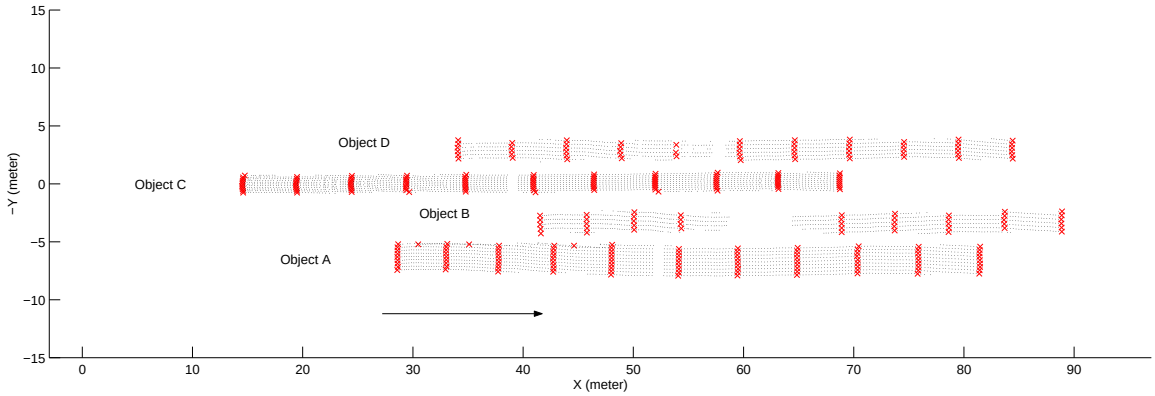


Figure 23: Raw data of 201 scans. Measurements associated with stationary objects are filtered out. Measurements are denoted by × every 20 scans. Note that object B was occluded during the tracking process.

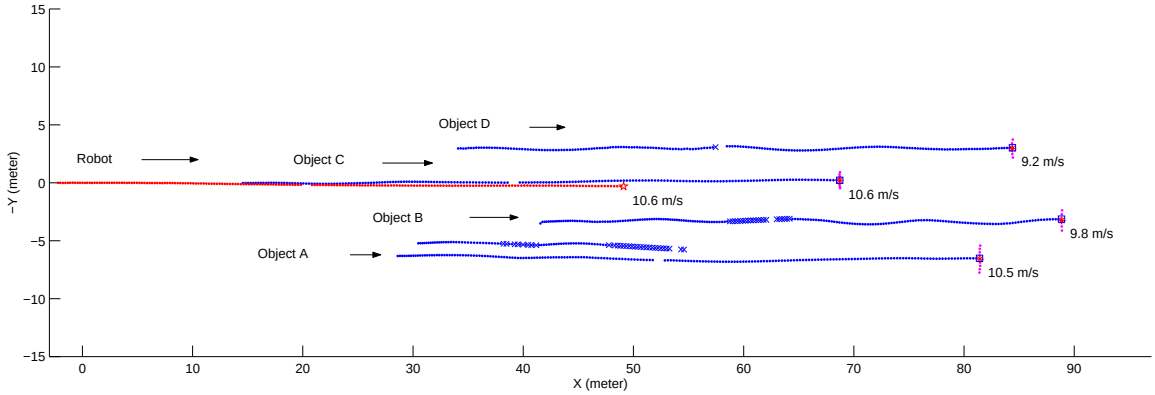


Figure 24: Results of multiple ground vehicle tracking. The trajectories of the robot and the tracked moving objects are denoted with the text labels. × denotes that the state estimates are not from the update stage but from the prediction stage because of occlusion.

Extension 1 provides a full reply of the SLAM with DATMO processing. As the experimental results of a city sized SLAM and single car tracking have been shown in the previous sections, multiple pedestrian and ground vehicle tracking will be shown to demonstrate the feasibility of the proposed SLAMMOT theory and SLAM with DATMO algorithms. In addition, the effects of 2-D environment assumption in 3-D environments, the issues about sensor selection and limitation and ground truth for evaluating SLAMMOT will be addressed in this section.

8.1 Multiple Target Tracking

Figure 23 and 24 illustrate an example of multiple car tracking for about 6 seconds. Figure 23 shows the raw data of the 201 scans in which object B was occluded during the tracking process. Figure 24 shows the tracking results. The occlusion did not affect tracking because the learned motion models provide reliable predictions of the object states. The association was established correctly when object B reappeared in this example. Extension 3 provides a full reply of the multiple car tracking processing.

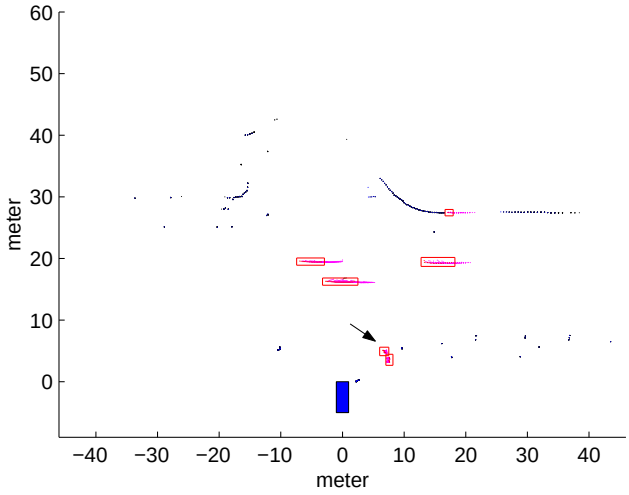


Figure 25: An intersection. Pedestrians are pointed out by the arrow.

Figure 26: Visual images from the tri-camera system. Block boxes indicate the detected and tracked pedestrians. Note that the images are only for visualization.

Figure 25-28 illustrate an example of pedestrian tracking. Figure 25 shows the scene in which there are three pedestrians. Figure 26 shows the visual images from the tri-camera system and Figure 27 show the 141 raw scans. Because of the selected distance criterion in segmentation, object B consists of two pedestrians. Figure 28 shows the tracking result which demonstrates the ability to deal with occlusion. Extension 4 provides a full reply of this multiple pedestrian tracking processing.

8.2 2-D Environment Assumption in 3-D Environments

We have demonstrated that it is feasible to accomplish city-sized SLAM, and Figure 13 shows a convincing 2-D map of a very large urban area. In order to build 3-D ($2\frac{1}{2}$-D) maps, we mounted another scanner on the top of the Navlab11 vehicle to perform vertical profiling. Accordingly, high quality 3D models can be produced. Figure 29, Figure 30 and Figure 31 show the 3-D models of different objects, which can be very useful to applications of civil engineering, architecture, landscape architecture, city planning, etc. Extension 5 provides a video that shows the 3D city modeling results.

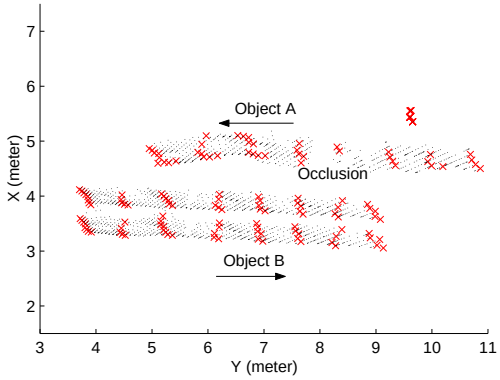


Figure 27: Raw data of 201 scans. Measurements are denoted by $\times$ every 20 scans.

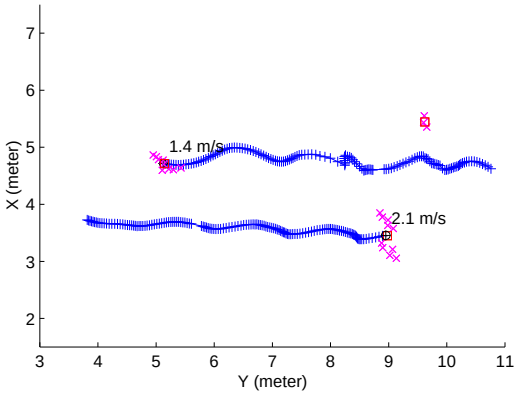


Figure 28: Results of multiple pedestrian tracking. The final scan points are denoted by magenta $\times$ and the estimates are denoted by blue $+$.

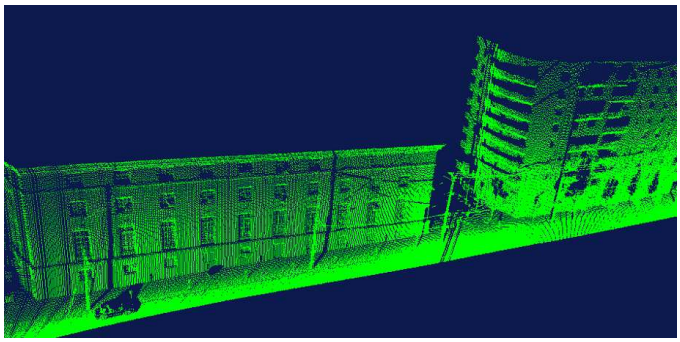


Figure 29: 3-D models of buildings on Filmore street.

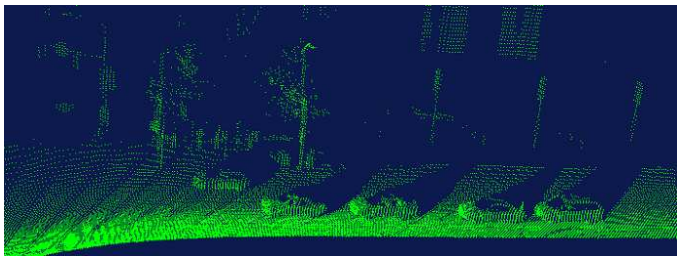


Figure 30: 3-D models of parked cars in front of the Carnegie Museum of Art.

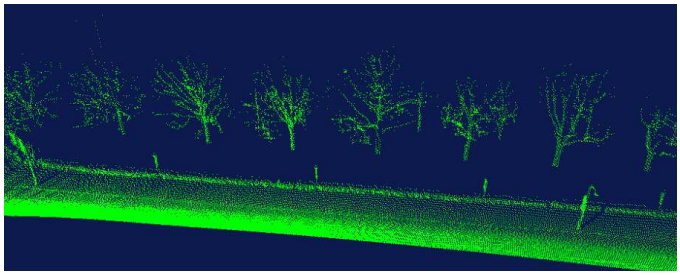


Figure 31: 3-D models of trees on S. Bellefield avenue.

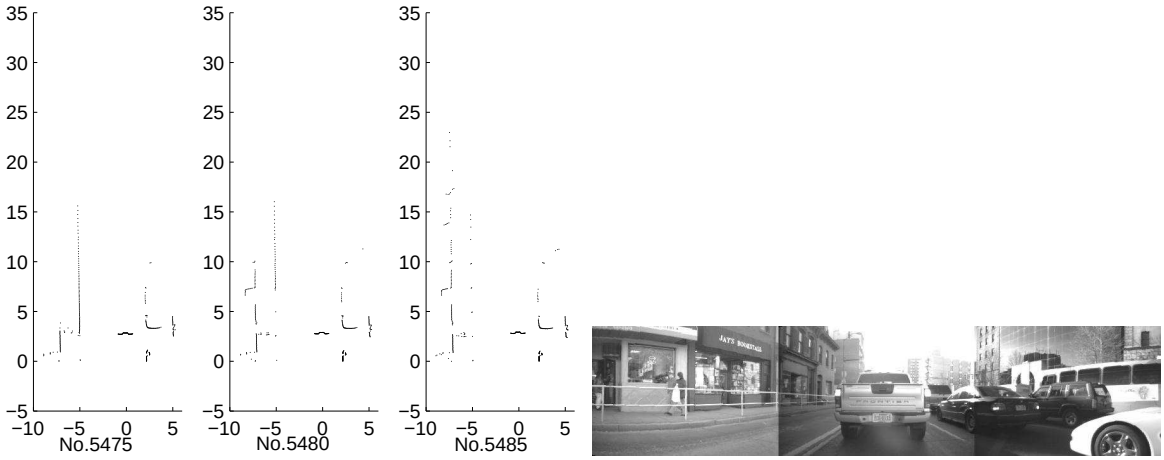


Figure 32: Dramatic changes between consecutive scans due to a sudden start.

Although the formulations derived in this paper are not restricted to two-dimensional applications, it is more practical and easier to solve the problem in real-time by assuming that the ground is flat. For most indoor applications, this assumption is fair. But for applications in urban, suburban or highway environments, this assumption is not always valid. False measurements due to this assumption are often observed in our experiments. One is from roll and pitch motions of the robot, which are unavoidable due to turns at high speeds or sudden stops or starts (see Figure 32). These motions may cause false measurements such as wrong scan data from the ground instead of other objects. Additionally, since the vehicle moves in 3-D environments, uphill environments may cause the laser beam to hit the ground as well (see Figure 33).

In order to accomplish 2-D SLAM with DATMO in 3-D environments, it is critical to detect and filter out these false measurements. Our algorithms can detect these false measurements implicitly without using other pitch and roll measurement. First, the false measurements are detected and initialized as new moving objects by our moving object detector. After data associating and tracking are applied to these measurements, the shape and motion inconsistency will tell us quickly that these are false measurements. Also these false measurements will disappear immediately once the motion of the vehicle is back to normal. The results using data from Navlab11 show that our 2-D algorithms can survive in urban and suburban environments. However, these big and fast moving *false alarms* may confuse the warning system and cause a sudden overwhelming fear before these false alarm are filtered out by

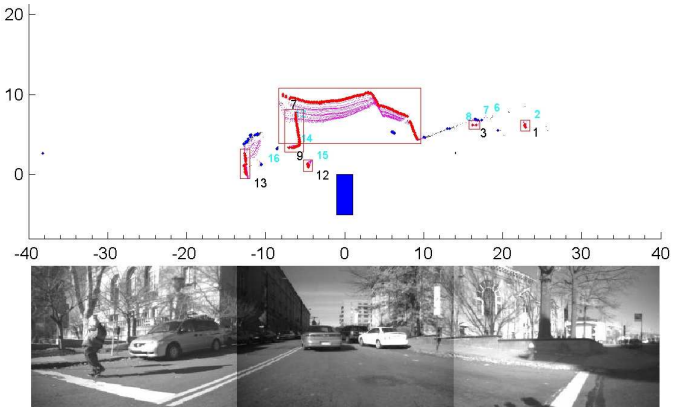


Figure 33: False measurements from a uphill environment.

the SLAM with DATMO processes. Using 3-D motion and/or 3-D perception sensors to compensate these effects should be necessary.

8.3 Sensor Selection and Limitation

The derived Bayesian formulations for solving the SLAMMOT problem are not restricted to any specific sensors. In this section, we discuss the issues on selection and limitations of perception and motion sensors.

8.3.1 Perception Sensors

In the tracking literature, there are a number of studies on issues of using different perception sensors (Bar-Shalom and Li, 1995; Blackman and Popoli, 1999). In the SLAM literature, use of different sensors has been proposed as well. For instance, bearing-only sensors such as cameras (Deans, 2005), and range-only sensors such as transceiver-transponders (Kantor and Singh, 2002; Newman and Leonard, 2003) have been used for SLAM. The fundamentals for using heterogeneous sensors for SLAM, MOT, and SLAM with DATMO are the same. The difference is *sensor modelling* according to sensor characteristics.

Although laser scanners are relatively accurate, some failure modes or limitations exist. Laser scanners can not detect some materials such as glass because the laser beam can go through these materials. Laser scanners may not detect black objects because laser light is absorbed. If the surface of objects is not diffusing enough, the laser beam can be reflected out and not returned to the devices. In our experiments these failure modes are rarely observed but do happen. In Figure 34, the measurement from the laser scanner missed two black and/or clean cars, which are shown clearly in the visual image form the tri-camera system. Heterogenous sensor fusion would be necessary to overcome these limitations.

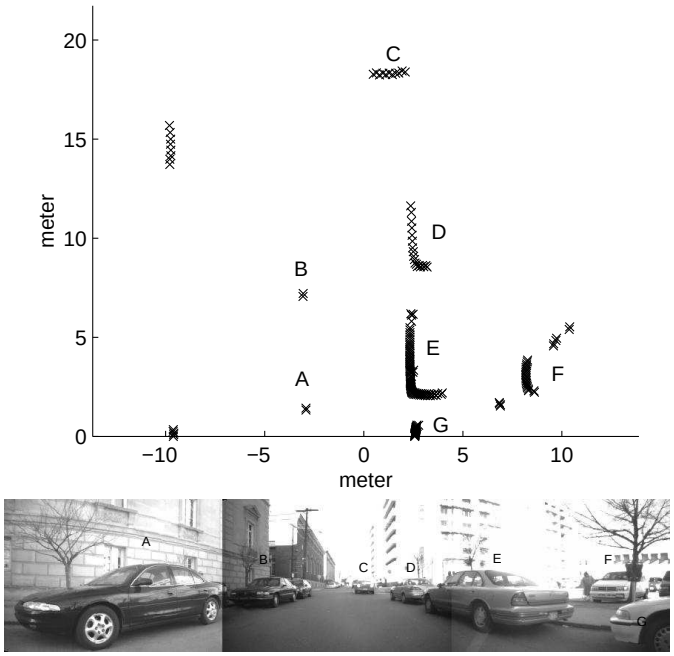


Figure 34: The failure mode of the laser scanners. Car A and Car B are not shown completely in the laser scanner measurement.

8.3.2 Motion Sensors

In this paper, we have demonstrated that it is indeed feasible to accomplish SLAMMOT using odometry and laser scanners. However, we do not suggest the totally abandonment of inexpensive sensors such as compasses and GPS if they are available. With extra information from these inaccurate but inexpensive sensors, inference and learning can be easier and faster. For instance, for the revisiting problem, the computational time for searching can be reduced dramatically in the orientation dimension with a rough global orientation estimate from a compass, and in the translation dimensions with a rough global location estimate from GPS. The saved computational power can be used for other functionalities such as warning and planning.

8.4 Ground Truth

It would of course be nice to have ground truth, to measure the quantitative improvement of localization, mapping and moving object tracking with the methods introduced in this paper. Unfortunately, getting accurate ground truth is difficult, and is beyond the scope of the work in this paper. Several factors make ground truth difficult:

- Localization: collecting GPS data in city environments is problematic, due to reflections from tall buildings and other corrupting effects.
- Mapping: the accuracy and resolution of the mapping results are better than available digital maps.

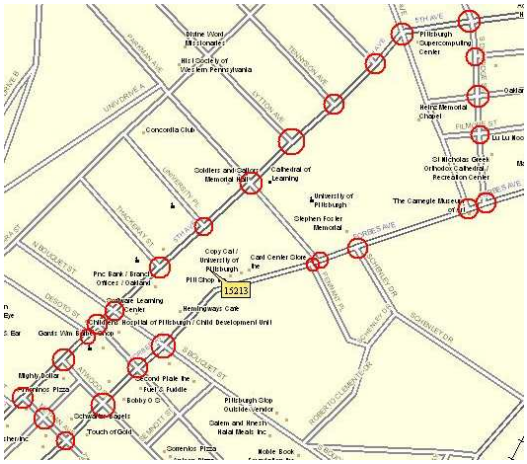


Figure 35: An available digital map. The locations of intersections are denoted by circles.

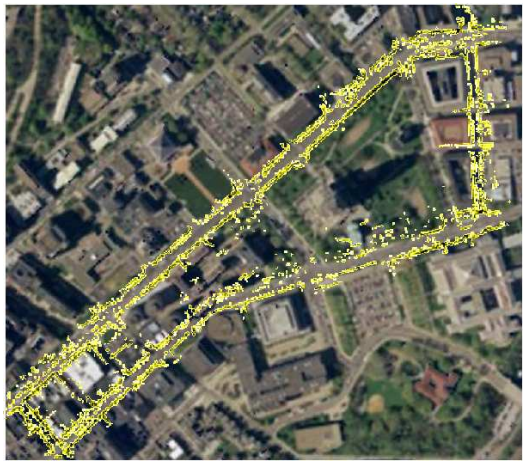


Figure 36: The reconstructed map is overlayed on an aerial photo.

- Moving object tracking: any system that works in the presence of uninstrumented moving objects will have a difficult time assessing the accuracy of tracking data.

Some of these difficulties are illustrated by Figures 35, 36, and 37. Figure 35 shows the locations of intersections on an available digital map. In Figure 37, those same intersections are overlayed on our reconstructed map. In Figure 36, the reconstructed map is overlayed on an aerial photo. Qualitatively, the maps line up, and the scale of the maps is consistent to within the resolution of the digital maps. Quantitative comparisons are much more difficult.

9 Conclusion and Future Work

In this paper, we have developed a theory for performing SLAMMOT. We first presented SLAM with generalized objects, which computes the joint posterior over all generalized objects and the robot. Such an approach is similar to existing SLAM algorithms. Unfortunately, it is computationally demanding and generally infeasible. Consequently, we developed

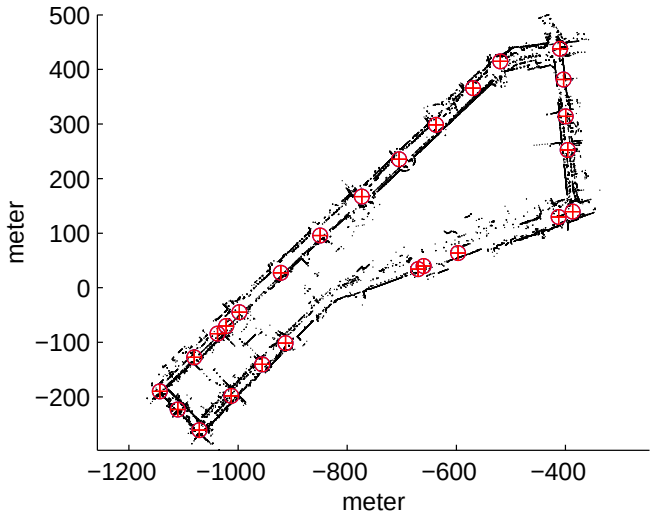


Figure 37: The same intersections shown in Figure 35 are overlayed on our reconstructed map.

SLAM with DATMO, in which the estimation problem is decomposed into two sperate estimators. By maintaining separate posteriors for the stationary and moving objects, the resulting problems are much lower dimensional than SLAM with generalized objects.

We implemented SLAM with DATMO and described practical algorithms which deal with the issues of perception modeling, data association and moving object detection. We have demonstrated that performing SLAM and moving object tracking concurrently satisfies both navigation and safety requirements in applications of intelligent transportation system and autonomous driving in urban areas. The substantial results indicated that it is indeed feasible to accomplish SLAMMOT from mobile robots at high speeds in crowded urban environments.

As future work, a project to generate quantitative SLAMMOT results would need to:

- characterize the sensors used and their errors.
- carefully characterize the errors of dead reckoning (odometry and heading measurements).
- instrument a few vehicles to be known moving objects, e.g. with accurate GPS or accurate pose estimation systems.
- carefully map a few points on the map to very high resolution, e.g. by using a theodolite to measure distances between corners of a few buildings, or by using carrier phase GPS at the level of the building rooftops, where multipath would not be a factor.

We also plan to investigate accomplishing SLAMMOT using heterogeneous sensors and performing higher level scene understanding such as interaction and activity learning.

A Derivation of SLAM with generic objects

In this section, we describe the formulation of SLAM with generic objects. Recall that the probabilistic formula of online SLAM with generic objects can be described as:

$$p(x_k, \mathbf{Y}_k \mid Z_k, U_k) \tag{16}$$

Using Bayes' rule, Equation 16 can be derived and rewritten as:

$$p(x_k, \mathbf{Y}_k \mid Z_k, U_k) \propto p(z_k \mid x_k, \mathbf{Y}_k)\, p(x_k, \mathbf{Y}_k \mid Z_{k-1}, U_k) \tag{17}$$

Using the total probability theorem, the rightmost term of Equation 17 can be rewritten as:

$$\begin{aligned} & p(x_k, \mathbf{Y}_k \mid Z_{k-1}, U_k) \\ = & \int\int p(x_k, x_{k-1}, \mathbf{Y}_k, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k)\, dx_{k-1}\, d\mathbf{Y}_{k-1} \end{aligned} \tag{18}$$

For deriving Equation 18 further, two situations are discussed: without and with interactions between the robot and generic objects.

A.1 Without Interaction

Here we assume that there is no interaction between the robot and generic objects. We can separate x_k and $\mathbf{Y}_k$ by:

$$\begin{aligned} & p(x_k, \mathbf{Y}_k, x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) \\ = \; & p(x_k \mid x_{k-1}, \mathbf{Y}_k, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \\ & \quad \cdot p(x_{k-1}, \mathbf{Y}_k, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) \\ = \; & p(x_k \mid x_{k-1}, \mathbf{Y}_k, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \\ & \quad \cdot p(\mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \\ & \quad \cdot p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) \end{aligned} \tag{19}$$

Following the no interaction assumption and the common independency assumptions in the SLAM literature, we can obtain the equations below:

$$p(x_k \mid x_{k-1}, \mathbf{Y}_k, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \;=\; p(x_k \mid x_{k-1}, u_k) \tag{20}$$

$$p(\mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \;=\; p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1}) \tag{21}$$

$$p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) = p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1}) \tag{22}$$

Putting all these equations into Equation 17, the Bayesian formula of online SLAM with generic objects is given as:

$$
\begin{aligned}
&\underbrace{p(x_k, \mathbf{Y}_k \mid Z_k, U_k)} \\
&\quad \text{Posterior at } k \\
&\propto \underbrace{p(z_k \mid x_k, \mathbf{Y}_k)} \int\int \underbrace{p(x_k \mid x_{k-1}, u_k)}\, \underbrace{p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1})} \\
&\qquad \text{Update} \qquad\qquad \text{Robot predict} \quad \text{generic objs} \\
&\qquad \cdot \underbrace{p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1})}\, dx_{k-1}\, d\mathbf{Y}_{k-1} \\
&\qquad\qquad \text{Posterior at } k-1
\end{aligned} \tag{23}
$$

Assuming that there is no interaction among generic objects, we can further derive:

$$
\begin{aligned}
p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1}) &= p(\mathbf{y}_k^1 \mid \mathbf{y}_{k-1}^1) p(\mathbf{y}_k^2 \mid \mathbf{y}_{k-1}^2) \cdots p(\mathbf{y}_k^l \mid \mathbf{y}_{k-1}^l) \\
&= \prod_{i=1}^{l} p(\mathbf{y}_k^i \mid \mathbf{y}_{k-1}^i)
\end{aligned} \tag{24}
$$

A.2 With Interactions

Assuming that interactions among the robot and the generic objects exist, x_k and $\mathbf{Y}_k$ are not decoupled and the right term of Equation 18 can be derived as

$$
\begin{aligned}
&p(x_k, \mathbf{Y}_k, x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) \\
&\quad = \; p(x_k, \mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \\
&\qquad \cdot p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_k) \cdot
\end{aligned} \tag{25}
$$

Then we take a reasonable independency assumption:

$$
\begin{aligned}
&p(x_k, \mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, Z_{k-1}, U_k) \\
&\quad = \; p(x_k, \mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, u_k)
\end{aligned} \tag{26}
$$

Putting all these equations into Equation 17, the Bayesian formula of online SLAM with generic objects is given as

$$
\begin{aligned}
&\underbrace{p(x_k, \mathbf{Y}_k \mid Z_k, U_k)} \propto \underbrace{p(z_k \mid x_k, \mathbf{Y}_k)} \\
&\quad \text{Posterior at } k \qquad\qquad \text{Update} \\
&\qquad \cdot \int\int \underbrace{p(x_k, \mathbf{Y}_k \mid x_{k-1}, \mathbf{Y}_{k-1}, u_k)} \\
&\qquad\qquad\qquad \text{Prediction with interactions} \\
&\qquad \cdot \underbrace{p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1})}\, dx_{k-1} d\mathbf{Y}_{k-1} \cdot \\
&\qquad\qquad \text{Posterior at } k-1
\end{aligned} \tag{27}
$$

B Derivation of SLAM with DATMO

In this section, we describe the formulation of online SLAM with DATMO.

B.1 Assumptions

Three assumptions are made to simplify the computation of SLAM with generic objects in which it is possible to update both the SLAM filter and the DATMO filter in real-time.

The first assumption is that measurements can be decomposed into measurements of stationary and moving objects.

$$z_k = z_k^o \cup z_k^m \quad \text{and hence} \quad Z_k = Z_k^o \cup Z_k^m \tag{28}$$

where z_k^m and z_k^o denote measurements of stationary and moving objects respectively. In particular this implies the following conditional independence

$$\begin{aligned} & p(z_k \mid x_k, O_k, M_k) \\ &= p(z_k^o \mid x_k, O_k, M_k)\, p(z_k^m \mid x_k, O_k, M_k) \\ &= p(z_k^o \mid x_k, O_k)\, p(z_k^m \mid x_k, M_k) \end{aligned} \tag{29}$$

where the variable $M_k = \{m_k^1, m_k^2, \ldots, m_k^q\}$ denotes the true locations of the stationary objects, of which there are q in the world at time k. The variable $O_k = \{\mathbf{o}_k^1, \mathbf{o}_k^2, \ldots, \mathbf{o}_k^n\}$ denotes the true hybrid states of the moving objects, of which there are n in the world at time k.

Now the general SLAM with DATMO problem can be posed as computing the posterior

$$p(x_k, M_k, O_k \mid Z_k, U_k) \tag{30}$$

The second assumption is that when estimating the posterior over the stationary object map and the robot pose, the measurements of moving objects carry no information about stationary landmarks and the robot pose, neither do their hybrid states O_k.

$$p(x_k, M_k \mid O_k, Z_k, U_k) = p(x_k, M_k \mid Z_k^m, U_k) \tag{31}$$

The third assumption is that there is no interaction among the robot and the moving objects. The robot and moving objects move independently of each other.

$$p(O_k \mid O_{k-1}) = \prod_{i=1}^{n} p(\mathbf{o}_k^i \mid \mathbf{o}_{k-1}^i) \tag{32}$$

B.2 Derivation

We begin by factoring out the most recent measurement:

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_k, U_k) \propto \\ & \quad p(z_k \mid x_k, O_k, M_k, Z_{k-1}, U_k)\, p(x_k, O_k, M_k \mid Z_{k-1}, U_k) \end{aligned} \tag{33}$$

Observing the standard Markov assumption, we note that $p(z_k \mid x_k, O_k, M_k, Z_{k-1}, U_k)$ does not depend on Z_{k-1}, U_k, hence we have

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_k, U_k) \propto \\ & \quad p(z_k \mid x_k, O_k, M_k)\, p(x_k, O_k, M_k \mid Z_{k-1}, U_k) \end{aligned} \tag{34}$$

Furthermore, we can now partition the measurement $z_k = z_k^o \cup z_k^m$ into moving and static, and obtain by exploiting the first assumption and Equation 29:

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_k, U_k) \propto \\ & \quad p(z_k^o \mid x_k, O_k)\; p(z_k^m \mid x_k, M_k)\; p(x_k, O_k, M_k \mid Z_{k-1}, U_k) \end{aligned} \tag{35}$$

The rightmost term $p(x_k, O_k, M_k \mid Z_{k-1}, U_k)$ can now be further developed, exploiting the second assumption

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_{k-1}, U_k) \\ & \quad = \; p(O_k \mid Z_{k-1}, U_k)\; p(x_k, M_k \mid O_k, Z_{k-1}, U_k) \\ & \quad = \; p(O_k \mid Z_{k-1}^o, U_k)\; p(x_k, M_k \mid Z_{k-1}^m, U_k) \end{aligned} \tag{36}$$

Hence we get for our desired posterior

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_k, U_k) \\ & \quad \propto \; p(z_k^o \mid x_k, O_k) p(z_k^m \mid x_k, M_k) \\ & \qquad \cdot\, p(O_k \mid Z_{k-1}^o, U_k) p(x_k, M_k \mid Z_{k-1}^m, U_k) \\ & \quad \propto \; \underbrace{p(z_k^o \mid x_k, O_k)\; p(O_k \mid Z_{k-1}^o, U_k)}_{\text{DATMO}} \\ & \qquad \cdot\, \underbrace{p(z_k^m \mid x_k, M_k)\; p(x_k, M_k \mid Z_{k-1}^m, U_k)}_{\text{SLAM}} \end{aligned} \tag{37}$$

The term $p(O_k \mid Z_{k-1}^o, U_k)$ resolves to the following prediction

$$\begin{aligned} & p(O_k \mid Z_{k-1}^o, U_k) \\ & \quad = \; \int p(O_k \mid Z_{k-1}^o, U_k, O_{k-1})\; p(O_{k-1} \mid Z_{k-1}^o, U_k)\; dO_{k-1} \\ & \quad = \; \int p(O_k \mid O_{k-1})\; p(O_{k-1} \mid Z_{k-1}^o, U_{k-1})\; dO_{k-1} \end{aligned} \tag{38}$$

Finally, the term $p(x_k, M_k \mid Z_{k-1}^m, U_k)$ in Equation 37 is obtained by the following step:

$$\begin{aligned} & p(x_k, M_k \mid Z_{k-1}^m, U_k) \\ & \quad = \; p(x_k \mid Z_{k-1}^m, U_k, M_k)\; p(M_k \mid Z_{k-1}^m, U_k) \\ & \quad = \; \int p(x_k \mid Z_{k-1}^m, U_k, M_k, x_{k-1})\; p(x_{k-1} \mid Z_{k-1}^m, U_k, M_k) \\ & \qquad \cdot\, p(M_k \mid Z_{k-1}^m, U_k)\; dx_{k-1} \\ & \quad = \; \int p(x_k \mid u_k, x_{k-1}) p(x_{k-1}, M_k \mid Z_{k-1}^m, U_{k-1}) dx_{k-1} \end{aligned} \tag{39}$$

which is the familiar SLAM prediction step. Putting everything back into Equation 37 we now obtain the final filter equation:

$$\begin{aligned} & p(x_k, O_k, M_k \mid Z_k, U_k) \\ & \quad \propto \; \underbrace{p(z_k^o \mid O_k, x_k)}_{\text{DATMO Update}} \end{aligned} \tag{40}$$

$$\cdot \quad \underbrace{\int p(O_k \mid O_{k-1})\, p(O_{k-1} \mid Z^o_{k-1}, U_{k-1})\, dO_{k-1}}_{\text{DATMO Prediction}}$$

$$\cdot \quad \underbrace{p(z^m_k \mid M_k, x_k)}_{\text{SLAM Update}}$$

$$\cdot \quad \underbrace{\int p(x_k \mid u_k, x_{k-1})\, p(x_{k-1}, M_{k-1} \mid Z^m_{k-1}, U_{k-1})\, dx_{k-1}}_{\text{SLAM Prediction}}$$

From Equation 40, input to this SLAM with DATMO filter are two separate posteriors, one of the conventional SLAM form and a separate one for DATMO. How to recover those SLAM and DATMO posteriors at time k is simple.

For the SLAM part, we get

$$\begin{aligned} & p(x_k, M_k \mid Z^m_k, U_k) \\ = \;& \int p(O_k, M_k, x_k \mid Z_k, U_k)\, dO_k \\ \propto \;& p(z^m_k \mid M_k, x_k) \cdot \\ & \int p(x_k \mid u_k, x_{k-1})\, p(x_{k-1}, M_{k-1} \mid Z^m_{k-1}, U_{k-1})\, dx_{k-1} \end{aligned} \tag{41}$$

For the DATMO part , we get

$$\begin{aligned} & p(O_k \mid Z_k, U_k) \\ = \;& \int\int p(O_k, M_k, x_k \mid Z_k, U_k) dM_k dx_k \\ \propto \;& \int (p(z^o_k \mid O_k, x_k) \int p(O_k \mid O_{k-1}) \\ & p(O_{k-1} \mid Z^o_{k-1}, U_{k-1}) dO_{k-1}) p(x_k \mid Z^m_k, U_k) dx_k \end{aligned} \tag{42}$$

where the posterior over the pose $p(x_k \mid Z^m_k, U_k)$ is simply the marginal of the joint calculated in Equation 41:

$$p(x_k \mid Z^m_k, U_k) \;=\; \int p(x_k, M_k \mid Z^m_k, U_k) dM_k \tag{43}$$

C Index to Multimedia Extensions

The multimedia extensions to this article are at: http://www.ijrr.org. Extension 1 is a video that illustrates the processing of the SLAM with DATMO approach (Figure 22). Extension 2 is a video that shows the processing of loop closing (Figure 13). Extension 3 and 4 are the videos illustrate the processing of multiple vehicle tracking (Figure 24) and multiple pedestrian tracking (Figure 28), respectively. Extension 5 is a video shows the 3D city modeling results (Figure 29, Figure 30 and Figure 31).

Extension	Type	Description
1	Video	SLAM with DATMO
2	Video	Loop closing
3	Video	Multiple vehicle tracking
4	Video	Multiple pedestrian tracking
5	Video	City mapping

Acknowledgment

Thanks to the members of the CMU Navlab group for their excellent work on building and maintaining the Navlab11 vehicles, and for their helps on collecting data. We also acknowledge the helpful suggestions by anonymous reviewers. This work was funded in part by the U.S. Department of Transportation; the Federal Transit Administration; by Bosch Corporation; by SAIC Inc.; and by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The ongoing project is partially supported by grants from Taiwan NSC (#94-2218-E-002-077, #94-2218-E-002-075,#95-2218-E-002-039, #95-2221-E-002-433); Excellent Research Projects of National Taiwan University (#95R0062-AE00-05); Taiwan DOIT TDPA Program (#95-EC-17-A-04-S1-054); Australia's CSIRO; and Intel.

References

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

Anguelov, D., Koller, D., Parker, E., and Thrun, S. (2004). Detecting and modeling doors with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Arnold, J., Shaw, S., and Pasternack, H. (1993). Efficient target tracking using dynamic programming. *IEEE Transactions on Aerospace and Electronic Systems*, 29(1):44–56.

Bar-Shalom, Y. and Li, X.-R. (1988). *Estimation and Tracking: Principles, Techniques, and Software*. YBS, Danvers, MA.

Bar-Shalom, Y. and Li, X.-R. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, Danvers, MA.

Biswas, R., Limketkai, B., Sanner, S., and Thrun, S. (2002). Towards object mapping in dynamic environments with mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland.

Blackman, S. and Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA.

Blom, H. A. P. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Trans. On Automatic Control*, 33(8).

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalavle mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Brand, M. (2001). Morphable 3d models from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Bregler, C., Hertzmann, A., and Biermann, H. (2000). Recovering non-rigid 3d shape from image streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Choset, H. and Nagatani, K. (2001). Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–136.

Christensen, H. I., editor (2002). *Lecture Notes: SLAM Summer School 2002*.

Coraluppi, S. and Carthel, C. (2001). Multiple-hypothesis IMM (MH-IMM) filter for moving and stationary targets. In *Proceedings of the International Conference on Information Fusion*, pages TuB1–18–23, Montréal, QC, Canada.

Coraluppi, S., Luettgen, M., and Carthel, C. (2000). A hybrid-state estimation algorithm for multi-sensor target tracking. In *Proceedings of the International Conference on Information Fusion*, pages TuB1–18–23, Paris, France.

Cox, I. J. and Hingorani, S. L. (1996). An efficient implemenation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 18(2).

Deans, M. C. (2005). *Bearings-Only Localization and Mapping*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Dellaert, F., Burgard, W., Fox, D., and Thrun, S. (1999). Using the condensation algorithm for robust, vision-based mobile robot localization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*.

Demirdjian, D. and Horaud, R. (2000). Motion-egomotion discrimination and motion segmentation from image-pair streams. *Computer Vision and Image Understanding*, pages 53–68.

Durrant-Whyte, H., Majumder, S., Thrun, S., de Battista, M., and Scheding, S. (2003). A bayesian algorithm for simultaneous localization and map building. In *The tenth International Symposium on Robotics Research*, volume 6, pages 44–66.

Elfes, A. (1988). *Occupancy Grids as a Spatial Representation for Mobile Robot Mapping and Navigation*. PhD thesis, Electrical and Computer Engineering/Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Elfes, A. (1990). *Autonomous Robot Vehicles*, chapter Sonar-based real-world mapping and navigation. Springer, Berlin.

Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8:173–184.

Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*.

Gish, H. and Mucci, R. (1987). Target state estimation in a multi-target environments. *IEEE Transactions on Aerospace and Electronic Systems*, 23:60–73.

Guivant, J. E. and Nebot, E. M. (2001). Optimization of the simultaneous localization and map building algorithm for real-time implementation. *IEEE Transaction on Robotics and Automation*, 17:242–257.

Hähnel, D., Burgard, W., Wegbreit, B., and Thrun, S. (2003). Towards lazy data association in SLAM. In Chatila, R. and Dario, P., editors, *Proceedings of the 11th International Symposium of Robotics Research*, Siena, Italy.

Hähnel, D., Schulz, D., and Burgard, W. (2002). Map building with mobile robots in populated environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland.

Hähnel, D., Triebel, R., Burgard, W., and Thrun, S. (2003). Map building with mobile robots in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., and Fisher, R. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kantor, G. A. and Singh, S. (2002). Preliminary results in range-only localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Kirubarajan, T. and Bar-Shalom, Y. (2000). Tracking evasive move-stop-move targets with an MTI radar using a VS-IMM estimator. In *Proceedings of the SPIE Signal and Data Processing of Small Targets*, volume 4048, pages 236–246.

Leonard, J. and Durrant-Whyte, H. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1442–1447.

Leonard, J. J. and Feder, H. J. S. (1999). A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach, D. K., editor, *Proceedings of the Ninth International Symposium on Robotics Research*.

Lu, F. and Milios, E. (1994). Robot pose estimation in unknown environments by matching 2d range scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–938, Seattle, WA.

Lu, F. and Milios, E. (1997). Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275.

Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123.

Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Montesano, L., Minguez, J., and Montano, L. (2005). Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Newman, P. M. and Leonard, J. J. (2003). Pure range-only subsea slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Olson, C. F. (2000). Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66.

Paskin, M. A. (2003). Thin junction tree filters for simultaneous localization and mapping. In Gottlob, G. and Walsh, T., editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1157–1164.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Trans. On Automatic Control*, 24(6).

Rogers, S. (1988). Tracking mulitple targets with correlated measurements and maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-24(3):313–315.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithms. In *Proc. the Third Int. Conf. on 3-D Digital Imaging and Modeling*, pages 145–152.

Shea, P. J., Zadra, T., Klamer, D., Frangione, E., and Brouillard, R. (2000). Percision tracking of ground targets. In *Proceedings of the IEEE Aerospace Conference*, volume 3, pages 473–482.

Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–58.

Smith, R. C., Self, M., and Cheeseman, P. (1990). *Autonomous Robot Vehicles*, chapter Estimating Uncertain Spatial Relationships in Robotics. Springer, Berlin.

Stewart, B., Ko, J., Fox, D., and Konolige, K. (2003). The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, Siena, Italy.

Thorpe, C. and Durrant-Whyte, H. (2001). Field robots. In *the 10th International Symposium of Robotics Research*, Lorne, Victoria, Australia. Springer-Verlag.

Thrun, S. (2002). Robotic mapping: A survey. Technical Report CMU-CS-02-111, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Thrun, S., Fox, D., and Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*.

Thrun, S., Koller, D., Ghahramani, Z., Durrant-Whyte, H., and Ng., A. (2002). Simultaneous mapping and localization with sparse extended information filters:theory and initial results. Technical Report CMU-CS-02-112, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA.

Thrun, S. and Liu, Y. (2003). Multi-robot SLAM with sparse extended information filers. In Chatila, R. and Dario, P., editors, *Proceedings of the 11th International Symposium of Robotics Research*, Siena, Italy.

Torresani, L., Yang, D., Alexander, G., and Bregler, C. (2001). Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tugnait, J. K. (1982). Detection and estimation for abruptly changing systems. *Automatica*, 18:607–615.

Wang, C.-C. (2004). *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Wang, C.-C., Duggins, D., Gowdy, J., Kozar, J., MacLachlan, R., Mertz, C., Suppe, A., and Thorpe, C. (2004). Navlab SLAMMOT datasets. www.cs.cmu.edu/~bobwang/datasets.html. Carnegie Mellon University.

Wang, C.-C., Lo, T.-C., and Yang, S.-W. (2007). Interacting object tracking in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Wang, C.-C. and Thorpe, C. (2002). Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC.

Wang, C.-C. and Thorpe, C. (2004). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan.

Wang, C.-C., Thorpe, C., and Suppe, A. (2003a). Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Columbus, OH.

Wang, C.-C., Thorpe, C., and Thrun, S. (2003b). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Wolf, D. F. and Sukhatme, G. S. (2005). Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65.

Xiao, J., Chai, J., and Kanade, T. (2004). A closed-form solution to non-rigid shape and motion recovery. In *The 8th European Conference on Computer Vision (ECCV 2004)*.