JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 5, May 2008

Real-time Obstacle Map Building with Target Tracking

Yeonsik Kang* , Derek S. Caveney† , and J. Karl Hedrick‡
University of California, Berkeley, Berkeley, CA, 94720, USA

DOI: 10.2514/1.29210

In this paper, a new method is proposed to build a probabilistic occupancy map for an unmanned aerial vehicle (UAV) equipped with a forward-looking sensor, such as a laser scanning sensor (known as lidar). For a UAV, target tracking as well as mapping of obstacles are both important. Instead of using raw measurements to build a map, the proposed algorithm uses the interacting multiple model (IMM)-based target formulation and tracking method first to process the noisy measurement data. The state estimates and true target probability of each point-mass target tracks are then used to build a probabilistic occupancy map. Therefore, simultaneous tracking and mapping of both moving and stationary obstacles are accomplished in real time. In addition, the mapping algorithm has the robustness to the noisy sensor measurements. The obtained probabilistic occupancy map shows good agreement with the physical layout of the obstacles in the field in simulations. This shows the potential that the developed method can be used to help an unmanned vehicle navigate the field without a previous database of obstacles.

I. Introduction

FOR an unmanned aerial vehicle (UAV), it is essential to be able to identify a safe path and to localize its own position to perform a mission. Also, to perform a collision avoidance maneuver, the tracking of moving targets is necessary. A forward-looking sensor such as a radar or laser scanning sensor (also known as lidar) can help a UAV detect moving or stationary obstacles, especially when flying in an outdoor environment. There are many tracking algorithms based on the Kalman filter specialized in tracking a moving target. However, it is also necessary to produce a two-dimensional (2-D) map of stationary obstacles to plan a safe path for the UAV. The method proposed in the present paper can achieve the tracking of both moving and stationary obstacles in a computationally efficient manner. The proposed method uses the outputs from the point mass target-tracking algorithm to generate a map of stationary obstacles. Therefore, the complex data association of a large set of measurements is solved only once.

In this paper, a new approach to build the probabilistic occupancy grid map based on the existing point-mass target-tracking algorithm will be presented, assuming that the position of the vehicle is known. Probabilistic occupancy grid mapping has been very popular in robotic navigation applications and many papers have dealt with methods to update the known or unknown map and localize the position of the robot based on the measurements from sensors such as sonar, lidar, or computer vision [1–6]. The map update law has been extended to filter out the measurements associated with dynamic obstacles outlined by Hahnel et al. [7] and Biswas et al. [8]. Recently, the probabilistic

Received 11 December 2006; accepted for publication 20 March 2008.

* Postdoctoral Reseracher, Department of Mechanical Engineering in University of California, Berkeley, 2162 Etcheverry Hall, Berkeley, CA 94720, USA, yeonsikkang@gmail.com.
† Postdoctoral Researcher Department of Mechanical Engineering in University of California, Berkeley, Berkeley, CA 94720, USA.
‡ James Marshall Wells Professor, Department of Mechanical Engineering in University of California, Berkeley, 5104 Etcheverry Hall, Berkeley, CA 94720, USA.

occupancy grid map was constructed using a quadtree-based approach, which facilitates access to the information in the map and manages the memory efficiently [9,10]. The quadtree algorithm has been extensively developed in the application of computer graphics and image processing [11,12].

In the current paper, the Kalman filter-based interacting multiple model (IMM) technique is used as a measurement model and the probabilistic map is built recursively. The localization of the unmanned vehicle is not addressed in this paper since our goal is to provide a fast and robust mapping algorithm for an unmanned vehicle equipped with a precise position sensor suite of global positioning system [GPS] and inertial navigation system (INS). Therefore, it is assumed that the position of the vehicle can be measured or estimated. There have been numerous efforts to localize the position of the UAV or the target relying on sensors, such as computer vision or laser scanner [5,13–16].

The IMM method was developed by the aerospace industry to track aircraft trajectories using radar measurements [17]. The application of this method to the automatic track formulation problem is addressed by Bar-Shalom [18].

In conventional mapping algorithms [1,4,5,19], noisy measurements are directly used to calculate the probability of occupancy grids, so the effect of noise may significantly deteriorate the result. The proposed mapping algorithm can reduce the effect of noise by setting a cutoff probability on targets to be used to populate the occupancy grid. In addition, a fast-moving obstacle can also be detected and the position of the moving obstacle is tracked using the Kalman-filter-based tracking algorithm. The derived recursive algorithm can be calculated fast enough to run in real-time applications such as UAV navigation. These advantages will be explained in the following sections by simulation results.

In Sec. II the IMM-based track formulation algorithm will be explained. In Sec. III the developed recursive map building algorithm will be presented. In Sec. IV a practical modification on the tracking algorithm will be presented to apply the method to the laser scanning finder type of sensor. Simulation results in 2-D cases will be presented in Sec. V followed by conclusions in Sec. VI.

II. Interacting Multiple Model Algorithm for Obstacle Detection

This section explains the IMM-based track-formulation algorithm introduced by Bar-Shalom [18]. In this algorithm, the true target probability (TTP) of each target can be calculated and the target with a TTP less than a certain lower threshold will be removed. After a TTP reaches a certain upper threshold, the algorithm will declare that an obstacle is detected. This TTP with the position estimates and estimation covariances will be used in Sec. III to build a probabilistic occupancy map.

To associate multiple measurements on a target, the probabilistic data association filter (PDAF) is used, which is a suboptimal Bayesian algorithm [20]. The PDAF is used to avoid the exponential growth of memory and computation requirements of the optimal data association, which would result in an algorithm that was difficult to implement on a fast real-time application.

The IMM algorithm is a stochastic state estimator using multiple Kalman filters to track and estimate the states of a dynamic system. In the current paper, an IMM algorithm uses two PDAFs to detect obstacles. Each depends on a different hypothesis but uses the same dynamics as follows

$$x(k+1) = Ax(k) + Bw(k) \tag{1}$$

$$z(k) = Cx(k) + v(k) \tag{2}$$

where $w(k)$ and $v(k)$ denote the mutually independent Gaussian noise processes with zero mean and covariances $Q(k)$ and $R(k)$ respectively. $x(k)$ and $z(k)$ denote the state and measurement vectors of the target as follows

$$x(k) = \left[\xi(k), \dot{\xi}(k), \eta(k), \dot{\eta}(k)\right]^T \tag{3}$$

$$z(k) = [\xi(k), \eta(k)]^T \tag{4}$$

where $\xi(k)$ and $\eta(k)$ denote the position of the target according to a Cartesian coordinate reference frame.

The details of the IMM algorithm for the obstacle detection problem are explained by Bar-Shalom [18], Bar-Shalom et al. [20], and Kang et al. [21]. The IMM algorithm is composed of four major steps and Fig. 1 shows the schematic of the IMM algorithm.

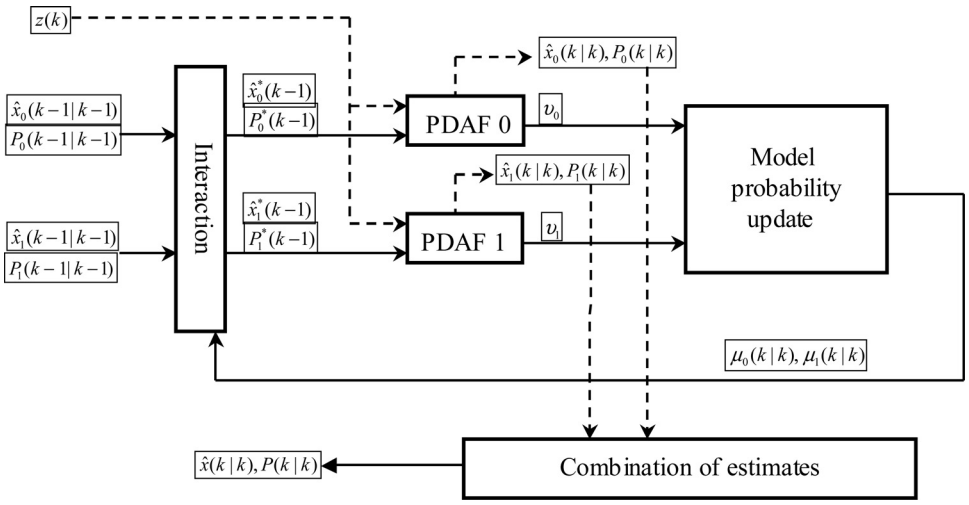

Fig. 1 Four major steps of IMM algorithm.

The two PDAFs shown in Fig. 1 are conditioned on two different events, respectively. In particular, the PDAF 0 is conditioned on the event that there is no obstacle in the area of interest while the PDAF 1 is conditioned on the event there is an obstacle in the area of interest. These two different events differentiate the two PDAFs by the values of detection probability P_D, which is defined as the probability that the true target is detected. For PDAF 0, the P_D is set to zero as it is not possible to obtain true measurement and for PDAF 1 the P_D is set to a value determined by sensor specifications.

In the first step of the IMM method, the initial conditions for each model PDAF $\hat{x}_j^*(k-1)$ and $P_j^*(k-1)$, are obtained by mixing the previous state estimates of each filter. The mixing is based on the model probability at the previous step and the Markov transition matrix between the events on which the PDAFs are based.

In the second step, the prediction through a Kalman-filter-based PDAF is made based on the mixed estimates of the previous iteration. The measurements validated through the validation gate are used for the innovation of the Kalman filter. The volume of the validation gate is

$$V(k) = g^2 \pi |S(k)|^{1/2} \tag{5}$$

where g is determined by the chi-square distribution depending on the gating probability P_G, and $S(k)$ denotes the measurement covariance matrix of the Kalman filter. Therefore, only the measurements that are close enough to the predicted measurement are associated with the estimates.

The model probability $\mu_j(k|k)$ of each filter is calculated recursively in the third step using the likelihood function, which represents the sensor model. The model probability $\mu_1(k|k)$ becomes the TTP of each track and the track is removed if the TTP is below a certain lower threshold. The TTPs with position estimates and covariances of the tracks will be used in calculating the probability of occupancy of each grid cell in Sec. III.

Finally, in the fourth step, the combined estimate and its covariances are calculated by using each filter's updated states, covariances, and model probabilities.

III. Probabilistic Map Building

In this section, the proposed algorithm to update the probabilistic occupancy map will be described. To use the obstacle information extracted from the sensor measurements for path planning or obstacle avoidance, it is desired to produce a map. In this study, we use the probabilistic occupancy grid map to represent the occupancy of the environment as it fits very well with the developed IMM-based obstacle detection as well as popular path planning algorithms for unmanned vehicles. In probabilistic occupancy grid mapping, the areas of interest are divided into grid cells and the value between 0 to 1 is assigned to each grid cell, which represents the probability that the cell is

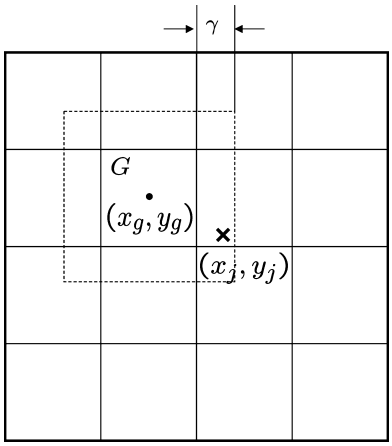

Fig. 2 The grid cell G and the nearby pth target.

occupied. If we want to convert these continuous values into binary values of either 0 or 1 to represent occupancy, we may set the threshold and assign occupancy of the grid cell if the probability is higher than this threshold.

The outputs from the IMM obstacle detection algorithm are used to update the probability of the grid cell that has nearby tracks. The necessary inputs for the developed probabilistic map update algorithm are the state estimates with covariances and the TTP of the tracks. For the grid cell without any nearby tracking, a different update law is developed based on Bayes rule. These two update laws will be described in subsections A and B, respectively.

A. Map Update Law with Nearby Tracking Information

The probability of occupancy of a grid cell, G, based on the cumulative sensor measurements Z^k is denoted by $P(\text{occ}|Z^k)$. The center of the grid cell G shown in Fig. 2 is denoted by (x_g, y_g) in Cartesian coordinates. The pth track position is denoted by $Y_p = (x_p, y_p)$. Suppose that the number of the tracks which exist near the cell G at time k is $N_p > 0$. A track is determined to be near the cell if the closest distance to the cell is smaller than a threshold γ. Therefore, the track farther than γ distance from the cell will not affect the probability of occupancy of the cell. In this paper, the square root of the maximum eigenvalue of $S(k|k)$ is used as γ since the track that is not located within this box has only a small portion of its distribution in the box. In the IMM algorithm, only the measurements validated through the validation gate, $V(k)$ are associated to update each track, therefore, the Z^k used to update a grid cell denotes the cumulative "validated" measurements of the nearby tracks.

In this paper, two basic assumptions are made to derive $P(occ|Z^k)$. The first assumption is that the probability of the occupancy of a cell is independent from the occupancy of the other cells. This assumption may not be realistic because outdoor environments are usually composed of larger objects and the knowledge of occupancy of one cell may help increase the efficiency of the algorithm on the neighboring cells. However, the probability of occupancy of a cell is calculated independently from other cells, which is common in many map update algorithms [6]. The second assumption is that each cell is occupied by only a single target track. There is a chance that the cell may be occupied by many target tracks, but in this paper, we will consider it as not possible. This assumption will not only simplify the derivation, but also make the algorithm more conservative. This second assumption can be written as

$$\text{occ} = \bigcup_{p=1}^{N_p} \text{occ}_p, \quad \text{occ}_p \cap \text{occ}_q = \phi \ \ if \ \ p \neq q \tag{6}$$

where occ_p denotes the event that the pth track exists and occupies the grid cell G. The event is defined by

$$\text{occ}_p = \left[(x_p, y_p)|(x_p, y_p) \in G, m_{1p}\right] \tag{7}$$

where m_{1p} denotes the event that the pth is a true target. Following these two assumptions, the probability of the occupancy of G is calculated by

$$\begin{aligned} P(\text{occ}|Z^k) &= P\left(\bigcup_{p=1}^{N_p} \text{occ}_p|Z^k\right) \\ &= \sum_{p=1}^{N_p} P(\text{occ}_p|Z^k) \\ &= \sum_{p=1}^{N_p} P_{Y_p}\left[(x_p, y_p) \in G, m_{1p}|Z^k\right] \end{aligned} \tag{8}$$

The second equality holds because of the assumption given by Eq. (6). The third equality is from the definition of the event occ_p. The joint density function in the third line of Eq. (8) is not easy to calculate directly. Bayes rule enables us to write this density function in an easier form to calculate

$$\begin{aligned} P_{Y_p}\left[(x_p, y_p) \in G, m_{1p}|Z^k\right] &= P_{Y_p}\left[(x_p, y_p) \in G|m_{1p}, Z^k\right] P(m_{1p}|Z^k) \\ &= \int_G \mathcal{N}(Y_p; \hat{Y}_{1p}; \hat{S}_p)\mu_{1p}(k|k) \end{aligned} \tag{9}$$

Conditioned on the event that the track is originating from a real obstacle, we can represent its distribution by using a Gaussian normal distribution with its mean and variance calculated from the Kalman filter estimates. The Gaussian distribution of the Y_p has its mean as $\hat{Y}_{1p} = C\hat{x}_{1p}(k|k)$ and covariance as $\hat{S}_p = C P_{1p} C^T$. The $P(m_{1p}|Z^k)$ is exactly the definition of the TTP, $\mu_{1p}(k|k)$. The estimates we need from the IMM algorithm to update the probabilistic occupancy grid map are the $\hat{Y}_p$, P_{1p} taken from the state and state covariance estimates, and the TTP $\mu_{1p}(k|k)$. The bivariate Gaussian distribution of $\mathcal{N}(Y_p; \hat{Y}_{1p}; \hat{S}_p)$ can be integrated over the grid cell numerically. The numerical integration procedure of the bivariate distribution is described by Genz [22].

The second assumption given by Eq. (6) greatly simplifies the derivation of the probability of occupancy. However, this assumption is violated in many cases and the resulting $P(occ|Z^k)$ is overestimated because the probability $P(occ_p \cap occ_q)$ is not zero generally. Although the assumption may be violated often, the overestimation of the probability of occupancy is not a significant problem, in fact it gives us a margin of safety.

Finally the update law is derived as

$$P(occ|Z^k) = \sum_{p=1}^{N_p} \int_G \mathcal{N}(Y_p; \hat{Y}_{1p}; \hat{S}_p)\mu_{1p}(k|k) \tag{10}$$

which is intuitive because it is calculated by the summation of integrations of all the targets' Gaussian probability distributions weighted by their TTP.

The map update law described in this subsection is the case when $N_p > 0$. If $N_p = 0$, no update will be made by the above law. The update law corresponding to such a case will be described in the next subsection.

B. Map Update Law without Nearby Tracking Information

The map update in the previous subsection is done using the validated measurements of the grid cell Z^k. However, if there is no track near the grid cell, there is no validated measurement around the grid cell. Even though, the observation of "empty" validated measurements can be used to develop map update law.

The update law is derived using Bayes rule by

$$
\begin{aligned}
P(\text{occ}|Z^k) = P(\text{occ}|Z(k)=\phi, Z^{k-1}) &= \frac{P(Z(k)=\phi\,|\text{occ}, Z^{k-1})P(\text{occ}|Z^{k-1})}{P(Z(k)=\phi\,|Z^{k-1})} \\
&= \frac{P(Z(k)=\phi\,|\text{occ})P(\text{occ}|Z^{k-1})}{P(Z(k)=\phi\,|\text{occ})P(\text{occ}|Z^{k-1}) + P(Z(k)=\phi\,|\text{unocc})P(\text{unocc}|Z^{k-1})} \\
&= \frac{(1-P_D^1)P(occ|Z^{k-1})}{(1-P_D^1)P(\text{occ}|Z^{k-1}) + (1-P_D^0)(1-P(\text{occ}|Z^{k-1}))}
\end{aligned} \tag{11}
$$

where P_D^1 is the detection probability, P_D^0 is the probability of false detection, and ϕ denotes the empty set. These variables are characteristics of the sensor type. The first line of Eq. (11) is the result of Bayes rule. The second line uses the assumption that the conditional probability distribution of measurement at k given the occupancy of the grid cell is independent from the previous measurement. This is a common assumption made in the occupancy grid map approach such as by Thrun [6]. Also in the second line of Eq. (11), the total probability theorem is used to expand $P(Z(k)=\phi\,|Z^{k-1})$. In the third line, an approximation is made to consider that the validated measurement $Z(k)$ is the same as the true measurement. Although they are not exactly same, the possibility of an incorrect measurement being validated is neglected because it is very small. In the third line of Eq. (11), P_D^1 represents the probability of getting a correct measurement of the grid cell which is occupied. This is $P(Z(k)\neq\phi\,|\text{occ})$ by definition. P_D^0 represents the probability that there is a true sensor return by the unoccupied grid cell, which is $P(Z(k)\neq\phi\,|\text{unocc})$ and should be zero or very small. If P_D^1 is high, then the algorithm tends to trust the sensor measurement, whereas if P_D^1 is low, then the algorithm is reluctant to increase TTP by a single measurement around the grid cell.

IV. Implementation of the Algorithm

The overall structure of the algorithm is shown in Fig. 3. First the sensor measurements are fed to the IMM-based obstacle detection algorithm. The algorithm outputs the point-mass state estimates, covariances, and TTP of the existing tracks. Also from the sensor measurements, we estimate the sensor's field of view and collect the grid cells that are within the sensor's field of view. The probabilities of occupancy of the grid cells are updated by the recursive map updating algorithm based on the outputs from IMM-based point-mass target tracking algorithm.

The map update is performed when the grid cell G is visible from the sensor. A specific type of sensor of interest in this paper is the lidar. Therefore, a lidar model is developed which mimics the physics of the real lidar. The lidar model does an intersection test between the sensor ray and the edge of an obstacle and if there is an intersection, it returns the distance and the relative angle of the intersecting point from the unmanned aircraft mixed with simulated random clutter and measurement noise. Each time step, the cells that are visible according to the position of the sensor and sensor specifications are collected and updated by the law described in Sec. III. Since it takes a few steps for the obstacle detection algorithm to initiate a new track or to terminate a track that is no longer in the sensor field of view, we give a margin when selecting the updated grid cells as in Fig. 4.

The tracking of a large obstacle induces the drifting of the tracks on the obstacle's surface and this is caused by movement of the reflection points as the unmanned vehicle moves. As a result, some of the tracks merge to the same spot of the surface and become redundant tracks. To eliminate redundant tracks around the same position, we used a routine that eliminates a track that is too close to another track which was generated earlier than itself.

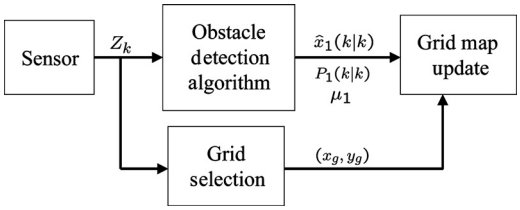

Fig. 3 Structure of the algorithm.

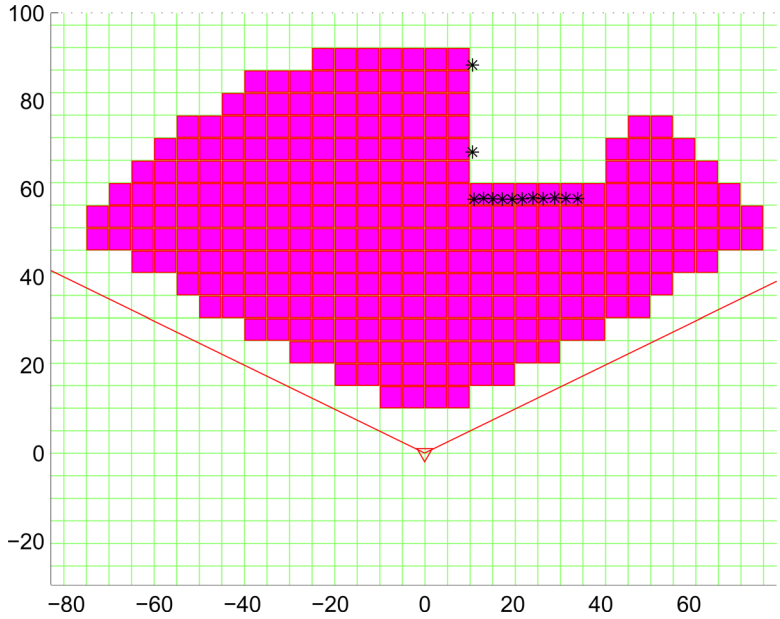

Fig. 4 The grid cells in the update region.

This routine avoids keeping too many tracks near the same position, therefore we can manage the number of the tracks we maintain. The lidar measurements from a large obstacle are difficult to associate since it is not guaranteed that measurements can be repeatedly obtained from the same spot. It is also possible that several measurements are associated with a single track, causing the size of the state covariance matrix to increase. In this paper, in addition to the existing measurement validation gate, association by a distance limit is applied, which limits the association of measurements to those within a certain distance of the track. This additional association limit lets us instantiate a new track near the grid cell from which the track drifted away, thereby maintaining good distribution of the tracks to represent the shape of the obstacle in the occupancy grid map. The association limits should be less than the size of the occupancy grid cell because of the resolution problem. Since we use the quantized map based on the occupancy grid, the distribution of the point-mass target tracks are important to represent a continuous probability distribution in a quantized map. In particular, if the estimated position of the obstacle is on the boundary of two grid cells, both of the cells should be declared to be occupied to prevent collision. However, if there is only one target track on the edge of two cells, two cells can have a probability of occupancy of 0.5 each at most. This is undesirable. In order to eliminate this possibility, at least a couple of target tracks should be maintained within one grid cell on the surface of a large obstacle. This was accomplished by putting association limits to the validation gate $V(k)$ and limiting the maximum size of the gate to the association limit. The effect of the association limit is discussed in Sec. V.

V. Simulation Results

Table 1 gives the specifications of the sensor model. Since a UAV is considered in a noisy outdoor environment, the specifications used in Table 1 may show larger standard deviation and heavier clutter noise than the specifications of a standard lidar. The sensor model scans in a 2-D horizontal plane, which is common in many lidar applications. In Fig. 5, the obstacle layouts and the path of the unmanned vehicle are plotted. The thick solid line shows the trajectory of the UAV and the thin dotted line shows the trajectory of the moving target. The velocity of the UAV in simulation is around 7–10 m/s following the path in clockwise direction. The velocity of the moving target is around 10 m/s. The obstacle layout used in the simulation is provided by the NASA Ames Research Center.

A. IMM Obstacle Detection Result

In this subsection, the outputs from the IMM-based obstacle detection algorithms are presented to show the performance and behavior of the algorithm using the lidar sensor model. The parameters for the IMM-based obstacle

Table 1 Lidar specification in Matlab simulation

Maximum range	100 m
Horizontal field of view	180°
Horizontal resolution	1°
Update rate	10 Hz
Clutter noise	2.5 per scan on average
Measurement noise	Gaussian distribution with 0 mean and 50 mm standard deviation

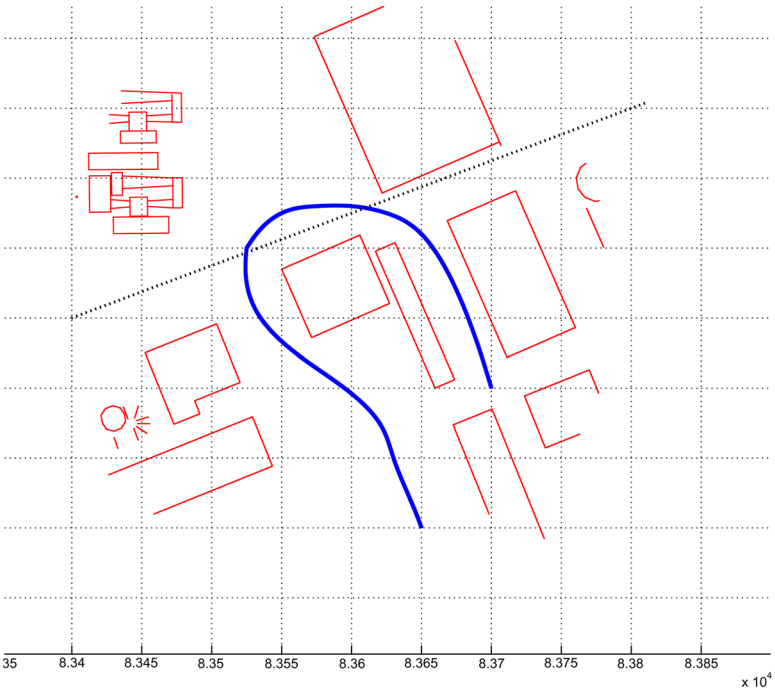

Fig. 5 The simulation trajectory of the vehicle (dots) and obstacle layouts (lines).

detection algorithm are as shown in Table 2. The P_D^0 should be zero in theory but, to avoid any divide by zero error, it is set to a small value. A track is terminated after its TTP $\mu_1(k)$ reaches the lower threshold of termination probability of 0.2. In Fig. 6, tracking results in the middle of the flight path are shown. The solid lines represent the cross sections of the obstacles in the field. The thick circles represent the tracks. The dotted lines represent the sensor beams. The position of the helicopter is represented by thin circles with its heading angle represented by a short line. It is shown that several tracks are generated only on the surface of the real obstacles, although there is clutter noise mixed in the measurements. In Fig. 6a, the algorithm ran with 3 m association limits mentioned in Sec. IV. With this association limit, the tracks are distributed evenly on the surface of the large obstacle. In Fig. 6b, the algorithm ran without the association limit. In Fig. 6b the tracks on the front wall drifted to the center, although they had been located evenly when they were generated. At this moment it was observed that the measurement covariances of the tracks on the front wall increased. Accordingly, the measurement gate size also increased and the measurements at the edge of the wall were associated to the tracks which drifted toward the center of the wall. Therefore, although there were only three tracks on the surface of the wall in front of the UAV's path, no more tracks were generated.

With the association limit, even though the tracks drifts toward the center, new tracks are generated at the edge of the wall since its measurements are not associated with the old tracks. Therefore, cells on the surface of the wall can maintain a high probability of occupancy. The number of the total tracks existing in Fig. 1a is 53, while there are only 12 tracks in Fig. 1b. Therefore, the obstacle layouts can be represented better with association limits.

Table 2 IMM-based obstacle detection algorithm parameters

P_D^0, P_D^1	0.01, 0.9
Track fusion distance	1 m
Association limit	3 m
The probability threshold of track termination	0.2

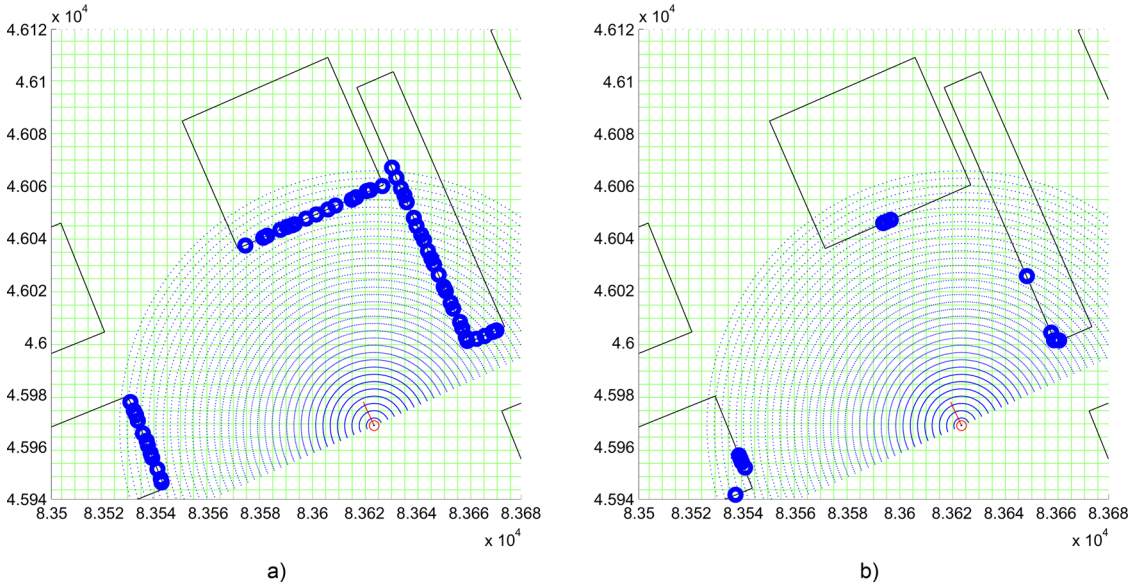

Fig. 6 The tracks generated by IMM-based obstacle detection algorithm: (a) with association limit and (b) without association limit.

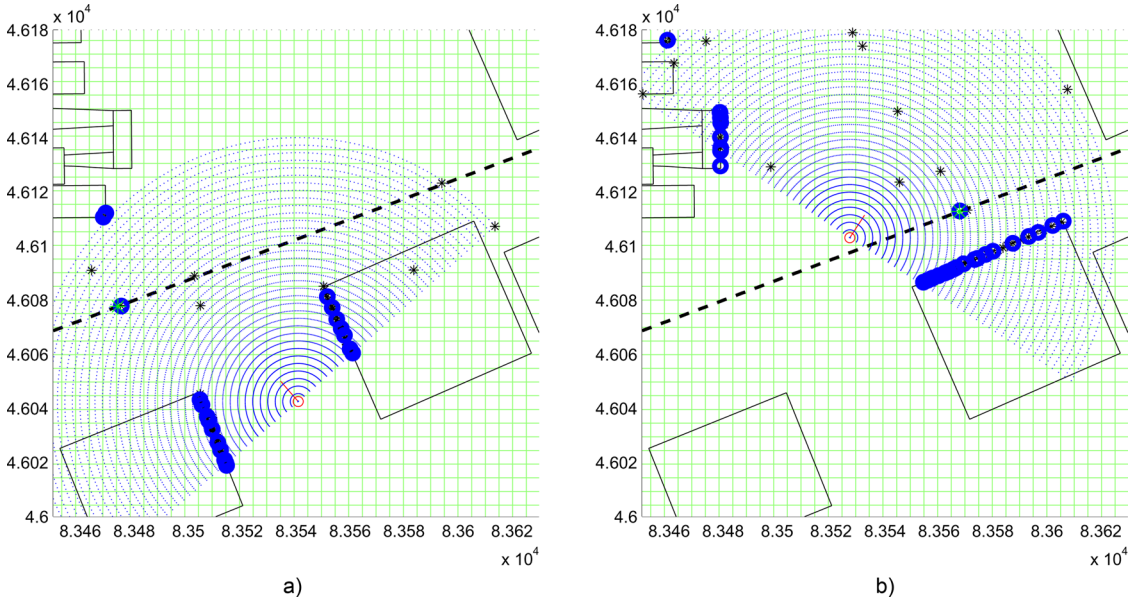

Fig. 7 Tracking of moving target and stationary obstacles: (a) at 24 s and (b) at 34 s.

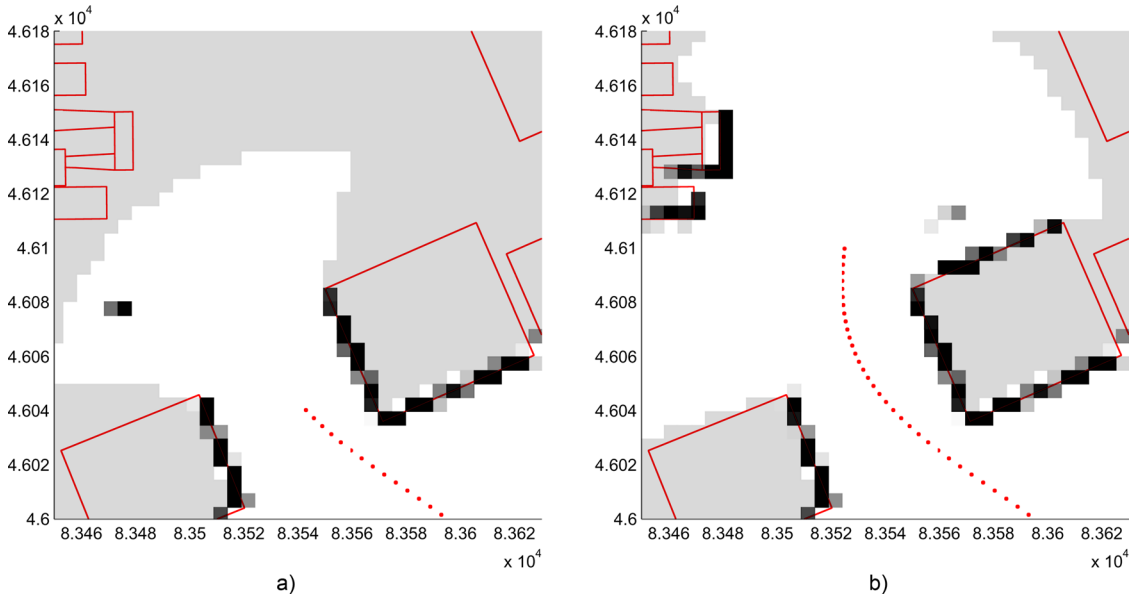

Fig. 8 The probabilistic occupancy grid map: (a) after 24 s and (b) after 34 s.

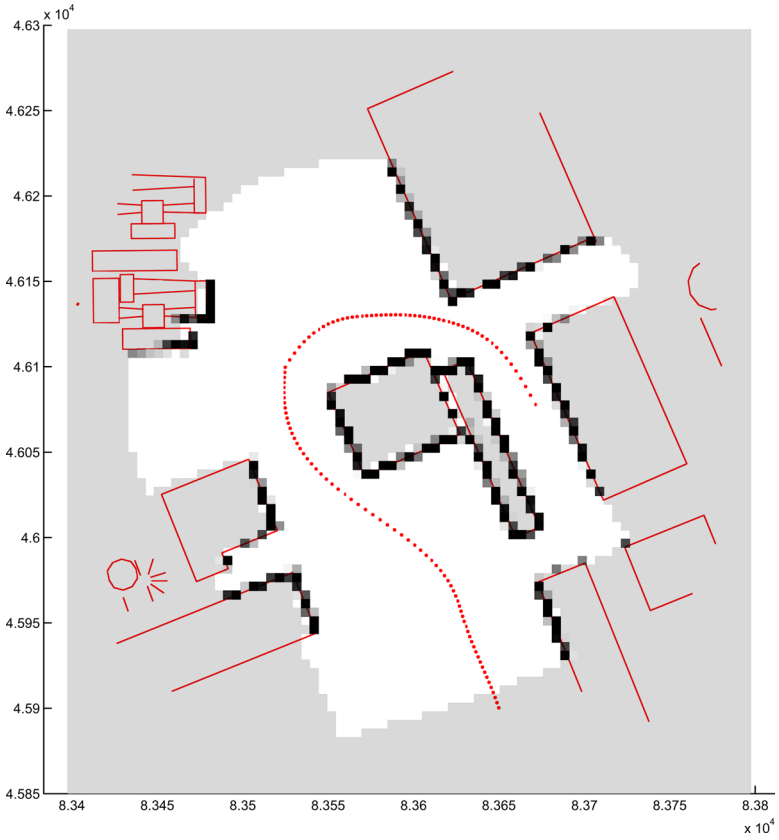

Fig. 9 Final probabilistic occupancy grid map result after 60 s run.

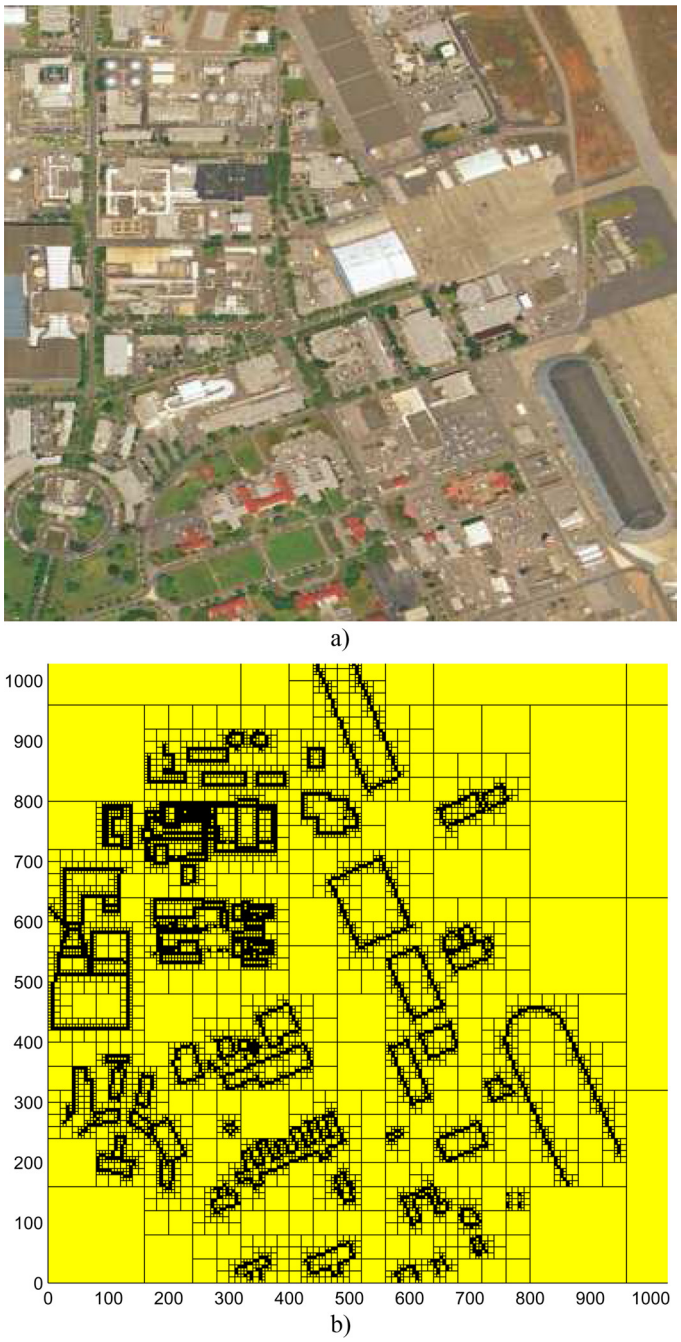

Fig. 10 (a) Image of the Moffett Field and (b) Pre-populated Quadtree.

In Figs. 7a and b, the tracking results at 24 and 34 s are plotted when the moving target is detected by the lidar. The trajectory of the moving target is plotted by a dashed line and raw sensor measurements with clutter noise are plotted by stars. In Fig. 7a the moving target is located on the left side of the UAV and in Fig. 7(b) the moving target is located on the right side of the UAV. In this result, it is shown that both the moving target and the stationary obstacles are detected and tracked by the IMM-based target-tracking algorithm.

B. Probabilistic Occupancy Grid Cell Updating

The occupancy grid maps are updated using the outputs from the IMM-based obstacle detection algorithm. The size of each grid cell in the map is $5 \times 5\,\mathrm{m}^2$ in simulation. In Figs. 8a and b, the probabilistic occupancy grid map results at the same moment with Fig. 7 are plotted. The dark color on the grid cell represents the high probability of occupancy, and the dots represent the UAV trajectory. The cells occupied by both the moving target and the stationary obstacles have high probability of occupancy. Since moving target occupies a cell for a while and moves to the neighboring cells, the probability of the cell currently occupying the moving target diminishes after the target advances to the neighboring cell. The final result after the vehicle finishes the whole flight path is shown in Fig. 9. The resulting probabilistic occupancy grid map shows close agreement with the physical layout of the obstacles in the field. The association of the raw measurements filtered out the clutter and measurement noise successfully.

C. Pre-populated Binary Occupancy Grid Representation

In this subsection, a probabilistic threshold of 0.5 is set and the continuous probability is quantized into binary occupancy of each grid cell. The binary occupancy value is more convenient to be managed in the onboard computer using algorithms such as quadtree [9,10], which is a tree-type data structure in which the cells with the same occupancy are merged into one cell. The simulation results in subsection B assumes that there is no previous knowledge about the field where the UAV will fly through. At the present time, geographical information is available through satellite images or satellite radar measurements. In many cases, these data are precise and hold a lot of geographical information. However, it is also possible that the data are outdated or some of the information is neglected through the data process. In the simulation results to follow, it is assumed that the map is prepopulated using previous information about the area. However, it does not contain the information about the building that was recently built in the area. In Fig. 10a, the picture of the Moffett Field is presented. In Fig. 10b, the database of the buildings in the field is used to prepopulate the map and the black cells denote the occupied grid cells while the yellow cells denote the empty cells. The size of the map is $1280\,\mathrm{m}^2$ and each cell is $5\,\mathrm{m}^2$.

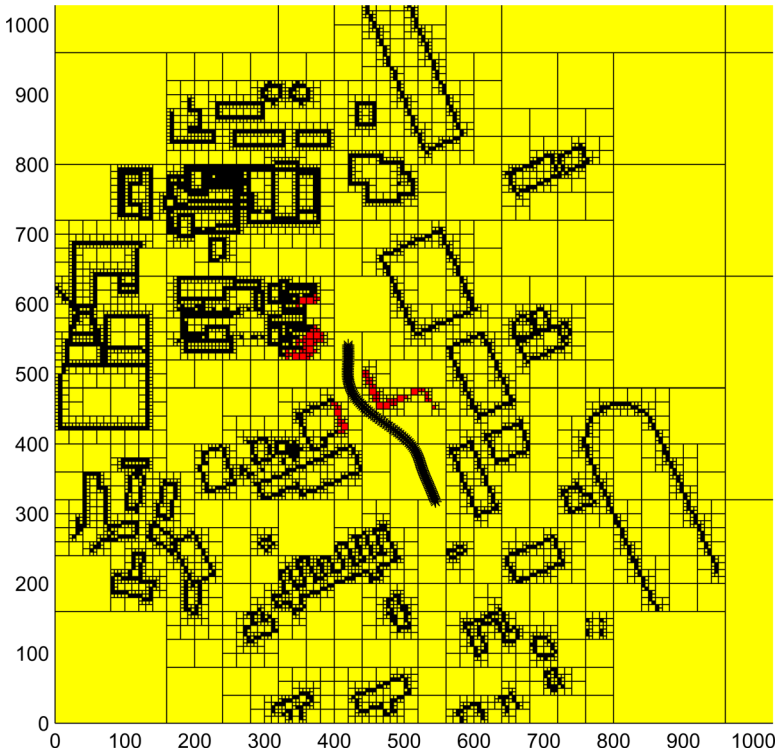

Fig. 11 Map after UAV sensor sweeps.

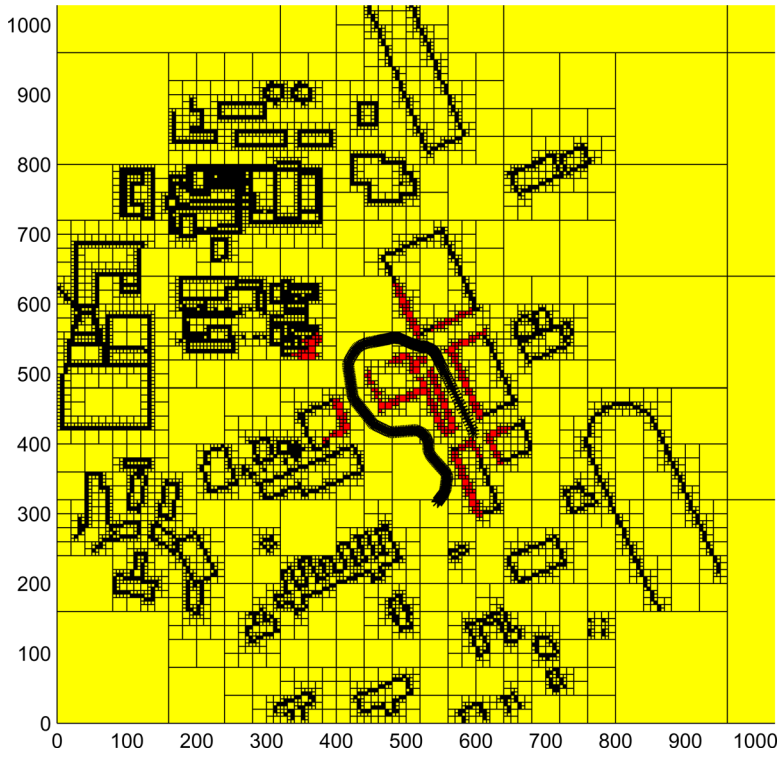

Fig. 12 Map after the flight of UAV.

In Fig. 11, a UAV equipped with a lidar sensor flies through a trajectory that is denoted by the stars. This trajectory is generated by connecting several way points using cubic-spline curves. The cells with probability of occupancy above the pre-determined threshold of 0.5 are denoted by red cells. Owing to sensor noise, sometimes, the adjacent cells are also declared to be occupied.

In Fig. 12, the UAV trajectory almost encircles the two buildings in the center and the final map acquired through this maneuver is plotted. The map shows the shape of the two buildings in the center of the map very well and indicates the potential of the algorithm to be used for real-time path planning of a UAV.

VI. Conclusion

In this study, a new method for building probabilistic occupancy maps is developed and its performance is evaluated in a simulation with realistic sensor models and real obstacle layouts in a 2-D plane. The algorithm developed here assumes that the position of the vehicle is known and focuses on the filtering of clutter and measurement noise using the existing machinery of an IMM algorithm for target tracking and formation. The outputs from the filters are used to produce a high-quality probabilistic occupancy grid map of obstacles. As the IMM algorithm uses Kalman filters that assume point-mass type obstacles, some practical ideas are implemented to handle the case with large-scale obstacles. In the simulation results, it is shown that the shape of the stationary large obstacles can be represented successfully on the resulting probabilistic occupancy map by using an algorithm that assumes point-mass type obstacles. The developed method also filters out sensor noise and outputs a probabilistic map of stationary outdoor obstacles that matches very well with the physical layout of the obstacles in the field. In case there is a moving obstacle, the obstacle detection algorithm based on Kalman filter estimates the position of the moving obstacle precisely. This is one of the advantages of the proposed algorithm, which can accurately estimate the position of the moving target and also map the stationary obstacles on the map. When a lidar is used to detect large obstacles, the targets being tracked by the estimator show drifting motion on the surface of the large obstacle. This is an inherent problem of the lidar since the

consecutive measurements are not guaranteed to originate from the same spot on the obstacle's surface. However, the simulation results showed that the effect of drift on the map is unnoticeable. The outcome from the mapping algorithm will be very useful in generating a safe path for UAVs.

Acknowledgments

The authors thank the researchers in NASA Ames Research Center and acknowledge the financial support of NASA STTR Program, Contract No. NNA04AA63C.

References

[1] Martin, M., and Moravec, H., "Robot Evidence Grids," *Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University*, 1996.

[2] Schultz, A., and Adams, W., "Continuous Localization Using Evidence Grids," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, IEEE, Vol. 4, 1998, pp. 16–20.

[3] Yamaguchi, B., "A Frontier-based Approach for Autonomous Exploration," *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, IEEE, 1997, pp. 146–151.

[4] Thrun, S., "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, Vol. 99, No. 1, 1998, pp. 21–71.
doi: 10.1016/S0004-3702(97)00078-7

[5] Thrun, S., Fox, D., and Burgard, W., "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Machine Learning*, Vol. 31, No. 1–3, 1998, pp. 29–53, also appeared in *Autonomous Robots*, Vol. 5, No. 3–4, 1998, pp. 253–271 (joint issue).
doi: 10.1023/A:1007436523611

[6] Thrun, S., "Learning Occupancy Grid Maps with Forward Sensor Models," *Autonomous Robots*, Vol. 15, No. 111-127, 2003, pp. 111–127.

[7] Hahnel, D., Trievel, R., Burgard, W., and Thrun, S., "Map Building with Mobile Robots in Dynamic Environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, Vol. 2, 2003, pp. 14–19.

[8] Biswas, R., Limketkai, B., Sanner, S., and Thrun, S., "Towards Object Mapping in Dynamic Environments with Mobile Robots," *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, IEEE, Vol. 1, No. 30, 2002, pp. 1014–1019.

[9] Frisken, S. F., and Perry, R. N., "Simple and Efficient Traversal Methods for Quadtrees and Octrees," *Journal of Graphics Tools*, Vol. 7, No. 3, 2002, pp. 1–11.

[10] Kraetzschmar, G. K., Gassull, G. P., and Uhl, K., "Probabilistic Quadtrees for Variable-resolution Mapping of Large Environments," *Proceedings of the 5th IFAC/EURON Synposium on Intelligent Autonomous Vehicles*, edited by M.I. Ribeiro and J. Santos Vicor.

[11] Samet, H., "The Quadtree and Related Hierarchical Data Structures," *ACM Computer Survey*, Vol. 16, No. 2, 1984, pp. 187–260.
doi: 10.1145/356924.356930

[12] Samet, H., "Spatial Data Structures" *Modern Database Systems, The Object Model, Interoperability and Beyond*, edited by W. Kim, ACM Press and Addison-Wesley, New York, 1995, pp. 361–385.

[13] Kim, J.-H., and Sukkarieh, S., "Airborne Simultaneous Localisation and Map Building," *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Vol. 1, No. 14–19, 2003, pp. 406– 411.

[14] Williams, S. B., Dissanayake, G., and Durrant-Whyte, H., "An Efficient Approach to the Simultaneous Localisation and Mapping Problem," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, No. 11–15, 2002, pp. 406–411.

[15] Campbell, M., and Wheeler, M., "A Vision Based Geolocation Tracking System for UAV's," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[16] Redding, J., Mclain, T. W., Beard, R. W., and Taylor, C., "A Vision Based Target Localization from a Fixed-wing Miniature Air Vehicle," *Proceedings of American Control Conference*, 2006, pp. 2862–2867.

[17] Houles, A., and Bar-Shalom, Y., "Multisensor Tracking of a Maneuvering Target in Clutter," *IEEE Transactions on Aerospace and Electronic Systems*, IEEE, Vol. AES-25, No. 2, March 1989, pp. 176–189.

[18] Bar-Shalom, Y., ed. *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Inc., Norwood, MA, 1995, ISBN 0-89006-377-X.

[19] Miller, J. R., and Amidi, O., "3-D Site Mapping with the CMU Autonomous Helicopter," *Proceedings of the 5th International Conference on Intelligent Autonomous Systems (IAS-5)*, 1998.

[20] Bar-shalom, Y., Chang, K. C., and Blom, H. A. P., "Automatic Track Formation in Cluter with a Recursive Algorithm," *IEEE Proceedings on Decision and Control*, IEEE, Vol. 2, December 1989, pp. 1402–1408.

[21] Kang, Y., Caveney, D., and Hedrick, J. K., "Performance Analysis of an IMM-based Obstacle Detection Algorithm," *Proceedings of ASME International Mechanical Engineering Conference and RD&D Expo, Anaheim*, American Society of Mechanical Engineering, 2004.

[22] Genz, A., "Numerical Computation of Rectangular Bivariate and Trivariate Normal and t Probabilities," *Statistics and Computing*, Vol. 14, No. 3, 2004, pp. 251–260.
doi: 10.1023/B:STCO.0000035304.20635.31

Roy Sterritt
Associate Editor