CIMCA 2004 Proceedings/ISBN :1740881885 M. Mohammadian (Ed.)
12-14 July 2004, Gold Coast - Australia

Incorporate Motion Tracking into Map Building in Dynamic Indoor Environments

G.Q. Huang, A.B. Rad, and Y.K. Wong
Department of Electrical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong
E-mail: {g.q.huang, eeabrad, eeykwong}@polyu.edu.hk

Abstract

In this paper, we present a new map-building algorithm for dynamic indoor environments. Instead of accurate but expensive laser ranger finder, we have employed ultrasonic sensors and an uncalibrated camera to obtain a map for dynamic indoor and structured environments. We model the environment via a Fuzzy-Tuned Grid-Based Map (FTGBM). To detect and track moving objects in the vicinity of the robot when mapping a dynamic environment, we propose two methods: Ultrasonic Temporal Difference (UTD) and Statistical Background Subtraction (SBS). The former is realized by monitoring a sequence of temporal lattice maps for a certain number of measurement periods to detect moving objects by using ultrasonic sensors. The latter is a statistical background subtraction technique which is realized with an EM learned 3-class mixture of Gaussians of background model based on sufficient updates. After finding the moving objects, we propose a Fuzzy-Tuned Integration (FTI) technique to incorporate the results of above two types of motion tracking to filter out the moving objects from the resulting map. Furthermore, since Bayesian update rule is used in FTGBM, our approach also has the capability to estimate and update the states of the dynamic objects which change their states slowly during map building process. The simulation and experimental studies demonstrate the capabilities and the robustness of our approach.

1 Introduction

Robotic map building is to generate spatial models of physical environments from sensor measurements through navigation in the environment. This procedure is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robot. Over the past two decades, the field has been received considerable attention and matured to a point where detailed maps of complex environments can be built in real-time, specifically indoors [1]. However, published works suggest that most approaches are designed for static environments, which assumes that the mobile robot is the only moving object in the map world. Nevertheless, the real world is usually dynamic, that is, the states of the objects in the environments do often change over time. Therefore, an autonomous mobile robot should be equipped with the capacity to be conscious of the changes around it and to filter out the spurious models of moving objects when building maps and to constantly update its map of the environment, if it is going to perform services in real world.

Recently, there has been work on updating maps in dynamic environments. Burgard *et al.* [2] update a given static map using the most recent sensor information to deal with people in the environment. Montemerlo *et al.* [3] present an approach to simultaneous localization and people tracking. Fox *et al.* [4] propose a probabilistic technique to identify range measurements that do not correspond to the given model of the environment. These approaches, however, only refine maps which are generated before to remove the influence of state changes, like door is open and later is close, or trash bin is relocated, etc. More recently, there also exist several approaches to map building in dynamic environments which contain moving objects in perceptual range of the robots. Wang and Thorpe [5] present a heuristic and feature-based approach to identify dynamic objects in range scans. The corresponding measurements are then filtered out during 2D and 3D scan registration. R. Biswas *et al.* [6] derive an approximate Expectation-Maximization (EM) algorithm for learning object shape parameters at both levels of the hierarchy, using local occupancy grid maps for representing shape. J. Andrade-cetto *et al.* [7] combine landmark strength validation and Kalman filtering for map updating and robot position estimation to learn moderately dynamic indoor environments. M. Di Marco *et al.* [8] adopt a set theoretic approach to address dynamic SLAM problem. D. Hähnel *et al.* [9] present a probabilistic approach to map building in populated environments by using Sample-based Joint Probability Data Association Filters (SJPDAFs) to track people in the data obtained with the laser range scanners of the robot. The results of the people tracking are integrated into the scan alignment process and into the map generation process, thus filtering out the spurious objects in the resulting maps. However, virtually all state-of-the-art approaches use SICK scanning laser range-finders. While the SICK is ideal for this because of the accurate and detailed range information provided, there are drawbacks. In particular, the SICK is expensive and quite heavy and bulky.

In our work, we respectively use sonar and camera to detect and track motions, and then fuzzy-tuned integrate the results of above two actions to filter out the moving objects from the resulting map. Furthermore, we use Bayesian update rule in fuzzy-tuned grid-based map to estimate and refine the states of some dynamic objects which change slowly. It should be noted that the similar ideas about fuzzy system and Fuzzy-Tuned Grid-Based Map (FTGBM) come from our previous work [10]. So, in this paper we omit this work and you can refer to [10] for details. The rest of this paper is organized as follows. We will describe the ultrasonic temporal difference by ultrasonic sensors in Section 2 and statistical background subtraction in Section 3, and then report the fuzzy-tuned integration to incorporate the both motion detection results into the resulting map update in Section 4. And Section 5 will contain our experiments carried on a real mobile robot and in simulation to illustrate the capabilities and the robustness of our approach. Finally, conclusion and future work will be presented in Section 6.

2 Ultrasonic Temporal Difference (UTD)

The fundamental idea to identify temporal changes in the surrounding environment of a robot is to monitor a temporal sequence of spatial observations and then to determine how these observations differ from each other. An inconsistency between two temporally subsequent observations is a strong indication of a potential motion in the environment. Such inconsistency is mainly caused by dynamic objects in a dynamic indoor environment. In computer vision literature, temporal difference is simple and popular method for detecting moving objects with a static observer. However, for a moving mobile robot, it in itself is not sufficient to unequivocally identify moving objects. Here, we propose a new scheme to detect moving objects using ultrasonic sensors called Ultrasonic Temporal Difference (UTD) borrowing from the etymology in the computer vision literature, which is realized by monitoring sensor-based information called Time-Variant Map (TVM) along the time axis with a certain time duration of τ ($\tau = nt$, t is sampling time) and simultaneously filtering out the same information, i.e. stationary objects, thus obtaining the trajectories and outliers of moving objects during time span τ. Note that all the sensor information has been transformed into the same global coordinate frame.

2.1 Time-Variant Map (TVM)

Ultrasonic Temporal Difference is realized by monitoring a temporal sequence of Time-Variant Maps (TVMs). This procedure is not new and we have borrowed it from [11-12]. We adopt occupancy grid model to represent time-variant map because it is easy to incorporate the result of ultrasonic temporal difference into the resulting map. In occupancy grid-based mapping procedure, each time that new measurements are available, a significant amount of time is spent in updating the posterior including free space or stationary object state from Bayesian update rule. However, it is less important in the context of short-term motion detection since all the sensor data information is synchronically registered in fuzzy-tuned grid-based map. Hence, we only update the occupancy probabilities of those cells in local sensor cone at time *t* while all other cells remain untouched. Fig.2 clearly shows the relevant transformation. It should be noted that our experimental platform is Pioneer 1 mobile robot [13] that only has seven sonar sensors, five in front and two at each side, separated by 15-degrees. Here, we are only concerned with the front 75-degree region. Therefore, there is only a 75-degree cone shown in Fig.1b. We call this representation a *time-variant map*. Building such maps is rather simple: in each sensor measurement at time *t*, the cell that corresponds to the object detection is labeled with this time tag *t*. The tag means that the cell *occupied at time t*. No other cells are updated during this operation. Therefore, the temporal changing features of the environment are captured by the sequence of time-variant maps: TVM_t, TVM_{t-1}, ..., TVM_{t-n}. An example of such a sequence is shown in Fig.2a-c. Note that the maps are already transformed into the same frame of reference.

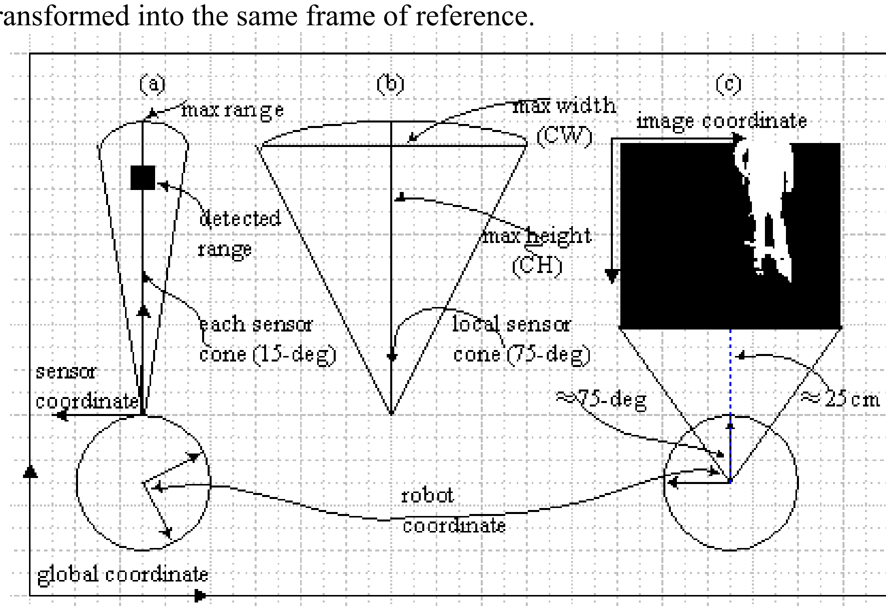
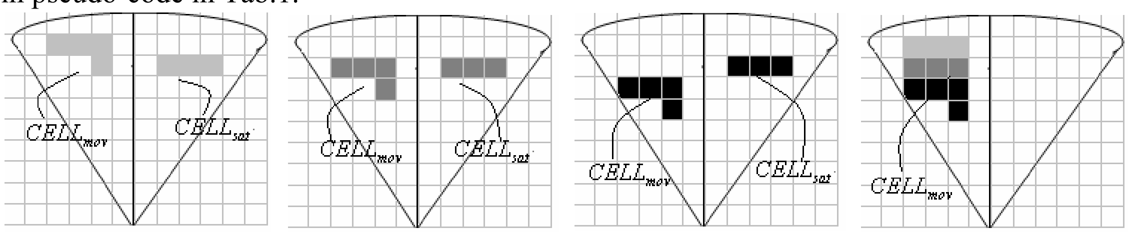


Figure 1: (a) The kinematical transformation; (b) Local sensor cone; (c) Image coordinate transformation

2.2 Detecting Moving Objects

Due to the noise and uncertainty inherently in the ultrasonic sensors, to keep false detection at a low rate, we do not track single cells that are apparently moving, but cluster ensembles of coherently moving cells into distinct objects even though which may be only part of certain object. The cell clustering algorithm is temporarily simple: to check the adjacent cells, if occupied, they are considered as the same class, otherwise considered as different ones. In this work, we use ultrasonic temporal difference with a sequence of time-variant maps to detect moving objects. We consider the set of cells in TVM_t which carry a time tag t (*occupied at time t*) and test whether the corresponding cells in TVM_{t-1} were occupied too, i.e., carry a time tag $t-1$. If corresponding cells in TVM_t, TVM_{t-1} carry time tags t and $t-1$, respectively, then we interpret the spatial region circumscribed by these cells occupied by a stationary object $CELL_{sat}$. If, however, the cells in TVM_{t-1} carry a time tag different from $t-1$ or no time tag at all, then the occupation of the cells in TVM_t must be due to a moving object $CELL_{mov}$. If it is detected as a stationary object, we filter this object out of the time-variant map by simply freeing the corresponding occupied cells, while the moving objects stay left in the time-variant map. Fig.2d shows the result of ultrasonic temporal difference based on the sequence of time-variant maps shown in Fig.2a-c. Note that here we consider only the two most recent maps, TVM_t and TVM_{t-1}, for detecting moving objects. This limits our motion detection resolution, since objects that move very slowly as compared to the sensor sampling rate will not be detected as moving. This problem can be alleviated by selecting an appropriate value of n ($n = \tau / t$) and by using Bayesian update rule which updates the dynamic object states in fuzzy-tuned grid-based map. The outline of ultrasonic temporal difference algorithm is shown in pseudo-code in Tab.1.



(a) (b) (c) (d)

Figure 2: A sequence of time-variant maps describing a simple environment, different gray levels represent the age of observation, darker ones corresponding to the more recent

3 Statistical Background Subtraction (SBS)

A common method to track motion in image sequences is background subtraction between an estimate of the image without moving objects and the current image. Previous researcher [14-16] have shown that the disruption can be somewhat suppressed by using statistical model of background in image-subtraction to find motion. Here, we also adopt a statistical method to model the background: a 3-class mixture of Gaussians, which is learned by using Expectation-Maximization (EM) algorithm [17]. We consider the intensity values of a particular pixel over time as an independent statistical process called "pixel process". In a structured indoor environment, due to the lighting changes, scene changes, and moving objects, the distribution of each pixel is fitted with multiple Gaussians. Since illumination is one of the important components which compose the indoor environments, it is necessary to discriminate the shadows from background and foreground. Therefore, we adopt a 3-class mixture of Gaussians to model the pixel process. Since it is hard to estimate the distribution of foreground along the image sequence, we adopt a widely distribution of Gaussian (here uniform distribution is used).

$$
\begin{aligned}
p(z) &= \omega_F * p_F(z) + \omega_B * p_B(z) + \omega_S * p_S(z) \\
&= \omega_F * \frac{1}{R} + \omega_B * \frac{1}{\sqrt{2\pi}\sigma_{x_B}} \exp\left(-\frac{(z-\mu_{x_B})^2}{2\sigma_{x_B}^2}\right) + \omega_S * \frac{1}{\sqrt{2\pi}\sigma_{x_S}} \exp\left(-\frac{(z-\mu_{x_S})^2}{2\sigma_{x_S}^2}\right)
\end{aligned} \tag{1}
$$

where

- $\omega_F, \omega_B, \omega_S$: Weights of the three distributions in the mixture, respectively.
- μ_{x_B}, μ_{x_S} : Means of the two Gaussians in the mixture, respectively.
- $\sigma_{x_B}, \sigma_{x_S}$: Standard deviations of the two Gaussians in the mixture, respectively.
- R : Parameter of the uniform distribution in the mixture, decided by the valid range of the intensity value. Usually it is 256.

In the learning stage, we use EM algorithm to estimate the model parameters by given a training sequence like [15]. It should be noted that EM algorithm is not guaranteed to find global maximum and very sensitive to the starting point. That is, the algorithm will not converge quickly and fail to fit the distribution properly, if given a poor initial estimate of the distribution. In our work, we empirically determine the initialization similar with [14].

Table1: Motion detection by ultrasonic temporal difference algorithm

Begin Algorithm *MotionDetection_UTD*
FOR each *cell class* $cc_{x,t}$ *representing an object x* in TVM_t
FOR each *cell* $c_{i,t}$ *in* $cc_{x,t}$
FOR each *corresponding cell* $c_{i,t-1}, \ldots, c_{i,t-k}, \ldots, c_{i,t-n}$ *in* $TVM_{t-1}, \ldots, TVM_{t-k}, \ldots, TVM_{t-n}$
IF $c_{i,t-k}$ *carries a time tag t-k* THEN
c_i *is occupied by a stationary object*
ELSE
c_i *is occupied by a moving object*
IF *majority of cells* $c_{i,t}$ in $cc_{x,t}$ is *moving* THEN
cell class $cc_{x,t}$ *is moving, i.e.* $CELL_{mov}$
ELSE
cell class $cc_{x,t}$ *is stationary, i.e.* $CELL_{sat}$
IF $CELL_{sat}$ THEN
free the corresponding occupied cells
ELSE IF $CELL_{mov}$ THEN
do nothing
End Algorithm

3.1 Motion Detection

In our work, each time new frame is available, we have to compensate the sensor motion in order to use background subtraction to detect foreground objects. That is, we map each pixel in current frame x_C into background frame. Note that due to the errors in feature localization, motion estimation etc., this map process is not very accurate. So, at best, we predicate a position $\hat{x}_B$, i.e. $\Gamma x_C = s\hat{x}_B$, where Γ is the transition matrix for background motion compensation and s is an arbitrary nonzero scalar. We use traditional method to determine Γ. That is, we extract corners from current frame and background and select the best n corresponding pairs, $C_C = \{c_{C,i}, i = 1,2,\cdots,n\}$, $C_B = \{c_{B,i}, i = 1,2,\cdots,n\}$. $c_{C,i}$ and $c_{B,i}$ are the corresponding corners in current and background frame respectively. Then we use least-square-estimation (LSE) method to estimate the transition matrix Γ according to the assumed transformation model, which is usually the affine or projective transformation.

Because the motion compensation is not accurate, that is $\hat{x}_B$ will not definitely the corresponding pixel x_C, in order to comprise this approximate alignment, we adopt another

Gaussian model called Alignment Gaussian Model (AGM), which centers at $\hat{x}_B$ with covariance matrix Σ in a validation region $\Re_{\hat{x}_B}$ similar with SGD in [16].

$$p(x_B \mid \hat{x}_B) = \frac{1}{2\pi \mid \Sigma \mid^{1/2}} \exp(-\frac{1}{2}(x_B - \hat{x}_B)^T \Sigma^{-1}(x_B - \hat{x}_B)) \quad (2)$$

$$\Re_{\hat{x}_B} = \{x_B : D_{x_B,\hat{x}_B} \leq \lambda\} \quad (3)$$

$$D_{x_B,\hat{x}_B} = (x_B - \hat{x}_B)^T \Sigma^{-1}(x_B - \hat{x}_B) \quad (4)$$

$$\Sigma = \alpha \hat{E} \quad (5)$$

$$\hat{E} = \frac{1}{n}\sum_{i=1}^{n}(c_{C,i} - c'_{C,i})(c_{C,i} - c'_{C,i})^T \quad (6)$$

$$s\,c'_{C,i} = \Gamma c_{B,i} \quad (7)$$

Σ is important for determining the size of AGM $\Re_{\hat{x}_B}$ and will be different from pixel to pixel. But here for computational simplicity, we assume it is constant and estimated by Eq.(8). With $\hat{E}$ is estimated, as coefficient α increases, the size of AGM increase and different results of the detection are obtained accordingly. In our work, since the environment is structured, we select a proper α empirically. Therefore, for a particular pixel in current frame x_C, there is a corresponding AGM in the background map. If x_C belongs to any of the background Gaussians of its AGM, it is labeled as background. If no corresponding background distribution can be found in its AGM, the pixel x_C is regarded as foreground.

3.2 Background Update

Everything above works well while the background is adequately updated. But this is not easy for a moving background, especially when there is an occlusion and/or uncovered background. Here we use the similiar approach with [16] to update non-stationary background. We briefly report the algorithm as Tab.2. For details, please refer to [16].

Table2: Background Update Algorithm

Begin Background_Update Algorithm
Initialization with the first frame. Number_of_Gausssian ← *1*
Gaussian[1].Mean ← *Pixel_Value of frame 1*
Gaussian[1].Variance ← σ^2
FOR *Frame 2 to N*
Motion compensation and obtaining $\hat{x}_C$
Find (x_C^*, ξ_j^*)
Gaussian_Number ← ξ_j^**; Value* ← x_C^*
IF $D_i > D$ THEN
Number_of_Gaussian++
Gaussian[Number_of_Gaussian].Mean ← $\hat{x}_C$
Gaussian[Number_of_Gaussian].Variance ← σ^2
ELSE
Gaussian[Gaussian_Number].Count++
Update the parameters of Gaussian[Gaussian_Number] with Value
END IF
Find $\max_{1 \leq j \leq Number\_of\_Gaussian} Gaussian[j]$
Update the background

END FOR
End Algorithm

4 Fuzzy-Tuned Integration (FTI)

After detecting moving objects, we need to integrate these two different sources into the result map with filtering out the spurious objects. However, since the camera is uncalibrated, we can not know the precise range information from images, additionally, the sonar is also not very accurate because of the uncertainty in radial and angular. Therefore, we can not directly use traditional multisensor fusion [18] which requires precise sensor-sensor calibration to integrate these two detection results into a common reference frame. In order to achieve a reliable integration, in this paper, we propose *Fuzzy-Tuned Integration (FTI)* algorithm to find out the spurious objects in the fuzzy-tuned grid-based map, which needs two necessary parameters: location and size of the spurious objects in the resulting map, and then filter them out. To design this fuzzy system, we first define the input variables and output variables as follows. The fuzzy rule base and membership design is similar with [10].

Input Variables

- $B\_Centroid(x,y)$: The centroid of $BLOB_{mov}$ in robot frame.
- $C\_Centroid(x,y)$: The centroid of $CELL_{mov}$ in robot frame.
- B_Size: The size of the $BLOB_{mov}$ in vision frame in number of pixels.
- C_Size: The size of the $CELL_{mov}$ in grid-based map in number of grids.

Output Variables

- $O\_Centroid(x,y)$: The centroid of update region in robot frames.
- O_Size: The size of update region in number of grids.

Everything above works fine under the assumption that motion correspondence problem has been well solved, that is the moving object pair respectively detected by ultrasonic sensor and uncalibrated camera already finely associated to each other. However, this problem can seriously damage the resulting map if the motion correspondence is not well done. In our work, we use the *Nearest-Neighbor* algorithm to solve the motion association. In our experiment, where closest is defined using the Euclidean distance of the centroids of the detected moving objects formulated by Eq.8. Note that $B\_Centroid(x,y)$ and $C\_Centroid(x,y)$ have been transformed into the same coordinate frame according to Fig.2.

$$DIST = |B\_Centroid(x,y) - C\_Centroid(x,y)| \quad (8)$$

Since our experiment is performed in the indoor dynamic environment which is structured and also has not many moving objects to detect and track, hence the nearest-neighbor algorithm can satisfy our need. So, our fuzzy-tuned integration works only when we find there exist corresponding moving objects, i.e. only when $DIST > \sigma_{correspond}$, where $\sigma_{correspond}$ is motion correspondence threshold obtained by trial and error. So, when making sure of the locations and sizes of the moving objects via above fuzzy system and nearest neighbor algorithm, we can easily filter out the moving objects from the resulting map just by freeing the corresponding occupied cells. The whole outline of proposed fuzzy-tuned integration algorithm is shown in pseudo-code in Tab.3.

Table 3: Fuzzy-tuned integration algorithm

Begin Algorithm *FuzzyTunedIntegration*

FOR *each detected moving objects by camera in image, i.e.* $B\_Centroid(x, y)$
FOR *each detected moving objects by sonar in grid map, i.e.* $C\_Centroid(x, y)$
Computing $DIST$
IF $DIST > \sigma_{correspond}$ THEN
Incorporating the results of two types of motion detection by using fuzzy-tuned integration and filter these spurious objects out of the grid-based map by freeing the corresponding occupied cells
ELSE
Do nothing
Improving the resulting grid-based map by freeing much isolated occupied grids
Waiting for next map update cycle
End Algorithm

5 Simulation and Experimental Results

5.1 To Update States of Dynamic Objects in Simulation Study

In this simulation study, we try to illustrate that the Bayesian update rule used in our proposed algorithm is capable to update dynamic object states. To explain more easily, we only consider the 1D environment and assume that the sensor measurement is r_i and there is a dynamic obstacle which changes its location over time with the motion profile:

- Location 1: Obstacle is stayed 1m away from the sensor. i.e. { r_i =1m, i=1,2,…,8}
- Location 2: Obstacle is moved to 1.6m away from the sensor. i.e. { r_i =1.6m, i =9,10,…,16}
- Location 3: Obstacle is moved to 0.5m away from the sensor. i.e. { r_i =0.5m, i =17,18,19,…}

The profiles of occupancy probabilities corresponding to 7th, 8th, 15th, 16th, 23rd, 24th readings of our algorithm are shown in Fig.3. From this figure, we can see that our algorithm can provide good estimates of the moving obstacle positions. Therefore, our proposed algorithm is suitable for updating the states of dynamic objects.

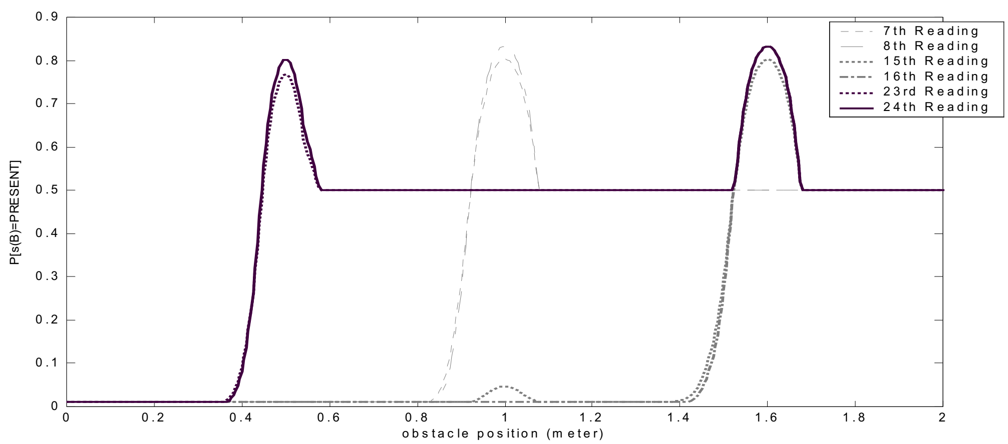


Figure 3: The result of the algorithm to update the state of dynamic object

5.2 Experimental Study

The goal of the experiment is to illustrate that incorporating motion tracking into the map building leads to a better global resulting map since spurious objects were filtered out. The experiment was carried out on the Pioneer 1 mobile robot [13] in a corridor at HKPU. The robot is equipped with one uncalibrated camera with a fixed angle and seven ultrasonic sensors, five locating front, separated by 15 degrees each, and two locating at each side. The software is written in C language and Saphira software [19] with API libraries. The navigation is not autonomous in current step and the robot is manually navigated to predefined locations to avoid the dead reckoning error. The range of sonar is up to 3 meters and the maximum distance of the camera visual zone we concerned in this experiment is also around 3 meters which can highly improve the quality of the fuzzy-tuned integration. Fig.4 shows the hand-measured model of this environment to be mapped.

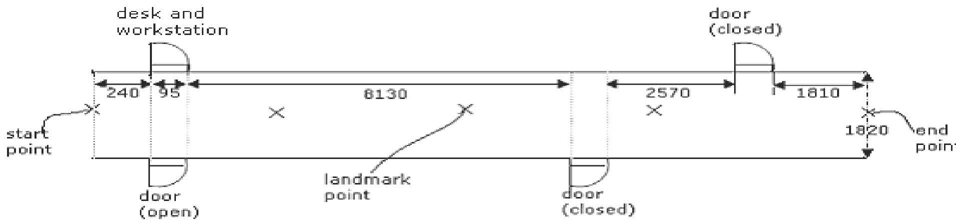


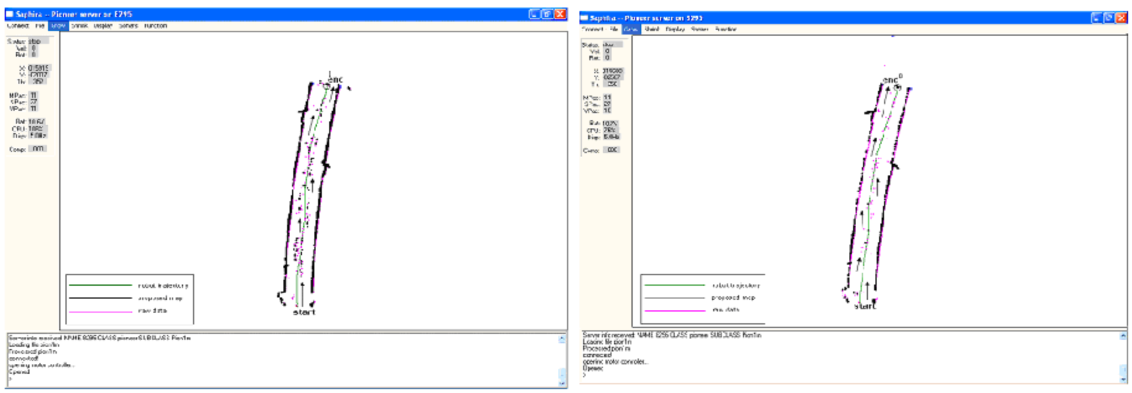
Figure 4: Hand-measured model of the corridor at HKPU

During map building, there were several (up to three) people walking in front of the robot. Fig.5 shows the robot during the map building process. Fig.6a shows the raw range ultrasonic measurements and the map obtained without people filtering. And the resulting map obtained with our proposed algorithm is shown in Fig.6b. Both maps have a resolution of 50mm per cell. As seen from Fig.6a, there are many cells in the resulting grid map, which have a high occupancy probability since people covered the corresponding area while the robot was mapping the environment. If, however, we use the proposed algorithm and filter out the most of moving objects (here is people), the effect of the people is seriously reduced in the resulting map. Therefore, the algorithm is more reliable for map building in dynamic environments than fuzzy-tuned grid-based map building method proposed in our previous work in [10], as well as other traditional map building methods. Note that since we temporarily used odemetric measurements to estimate the robot's position with predefined landmarks, there must be some dead reckoning errors. Thus the resulting map was not rectangular compared with the hand-measured map. The localization problem will be addressed in future work.

6 Conclusions and Future Work

In this paper, we presented a new solution to map indoor dynamic environments by incorporating motion tracking: ultrasonic temporal difference and statistical background subtraction. The former is constructed by monitoring a sequence of temporal local lattice maps. The latter is achieved based on sufficient background update with a 3-class mixture of Gaussians. After detecting the moving objects, due to the inaccuracy of both methods, we used a fuzzy system to integrate the results to filter the spurious objects out of the resulting map. Additionally, the motion correspondence problem is also addressed and solved by the nearest-neighbor algorithm. Preliminary simulation and experiment results demonstrated the capabilities and the robustness of our approach. And the further experimental studies are still ongoing now. In the future work, we try to design much more robust methods to solve the motion correspondence problem and extend our work to address SLAM in dynamic environments.

Figure 5: Pioneer 1 mobile robot map building in a dynamic corridor environment at HKPU



(a) (b)

Figure 6: (a) Fuzzy-tuned grid-based map without filtering out moving objects; (b) Resulting map of our proposed algorithm

References

[1] S. Thrun, Robotic mapping: A survey, *Technical Report CMU-CS-02-111*, CMU, 2002.

[2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, D. Schulz, W. Steiner, and S. Thrun, Experiences with an interactive museum tour-guide robot, *AI* 114(1-2):3-55, 1999.

[3] M. Montemerlo, W. Whittaker, and S. Thrun, Conditional particle filters for simultaneous mobile robot localization and people-tracking, *ICRA*, Washington, DC, 2002.

[4] D. Fox, W. Burgard, and S. Thrun, Markov localization for mobile robots in dynamic environments, *Journal of Artificial Intelligence Research*, 11:391-427, 1999.

[5] C.-C. Wang, C. Thorpe and S. Thrun, Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas, *ICRA*, Sep., 2003.

[6] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, Towards object mapping in non-stationary environments with mobile robots, *IROS,* Lausanne, Switzerland, 2002.

[7] Juan Andrade-Cetto, and Alberto Sanfeliu, Concurrent map building and localization on indoor dynamic environments, *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3): 361-374, 2002.

[8] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, A set theoretic approach to dynamic robot localization and mapping, *Autonomous Robots,* 16: 23-47, 2004.

[9] D. Hähnel, D. Schulz, and W. Burgard, Mobile robot mapping in populated environments, *Advanced Robotics*, 17 (7): 579-598, 2003.

[10] K.M. Chow, A.B. Rad, and Y.L. Ip, Enhancement of probabilistic grid-based map for mobile robot applications, *Journal of Intelligent and Robotic Systems*, 34(2):155-174, 2002.

[11] Y. Han, M. Han, H. Cha, M. Hong, and H. Hahn, Tracking of a moving object using ultrasonic sensors based on a virtual ultrasonic image, *Robotics and Autonomous Systems*, 36: 11-19, 2001.

[12] Erwin Prassler, and Jens Scholz, Tracking multiple moving objects for real-time robot navigation, *Autonomous Robots*, 8: 105-116, 2000.

[13] Pioneer Mobile Robot Operation Manual, Edition 2, Active media, Inc., January 1998.

[14] J. Rittscher, J. Kato, S. Joga, and A. Blake, A probabilistic background model for tracking, *Proceedings of the European Conference on Computer Vision*, 336–351,2000

[15] Rowe S, Blake A, Statistical mosaic for tracking. *Image Vision Comput* 14:549–564, 1996

[16] Ren Y, Chua CS, Ho YK, Motion detection with nonstationary background. *Machine Vision and Application* (2003)13: 332-343

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

[18] J.A. Castellanos, J. Neira, and J.D. Tarós, Multisensor fusion for simultaneous localization and map building, *IEEE Robotics and Automation*, 17(6), Dec. 2001.

[19] Saphira Manual, Version 6.1e, Active media, Inc., April 1998.