Online SLAM in Dynamic Environments

G.Q. HUANG, A.B. RAD, and Y.K. WONG
Department of Electrical Engineering
The Hong Kong Polytechnic University, Hong Kong
{g.q.huang, eeabrad, eeykwong}@polyu.edu.hk

Abstract - **In this paper, we propose a novel online algorithm for Simultaneous Localization and Mapping (SLAM) in dynamic environments. We first formulate the problem with two interdependent parts: SLAM and Multiple Target Tracking (MTT). To pursue online performance, we propose a hierarchical hybrid method to solve SLAM: locally by Maximum Likelihood (ML) with occupancy grid map, and globally by Extended Kalman Filter (EKF) with feature-based map. Meanwhile we apply a straightforward Nearest Neighborhood (NN) algorithm based on Euclidean metric to address MTT. In order to track multiple moving objects reliably, we propose an Enhanced Fuzzy Clustering (EFC) method to segment 2D range images and reliably group objects. Experiments validated on Pioneer 2DX mobile robot with SICK LMS200 demonstrate the capability and robustness of the proposed algorithm.**

Index Terms - **Simultaneous Localization and Mapping (SLAM), Multiple Target Tracking (MTT), Maximum Likelihood (ML), Extended Kalman Filter (EKF), Nearest Neighborhood (NN), Fuzzy Clustering (FC).**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is an essential capability for autonomous mobile robots to explore unknown environments, and has been attracted immense attention in the literature in the past decades. Virtually most of the state-in-the-art algorithms of SLAM are based on static environments. The web site of the SLAM summer school 2002 [1] provides a comprehensive coverage of the key topics and state of the art in SLAM. But the real worlds where the robots will be deployed are usually dynamic. In order to design a robot not only to work *for* human, but hopefully to work *with* human, it should have the ability to be conscious with the dynamic surrounding. Recently, there exist several approaches to map building and updating in dynamic environments which contain moving objects in perceptual range of the robots. 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. Andrade-cetto *et al.* [4] combine feature strength validation and Kalman filtering for map updating and robot position estimation to learn moderately dynamic indoor environments. Hähnel *et al.* [5] present a probabilistic offline approach to map building in populated environments by using Sample-based Joint Probability Data Association Filters (SJPDAFs) and the moving people are filtered out from the resulting maps. All previously mentioned approaches to address SLAM in dynamic environments do simply filter out the information of moving objects from resulting maps. More recently, as the best we know, Wang [6] is the first to address the problem of simultaneous localization, mapping and moving object tracking. His work mainly focused on representation of the world and data association. Wolf, *et al* [7] proposed to solve SLAM in dynamic environments by maintaining one landmark map for localization purpose and two occupancy grid maps: one for static objects and another for moving objects. Obviously this method is limited to moderately dynamic indoor environments.

In our work, we simultaneously incorporate Multiple Target Tracking (MTT) with SLAM into a coherent strategy, and not just simply filter out the information of moving objects after detecting them, because we find that both SLAM and MTT are mutually beneficial from each other (shown in Figure 1): SLAM provides a more accurate pose estimate and a world map, which is used by MTT to detect moving objects more reliably; MTT can detect and predict the locations of the moving objects, so SLAM can filter out moving objects and get more accurate localization and surrounding map.

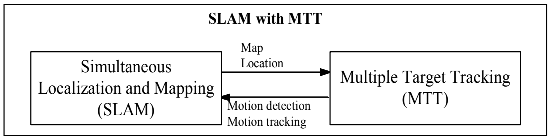

Figure 1. SLAM with MTT is mutually beneficial from each other.

In our work, we explore the theoretical framework of SLAM in dynamic environments from a Bayesian point of view, and mathematically decompose the problem into two

0-7803-9177-2/05/$20.00/©2005 IEEE

interdependent parts: SLAM and MTT. Thereby, we can address SLAM and MTT, respectively. To pursue online performance, we propose a novel hierarchical hybrid method for SLAM. We employ Maximum Likelihood (ML) to solve SLAM basically, thanks to its simplicity and fast computation, and adopt occupancy grid map to represent the environment. However, it is well known that grid-based approach does not provide a mechanism for loop closing and also is suffered from too much storage and computation load for large scale environments. So, to overcome these limitations, we *locally* solve SLAM by ML with occupancy grid map, and *globally* solve it by Extended Kalman Filter (EKF) with feature-based map where feature is local grid map with 3-Degree of Freedom (3-DOF) state. EKF feature-based algorithm can smoothly solve the loop closing problem, which is a well-known point in the SLAM literature. Also, to maintain local maps is more efficient than to update a whole global grid map. Again for online performance reasons, we apply a straightforward Nearest Neighborhood (NN) algorithm based on Euclidean metric to solve MTT problems. However, in order to track multiple moving objects reliably by NN, it must be able to detect moving objects reliably, which also compensates some potential limitations of NN criterion. So, we propose an Enhanced Fuzzy Clustering (EFC) method to segment 2D range images and reliably group objects, and then detect moving objects with eigenvalue-based method and consistency-based method. When information of moving objects available, we will track these moving objects right away and update SLAM map by filtering out this information from the resulting SLAM map, thus make the resulting map more reliable and accurate.

This paper is organized as follows. In the following section, we will present the problem statement of SLAM in dynamic environments from Bayesian perspective, and a theoretical Bayesian formulation with two parts: SLAM and MTT. In Section III, we will describe our hierarchical hybrid approach to SLAM. And the enhanced fuzzy clustering based nearest neighbourhood method for MTT will presented in Section IV. Section V presents some preliminary simulation and experiment results to demonstrate the feasibility and robustness of our algorithm. Conclusion will come into Section VI.

II. PROBLEM STATEMENT

A. Dynamic Bayesian Network (DBN)

We approach the problem from a Bayesian point of view. Figure 2 illustrates a generative Dynamic Bayesian Network (DBN) [8-9] of the problem of SLAM in dynamic environments, which is derived by SLAM and MTT here. In particular, we denote the discrete time index by the variable *t*, odometry measurement from *t-1* to *t* by u_t, sensor measurement at *t* by z_t, true location of the robot by x_t, the map containing *l* features by *m,* locations of *n* moving objects at time *t* by y_t, and motion model of moving object by w_t. And the following sets refer to data leading up to time *t*.

$$\boxed{\begin{array}{l} u_{0:t} \equiv \{u_0, u_1, ..., u_t\} = \{u_{0:t-1}, u_t\} \\ z_{0:t} \equiv \{z_0, z_1, ..., z_t\} = \{z_{0:t-1}, z_t\} \\ x_{0:t} \equiv \{x_0, x_1, ..., x_t\} = \{x_{0:t-1}, x_t\} \\ m \equiv \{m^1, m^2, ..., m^l\} \\ y_t \equiv \{y_t^1, y_t^2, ..., y_t^n\} \\ w_t \equiv \{w_t^1, w_t^2, ..., w_t^n\} \end{array}} \tag{1}$$

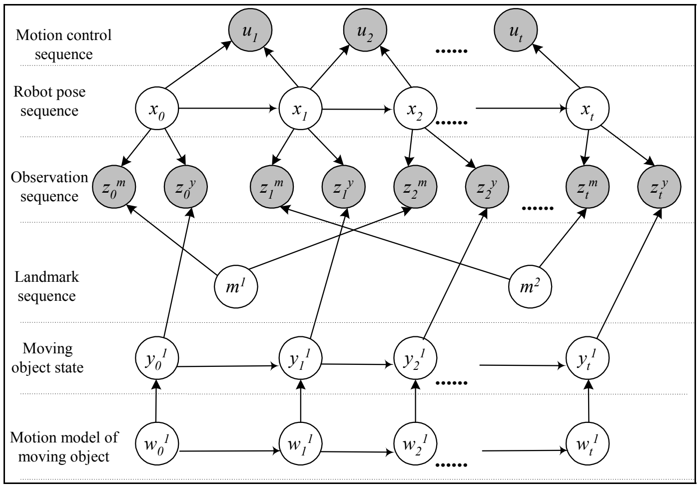

Figure 2. A DBN for SLAM with MTT, with one moving object and two stationary objects (features) as an example. Shaded circles denote explicit states and clear circles denote hidden or implicit states which should be inferenced from explicit ones.

We also assume the dynamic system as follows:

$$x_t = f_t(x_{t-1}, u_t, t) + \upsilon_t \tag{2}$$

$$z_t = h_t(x_t) + \omega_t \tag{3}$$

where: f_t and h_t are non-linear state transition model and observation model, respectively. υ_t and ω_t are noise vectors which are Gaussian, temporally uncorrelated and zero-mean.

B. Bayesian Formulation

Please NOTE that the formulation derivation here is similar with Wang [6].

1) Assumptions

Besides common assumption in SLAM literature, that is, SLAM problem is a Markov process, which states that given knowledge of the current state, the future is independent of the past, there are two special assumptions in this problem. The first assumption is that measurement can be decomposed into measurements of static and moving objects. That is, we can reliably detect moving objects from stationary objects, which is the foundation for all of the following formulations. And we put it into MTT part, so the MTT in our work is more general than normal MTT.

$$\begin{aligned} & z_t = z_t^m + z_t^y \\ & \Rightarrow z_{0:t} = z_{0:t}^m + z_{0:t}^y \\ & \Rightarrow p(z_t \mid x_t, m, y_t) = p(z_t^m \mid x_t, m, y_t)\, p(z_t^y \mid x_t, m, y_t) \\ & \qquad\qquad\qquad\quad = p(z_t^m \mid x_t, m)\, p(z_t^y \mid x_t, y_t) \end{aligned} \tag{4}$$

The second assumption is that the measurements of moving objects carry no information for SLAM and their positions:

$$p(x_t, m \mid z_{0:t}, u_{0:t}, y_t) = p(x_t, m \mid z_{0:t}^m, u_{0:t}) \tag{5}$$

2) *Derivation*

Based on above assumptions, we derive the problem as follows:

$$\begin{aligned} & p(x_t, m, y_t \mid z_{0:t}, u_{0:t}) \\ & \overset{Bayes}{\propto} p(z_t \mid x_t, m, y_t, z_{0:t-1}, u_{0:t})\, p(x_t, m, y_t \mid z_{0:t-1}, u_{0:t}) \\ & \overset{Markov}{\propto} p(z_t \mid x_t, m, y_t)\, p(x_t, m, y_t \mid z_{0:t-1}, u_{0:t}) \\ & \overset{Assume1}{=} p(z_t^m \mid x_t, m)\, p(z_t^y \mid x_t, y_t)\, p(x_t, m, y_t \mid z_{0:t-1}, u_{0:t}) \\ & \overset{Bayes}{=} p(z_t^m \mid x_t, m)\, p(z_t^y \mid x_t, y_t)\, p(x_t, m \mid z_{0:t-1}, u_{0:t}, y_t)\, p(y_t \mid z_{0:t-1}, u_{0:t}) \\ & \overset{Assume2}{=} p(z_t^m \mid x_t, m)\, p(z_t^y \mid x_t, y_t)\, p(x_t, m \mid z_{0:t}^m, u_{0:t})\, p(y_t \mid z_{0:t-1}, u_{0:t}) \\ & = p(z_t^m \mid x_t, m)\, p(x_t, m \mid z_{0:t}^m, u_{0:t})\, p(z_t^y \mid x_t, y_t)\, p(y_t \mid z_{0:t-1}, u_{0:t}) \\ & = p(z_t^m \mid x_t, m)\, p(x_t \mid z_{0:t-1}^m, u_{0:t}, m)\, p(m \mid z_{0:t-1}^m, u_{0:t}) \\ & \quad * p(z_t^y \mid x_t, y_t) \int p(y_t \mid z_{0:t-1}, u_{0:t}, y_{t-1})\, p(y_{t-1} \mid z_{0:t-1}, u_{0:t})\, dy_{t-1} \\ & = p(z_t^m \mid x_t, m) \int p(x_t \mid z_{0:t-1}^m, u_{0:t}, m, x_{t-1})\, p(x_{t-1} \mid z_{0:t-1}^m, u_{0:t}, m)\, p(m \mid z_{0:t-1}^m, u_{0:t})\, dx_{t-1} \\ & \quad * p(z_t^y \mid x_t, y_t) \int p(y_t \mid y_{t-1})\, p(y_{t-1} \mid z_{0:t-1}, u_{0:t-1})\, dy_{t-1} \\ & = p(z_t^m \mid x_t, m) \int p(x_t \mid u_{0:t}, x_{t-1})\, p(x_{t-1}, m \mid z_{0:t-1}^m, u_{0:t-1})\, dx_{t-1} \quad \underline{\underline{SLAM: p(x_t, m \mid z_{0:t}^m, u_{0:t})}} \\ & \quad * p(z_t^y \mid x_t, y_t) \int p(y_t \mid y_{t-1})\, p(y_{t-1} \mid z_{0:t-1}, u_{0:t-1})\, dy_{t-1} \quad \underline{\underline{MTT: p(y_t \mid z_{0:t}, u_{0:t})}} \end{aligned} \tag{6}$$

III. HIERARCHICAL HYBRID METHOD FOR SLAM

A. *Local ML with Occupancy Grid Map*

1) *ML Estimation*

The idea and implementation of ML is simple, thus meeting the online computing requirements: Given a sensor measurement and odometry reading, determine the most likely pose. Then append the pose and build the map. Particularly for SLAM in (6), that means to maximize the marginal likelihoods of pose and map given previous pose and map. We just use following function (7) to update the map, and in practice, we employ occupancy grid map to implement it, which will be discussed later.

$$\hat{m}(x_{0:t}, z_{0:t}) = \arg\max p(m \mid x_{0:t}, z_{0:t}) \tag{7}$$

So, when the map available, SLAM problem in (6) can be reduced further at each time step, with additional assumption that previous step pose $\hat{x}_{t-1}$ is known too.

$$\begin{aligned} & p(x_t, \hat{m}(x_{0:t}, z_{0:t}) \mid z_{0:t}, u_{0:t}) \\ & = p(z_t \mid x_t, \hat{m}(x_{0:t}, z_{0:t})) \\ & \quad \bullet \int p(x_t \mid u_t, x_{t-1})\, p(x_{t-1}, \hat{m}(x_{0:t-1}, z_{0:t-1}) \mid z_{0:t-1}, u_{0:t-1})\, dx_{t-1} \\ & = p(z_t \mid x_t, \hat{m}(x_{0:t}, z_{0:t}))\, p(x_t \mid u_t, \hat{x}_{t-1}) \end{aligned} \tag{8}$$

Now, what is left to do is to calculate the *t*-th pose by ML estimate.

$$\hat{x}_t = \arg\max_{x_t} p(z_t \mid x_t, \hat{m}(x_{0:t}, z_{0:t}))\, p(x_t \mid u_t, \hat{x}_{t-1}) \tag{9}$$

To calculate (9) is only a complex mathematics exercise. We omit the detailed information here.

2) *Occupancy Grid Map*

As discussed previously, we apply occupancy grid map method to calculate (7). So here we establish the standard occupancy grid map approach [10]. As the name suggests, occupancy grid maps usually are represented by two-dimensional grids and generate probabilistic maps. Let $m_{x\,y}$ denote the occupancy of the grid cell at <*x, y*> in the map *m*. Occupancy is a binary variable: Either the cell is occupied or it is free. The problem, thus, is to calculate a posterior over a set of binary variables, each of which is a single numerical probability $p(m_{x\,y} \mid x_{0:t}, y_{0:t})$. Then, we apply Bayes filters to calculate these posteriors. For computational reasons, it is common practice to calculate the so-called *log-odds*:

$$\begin{aligned} l_{x,y}^T &= \log \frac{p(m_{x,y} \mid x_{0:t}, z_{0:t})}{1 - p(m_{x,y} \mid x_{0:t}, z_{0:t})} \\ &= \log \frac{p(m_{x,y} \mid x_t, z_t)}{1 - p(m_{x,y} \mid x_t, z_t)} + \log \frac{p(m_{x,y})}{1 - p(m_{x,y})} + \log \frac{p(m_{x,y} \mid x_{0:t-1}, z_{0:t-1})}{1 - p(m_{x,y} \mid x_{0:t-1}, z_{0:t-1})} \\ &= \log \frac{p(m_{x,y} \mid x_t, z_t)}{1 - p(m_{x,y} \mid x_t, z_t)} + \log \frac{p(m_{x,y})}{1 - p(m_{x,y})} + l_{x,y}^{T-1} \end{aligned} \tag{10}$$

$$l_{x,y}^0 = \log \frac{p(m_{x,y})}{1 - p(m_{x,y})} \tag{11}$$

Obviously, the occupancy grid mapping algorithm is recursive with the initialization (11). So, to compute the desired probability only requires two entities: *occupancy prior* $p(m_{x\,y})$ and *inverse sensor model* $p(m_{x\,y} \mid x_t, y_t)$ which specifies the probability that a grid cell $m_{x\,y}$ is occupied based on a single sensor measurement z_t taken at location x_t. When log-odd is available, we also need to recover the desired posterior from the calculated log-odd:

$$p(m_{x,y} \mid x_{0:t}, z_{0:t}) = 1 - [1 + e^{l_{x,y}^T}]^{-1} \tag{12}$$

B. *Global EKF with Feature-based Map*

As we mentioned repeatedly, grid-based approach does not provide a mechanism for loop closing and also is suffered from too much storage and computation load for large scale environments. So, we build occupancy grid map locally, and treat each local grid map as a 3 Degree of Freedom (3-DOF) feature state represented by the gravity and orientation of the local map, then employ EKF to update these features globally. The advantage of this hierarchical scheme is to overcome the storage and inconsistency problems. To consistently close a large cyclic map, we must recognize the already visited place. Specifically, we must know whether or not current local grid map is in a pre-visited place. To do so, we adopt covariance increasing [11] method when there is an inconsistency in the global map (Figure 3).

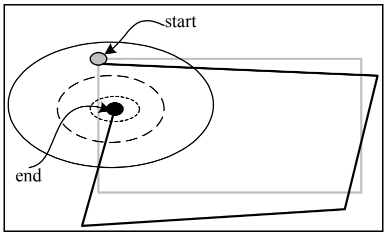

Figure 3. Covariance increasing activated when inconsistency is detected.

EKF feature-based map is a dominant method in SLAM literature in the past decades. The overall algorithm is summarized as follows.

Algorithm EKF

1. Initialization step

- Initialize the mean square error covariance $P_{0|0}$, predict the position $x_{0|0}$, state noise covariance model Q_0 and measurement noise covariance model R_0.

2. Prediction step

- $$\begin{aligned}\hat{x}_{t|t-1} &= E[x_t \mid z_{1:t-1}] \\ &\approx E[f_t(\hat{x}_{t-1|t-1}, u_t, t) + (x_{t-1} - \hat{x}_{t-1|t-1})\nabla F_t + O[(x_{t-1} - \hat{x}_{t-1|t-1})^2] + \upsilon_t \mid z_{1:t-1}] \\ &= f_t(\hat{x}_{t-1|t-1}, u_t, t)\end{aligned}$$

- $$\begin{aligned}P_{t|t-1} &= E[(x_t - \hat{x}_{t|t-1})(x_t - \hat{x}_{t|t-1})^T \mid z_{1:t-1}] \\ &= \nabla F_t P_{t-1|t-1} \nabla F_t^T + Q_t\end{aligned}$$

where:

$\nabla F_t = \nabla f_t(x_{t-1})|_{\hat{x}_{t-1|t-1}}$: Jacobian matrix of state transition model with respect to robot state

Q_t: state noise covariance at time t

3. Update step

- Innovation: $\upsilon_t \equiv z_t - \hat{z}_{t|t-1} \equiv z_t - E[z_t \mid z_{1:t-1}] = z_t - h_t(\hat{x}_{t|t-1})$
- Innovation covariance:

$S_t \equiv E[\upsilon_t \upsilon_t^T] = \nabla H_t P_{t|t-1} \nabla H_t^T + R_t$

- Kalman gain:

$K_t = P_{t|t-1} \nabla H_t^T S_t^{-1}$

$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - h_t(\hat{x}_{t|t-1}))$

$P_{t|t} = P_{t|t-1} - K_t S_t K_t^T$

where:

$\nabla H_t = \nabla h_t(x_t)|_{\hat{x}_{t|t-1}}$: Jacobian matrix of measurement model with respect to robot state

R_t: measurement noise covariance at time t

IV. NN WITH FC FOR MTT

A. *EFC for Object Detection*

As we discussed earlier, in order to track multiple moving objects reliably, we must be able to detect objects reliably first. In our work, we propose an online Enhanced Fuzzy Clustering (EFC) algorithm. Fuzzy Clustering (FC) algorithms have been used in many applications involving data segmentation. The basic fuzzy clustering algorithm is the fuzzy c-means (FCM) method developed by Bezdek [12]. FCM is aimed at minimizing the sum of the squared distances between the data points and the c cluster centers V_i. The objective function to be minimized is

$$J_m : M_{fc} \times R^{cp} \rightarrow R^+$$

$$J_m(\{U_{ik}\},\{V_i\};\{X\}) = \sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m (d_{ik})^2 \mid \sum_{i=1}^{c} U_{ik} = 1,\ \mathrm{k} = 1,2,\ldots,\mathrm{n} \tag{13}$$

where, M_{fc} is the fuzzy partition space; $U \in M_{fc}$ is a fuzzy c-partition of X; $V=\{V_1, V_2, \ldots, V_c\}$ are the cluster center vectors, V_i is a cluster center of p features, $V_i \in \Re^p$; X is the matrix of feature vectors, where $X=\{X_1, X_2, \ldots, X_n\}$, $X_i \in \Re^p$; and d_{ik} is given by

$$(d_{ik})^2 = \|x_k - v_i\|^2 \tag{14}$$

Here, n is the total number of feature vectors (data points) and c is the number of cluster centers. The index i denotes each clusters, index k denotes each of n number of data points and p is the dimension of $\Re$ space. The exponent $m \in [1, \infty)$ is used to adjust the weighting effect of the membership value. The cluster centers v_i can be obtained from the following:

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m}\ ,\ \mathrm{i} = 1,2,\ldots,\mathrm{c} \tag{15}$$

The fuzzy membership matrix U can be obtained from the following:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}}\ ,\ \mathrm{i} = 1,2,\ldots,\mathrm{c}\ ;\ \mathrm{k} = 1,2,\ldots,\mathrm{n} \tag{16}$$

if $d_{ik} = 0$ then $u_{ik} = 1$ and $u_{jk} = 0$ for $j \neq i$.

It is well known that due to the iterative nature and often large number of feature vectors in FCM (13), it is computationally intensive, which does not qualify for our real-time requirement. We mitigate this time problem by carefully selecting initial cluster centers and cluster number c to reduce the number of iteration required to convergence. To do so, we employ a simple distance criterion to segment 2D range image into different objects, as shown in Figure 4. Then we pass the gravity of each object as initial cluster center, and number of the objects as initial cluster number c into FCM. Because the simple segmentation can not produce perfect results, which will damage the performance of FCM, we also apply *Compatible Cluster Merging (CCM)* [13] to evaluate and enhance the results of FCM by merging compatible clusters. Hereby, we get the objects more reliably and efficiently than common FC algorithm documented in [12] which randomly initialized cluster center matrix V.

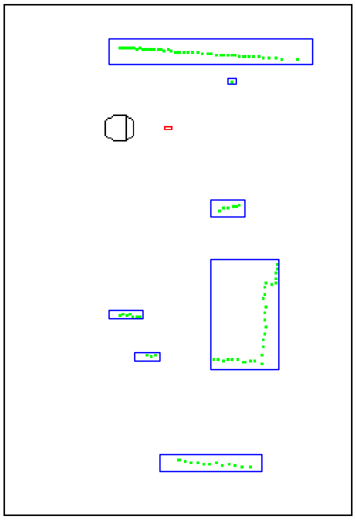

Figure 4. Segment range image: black circle denotes robot, blue rectangles denote objects

To achieve moving object tracking, we must be able to discern moving objects from stationary ones. We propose two methods to detect moving objects: eigenvalue-based approach and consistency-based approach. A cluster representative of moving object has elliptical or rectangular shape, while the static cluster is linear. Since the eigenvalue and eigenvector of each cluster is computed in the above EFC procedure, it is simple to do eigenvalue-base approach.

Eigenvalue-based Approach

```
FOR i=1 to $c$
IF $v_{2i}/v_{1i}>k$, THEN $c_i$ is a moving object
ELSE it is a static object
END
```

---

where, v_{1i} is the eigenvalue of the first principal component and k indicates the shape of the cluster and for linear cluster is close to zero.

The consistency-based approach is relied on the occupancy grid map provided by SLAM. For a static object, when mapped into grid map, the corresponding grid cells would not change their probabilities over time. So, it is easy to detect moving objects by comparing probabilities over consecutive laser scans (SICK LMS200 is used in our work). A cluster is identified as a potential moving object if the greater than 50% corresponding grid cells changed much.

B. *NN for Object Tracking*

Now, we detected both static and moving objects. In tracking scheme, we parameterize each moving object as three feature-points: two endpoints which are two most distant points in the cluster, and one gravity point. So, we can track these feature-points as track moving objects by NN algorithm. NN is the simplest suboptimal data association algorithm, which assumes that each measurement originates from the closest corresponding feature, in our work, where closest is defined using the Euclidean distance of the gravity points. If the distance to the nearest neighbor is smaller than a given threshold, we assume both clusters represent the same object. An important point should be noted that we only track moving objects in local maps, which could save much computation expense. There are still some complicated practical issues should be carefully addressed like occlusion effects, crossover track problems. There problems are under studying now.

V. EXPERIMENTAL STUDY

We validate the proposed algorithm on a Pioneer 2DX mobile robot equipped with SICK LMS200 (Figure 5) and further experiments are still on-going now. We first performed an experiment in a static indoor environment to test the hierarchical hybrid algorithm for SLAM. The hand-measured world model is shown in Figure 6. And the map generated by raw laser scans by using direct method is shown in Figure 7. As seen from it, there are obvious inconsistencies among the map, especially when the robot moves into the pre-visited places such as corners of the map.

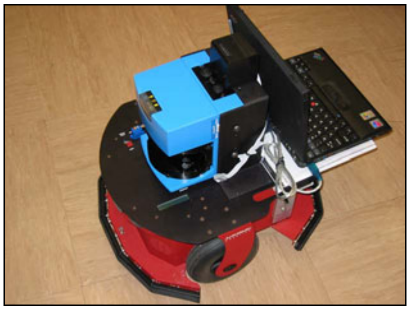

Figure 5. Pioneer 2DX mobile robot with SICK LMS200.

The map generated by proposed SLAM algorithm is shown in Figure 8. In Figure 8, the green place denotes unknown region, the gray place denotes free region, the dark place denotes the objects, and the red line just denotes the robot trajectory. It is easy to see from Figure 8 that the inconsistency problem is well solved, even robot frequently moves into pre-visited regions. In Figure 9, we also show two local grid maps how to work in global level.

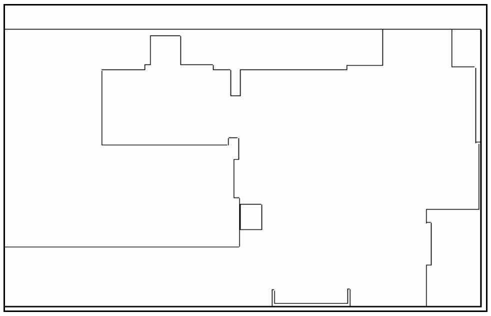

Figure 6. Hand-measured world model.

Figure 7. Map generated by raw scans of the laser

VI. CONCLUSION

In this paper, we have presented an online algorithm to solve SLAM problem in dynamic environments. Specifically, we have proposed a hierarchical hybrid method integrating ML and EKF with occupancy grid map and feature-based map to solve SLAM, and EFC based NN to solve MTT. Some preliminary experimental results have demonstrated the capability and robustness of our algorithm. Further experiments are still ongoing now. In the future work, we plan to explore autonomous navigation in dynamic environments based on this work.

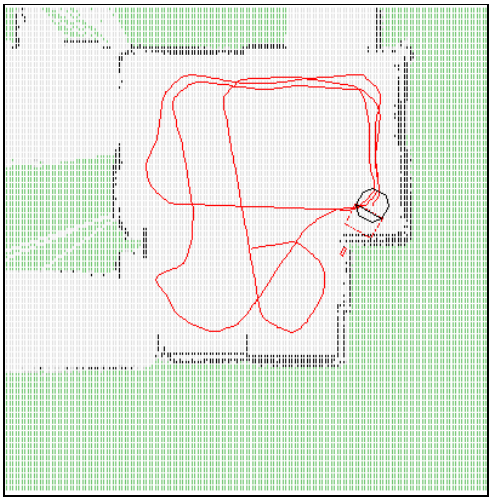

Figure 8. Map generated by the proposed algorithm.

Figure 9. Two local grid maps work in global level

REFERENCES

[1] H. I. Christensen, Lecture Notes: SLAM Summer School 2002. URL http://www.cas.kth.se/SLAM/.

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

[3] M. Montemerlo, W. Whittaker, and S. Thrun, "Conditional particle filters for simultaneous mobile robot localization and people-tracking", In Proc. *IEEE International Conference on Robotics and Automation*, Washington D.C., May 11-15, 2002

[4] J. Andrade-cetto, and S. Alberto, "Concurrent map building and localization on indoor dynamic environments", *International Journal of Pattern Recognition and Artificial Intelligence*, 2002, 16(3): 361-374.

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

[6] C.C. Wang, "Simultaneous Localization, Mapping and Moving Object Tracking". PhD thesis, Carnegie Mellon University. 2004.

[7] D. Wolf and G. S. Sukhatme, "Online simultaneous localization and mapping in dynamic environments", In Proc. *IEEE International Conference on Robotics and Automation*, New Orleans, LA April 26-May 1, *2004*.

[8] M. I. Jordan (ed.): Learning in Graphical Models. MIT Press. 1998.

[9] Kevin Patrick Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *PhD Thesis, UC Berkeley.* 2002.

[10] A. Elfes, "Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation", PhD thesis, Carnegie Mellon University, 1989

[11] X.R. Li, "Tracking in clutter with strongest neighbor measurement: i-theoretical analysis", *IEEE Transactions on Automatic Control*, 1998, 43(11): 1560-1578.

[12] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, 1981.

[13] R. Krishnapuram, and C. P. Freq, "Fitting an unknown number of lines and planes to image data through compatible cluster merging", *Pattern Recognition* 25(4), 385–400.