Abstract—To make a robot to work for and with human, the ability to simultaneously localize itself, accurately map its surroundings, and safely detect and track moving objects around it is a key prerequisite for a truly autonomous robot. In this paper, we explore the theoretical framework of this problem, i.e. Simultaneous Localization and Mapping (SLAM) with Multiple Target Tracking (MTT), and propose to employ Sequential Monte Carlo Methods (SMCM) as robust and computationally efficient algorithm. After mathematically formulating the problem, we apply a Rao-Blackwellized particle filter to solve SLAM which is partitioned into robot pose and feature location estimations and a conditioned particle filter to solve MTT which is partitioned into robot pose and moving object state estimations, both filters conditioned on robot pose. In detail, we propose Sampling Importance Resampling (SIR) method to estimate robot pose, Extended Kalman Filter (EKF) to estimate feature location, and Hybrid Independent/Coupled Sample-based Joint Probability Data Association Filter (Hyb-SJPDAF) to solve tracking and data association problem. We present some preliminary experimental results to demonstrate the capabilities of our approach.

Key words—Simultaneous Localization and Mapping (SLAM), Multiple Target Tracking (MTT), Sequential Monte Carlo Method (SMCM), Sampling Importance Resampling (SIR), Extended Kalman Filter (EKF), Joint Probability Data Association (JPDA)

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 [1]. But the real worlds the robots will be deployed in are usually dynamic. In order to design a robot not only to work for human, but also hopefully to work with human, it should have the ability to be conscious with the dynamics to achieve the goal of working for and with human. Recently, there exist several approaches to map building/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 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’s work [6] is the first to address the problem of simultaneous localization, mapping and moving object tracking, i.e., SLAM with DATMO (Detection and Tracking of Moving Objects).

But his work mainly focuses on representation of the world and data association. And the state estimation of the whole SLAM and DATMO is solved simply by using Extended Kalman Filter (EKF), which suffers from two problems that complicate its application in large, real-world environments: quadratic complexity, i.e., the computation complexity of EKF scales quadratically with the number of features in the map, which is always the case in real world; and sensitivity to failures in data association.

In this paper, we simultaneously incorporate MTT (Multiple Target Tracking) into SLAM, 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, as 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, SLAM can filter out moving objects and get more accurate localization and surrounding map.

<table>
<thead>
<tr>
<th>SLAM with MTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simultaneous Localization and Mapping (SLAM)</td>
</tr>
<tr>
<td>Motion detection</td>
</tr>
</tbody>
</table>

Figure 1: SLAM with MTT: mutually beneficial from each other

In this paper, we explore the theoretical framework of SLAM with MTT from a Bayesian point of view, and propose to use Sequential Monte Carlo Methods (SMCM) [7] as robust and computationally efficient algorithm to solve this problem. First, we will mathematically formulate this problem under some reasonable assumptions. So, our approach applies a Rao-Blackwellized particle filter to solve SLAM, like previously proposed by [8-9], which factors the SLAM problem into a robot localization problem, and a collection of feature estimation problems that are conditioned on the robot pose estimate, and then use a particle filter, which in our work is sampling importance resampling (SIR) method, for solving the former and EKF for the latter. At the same time, we use a conditioned particle filter to solve MTT problem, which tracks multiple moving objects also conditioned upon robot pose. Namely, to solve tracking and data association problem, we propose a Hybrid Independent/Coupled Sample-based Joint Probability Data Association Filter (Hyb-SJPDAF) to take the advantages of both Coupled-SJPDAF which assumes the target states are correlated and Independent-SJPDAF which assumes the target states are independent, similarly with [10].

This paper is organized as follows. In the following section, we will present the problem statement of SLAM with MTT from Bayesian perspective, with a theoretical Bayesian formulation. In section III, we will describe our solutions to SLAM and MTT respectively. Section IV presents some preliminary simulation and experiment results to demonstrate the capabilities and the robustness of our approach. Conclusion will come into section V.

II. PROBLEM STATEMENT

A. DBN

From Bayesian perspective, we approach the problem of concurrent localization, mapping and tracking. Figure 2 illustrates a generative Dynamic Bayesian Network (DBN). 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 $t$.

$$u_{0:t} = \{u_{0}, u_{1}, \ldots, u_{t}\} = \{u_{0}, u_{1}, u_{2}\}$$
$$z_{0:t} = \{z_{0}, z_{1}, \ldots, z_{t}\} = \{z_{0}, z_{1}, z_{2}\}$$
$$x_{0:t} = \{x_{0}, x_{1}, \ldots, x_{t}\} = \{x_{0}, x_{1}, x_{2}\}$$
$$m = \{m_{1}, m_{2}, \ldots, m_{l}\}$$
$$y_{t} = \{y_{1}, y_{2}, \ldots, y_{n}\}$$
$$w_{t} = \{w_{1}, w_{2}, \ldots, w_{n}\}$$

Figure 2: A DBN for concurrent localization, mapping and tracking 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 inferred from explicit ones.

We also assume the dynamic system as follows:

$$x_t = f_t(x_{t-1}, u_t, t) + \nu_t \quad \text{(2)}$$
$$z_t = h_t(x_t) + \omega_t \quad \text{(3)}$$

where: $f_t$ and $h_t$ are non-linear state transition model and observation model, respectively. $\nu_t$ and $\omega_t$ are noise vectors which are Gaussian, temporally uncorrelated and zero-mean.
B. Bayesian Formulation

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

1) Assumptions

Besides common assumption in SLAM literature, that is, SLAM problem is a Markov process, 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, 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.

\[ z_i = z^m_i + z^r_i \]
\[ \Rightarrow z^0_{i0} = z^m_{i0} + z^r_{i0} \] (4)
\[ \Rightarrow p(z_i | x_i, m, y_i) = p(z^m_i | x_i, m, y_i) p(z^r_i | x_i, m, y_i) = p(z^m_i | x_i, m) p(z^r_i | x_i, y_i) \]

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

\[ p(x_i, m | z^0_{i0}, u_{0:j}, y_i) = p(x_i, m | z^m_{i0}, u_{0:j}) \] (5)

2) Derivation

\[ p(x_i, m, y_i | z_{i0}, u_{0:i}) \]  
\[ \text{Bayes} \]
\[ \propto p(z_i | x_i, m, y_i) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]  
\[ \text{Markov} \]
\[ \propto p(z^m_i | x_i, m) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]  
\[ \text{Assumed} \]
\[ = p(z^m_i | x_i, m) p(z^r_i | x_i, y_i) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]  
\[ \text{Bayes} \]
\[ \propto p(z^m_i | x_i, m) p(z^r_i | x_i, y_i) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]  
\[ \text{Markov} \]
\[ = p(z^m_i | x_i, m) p(z^r_i | x_i, y_i) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]
\[ \text{Assumed} \]
\[ = p(z^m_i | x_i, m) p(x_i, m, y_i | z_{i0}, u_{0:i}) \]
\[ * p(z^r_i | x_i, y_i) \prod_j \left( p(y_j | x_j, u_{0:j}, y_{0:j}) p(x_j | z^0_{j0}, u_{0:j}, y_{0:j}) d\tilde{x} \right) \]
\[ = p(z^m_i | x_i, m) \prod_j \left( p(y_j | x_j, u_{0:j}) \right) \text{SLAM: } p(y_j | z^m_{0:j}, u_{0:j}) \]
\[ * p(z^r_i | x_i, y_i) \prod_j \left( p(y_j | x_j, u_{0:j}) \right) \text{MTT: } p(y_j | z^r_{0:j}, u_{0:j}) \]

III. SEQUENTIAL MONTE CARLO METHODS FOR SLAM WITH MTT

EKF provides an effective solution for the non-linear Gaussian case. However, it is always the case including non-linear and non-Gaussian elements in SLAM problem, especially when interacting with MTT. Sequential Monte Carlo Methods (SMCM) is a set of simulation-based methods which provide a convenient approach to compute posterior distributions [7]. So, we employ this approach in our work. As for SLAM, we use Rao-Blackwellised Particle Filter [9], which partitions SLAM into a localization and a mapping problem, and then uses particle filter to estimate the robot pose \( p(x_{0:i}, z^m_{0:j}, u_{0:i}) \) and EKF filter to estimate the location of features in the map \( p(m_i | z^m_{0:i}, u_{0:i}, x_{0:i}) \). Each particle has an independent EKF running for each feature in the map to estimate its position. Following factorization formulation (Eq.7) just shows decomposed SLAM, similar with FastSLAM [8]. NOTE that robot pose estimation will also be used in MTT.

\[ p(x_{0:i}, m | z^m_{0:i}, u_{0:i}) \]
\[ = p(x_{0:i} | z^m_{0:i}, u_{0:i}) p(m | z^m_{0:i}, u_{0:i}, x_{0:i}) \] (7)
\[ = p(x_{0:i} | z^m_{0:i}, u_{0:i}) \prod_i p(m_i | z^m_{0:i}, u_{0:i}, x_{0:i}) \]

As for MTT, similarly, we use a conditional particle filter by using following representation (Eq.8), because there are strong interaction between the robot pose estimate and moving target location estimates, depending upon where the robot is, sensor measurements may be explained by moving objects or stationary objects in the map; which is the key difference between MTT in concurrent localization, mapping and tracking problem, and normal MTT.

\[ p(y_i | z^r_{0:i}, u_{0:i}) \]
\[ = p(x_i | z^r_{0:i}, u_{0:i}) \prod_j p(y_{i,j} | z^m_{0:j}, u_{0:j}, x_i) \] (8)

As seen from above equation, now the moving object states \( p(y_{i,j} | z^m_{0:j}, u_{0:j}, x_i) \) are dependent on robot pose \( x_i \). Conditional particle filter represent both posterior of robot pose and moving object state using separate sets of particles. That is, moving object states \( p(y_{i,j} | z^r_{0:j}, u_{0:j}, x_i) \) also are represented by particle sets, just like using particle filter to estimate robot pose. NOTE that here the robot pose estimation is identical with that one in SLAM.
As so far, we have decomposed the problem concurrent localization, mapping and tracking into three posteriors: robot pose posterior \( p(x_t | z_{0:t}, u_{0:t}) \), feature location posterior \( p(m_i | z_{0:t}, u_{0:t}, x_{0:t}) \) and moving object state posterior \( p(y_{t,i} | z_{0:t}, u_{0:t}, x_t) \); and also generally propose to use particle filter to solve robot pose and moving object state posteriors and EKF to solve feature location posterior. So, now what we should to do is how to implement these posteriors.

A. SIR for Robot Pose Posterior

Algorithm SIR

1. For \( i=1, \ldots, N \), sampling \( x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}) \) which is the robot motion model.
2. For each particle calculate importance weight \( \alpha_t^{(i)} = p(z_t | x_t^{(i)}) \) which is the sensor measurement model.
3. Normalize the importance weights \( \tilde{\alpha}_t^{(i)} = \frac{\alpha_t^{(i)}}{\sum_{i=1}^{N} \alpha_t^{(i)}} \).
4. Evaluate \( \tilde{N}_t = \frac{1}{\sum_{i=1}^{N} (\tilde{\alpha}_t^{(i)})^2} \).
5. If \( \tilde{N}_t < N_{	ext{thresh}} \), then resample using the systemic resampling technique.

After generating \( N \) particles in this way, the resulting sample set \( S_t \) is distributed according to an approximation to the desired robot pose posterior \( p(x_t | z_{0:t}, u_{0:t}) \), an approximation which is correct as the numbers of particles \( N \) goes to infinity. NOTE that only the most recent robot pose estimate at time \( t-1 \) is used when generating the sample set at time \( t \), which allows us to silently “forget” all other pose estimate and make this robot pose estimate is identical with that one in MTT.

B. EKF for Feature Location Posterior

We use EKF to represent the conditional feature location estimates \( p(m_i | z_{0:t}, u_{0:t}, x_{0:t}) \). Since this estimate is conditioned on the robot pose, the Kalman filters are attached to each pose particle in the sample set \( S_t \). The overall algorithm in this part 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
   - \( \hat{x}_{t|t-1} = \mathbb{E}[\hat{x}_t | z_{0:t-1}] = H(z_{t|t-1}) + Q_{t|t-1} \)
   - \( P_{t|t-1} = \mathbb{E}[(x_t - x_{t|t-1})(x_t - x_{t|t-1})^T | z_{t-1}] \)
   
   where:
   \( \mathbb{E} \) : Jacobian matrix of state transition model with respect to robot state
   \( Q_t \) : state noise covariance model at time \( t \)
3. Update step
   - Innovation: \( \nu_t = z_t - \hat{x}_{t|t-1} \)
   - Innovation covariance: \( S_t = \mathbb{E}[\nu_t \nu_t^T] = \nabla H \nabla H^T + R_t \)
   - Kalman gain: \( K_t = P_{t|t-1}\nabla H^T S_t^{-1} \)
   - \( \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (z_t - \hat{h}(\hat{x}_{t|t-1})) \)
   - \( P_{t|t} = P_{t|t-1} - K_t S_t K_t^T \)
   
   where:
   \( \nabla H_t \) : Jacobian matrix of measurement model with respect to robot state
   \( R_t \) : measurement noise covariance model at time \( t \)

C. Hyb-SJPDAF for Moving Object State Posterior

When incorporating MTT into SLAM, the fundamental issue is motion detection, i.e. we already discriminate which measurement data belongs to moving objects and which belongs to stationary objects, also should address data association problem, i.e. which measurement is caused by which object. Since motion detection is not the main point in this paper, a fairly simple consistency-based method is used to solve this problem. Any inconsistency between the map and the new measurement should belong to a potential moving object. So, we simply compare the values of new measurement and local map after transforming them into...
same coordination to detect moving objects. To track multiple moving objects, here consider \( k \) targets, one generally has to evaluate the joint probability distribution of the state of all objects. Firstly, we have following notations:

Let \( z_t \) denote the measurement at time \( t \) with \( m \) measurements in the validation region

Let \( X_t \) denote the state of all \( k \) objects at time \( t \)

Let \( \theta \) denote joint association event and \( \theta_{ji} \) is the particular event which assigns measurement \( j \) to target \( i \).

Let \( \beta_{ji} \) denote the association probability

Let \( z_{0d} \) denote Null-measurement in which no target has been detected.

In MTT literature [10], we have

\[
\beta_{ji} = P(\theta \mid Z_{ti}) = P(\theta \mid Z_t, Z_{t-1}, \ldots, Z_0)
\]

Markov \( P(\theta \mid Z_t, X_{ti}) \)

\[
= \frac{1}{c} p(Z_t \mid \theta, X_{ti}) p(\theta \mid X_{ti})
\]

\[
= P_{D}^{k-1}(1-P_{D})^{s} \prod_{\theta_{ji}=0} p_{ji}
\]

where

- \( P_{D} \) is the probability of detection
- \( P_{F} \) is the probability of false alarm
- \( p_{ji} \) is the association likelihood that measurement \( j \) assigns to target \( I \), which evaluated based on particles:

\[
p^{s}_{ji} = \frac{1}{(2\pi)^{d/2} |S|} \exp \left( -\frac{d^{ij}}{2} \right)
\]

And now, we can use Hyb-SJPDAF which combines independent-SJPDAF and coupled-SJPDAF, one of which assumes target states are independent and another assumes they are correlated, to address MTT problem. NOTE that data association problem for features in the map is simultaneously evaluated based on particles:

\[
\theta_{ji} = \sum_{\theta_{ji}} p(\theta \mid Z_{si})
\]

\[
P(\theta \mid Z_{si}) = P_{D}^{k-1}(1-P_{D})^{s} \prod_{\theta_{ji}=0} p_{ji}
\]

- For each target \( i=1, \ldots, k \) generate a new set \( \{X_{i}^{(s)}i\}_{s=1}^{N} \) by resampling with \( N \) times from

\[
\{X_{i}^{(s)}\}_{s=1}^{N} \quad \text{where} \quad P(X_{i}^{(s)}i) = X_{i}^{(s)}i = \theta_{ji}^{(s)}
\]

- For \( s=1, \ldots, N \) and for each target \( i=1, \ldots, k \), predict new particles as Eq.2

\[
X_{i}^{(s)}i = f(X_{i}^{(s)}i, u^{(s)}i)
\]

and calculate \( \hat{N}_{d} \) as Eq.10

3. \( \hat{N}_{d} > N_{min} \), adopt Coupled-SJPDAF as follows:

- For each particle \( s=1, \ldots, N \), calculate the weights for all measurement to track association

\[
\theta_{ji} = \sum_{s=1}^{N} p(H_{ji}^{s}) \quad \text{where} \quad P(H_{ji}^{s}) = P(\theta \mid Z_{si})
\]

- Predict new particles (Eq.23), for \( s=1, \ldots, N \), and calculate \( \hat{N}_{d} \) (Eq.10)

4. \( t=t+1 \), go to step 2

IV. EXPERIMENTAL RESULTS

The proposed algorithms were tested and are being tested under both simulation and experiment. For SLAM part in concurrent localization, mapping and tracking problem, we proposed to employ Rao-Blackwellised particle filter, i.e. using SIR technique to approximate robot pose and EKF to compute feature location based on each particle. We validated this method in 1D simulation with 2 features by comparing EKF-SLAM algorithm. In this simulation, we assume that the true robot pos is 20, two feature locations are 50, 70; and measurement model is a simple linear uncorrelated model and robot motion model is also a simple linear model; and here we use 1000 particles. Table 1 shows the results as proposed filter works, and from it, we can see our algorithm can well approximate the posteriors of robot pose and feature locations.

Currently we are validating our proposed algorithm on a pioneer 2 mobile robot equipped with SICK LMS200 (Figure 3 left), and Figure 3 right shows the SLAM by raw data. Both dynamic SLAM and MTT have to be tested in real dynamic environments which contain moving objects. And these works are on-going now.
TABLE I: SIMULATION RESULTS BY COMPARING PROPOSED RBBF WITH EKF SLAM

<table>
<thead>
<tr>
<th>Particle num=1000</th>
<th>Rao-Blackwellised Particle Filter</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Robot pose</td>
<td>1st feature</td>
</tr>
<tr>
<td>initial pose</td>
<td>(20.0000,50.5000,69.0000)</td>
<td>((0.0000,0.0000,0.0000),(0.0000,0.0000,0.0000),(0.0000,0.0000,3.0030))</td>
</tr>
<tr>
<td>predicted pose</td>
<td>(18.1088,50.5000,69.0000)</td>
<td>((10.644918,0.0000,0.0000),(0.0000,5.0050,0.0000),(0.0000,0.0000,3.0030))</td>
</tr>
<tr>
<td>1st observed pose</td>
<td>(20.2804,50.6403,69.0000)</td>
<td>((4.5361,5.2629,0.0000),(5.2629,5.1366,0.0000),(0.0000,0.0000,3.0030))</td>
</tr>
<tr>
<td>2nd observed pose</td>
<td>(18.6925,49.8463,69.3462)</td>
<td>((4.1223,2.0592,2.0591),(2.0592,3.5332,1.0296),(2.0591,1.0296,2.5321))</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we presented the theoretical framework SLAM with MTT. AND we proposed to employ sequential Monte Carlo methods (SMCM) to solve this rather new problem, which build a bridge between SLAM literature and MTT literature. In detail, under the framework, sampling importance resampling (SIR) technique is used to approximate the robot pose posterior; extended Kalman filter (EKF) based on each particle of robot pose is used to compute feature or stationary object location state; a hybrid independent/coupled sample-based joint probability data association filter (Hyb-SJPDAF) which is also conditioned on robot pose, is used to solve moving object tracking and data association problem.

REFERENCES