Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Data Fusion and Efficient Algorithm for Moving Target Tracking

Guocheng Liu, and Yongji Wang

Chinese Education Ministry Key Lab. for Image Processing and Intelligence Control
Department of Control Science and Engineering, Huazhong University of Science and Technology
P.R. China (Tel: 0086-27-68772267; e-mail: gcliu2004@hotmail.com)

Abstract: The principle of target tracking and data fusion techniques are discussed. To resolve high uncertainty that exists in sensors of mobile robots, one multi-sensor data fusion algorithm is presented. The algorithm is based on particle filter techniques, fuses the information coming from multiple sensors and merges different state space models. So it can be used to eliminate system and measurement noise and estimate value of position and headings of mobile robot. On simulation experiments, we compare different cases such as single sensors and multi-sensor data fusion, the results demonstrate the feasibility and effectiveness of this algorithm and exhibits good tracking performance.

1. INTRODUCTION

Target tracking is the estimation of unknown target kinematical state based on indirect, inaccurate and uncertain measurements from sensors. The unknown target kinematics of interest is usually the position, velocity, and acceleration of the moving target. The sensor measurements are usually perturbed by noise and contain information about the target kinematics. The objective of target tracking is to collect sensors data from a field of view containing potential target information and to partition the sensor data into sets of observations or tracks that are originated from the same sources. Figure 1 depicts a target tracking scenario.

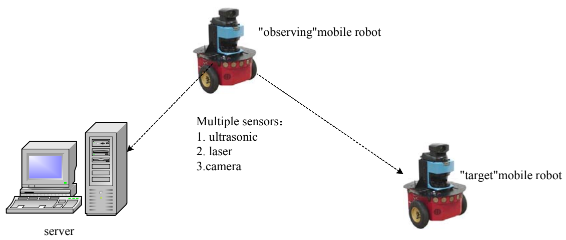

Fig. 1 An illustration of a target tracking scenario

One complex tracking system (e.g. multiple robots system) usually has multiple sensors in order to improve the accuracy of tracking. For the sensor networks consisting of multiple sensors, there exist two kinds of architectures for tracking. One is hierarchical, consisting of lead nodes and sensing nodes. The lead nodes perform data fusion, keep records of the targets and communicate with the neighboring lead nodes; the sensing nodes collect signals and report to their leader nodes. Another architecture has no limits about primary and secondary sensors. A neighborhood of sensors nodes is dynamically formed when a target presented. All sensors receiving signals with strength above a threshold constitute an active neighborhood. The one with strongest signals plays the leading role. Often several sensors of the same type are used, but different types of sensors could be used. The sensors may be placed at fixed location or on moving platform. In multiple sensors systems, the information from each sensor may be have different modalities, it need use computer techniques to analyze automatically the information from those sensors under some rules; optimize and synthesize them to finish the estimation and make the decision. All of these improve rapidly the development of multiple sensors data fusion technique (Blanc *et al*. 2005; Hall *et al*. 1992; Prahlad *et al*. 2006).

Sensor fusion, i.e., combining data from several sensors to improve the estimate, can easily be incorporated in the Bayesian approach. Recently, particle filter, also known as Sequential Monte Carlo Methods, have become popular tools to solve tracking problems (Doucet *et al*. 2001; Ristic *et al*. 2004). The popularity stems from their simplicity, flexibility, and systematic treatment of non-linearity and non-Gaussian. This technique is a numerical method based on simulation and to use discrete hidden Markov chain for modeling, the system model describes the evolvement of the unknown state (target) over time and the measurement model associates the available measurements to the target state. Using the past and present measurements, basis on prior information, we can perform the prediction and update step, provide an approximate distribution for the target state. These have significantly increased mobile robots in tracking, navigation, map building etc. (Thrun *et al*. 2005).

This paper applies particle filter to implement the state estimation and multi-sensor data fusion, and presents a new data fusion algorithm. It is organized as follows. In Section II, the sensor system of the mobile robots is introduced; the data fusion technique is discussed and the target tracking formulations are explained. In Section III, we propose the detailed algorithm and data model. The simulation results are given in Section IV followed by the conclusion in Section V.

2. DATA FUSION AND TARGET TRACKING

2.1 Sensor Systems

The sensor systems are the medium of the robots probe the environment and it is the prominent embodiment about robots' intelligence. In Fig. 1, the robots have these sensors: ultrasonic sensors; laser measurement system and vision system.

In order to simplify the algorithm, due to the particularity of vision sensors, only ultrasonic and laser sensors are used in this paper.

2.2 Data Fusion

Data fusion for estimation, or estimation fusion, is the problem of how to best utilize useful information contained in multiple sets of data for the purpose of estimation an unknown parameter or process. These measurements may be of different types or include conflicting information. The multiple sets of measurements are usually but not necessarily obtained from multiple sensors. Even if the measurements coming from single sensors, we can artificially treat them coming from different locations and view it as the fusion problem. In this sense, estimation itself is fusion by fusing the prior and posterior information, and filtering is fusion by fusing the prediction and current measurement.

Depending on whether the raw measurements are sent to the fusion center or not, there are two basic estimation architectures: centralized and distributed (also referred to as measurement fusion and track fusion in target tracking, respectively). As in Fig. 2(a), 2(b) show, for measurement fusion, all raw measurements are sent to the fusion center; for track fusion, each sensor only sends the processed data to the fusion center.

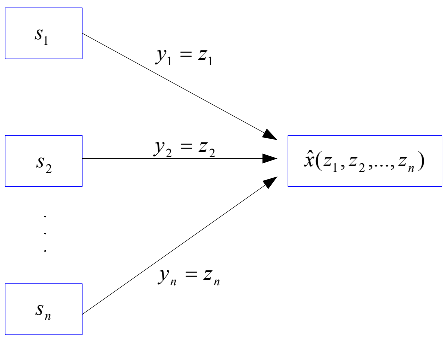

Fig. 2(a) Measurement (Centralized) Fusion

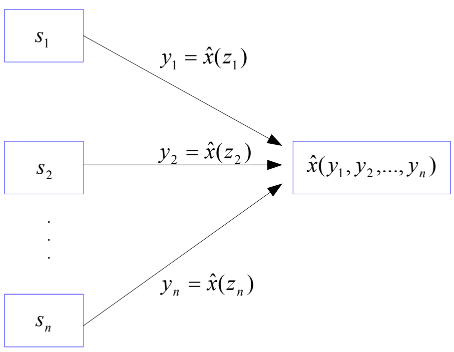

Fig. 2(b) Track (Distributed) Fusion

2.3 Target Tracking Formulations

A state space approach (as in Fig. 3) is used to formulate the tracking problem. In this framework, the unknown kinematics of the target is called the state. Supposing the sequence of the state is belong to one-order Markov process, the evolution of the target state x_k with time k is described by a system model as in

$$x_{k+1} = f(x_k) + w_{k+1} \tag{1}$$

where x_k is a vector of size n_x, $f : R^{n_x} \mapsto R^{n_x}$ is a known nonlinear function of the state, w_{k+1} is an additive, independent and identically distributed process-noise vector of dimension n_x with covariance matrix Q_{k+1}.

The target measurement vector z_k^l of size n_l is obtained by using the l^{th} ($l = 1,...,L$) sensor, and it is related to x_k by the measurement model

$$z_k^l = h_k^l(x_k) + v_k^l \tag{2}$$

where h_k^l is a known nonlinear measurement function for the l^{th} sensor and v_k^l is an additive independent and identically distributed measurement noise sequence vector of size n_l with covariance matrix R_k^l. The system and measurement models in (1) and (2) are collectively called the dynamic model.

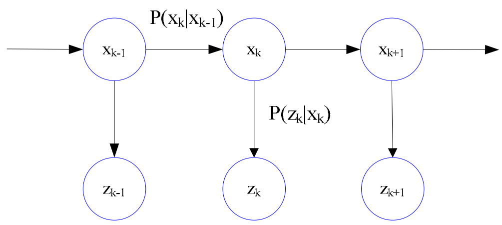

Fig. 3 the state space model using Bayesian networks

In sensor networks, the type of sensors may be the same or different, so there exists cross-sensor and cross-modality data fusion to derive the composite measurement at the fusion center and its corresponding likelihood density function (Wu *et al.* 2006). As shown in Fig. 4, the signals using the same type denoted as $Z^m = (Z_1^m, Z_2^m, ..., Z_N^m)$ have the same modality. Because data elements in Z^m are collected by different sensors at different locations, they will contribute collectively in the determination of source location. In the application these sensors are dependent. As such, cross-sensor data fusion is usually embedded in the measurement model and incorporated into the derivation of likelihood function. The multiple modality data $Z = (Z^1, Z^2, ..., Z^L)$ comes from different types of sensors and they contribute complementary to the estimation. We refer this type of data fusion as cross-modality data fusion. These sensors work independently and the data fusion can be formulated as in

$$p(Z|x) = p(Z^1|x)p(Z^2|x)...p(Z^L|x) \tag{3}$$

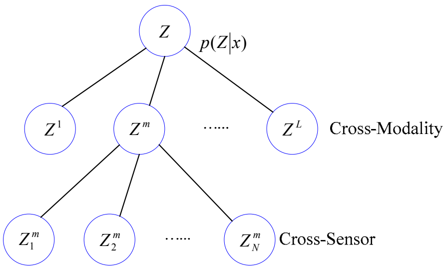

Fig. 4 Cross-Sensor and Cross-Modality in sensor networks

3. DATA MODEL AND ALGORITHM

From a Bayesian perspective, the tracking problem is to recursively evaluate the maximum a posterior (MAP) distribution $p(x_k|z_{1:k})$, given a sequence of measurement $z_{1:k}$. It is assumed that the initial probability density for the state, $p(x_0)$, is available. Then, in principle, $p(x_k|z_{1:k})$ may be obtained recursively in two stages: the prediction stage and the update stage. The prediction stage involves using the system model to obtain the estimate of the state at the next time interval. Then in the update stage, the likelihood density function can be calculated using the measurement model to update the state generated at prediction stage. The two stages can be formulated as

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{4}$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{5}$$

Principally, using above model and equation, for a given initial state and measurement sequences, we can evaluate states at any time instance k. Unfortunately, most of the presented nonlinear filtering methods (EKF, UKF and DDF), which are based on local linearization of the nonlinear system equations or local approximation of the probability density of the state variables, have yet been universally effective algorithms for dealing with both nonlinear and non-Gaussian system. For these nonlinear and non-Gaussian problems, the sequential Monte Carlo method was investigated (Doucet *et al.* 2001; Gordon *et al.* 1993).The sequential Monte Carlo filter can be loosely defined as a simulation-based method that uses a Monte Carlo simulation scheme in order to solve on-line estimation and prediction problems, also known as particle filter.

3.1 Particle Filter

According to Monte Carlo methods (Metropolis *et al.* 1949), assume that we are able to draw N identical independently distributed random samples, also named particles, $\{X_k^{(i)}, i = 1,..., N\}$ from $p(x_k|z_{1:k})$, the PDF of the state can be approximated as

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} (1/N)\delta(x_k - x_k^{(i)}) \tag{6}$$

where $\delta(x_k - x_k^{(i)})$ denotes the Dirac delta function.

Unfortunately, it is usually impossible to sample efficiently from the posterior distribution $p(x_k|z_{1:k})$ at any time. An classical solution consists of using the importance sampling (IS) which draw N independent samples from the importance function $q(\cdot)$, so the $p(x_k|z_{1:k})$ is approximated by

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} \tilde{\omega}_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{7}$$

where the normalized importance weight $\tilde{\omega}_k^{(i)}$ are given by

$$\tilde{\omega}_k^{(i)} = \omega_k^{(i)} \Big/ \sum_{i=0}^{N} \omega_k^{(i)} \tag{8}$$

If the samples $x_k^{(i)}$ were drawn from the importance function $q(x_k|z_{1:k})$, then the importance weights can be calculated as

$$\omega_k^{(i)}(x_{0:k}) = \frac{p(x_{0:k}|z_{1:k})}{q(x_{0:k}|z_{1:k})}.$$

Assume that the importance function is chosen such that

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}) \tag{9}$$

The weights are recursively updated as

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, z_{1:k})} \tag{10}$$

Prediction, update and evaluation of importance weights constitute the sequential importance sampling (SIS). In any SIS framework after recursions, it can be proved that the variance of the weights increase systematically over time with the consequence that we associate unit weight to one particle and zero to the other. This degeneracy problem can be monitored by increasing the number of particles. But this method has no benefit for real-time applications. So the ordinary solutions are to choose the importance function and to use a resampling operation.

In order to get good estimation, the distribution of importance function should be close to the posterior distribution of the true state, so the variance of the importance weights should small as soon as possible, and having proved that, as to Markovian system, the optimal importance function is

$$q(x_k|x_{k-1}, z_{1:k}) = p(x_k|x_{k-1}, z_k) \tag{11}$$

N. Gordon proposes the strategy of resampling to control the degeneracy problem (Gordon *et al*. 1993). This resampling probabilistically replicates particles with high weights and discards particles with low weights. SIS and resampling constitute the SIR particle filter. The resampling is an optional step based on weights and it could bring the impoverishment problem. In order to check whether resampling is necessary or not, a threshold criterion is defined, $N_{eff} < N_{th}$, with the threshold N_{th} and the effective sample size N_{eff} defined by $N_{eff} = 1/\sum_{i=1}^{N}(\tilde{\omega}_k^{(i)})^2$.

3.2 System Model

In the tracking scenario of Fig. 1, there are two Pioneer 2 mobile robots, one is the observing robot, its position is fixed and it can probe another mobile robot (target) moving trajectory using the laser sensor. The state variable of the target at time k is defined as $\chi_k = (x_k, y_k, \theta_k)^T$, here, let x_k, y_k, θ_k represent the position and heading angle of the target in the Cartesian coordinates. During the observed robot (target) is moving, there exists an obstacle (wall), the distance is measured by the ultrasonic sensors.

The dynamics of the mobile robot can be modelled by two parts, the rotation and the translation. The rotation and its uncertainty can be modelled as

$$\theta_{k+1} = \theta_k + \delta\theta + N(\mu_{rotation}, \sigma_{rotation}) \tag{12}$$

where $\delta\theta = \mathrm{atan}\frac{\Delta y}{\Delta x}$ denotes the changes in angle of the target at time k. Assume the noise is the independent and identically distributed Gaussian random process.

The translation has two sources of errors: the actual travelled distance and the changes in orientation during the forward translation (drift). The dynamics are defined in this model:

$$\theta_{k+1} = \theta_k + \delta\theta + N(\mu_{drift}, \sigma_{drift}) \tag{13}$$

$$x_{k+1} = x_k + (\delta\rho + N(\mu_{translation} \times \delta\rho, \sigma_{translation} \times \delta\rho)) \times \cos(\theta_{k+1}) \tag{14}$$

$$y_{k+1} = y_k + (\delta\rho + N(\mu_{translation} \times \delta\rho, \sigma_{translation} \times \delta\rho)) \times \sin(\theta_{k+1}) \tag{15}$$

Also assume the noise is the independent and identically distributed Gaussian random process. We use the format below to denote the (12),(13),(14) and (15)

$$x_{k+1} = x_k + \delta x_k + N(\mu, \sigma) \tag{16}$$

where δx_k represents the changes in the distance and orientation at time k.

For the sake of simplicity and effectiveness, we choose the SIR particle filter for the estimation of the state. The importance function and update equation are

$$q(x_{k+1} | x_k^{(i)}, z_{k+1}) \rightarrow p(x_{k+1} | x_k^{(i)}) \tag{17}$$

$$\omega_{k+1}^{(i)} = p(z_{k+1} | x_{k+1}^{(i)}), \quad \sum_{i} \tilde{\omega}_k^{(i)} = 1 \tag{18}$$

3.3 Measurement Model

Although the tracking procedure in Fig. 1 can be applied under quite general measurement models, the assumption of a practical model is necessary for the simulation analysis.

There are 16 ultrasonic sensors mounted on a ring around the Pioneer 2 so that a 360° environment can be explored. Assume the position of the obstacle (wall) is known and its projection on the Cartesian coordinates is one line which can be denoted as $y = hx + d$, so the distance is measured that leads to the following measurement model about the n^{th} ultrasonic sensor

$$z_k^n = \left(\frac{2h}{\sqrt{h^2+1}}\right)x_k + \left(\frac{-2}{\sqrt{h^2+1}}\right)y_k + \frac{2d}{\sqrt{h^2+1}} \tag{19}$$

The measurement of ultrasonic ring is denoted using one vector

$$z^{us} = (z^1, z^2, \ldots, z^N)^T \tag{20}$$

The measurement model about the laser sensor is

$$z_k^{las} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} = \begin{pmatrix} \sqrt{(x_k - x_s)^2 + (y_k - y_s)^2} \\ a\tan(\frac{y_k - y_s}{x_k - x_s}) - \theta_k \end{pmatrix} \tag{21}$$

where (x_s, y_s) is the position of the observing robot which is fixed. Also, the appropriate noise will be added on these measurement models and it belongs to the independent and identically distributed Gaussian random process. The system model and the measurement model are independent. The equation to update the importance weight for fusing the multi-modality measurement data is given by

$$\omega_k^{(i)} = p(z^{us} | x_k^{(i)}) p(z^{las} | x_k^{(i)}) \tag{22}$$

3.4 Algorithm Implementation

Here summarizing the algorithm implementation for target tracking in sensor networks using ultrasonic and laser sensors. The Cross- Sensor and Cross- Modality (CSCM) data fusion algorithm is presented below.

Step1: Initialization: Randomly generate an initial pose of mobile robot, $x^{(i)}, i = 1,2,\ldots,N$, in location space, It is assumed that a mobile robot move on the plane.

Step2: Prediction: for each particle $i = 1:N$, draw $x_k^{(i)} = x_k^{(i)} + \delta x_{k-1}$.

Step3: Determine active neighborhood by threshold of received sensor signals to generate vector z^{us} and z^{las}.

Step4: Update the weights $\omega_k^{(i)} = p(z^{us} \left| x_k^{(i)} \right.) p(z^{las} \left| x_k^{(i)} \right.)$ and normalize the weights.

Step5: Compute the expectation of the state of the target $\hat{E}(x_k) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\omega}_k^{(i)} x_k^{(i)}$.

Step6: Compute the N_{eff} and perform particle resampling whenever $N_{eff} < N_{th}$

4. SIMULATION AND RESULTS

Given a particle distribution, we need to find the state which defines with accuracy the target position. We use 3 different methods: For weighted particle set $\{(x_t^{(i)}, \omega_t^{(i)})\}_{i=1}^{N}$, firstly, the best particle (the $x_t^{(j)}$ such that $\omega_t^{(j)} = \max(\omega_t^{(i)})$) be used; secondly, the weighted mean ($\bar{x}_t = \sum_{i=1}^{N} x_t^{(i)} \omega_t^{(i)}$) and, lastly, the robust mean ($\bar{x}_t = \sum_{i=1}^{N} x_t^{(i)} \omega_t^{(i)} and \left| x_t^{(i)} - x_t^{(j)} \right| < \varepsilon$). Each method is of its advantages and disadvantages: the best particle introduces a discretization error, while the weighted mean fails when faced with multi-modal distribution; the best method is the robust mean but it is also the most computationally expensive. In cases where the target is surrounded of objects whose characteristic is similar, the best method is to use as state that defines the target position the best particle.

4.1 Single Sensor

1. Target tracking using ultrasonic sensor only: tracking result as shown in Fig. 5(a), 5(b), 5(c), 5(d).
2. Target tracking using laser sensor only: tracking result as shown in Fig. 6(a), 6(b), 6(c), 6(d).

4.2 Multi- Sensor Data Fusion

Target tracking using ultrasonic and laser sensors, tracking result as shown in Fig. 7(a), 7(b), 7(c), 7(d).

4.3 Results

In this paper, we have two data fusion approaches for moving target tracking. The performance under 3 different cases is compared. The results are shown on Table 1. Our particle filter algorithm is run with 500 particles and 100 experiments. Performance is measured in terms of the Root Mean Square Error (RMSE), defined as:

$$RMSE = \sqrt{\frac{1}{S}\sum_{i=1}^{S}\frac{1}{N_{mc}}\sum_{m=1}^{N_{mc}}(x_k^{i,m} - x_k^{true})^2} \quad (20)$$

where $x_k^{i,m}$ is the estimated target state (position and heading angle) at time k for m^{th} run. $N_{mc} = 100$ is the total independent runs. $S = 25$ is the number of steps. Hence, by fusing multi-information coming from multi-sensors, we were able to obtain much better tracking results, its mean square error are smaller than tracking using single sensor only. Also from the Fig. 5(a), 6(a) and 7(a) (the moving mobile robot is at the same position), we could get the conclusion intuitively. What is more, the estimation based on our algorithm is better than EKF algorithm, but the EKF has much less computational load than the particle filter.

.

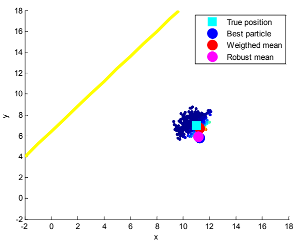

Fig. 5(a) tracking using ultrasonic sensor, the yellow line is the wall

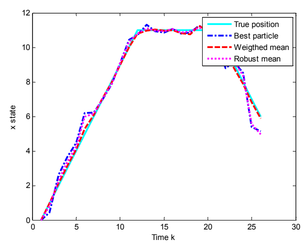

Fig. 5(b) X state estimation

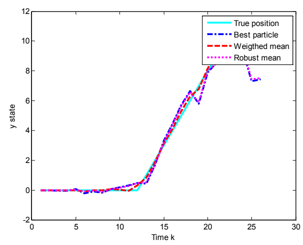

Fig. 5(c) Y state estimation

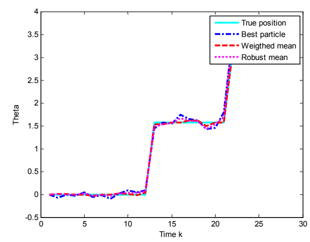

Fig. 5(d) Theta state estimation

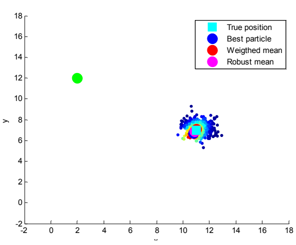

Fig. 6(a) tracking using laser sensor, the green dot is the observing robot

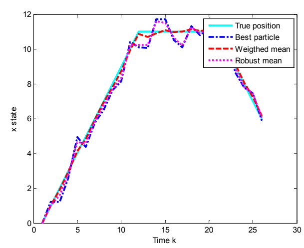

Fig. 6(b) X state estimation

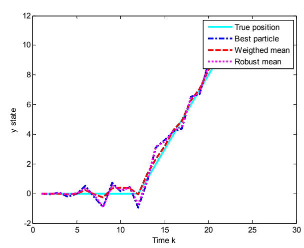

Fig. 6(c) Y state estimation

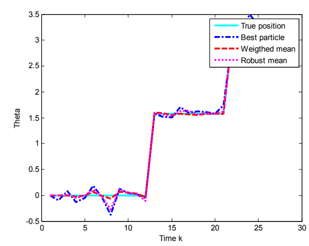

Fig. 6(d) Theta state estimation

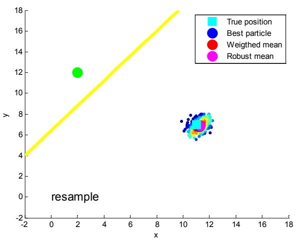

Fig. 7(a) data fusion for target tracking process

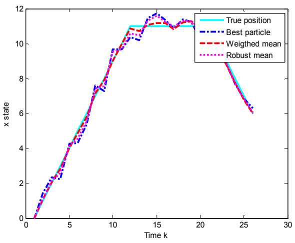

Fig. 7(b) X state estimation

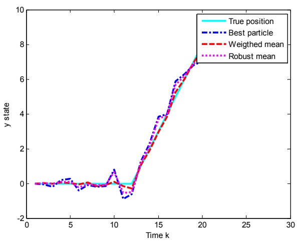

Fig. 7(c) Y state estimation

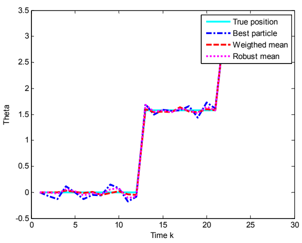

Fig. 7(d) Theta state estimation

Table 1. Performance comparison using EKF and our PF algorithm

Sensors	Algorithms	Time (s)	RMSE		
			x	y	θ
Ultrasonic Sensors (US)	EKF	10.1143	4.4961e-004	5.9919e-004	1.8586e-005
	PF	12.8085	6.2778e-005	5.2895e-005	1.6088e-006
Laser	EKF	7.9536	8.720e-005	7.4365e-005	6.5030e-006
	PF	10.5551	2.8555e-006	5.8685e-006	6.2522e-007
Fusion of US and Laser	EKF	12.1047	7.5947e-005	6.7309e-005	5.9097e-006
	PF	14.5279	1.5268e-006	2.4359e-006	5.5364e-007

5. CONCLUSIONS

In tracking application, the target state (e.g. position, velocity, acceleration) can be estimated by processing the measurements collected from all deployed sensors. The estimation performance significantly relies on the accuracy of the sensor when data fusion is conducted. In this paper we have presented a Cross-Sensor and Cross-Modality (CSCM) data fusion algorithm based on sequential Monte Carlo methods. By fusing multi-information coming from multi-sensors, integrating different state-space models, we can track the moving mobile robot. In the simulation experiments, we compare different cases and the results show the feasibility and the effectiveness of the algorithm. Our methods can be extended to multi-target tracking, which is our future work.

ACKNOWLEDGMENT

This work was supported by a grant from the National Natural Science Foundation of China (No. 60674105) and the Ph. D. Program Foundation of Ministry of Education of China (No. 20054870013).

REFERENCES

Blanc C., L. Trassoudaine and J. Gallice, EKF and particle filter track to track fusion : a quantitative comparson from radar/ lidar obstacle tracks, *2005 7th International Conference on Information Fusion (FUSION)*, pp. 1303-1310.

Doucet, A., Nando de Freitas, and N. Gordon (2001). *sequential Monte Carlo methods in practice*, Springer-Verlag.

Gordon, N., D. J. Salmond, and A. F. M. Smith (1993). Novel approach to nonlinear/non-Gaussian state estimation, *IEE Proceedings-F*, **vol. 140**, no. 2, pp. 107-113

Hall, D. L. (1992). *Mathematical techniques in multisensor data fusion*, Artech House, Inc.

Metropolis, N., and S. Ulam (1949). The Monte Carlo method, *Journal of American Statistical Association*, **vol. 44,** no. 247, pp. 335-341.

Prahlad V., and J. Liu (2006). Improved particle filter in sensor fusion for tracking randomly moving object, *IEEE Transaction on instrumentation and measurement*, **vol. 55,** no. 5, pp. 1823-1832.

Ristic, B., S. Arulampalam, and N. Gordon (2004). *Beyond the Kalman filter: particle filters for tracking applications*, Boston, MA: Artech House Publishers.

Thrun, S., W. Burgard, and D. Fox (2005). *Probabilistic robotics*, The MIT Press.

Wu, J. K., and Y. F. Wong (2006). Bayesian approach for data fusion in sensor networks, *Information Fusion, ICIF'06 9th International Conference*, pp. 1-5.