

Rao-Blackwellised Particle Filter with Adaptive System Noise and its Evaluation for Tracking in Surveillance

Xinyu Xu, Baoxin Li

Center for Cognitive Ubiquitous Computing

Dept. of Computer Science and Engineering, Arizona State University

699 South Mill Avenue, Tempe, AZ, USA 85281

{xinyu.xu and baoxin.li}@asu.edu

ABSTRACT

In the visual tracking domain, Particle Filtering (PF) can become quite inefficient when being applied into high dimensional state space. Rao-Blackwellisation^[1] has been shown to be an effective method to reduce the size of the state space by marginalizing out some of the variables analytically^[2]. In this paper based on our previous work^[3] we propose an RBPF tracking algorithm with adaptive system noise model. Experiments using both simulation data and real data show that the proposed RBPF algorithm with adaptive noise variance improves its performance significantly over conventional Particle Filter tracking algorithm. The improvements manifest in three aspects: increased estimation accuracy, reduced variance for estimates and reduced particle numbers are needed to achieve the same level of accuracy. The last two performance improvements are evaluated in this paper using simulation data.

1. INTRODUCTION

Visual tracking is an important step in many practical applications including video-based surveillance. In recent years, particle-filter-based visual tracking has been extensively studied^[4, 5, 6 and 7]. Particle filtering has been shown to offer improvements in performance over some conventional methods such as the Kalman filter, especially in non-linear/non-Gaussian environments^[8], but the large number of samples required to represent the posterior density prevent its use in high dimensional state-space. However, in some cases, the model may have “tractable structure” with some components having linear dynamics and so can be marginalized out and analytically estimated using exact filters conditional on certain other components. The exact filters could be the Kalman filter, the HMM filter, or any other finite dimensional optimal filters^[1]. This technique is called Rao-Blackwellisation^[2]. The resultant method is often called Rao-Blackwellised particle filter (RBPF).

RBPF has been studied in the context of multi-target tracking^[9], fault diagnosis for robot and robot localization^[10, 11] and signal processing^[12]. In addition, in our previous work^[3], with video-based surveillance as a case study, we utilized the constraints imposed by typical camera-scene configuration to partition the original state space into two sub-spaces, and then we proposed a RBPF algorithm for surveillance tracking. However, in most existing tracking algorithms including both RBPF and PF, the noise variance in the system motion model is typically static which makes the tracker unstable and un-robust when tracking objects with dramatically and fast changing velocity. To overcome this drawback and based on our existing work, in this paper we present a RBPF algorithm with adaptive system noise model. Furthermore, a systematic evaluation of a RBPF algorithm over conventional particle filtering is not available in the visual tracking literature. Motivated by this fact, in this paper we evaluate the improvements of RBPF over conventional particle filter based on the RBPF algorithm with adaptive system noise model. Experimental results show that the improvements of RBPF over conventional PF manifest in the following three aspects: reduced variance for estimates, reduced number of particles needed to achieve the same level of accuracy, and increased estimation accuracy.

Denote the state to be tracked as X_t and observation as Z_t with subscript t the time index. The key idea of RBPF for tracking is to partition the original state-space X_t into two parts R_t (*root variables*), and L_t (*leaf variables*), such that $p(L_{1:t}|R_{1:t}, Z_{1:t})$ can be analytically updated using an exact filter, and thus the approximation to $p(R_{1:t}|Z_{1:t})$ using a Monte Carlo method yields straightforwardly an approximation to joint posterior $p(R_{1:t}, L_{1:t}|Z_{1:t})$. The justification for this decomposition follows from the factorization of the probability:

$$p(R_{1:t}, L_{1:t}|Z_{1:t}) = p(L_{1:t}|R_{1:t}, Z_{1:t})p(R_{1:t}|Z_{1:t}) \quad (1)$$

2. PRIOR WORK

In a typical surveillance application, the camera is mounted relatively much higher than the dominant plane and the to-be-tracked object is constrained to move on a dominant plane (e.g. ground plane). So if the to-be-tracked object is not higher than the camera to the ground, the image of the object will always lie ‘below’ the vanishing line of the ground plane, and when it moves towards to the camera, the size (or scale) of the object on the image will get bigger as the y coordinate on image plane gets bigger, and vice versa.

Formally, in our work we use an ellipse to model the tracked object, and an 8-D state model is used to describe the dynamics of the ellipse, following ^[13]:

$$\{x, \tilde{x}, y, \tilde{y}, H_y, \widetilde{H}_y, H_x, \widetilde{H}_x\}$$

where (x,y) represent the center location of the ellipse, $\{\tilde{x}, \tilde{y}\}$ represent the motion velocity, $\{H_y, H_x\}$ are the lengths of the ellipse half axes, and $\{\widetilde{H}_y, \widetilde{H}_x\}$ are the corresponding rates of scale change on the axes. With the above idea, the scale change of a moving object is related to its position along the y-axis (i.e., the vertical axis in the image domain). The length of the y-axis of the ellipse will become bigger as the y location of the object gets bigger (the object is moving toward the camera), while the length of the y-axis will become smaller as the y location gets smaller (the object is moving further from the camera). This facilitate us to partition the original 8-D state space into two groups: the root variables R containing the motion information (including location and velocity), which will be sampled by a regular particle filter, and the leaf variables L consisting of the scale parameters (including lengths of the ellipse half axes and its scale change), which will be estimated by Kalman Filter in our work.

Just like regular particle filters, RBPFs represent posterior by a set of weighted samples: $S_t = \{s_t^i, w_t^i \mid 1 \leq i \leq N\}$. Each particle maintains not just a sample from $p(R_t | Z_t)$, which we denote by R_t^i , but also a parametric representation of the distribution $p(L_t | R_t^i, Z_t)$ which consists of the mean vector of the leaf state distribution, $\mu_t^i = E[L_t]$, and the estimation error covariance reflecting the variance of the leaf state distribution, $\sigma_t^i = E[(L_t^i - \mu_t^i)(L_t^i - \mu_t^i)^T]$ ^[11]. So each particle is represented by: $s_t^i = \langle R_t^i, \mu_t^i, \sigma_t^i \rangle$. The RBPF algorithm will sample the non-linear non-Gaussian motion (root variables) using particle filter, while apply Kalman filter to estimate the scale parameters (leaf variables) conditional on the motion state which have already been estimated. The proposed RBPF algorithm with adaptive system noise is illustrated in Fig.1. Our previously proposed RBPF tracking algorithm was completely reported in [3] which include all the steps in Fig.1 excluding the last step.

Input:: previous samples set and observations Z_t .

For $i = 1:N$ do

1. Propagate samples

a) **Sample object motion**: $R_t^{i-} \sim p(R_t | R_{t-1}^i, Z_t) = p(Z_t | R_t) p(R_t | R_{t-1}^i)$

b) **Kalman prediction**: $L_t^{i-} \sim p(L_t | R_t^i, R_{t-1}^i, L_{t-1}^i, Z_t)$

2. Evaluate weight: $w_t^i \propto p(Z_t^{CH} | R_t^i, L_t^i) \bullet p(Z_t^G | R_t^i, L_t^i)$

a) **Compute the color histogram.**

b) **Compute the gradient.**

c) **Compute the weigh.**

End for loop

3. Select samples: multiply /discard samples proportional to its weight.

For $i = 1:N$ do

4. Kalman update.

End for loop

5. Compute the mean state : $E[S_t] = \sum_{i=1}^N w_t^i s_t^i$

6. Compute a new system noise variance : $N_t = M_t N_0$
 $M_t = \max(\min(R_0 * \sqrt{1/\phi_t}, R_{\max}), R_{\min})$

Fig. 1. Rao-Blackwellised Particle Filter for tracking in surveillance

3. ADAPTIVE NOISE VARIANCE FOR SYSTEM MOTION IN RBPF

3.1 Why adaptive system noise is needed in RBPF?

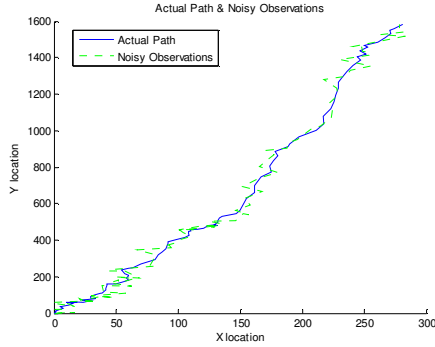


Fig. 2 Actual path with noisy observations

In our previously proposed RBPF algorithm, the root variables are estimated by propagating the state of prior step according to system motion model using Particle Filtering. If fixed noise variance in system motion is employed to track objects moving with dramatically and fast changing velocity, the range of sample set will not be sufficient to cover a large motion jump. This argument has been confirmed by the following simulation experiments. In these simulations, a point moving on a 2-D plane with a certain velocity generates an actual path; noise is then added to the actual path to simulate a noisy measurement of the actual path (see Fig. 2). After that the proposed RBPF is invoked to ‘track’ the actual path. The state of the moving point at any time is given by $s_t = \{x_t, u_t, y_t, v_t\}$, in which

$\{x_t, y_t\}$ corresponds to the position and $\{u_t, v_t\}$ represents velocity in x and y direction. When applying RBPF algorithm, the root, leaf and observations are established as:

$$R_t = \{u_t\} \quad L_t = \{x_t, y_t, v_t\} \quad Z_t = \{x_t, y_t\}$$

And v_t is assumed two times of u_t , this is to facilitate utilizing dependency relation between root and leaf in applying RBPF. Note that the adaptive system noise variance is essentially operated on the root variables which are estimated by particle filtering (i.e. Monte Carlo sampling method), therefore in a conventional particle filtering algorithm, the estimation accuracy may also be increased by making the system noise adaptive as like here for RBPF. Fig. 3 shows the estimation Mean Square Error (MSE) for u_t (top row) and location (bottom row) versus different noise variances under 3 motion settings: moving with small constant velocity (left column), moving with non-constant velocity within a small range (middle column), moving with dramatically changing velocity (right column).

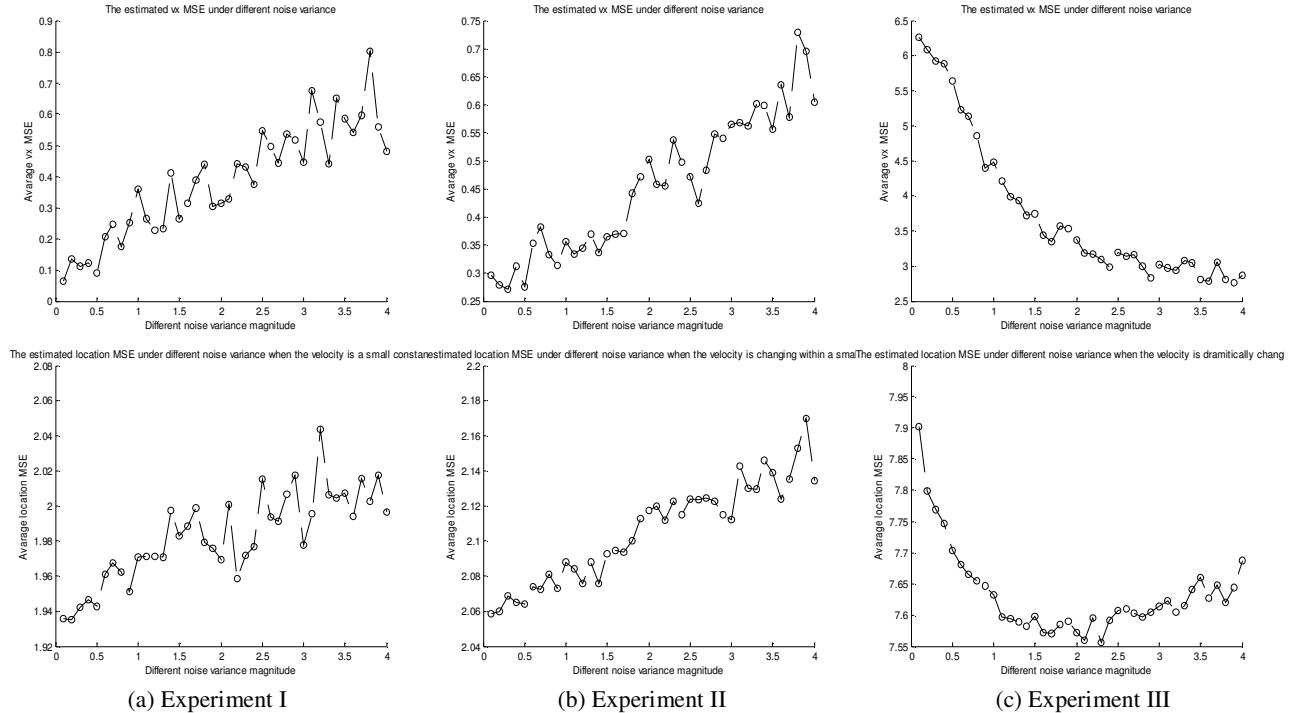


Fig. 3. The estimation error for velocity in x -axis (top row) and location (bottom row) under various noise variances. Left column: motion velocity is small constant; middle column: motion velocity varies in a small range. Right column: motion velocity changes a lot.

Experiment I: A path with constant velocity is generated, the initial velocity at time zero in the y -axis and x -axis are 2 and 1 respectively. The top figure in Fig. 3(a) depicts that, when noise variance is equal to 0.1, the estimation error for x -axis velocity reaches its lowest value (around 0.08). And for the bottom figure in Fig. 3(a), the estimated location MSE reaches its lowest value when the variance is 0.2. So we only need a very small noise variance to get a good estimation.

Experiment II: A path with non-constant velocity is generated; the x -axis velocity for the true path is uniformly distributed within a reasonably small range. The top figure in Fig. 3(b) illustrates that noise variance needs to be increased to 0.3 in order to reach the lowest estimation error for x -axis velocity (around 0.26). For bottom figure the variance where the lowest estimated location error is reached stays 0.2, the same with Experiment I.

Experiment III: A path with non-constant velocity is generated in which velocity in x -axis contains large jump. Fig. 3(c) illustrates that noise variance should be increased to 3.9 and 2.4 respectively where the lowest estimation error (around 2.7 and 7.55 respectively) is reached for both x -axis velocity and location.

3.2 Implement adaptive noise variance for system motion in RBPF

Experiment III in Section 3.1 clearly shows that adaptive noise is needed to tracked objects moving with dramatically and fast changing velocity. With adaptive system noise model, the range covered by the particles varies as the moving velocity changes. Considering this fact, the noise variance should be proportional to the prediction error (the prediction error determines the quality of tracking). If the prediction error is small, which indicates a good prediction, we only need noise with small variance to absorb the residual motion; if prediction error is large which implies a poor prediction, we then need noise with large variance to cover potentially large jumps in the motion state. To this end, the noise model has the form $N_t = M_t N_0$, where M_t is the magnitude of the noise variance, and M_t is a function of prediction error. In both the simulation and real data experiment, N_0 is a Gaussian distribution with mean zero and variance M_t .

In our experiment on real data, a *similarity* is computed which is inversely proportional to the prediction error: first at each time step, the mean state is obtained based on the combined weight for color histogram and gradient, then we compute the similarity between the *current mean state* and *the state of the previous time step* ($t-1$). Intuitively, if this similarity is large, it means that with the current noise variance the algorithm still maintains a good tracking, then we only need to adjust the variance a little bit; but if the similarity is small, this may well be due to the fact that the object is moving in a dramatically different velocity to such an extent that the tracker have obviated from the true object location, then we need to enlarge the noise variance. Therefore the noise variance should be inversely proportional to the *similarity*. Formally, suppose the similarity at time t is φ_t . At time t , the new noise variance is computed as:

$$M_t = \max(\min(R_0 * \sqrt{1/\varphi_t}, R_{\max}), R_{\min}) \quad (1)$$

where R_{\min} is the lower bound to maintain a reasonable sample coverage and R_{\max} is the upper bound to constrain the variance below a certain threshold if φ_t is very small. The algorithm needs to run several iterations before it finds the appropriate noise variance and stabilizes on the object. This is especially true when the velocities are dramatically different between two successive frames.

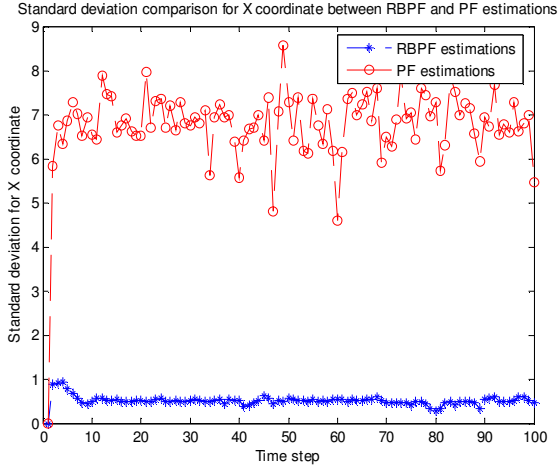
In the experiments on simulation data, the way we make the noise variance adaptive is slightly different with that used in the real video sequences. At each time step, a *similarity* between the *mean location* of the moving points at time t and the *observation* at time t is computed using Eq. (2), then we plug φ_t computed by Eq. (2) into Eq.(1) to get the new variance.

$$\varphi_t = \exp((-0.5) * (\mu_{t_x} - Z_t)^T * \Sigma^{-1} * (\mu_{t_x} - Z_t)) \quad (2)$$

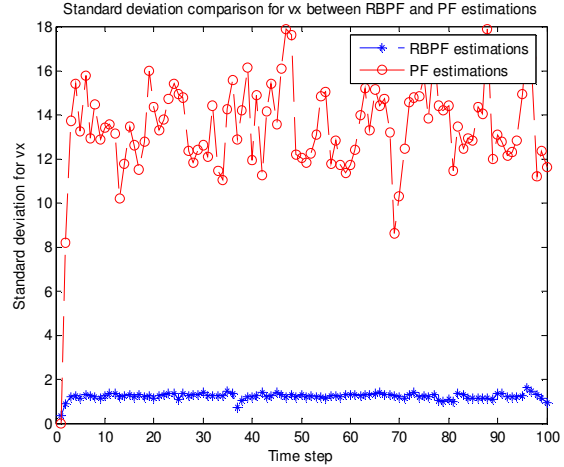
In Eq. (2), μ_{t_x} is the mean of $[x_t \ y_t]$ estimated by the Kalman Filter, Z_t is the observation at time t , and Σ is the estimation error covariance of $[x_t \ y_t]$. Essentially, Eq. (2) first computes the Mahanalobis distance from the ESTIMATED mean location μ_{t_x} to the OBSERVED location Z_t , the larger this Mahanalobis distance the smaller φ_t is.

4. EVALUATION OF PROPOSED RBPF ALGORITHM

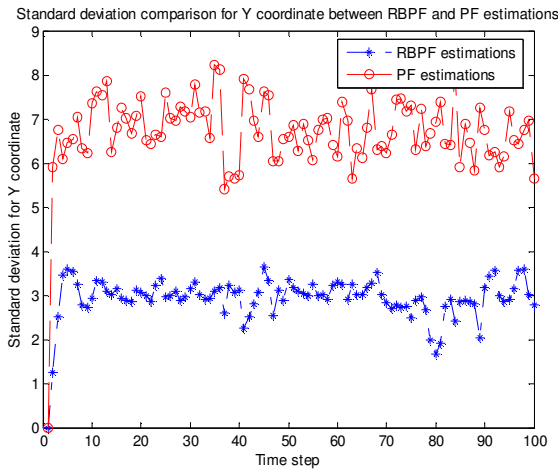
The evaluation of the proposed RBPF algorithm is mainly carried out using simulation data as it provides us quantitative analysis. Experiments show that the improvements of proposed RBPF over regular PF manifest in three aspects: increased estimation accuracy, reduced variance for estimates, reduced particle numbers are needed to achieve the same level of accuracy. In the following sub-sections, we are going to evaluate the improvements of proposed RBPF over conventional PF from two aspects: first variance for estimates, second particle numbers required to maintain good tracking (readers are suggested to refer to [15] for the evaluation of increased estimation accuracy). And in Section 4.3, the proposed RBPF with adaptive noise variance and RBPF with non-adaptive noise variance are compared.



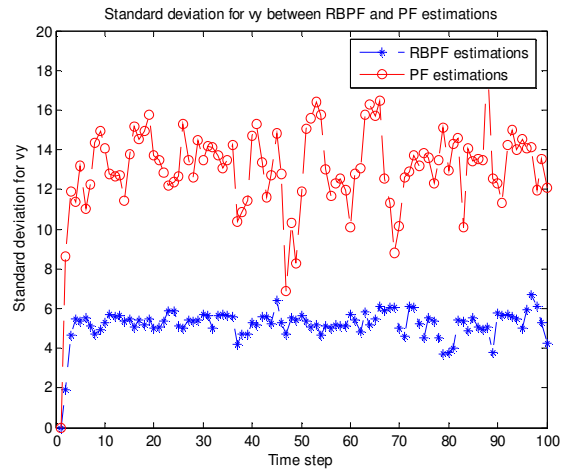
(a) Standard deviation for location in the X direction.



(b) Standard deviation for velocity in the X direction



(c) Standard deviation for location in the Y direction



(d) Standard deviation for velocity in the Y direction

Fig. 4. Variance comparison between RBPF and regular PF

4.1 Reduced estimates variance

One advantage of Rao-Blackwellised particle filter is that it can greatly reduce the variance of the state estimates. Intuitively this is because an exact filter is employed to estimate the linear dynamics. For a formal justification please refer to [14] and [15]. To evaluate whether our proposed RBPF algorithm has reduced the variance of state estimates, simulations with the same settings as Section 3.1 experiment III are conducted. Next, the proposed RBPF algorithm and conventional PF are invoked to ‘track’ the true path, 200 particles are used for both algorithms to approximate the posterior. After that, we compute the standard deviation (which is the square root of variance) of the 200 particles, each particle contains four components: x -coordinate, x -axis velocity, y -coordinate and y -axis velocity. As is shown in Fig. 4, the standard deviation of x -coordinate, x -axis velocity, y -coordinate and y -axis velocity computed by RBPF (blue curve in Fig.4 (a),(b),(c) and (d)) are around 0.7, 1.5, 3 and 5 respectively, while the standard deviation of corresponding components computed by conventional PF are around 7,14,7 and 12 respectively (red curve in Fig.4 (a),(b),(c) and (d)). So in terms of any of the four state components, it’s obvious that the variance of RBPF estimates is much lower than that of PF.

4.2 Reduced particle numbers

Another advantage of RBPF is that it can dramatically reduce the size of the space over which we need to sample. This in turn brings the benefit that far fewer particles are needed to reach the same level of accuracy as opposed to conventional PF. In Table 1, for particle filter running with 200 particles, the mean square error for location and velocity

are 54.0366 and 17.2754 respectively. As the number of particles increases (in the order of 200,300,400,500) in PF, the MSEs for both location and velocity are decreasing (location MSE decreases in the order of 54.0366,32.3036,25.4585,18.1270 and velocity MSE decreases in the order of 17.2754,15.0502,14.8343,13.0881) , however the MSEs can not be decreased any more by increasing the number of particles when it reaches 500. At that time the MSEs for both location and velocity are still larger than those obtained by RBPF with 200 particles (location MSE for RBPF with 200 particles is 8.5450, velocity MSE for RBPF with 200 particles is 8.4798). Thus we may conclude that our proposed RBPF only needs 200 particles to reach the same of level of estimation accuracy obtained by PF running with 500 particles.

Table 1. MSE for RBPF and PF running with different number of particles

	Mean Square Error for location	Mean Square Error for velocity
PF with 200 particles	54.0366	17.2754
PF with 300 particles	32.3036	15.0502
PF with 400 particles	25.4585	14.8343
PF with 500 particles	18.1270	13.0881
RBPF with 200 particles	8.5450	8.4798

4.3 Performance comparison between RBPFs with adaptive and non-adaptive noise variance

Adaptive RBPF means the algorithm in Fig. 1 with the last step, and non-adaptive RBPF is the algorithm without the last step of Fig. 1, i.e. the system noise variance is fixed during the tracking process. Our purpose of this analysis is to study whether the adaptive RBPF indeed performs better than the non-adaptive RBPF when the velocity of moving object between two successive frames is changing very fast. Simulation experiments are run 10 times to obtain a statistical comparison between adaptive and non-adaptive motion model. Results show that the estimation error of x -axis velocity is notably reduced (from 5.3536 for RBPF with non-adaptive motion to 1.7756 for RBPF with adaptive motion). Fig. 5 (a) and (b) depict the true x -axis velocity versus the x -axis velocity estimated by adaptive noise variance and non-adaptive noise variance respectively. At each time instance the true velocity in Fig. 5(a) and (b) is identical. In Fig. 5(a), the velocity estimated by RBPF with adaptive noise variance (blue curve) has roughly the same fluctuation tendency as that of the true velocity (black curve), while in (b) when the noise variance is fixed, the estimated velocity presents a flat changing tendency with a large gap from the true velocity. Our real data experiments also show that the adaptive RBPF, in face of dramatically changing velocity, performs better than that of non-adaptive RBPF (see Section 5). However, if the motion velocity of to-be-tracked object is almost constant or the acceleration is constant, then these two RBPF variants perform as well as each other.

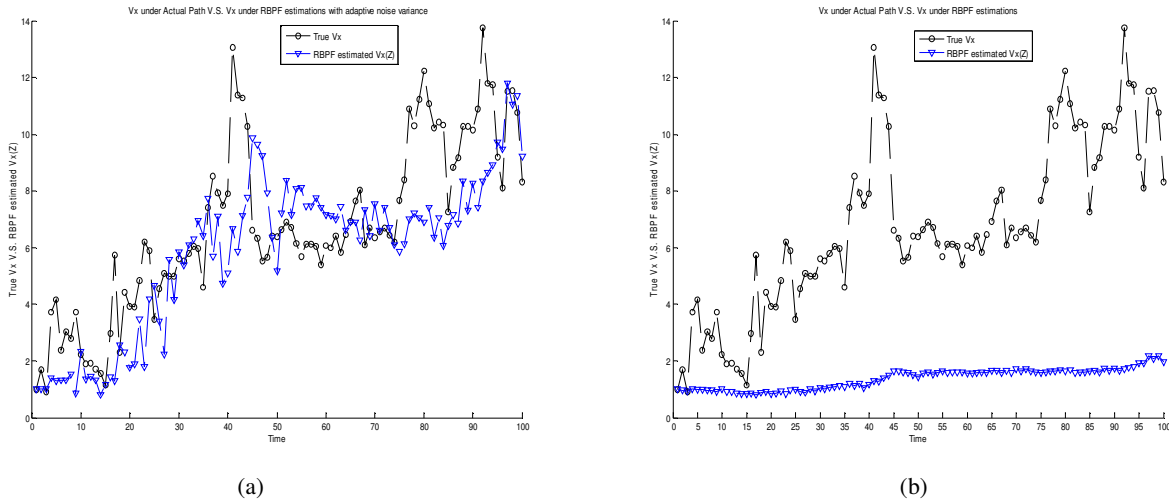


Fig.5. (a) True velocity in x -axis (black) versus velocity in x -axis estimated by RBPF with adaptive noise variance (blue). (b) True velocity in x -axis (black) versus velocity in x -axis estimated by RBPF with non-adaptive noise variance (blue).

5. RESULTS ON REAL VIDEO SEQUENCE

One surveillance scenario is outdoor human tracking. The test sequence for this scenario is from EC Funded CAVIAR project (<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>), which has also been used by PETS04 (Performance Evaluation for

tracking and Surveillance in conjunction with ECCV2004). In this sequence, the motion velocity of the person is changing rapidly, so we can use this sequence to compare the performance of adaptive RBPF with that of non-adaptive RBPF. Note that if the to-be-tracked object is moving with constant velocity, usually non-adaptive RBPF performs as good as adaptive RBPF. But when the object is moving with fast-changing velocity, then the tracking accuracy of adaptive RBPF is much higher than that of non-adaptive RBPF, and in cases when the non-adaptive RBPF tracker was lost due to the fast-changing movement, the adaptive RBPF tracker is still able to maintain good tracking. Some frames have been shown in Fig. 6. We can tell that after the person accelerates his motion at frame 430, the RBPF tracker with non-adaptive noise variance is not able to track the person accurately reflected by lagging behind the person (the PF tracker scale only on the legs and bottom half of the torso), in contrast the RBPF tracker with adaptive noise variance can still catch up the movement even after the person increases the motion velocity.

Another test scenario in surveillance is vehicle tracking. The sequence in Fig. 7 is used to illustrate the performance improvement of proposed RBPF over conventional PF. In Fig.7 the car is involved with complex motion such as turning (rotation), translation, and large scale change. It can be clearly seen that the RBPF tracker tracks the car all the way along its turning by scaling accurately on the car, in contrast the size of PF tracker gets very large and can not realistically reflect the scale change of the car, this leads to the result that the PF tracker deviate from the car eventually. One key point worth mentioning is that the PF tracker does not perform well especially when the object becomes too small as it moves away from the camera. On the other hand, the proposed method is able to keep tracking even in face of dramatic scale change.



Fig. 6. Comparison between RBPF with adaptive noise variance and RBPF with non-adaptive noise variance. Top row: Results from RBPF with adaptive noise variance; Bottom row: results from RBPF with non-adaptive noise variance. From left to right, the frame index is 396,430,467,487,558.



Fig. 7. Vehicle tracking. Top row: RBPF tracker. Bottom row: PF tracker.

6. CONCLUSION

In this paper, we present a RBPF tracking algorithm with adaptive noise variance for system motion, and we also discuss the evaluation of the proposed algorithm using simulation data. The algorithm evaluation and the experiments on real video sequence show that the proposed RBPF algorithm outperforms conventional particle filter in the following aspects:

1. The tracking (estimation) accuracy has been increased (which is being reported by the authors elsewhere).
2. Less number of particles is needed to achieve the same of level estimation accuracy as standard Particle Filter.
3. The variances of RBPF estimates are decreased compared to those of PF.

Additionally, as one of our future working direction, the tracker can be avoided from being distracted by the static background through comparing the on-line gradient with a pre-stored background gradient map.

REFERENCES

- [1]K. Murphy and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks", *sequential Monte Carlo methods in practice*, Springer-Verlag, New York, 2001, pp.500-512.
- [2]G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes", *Biometrika*, **83(1)**:81--94, 1996.
- [3]X. Xu, B.Li, "Rao-Blackwellised Particle Filter for Tracking with Application in Visual Surveillance", the *Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, in conjunction with ICCV 2005, Oct 15-21, Beijing.
- [4]A. Doucet, J. Freitas, and N. Gordon, *sequential Monte Carlo methods in practice*, Springer-Verlag, New York, 2001.
- [5]S.Arulampalam, S.Maskell, N.Gordon, and T. Clapp, "A tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking", *IEEE Transactions of Signal Processing*, 2002, 50(2):174—188.
- [6]N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *Proc. Inst. Elect. Eng. F*, vol. 140, no. 2, Apr. 1993, pp. 107--113.
- [7]M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density", In *Europe Conference On Computer Vision (ECCV)*, 1996, pp. 343–356.
- [8] S. Arulampalam, B. Ristic, "Comparison of the Particle Filter with Range Parameterized and Modified Polar ekf for Angle-Only Tracking", *Signal and Data Processing of Small Targets*, vol. 4048, 2000, pp. 288–299.
- [9]D. Schulz, D. Fox, and J. Hightower, "People Tracking with Anonymous and Id-sensors Using Rao-Blackwellised Particle Filters", in *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.
- [10]N. de Freitas, R. Dearden, F. Hutter, R. Morales-Menendez, J. Mutch and D. Poole, "Diagnosis by a Waiter and a Mars Explorer", in *Proceedings of the IEEE Special Issue on Sequential State Estimation*, 2003.
- [11] C. Kwok, D. Fox, "Map-Based Multiple Model Tracking of a Moving Object", *RobuCup 2004*, pp18-33.
- [12]T. Soma, K. Yosui and T. Mutsumoto, "Reconstructions and predictions of nonlinear dynamical systems by Rao-Blackwellised sequential Monte Carlo", *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 Proceedings (ICASSP '03)*.
- [13]K.Nummiaro, E. Koller-Meier, and L.V.Gool, "Object Tracking with an Adaptive Color-Based Particle Filter", *Proc. Symposium for Pattern Recognition of the DAGM*, pp.591-599, 2003.
- [14] N. Mukhopadhyay, *Probability and Statistical Inference*, Marcel Dekker, Inc., New York, 2000.
- [15]A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, Dec. 2000.