Implementing Manifold Learning in Adaptive MCMC for Tracking Vehicle under Disturbances

Wei Yeang Kow, Yit Kwong Chin, Wei Leong Khong, Hui Keng Lau, Kenneth Tze Kin Teo
Modelling, Simulation & Computing Laboratory, Material & Mineral Research Unit
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Malaysia
msclab@ums.edu.my, ktkteo@ieee.org

***Abstract*—In recent years, tracking vehicle with overlapping and maneuvering disturbances has become a challenging task in visual tracking. Markov Chain Monte Carlo (MCMC) is proved to be effective in tracking vehicle under disturbances by probabilistically estimating the vehicle position. However the sampling based tracking algorithm is highly depending on the sampling efficiencies where adequate chain length is necessary to sustain the tracking accuracy. Therefore variance ratio (VR) based MCMC has been implemented in this study to adapt the chain length according to the disturbances encountered. Isomap manifold learning is further implemented to update the vehicle model and accurately track the vehicle with maneuvering disturbances. Multiple vehicle models with different viewing angles are represented by Isomap under low dimensional manifold. The suitable vehicle model will be selected according to the estimated vehicle position. Experimental results have shown that Isomap-VR-MCMC have better tracking performances compared to VR-MCMC with smaller RMSE value.**

***Keywords*—Markov Chain Monte Carlo (MCMC), Isometric Feature Mapping (Isomap), variance ratio (VR).**

I. Introduction

The strong growth rate of the vehicle usage in urban area over the recent years has emerged traffic surveillance to an important feature in traffic flow monitoring. Various sensors has been developed for traffic surveillance purposes and among all, video sensors has shown great capability in obtain wide range of vehicle information. The sensors is able of obtaining various types of vehicle parameters such as vehicle speed, vehicle count and determine the traffic irregularities such as vehicle accidents and slow traffic flows [1]. Since the video sensors track the vehicle based on the observable vehicle outlook, it faces limitations when the target vehicle is undergoing overlapping and maneuvering disturbances. The vehicle observable information will be lost during the overlapping situation and maneuvering vehicle will gives varying vehicle outlook which increases the tracking difficulties [2].

Various researches have been carried out to tracked vehicle undergoing overlapping disturbances. MCMC has been widely implemented to track vehicle undergoes overlapping disturbances as it is capable of estimating the vehicle position by probabilistic sampling [3]. The sampling based tracking algorithm has been implemented to track overlapping vehicle during the vehicle is on side and front view particularly during heavy traffic jams occurred [4, 5]. The implementation result has shown great detection and tracking accuracy using pre-determined vehicle model. However the tracking algorithm will have tracking limitations if the vehicle is undergoing maneuvering disturbances where the vehicle outlook is varying. The implemented fixed vehicle model has only single view angle which only enable the algorithm to track target vehicle at limited angle. Thus manifold learning has been implemented to update the vehicle model according to the estimated vehicle position to assist the tracking algorithm in handling the maneuvering disturbances.

Manifold learning has been implemented in visual tracking to enable the tracking to be performed on objects with deformable outlook where the appearances of the object are varying as time changes. Research [6] has proposed a new concept of identity and view manifold to recognize maneuvering vehicle. The developed method is capable of interpolate the appearance of an unknown target based on the training sets learned by capturing inter-class and intra-class of the target vehicle appearances. Hence it is able of automatically update the vehicle model by interpolate the latest vehicle shape with the manifold representation which is suitable to be implemented for maneuvering vehicle tracking.

The manifold learning representation is further implemented as efficient model update to track target vehicle in real time [7]. The implementation model the variation of vehicle appearances by representing multiple vehicle feature vector under the low dimension manifold. Suitable vehicle model will be selected according to the computed vehicle feature vector which enabled the target vehicle experiencing pose variation and overlapping disturbances to be tracked accurately. On the other hand, Isomap manifold representation has been implemented to learn the pose variation of target object that experiencing random movement. The algorithm obtains the manifold representation by applying multidimensional scaling computations on the geodesic distances of the vehicle parameters [8]. The implementations has enable better manifold representations on the training data thus yielding better tracking accuracy compared to the conventional principal component analysis (PCA) training algorithm. Hence the Isomap is suitable to be implemented for the tracking of maneuvering vehicle where multiple vehicle models are needed to be trained and represented effectively.

In this paper, MCMC will be implemented to track the target vehicle undergoes both overlapping and maneuvering disturbances. The Isomap manifold representation will be implemented to learn vehicle model with multiple view angle to aid MCMC in tracking maneuvering vehicle. Since the performance of MCMC is depending on the sampling efficiency, VR convergence diagnostic algorithm is implemented to adapt the chain length of the MCMC according to the disturbances encountered. The performance of Isomap-VR-MCMC tracking algorithm will be compared with the VR-MCMC without manifold representations. The proposed tracking algorithm is expected to have better tracking accuracy with lower RMSE value.

II. MCMC TRACKING ALGORITHM

MCMC is widely implemented to solve high dimensional distribution based on probabilistic sampling. To build a single sequence of Markov Chain, vehicle position samples θ will be proposed based on the proposal distribution as shown in (1). The position samples consist of the target vehicle centroid coordinate $\{x, y\}$ which is the x-axis and y-axis pixel coordinate of the image frames.

$$Q(\theta^* \mid \theta_t^{i-1}) = \frac{1}{2\pi\sigma_q^{\ 2}} e^{-\frac{(\theta^* - \theta_t^{i-1})^2}{2\sigma_q^{\ 2}}} \tag{1}$$

The proposal distribution is a Gaussian distribution which prevents the algorithm from proposing position samples that are out of the target range. θ^* is the new proposed position sample and θ_t^{i-1} is the previous accepted sample. Therefore it is notable that the proposed position will be within the range of variance σ_q using the previous accepted position sample as the mean value. This also indicates that each new position sample is estimated according to the latest accepted sample and overlooks the samples accepted earlier. Prior probability will then be implemented to compute the probability of acceptance of the current estimated position sample to the vehicle position computed at the previous vehicle frames. The prior probability computation is shown in (2).

$$p(\theta^*) = \frac{1}{2\pi\sigma_p^{\ 2}} e^{-\frac{(\theta^* - \theta_{t-1})^2}{2\sigma_p^{\ 2}}} \tag{2}$$

Similar to the proposal distribution, the prior probability is a Gaussian distribution as well. The balance characteristic of the normal distribution has enabled the algorithm to compute the acceptance probability of vehicle position that may move in both forward and backward directions. From (2), θ_{t-1} is the vehicle position at the previous tracking frame. Hence the range of acceptance of the proposed vehicle position will be within the range of variance σ_p with θ_{t-1} as the mean of distributions. To ensure the proposed position sample is near to the target vehicle position, observation likelihood is computed as shown in (3), (4) and (5).

$$\pi(E \mid \theta) = \frac{1}{2\pi\sigma_d^{\ 2}} e^{-\frac{d}{2\sigma_d^{\ 2}}} \tag{3}$$

$$\pi(C \mid \theta) = \frac{1}{2\pi\sigma_d^{\ 2}} e^{-\frac{B}{2\sigma_b^{\ 2}}} \tag{4}$$

$$\pi(\theta) = \beta[\pi(C \mid \theta)] \cdot \gamma[\pi(E \mid \theta)] \tag{5}$$

The observation likelihood in (5) is the fusion of the edge distance likelihood in (3) and color likelihood in (4). The edge distance likelihood is computed using the edge distance transform. The distance transform is defined on a binary image where the distance of each pixel to the nearest edge pixel is computed. Hence the edge pixel will have the distance value of zero and the distance values will gradually increases as the neighboring pixel is distance away from the vehicle edge. When the proposed position sample is closed to the target vehicle, the edge distance d will be smaller and hence giving larger edge distance likelihood $\pi(E \mid \theta)$. Inversely, if the proposed position is not near to the target vehicle, the edge distance will be large and hence reduces the likelihood value.

For color likelihood computation, HSV color space has been selected due to its capability in manipulating the color illuminations [9]. The color likelihood is determined by computing the similarity of the color histogram of vehicle model to the color histogram of the target vehicle appearance at the proposed vehicle position. Similarity between the color histogram is determined by using the Bhattacharyya distance B where similar histogram will give small distance value. Smaller distance value will give larger color likelihood value of $\pi(C \mid \theta)$ whereas larger distance value indicates that the proposed position is unlikely to the vehicle model.

The weight constant β and γ in (5) is implemented to calibrate the computation priority of the observation likelihood. Setting β with larger value will enable the MCMC to track the target vehicle more prior to the color changes and larger γ weight will ensure that the target vehicle is tracked with priority on the edge likelihood. The proposed position sample is then computed by the Metropolis-Hasting algorithm in (6) to determine the sample acceptance rate into the MCMC.

$$\alpha = \min\left(1, \frac{P(\theta^*)Q(\theta^{i-1} \mid \theta^*)\pi(\theta^*)}{P(\theta^{i-1})Q(\theta^* \mid \theta^{i-1})\pi(\theta^{i-1})}\right) \qquad (6)$$

The Metropolis-Hasting algorithm determines the acceptance rate of the proposed position sample based on the prior probability, proposal distribution and the observation likelihood. If the proposed position sample has achieved the acceptance ratio, the sample will be accepted in to the MCMC. Inversely if the sample does not reach to the ratio, the proposed sample will be rejected and the previous accepted position sample will be replicated as the new accepted sample into the MCMC. Samples that has been accepted will form a MCMC samples set of $\{\theta^1, \theta^2, \theta^3, ..., \theta^n\}$. Hence MCMC with longer chain length will have larger number of samples and short chain length MCMC will have lesser samples in the chain. The final vehicle position is the computed by using the Monte Carlo integration in (7).

$$E(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta^i\right) \qquad (7)$$

The performance of MCMC is highly dependent on the sampling efficiency. Chain length that are too small may cause the sample sets are premature for evaluation whereas chain length that are too long may lead to high computational cost which is not desirable. Therefore VR convergence diagnostic algorithm is implemented to determine the steady state of the MCMC samples where suitable chain length will be computed to accurately track the target vehicle.

III. VR-MCMC Tracking Algorithm

VR-MCMC tracking algorithm is developed where VR convergence diagnostic algorithm will be implemented in to MCMC to adapt the chain length according to the disturbances occurred. Since the VR determine the convergence rate based on multiple chain of MCMC, two sequences of MCMC will be implemented in for the convergence diagnostic process. Both MCMC sequences will be initialized at different starting point to enable better exploration rate. The MCMC is determined as converged by computing the ratio between the variance within a MCMC sequence and variance between the MCMC sequences [10]. The variances computations are defined in (8) and (9) as shown below.

$$\frac{B}{n} = \frac{1}{m-1}\sum_{j=1}^{m}\left(\bar{\theta}_j - \bar{\theta}_{.}\right)^2 \qquad (8)$$

$$W = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n-1}\sum_{i=1}^{n}\left(\theta_j^i - \bar{\theta}_j\right)^2\right) \qquad (9)$$

The computation in (8) is to determine the variance between the MCMC sequences and (9) is to compute the variance between MCMC sequences. The variable $\bar{\theta}_j$ is the mean value of a single sequence MCMC and the mean value between the MCMC sequences is defined as $\bar{\theta}_{.}$ Variable n is the number of accepted samples in the MCMC sequence and m is amount of MCMC sequences that has been executed. The MCMC is diagnosed as converged when the estimator R is approximate to 1 as shown in (10).

$$R = D\left(\left(\frac{n-1}{n}\right) + \left(1 + \frac{1}{m}\right)\frac{B}{nW}\right) \qquad (10)$$

The estimator in (10) is the ratio value between both the variances computed in (8) and (9). Variable D is a constant that calibrates the inherent approximation for the variance ratio. Thus, if the proposed position samples are accurate and continuously accepted into the MCMC, the MCMC samples will be clustered together and near to each other. Consequently, the variance within and between MCMC sequences will be small and the MCMC are determine as strongly mixed and converging to the desired distribution. However if the proposed sample positions are inaccurate, the MCMC samples will experience slow mixing rate with large variances between the samples. As a result, more samples are needed to be generated to achieve the convergence and hence longer chain will be acquired for vehicle position evaluation. The developed VR-MCMC tracking algorithm is illustrated in TABLE I below.

TABLE I. VR-MCMC Tracking Algorithm.

VR-MCMC Tracking Algorithm
for frame t = 1 to end
Initialize two sample θ^1 at time t for two sequence of MCMC
Loop 1
Loop 2 (for two MCMC)
Proposed two new states θ^* with proposal distribution
Compute prior probability
Compute observation likelihood.
Compute Metropolis-Hasitng acceptance ratio, α
Generate random value $U = rand(0,1)$
if $U \le \alpha$, add $\theta_t^i = \theta^*$
else add $\theta_t^i = \theta_t^{i-1}$.
end if
end Loop 2
Compute B/n, W and R
if $R \approx 1$, go to end Loop 1
else go to Loop 2
end if
end Loop 1
Compute vehicle position with Mont Carlo integration
end for

IV. Isometric Feature Mapping

Isomap represented the higher dimensional data by using multidimensional scaling (MDS) where the geodesic distance space of the vehicle parameters will be computed to obtain the manifold representations. Geodesic distance is the shortest path on the curve surface of the manifold data. It is determined based on the approximation of short steps between the neighboring data points where the number of neighboring data is selected depends on the type of data that need to be represented. Small neighbor number will correlate less data point together while large neighbor value is capable of correlate more data point and provides a more complicated manifold representation.

To build to Isomap vehicle model representation, *K* number of nearest neighbors will be set to connect the data point according to the computed pair wise distances. The implemented intrinsic data point is the possible coordinate position of the vehicle model. The geodesic pair wise distances of all pairs of data point is obtained by determining the Euclidean distance d_x of all pairs of data points. Shortest distance path d_g between the neighboring data is then determined by preserve the computed Euclidean distance and the remaining distances is filled up by using Floyd's algorithm as defined in (11).

$$d_g(i,j) = \min\{d_g(i,j), d_g(i,k) + d_g(k,j)\} \tag{11}$$

The computed shortest distance path in (11) will form a matrix of shortest path D_G that represents the relations of all pairs of neighboring data points. The shortest path matrix will then be implemented into the MDS algorithm to obtain the represented data in the low dimension manifold. The similarity matrix s is first computed as shown in (12) and then inner product matrix and centering matrix is then determined as defined in (13) and (14).

$$s(i,j) = D^2_{G(i,j)} \tag{12}$$

$$b = -\frac{1}{2} hsh \tag{13}$$

$$h = I - \frac{1}{N} h \tag{14}$$

From (12), it can be seen hat the similarity matrix is the squared value of the distance matrix. The inner product matrix in (13) is determined based on the centering matrix in (14) where I is the identity matrix and N is the length of the matrix. Eigenvalue and eigenvector of the obtained inner product matrix is then computed and the obtained eigenvalue will be in the form of diagonal matrix of $\lambda = diag(\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n)$. The final dimensional representation is obtained as shown in (15).

$$M = \sqrt{\lambda_p} \cdot v_p^i \tag{15}$$

The variable v_p^i in (15) is the eigenvectors where i is the i-th eigenvector component to the p-th eigenvalue. The eigenvalue with the highest dimensional value will be selected to recover the new low dimensional representation. In this implementation, the recovery dimension is two as the vehicle motion is changing in two dimensions. Hence eigenvalue with the largest two values will be computed to acquire the manifold representations. Thus the high dimensional vehicle model coordinate point with *K* nearest neighbours has been preserved by forming new coordinate manifold representations M .

V. Isomap Vehicle Model

Multiple vehicle models are represented under manifold dimension by using the Isomap algorithm. The implemented vehicle models are at left turning maneuver where the vehicle model will be changing from side view to rear view. An overall of twenty three vehicle models have been selected as the training data and part of the vehicle model are illustrated in Fig. 1 below.

The number of nearest neighbor is set to three which enable three neighboring vehicle models with the closest coordinate to be correlated to each other. The Euclidean distance matrix of the coordinate of each vehicle model is computed and implemented in to the Isomap computations. The residual variance is then computed to determine the dimension that has been recovered by the Isomap. It is determine by calculating the correlation coefficient of the similarity matrix as shown in (16) and the computed values are plotted in Fig. 2.

$$R_d = 1 - C_d \tag{16}$$

In Fig. 2, it is observable that the residual variance value stops decreasing at the value of 2 which shows that two dimension of coordinate manifold has been recovered. This indicated that the computed manifold representation has been accurate since the vehicle models are changing in two dimensions. The new coordinate representation of the vehicle model is plotted in Fig. 3.

Fig. 1. Maneuvering vehicle models

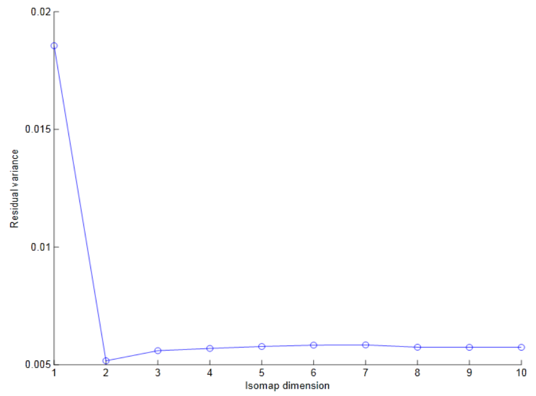

Fig. 2. Residual variance of Isomap

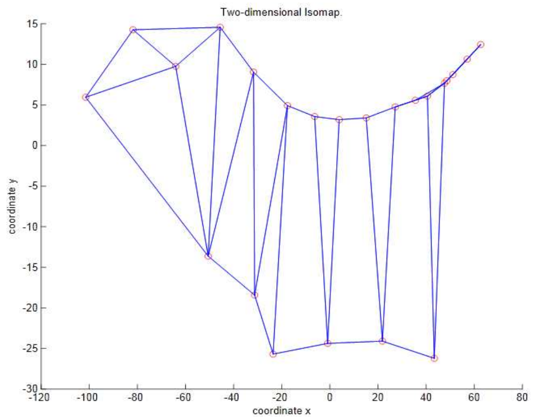

Fig. 3. Isomap representation of the vehicle model

In Fig. 3, it is observable that the plotted data points are the newly represented vehicle model coordinates. Each data points are correlate to the three nearest neighboring points since the nearest neighbor value is set to three. The plotted points with negative y-coordinates represents the vehicle model that moving in straight directions and the points with positive y-coordinate represents the vehicle model in maneuvering situation. The distance of the proposed position sample to each coordinate point will be calculated and the coordinate with the smallest distance value will be determined. Vehicle model that corresponds to the coordinate will be selected as the reference model to compute the observation likelihood of the MCMC. Hence the vehicle model will be updated according to the estimated position samples which enhance the capability of the MCMC to track the vehicle under disturbances more accurately.

VI. Results and Discussions

The Isomap vehicle model is preprocessed and the obtained manifold representation is implemented into the VR-MCMC tracking algorithm to update the vehicle model according to the proposed position sample. The developed Isomap-VR-MCMC tracking algorithm is implemented to track a target vehicle that undergoes both overlapping and maneuvering disturbances at the same time. VR-MCMC tracking algorithm without using Isomap vehicle model is also implemented to track the vehicle under the same disturbances where only one vehicle model with side view is used for the observation likelihood computations. The performance of the Isomap-VR-MCMC is shown in Fig. 4.

It is observable in Fig. 4 that the Isomap-VR-MCMC is capable to keep track on the target vehicle especially when the vehicle is undergoes maneuvering situations. At frame 4, the target vehicle is clear from disturbances and is tracked accurately. The target vehicle is overlapped by another vehicle on the front at frame 7 and frame 8. It can be seen that in frame 8, the target vehicle is undergoing slight maneuvering motion turning into the side lane. This has cause to the changes of vehicle appearance which will affect the accuracy of the observation likelihood computations in MCMC. However the target vehicle is still keep tracked accurately. This is due to the vehicle model has been updated from the represented manifold according to the proposed position which provides more accurate computations on the observation likelihood.

The target vehicle is finished overlapped at frame 11 but another approaching vehicle begins to overlap in front at frame 12, 13 and 14 for the second time. The disturbance in these frames are more serious compared to the previous overlapping disturbance as the vehicle has change from side view to rear view. The changes of the vehicle outlook have increases the tracking difficulties and cause the algorithm to experience slight tracking error which can be seen in frame 12 and 13. However the developed algorithm is still capable to keep track on the target vehicle with better RMSE value compared to the VR-MCMC which did not update the vehicle model accordingly. The comparison of RMSE between Isomap-VR-MCMC and VR-MCMC is shown in Fig. 6 and the adapted chain length is shown in Fig. 7 respectively.

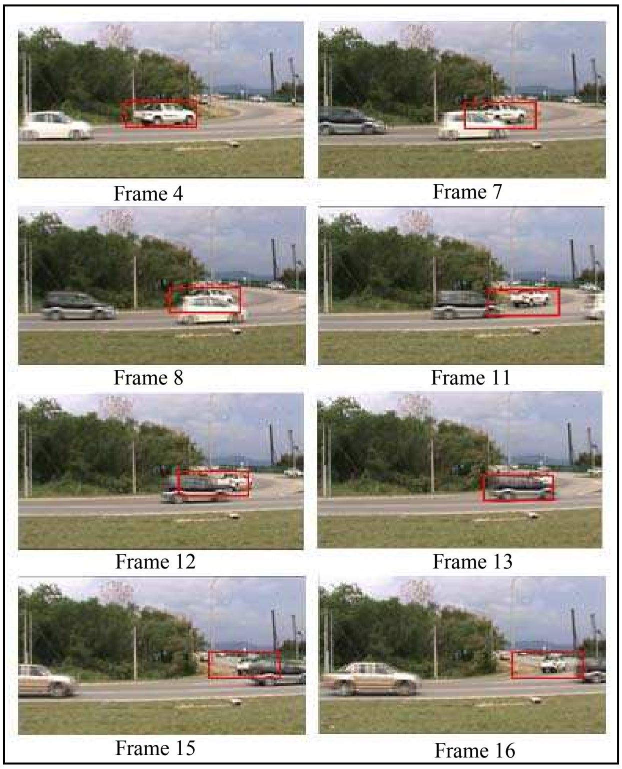

Fig. 4. Tracking performances of Isomap-VR-MCMC

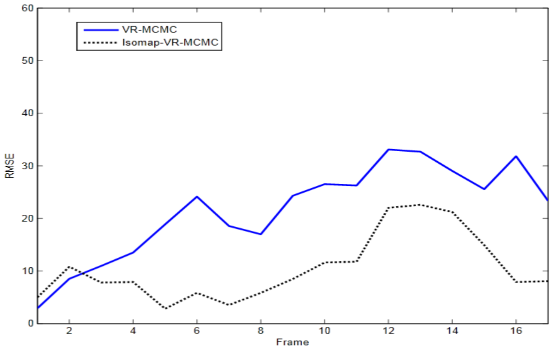

Fig. 5. RMSE of VR-MCMC and Isomap-VR-MCMC

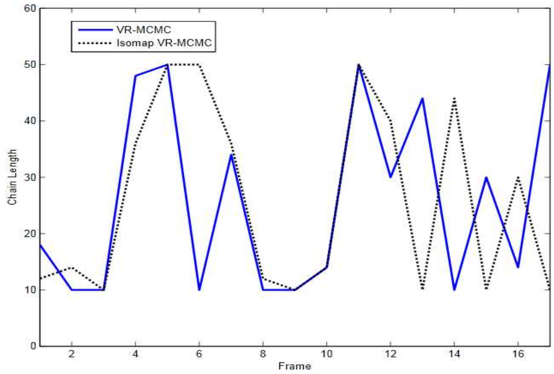

Fig. 6. Chain length of VR-MCMC and Isomap-VR-MCMC

In Fig. 5, it can be seen that the Isomap-VR-MCMC has lower RMSE compared to the VR-MCMC tracking algorithm. The RMSE value that is over 25 indicates that the target vehicle is lost tracked. Thus it is observable that the VR-MCMC has lost track on the target vehicle from frame 10 to 17 where the target vehicle is undergoes heavy outlook changes. This is due to the implemented single vehicle model with side view is not suitable to be used to compare to the target vehicle that has changed to rear view during the maneuvering process. However the VR-MCMC still capable to keep track on the target vehicle between frame 1 to 9 where the vehicle is still moving in straight line and the single vehicle model is suffice for the observation likelihood computation.

For Isomap-VR-MCMC, the algorithm has successfully tracked the target vehicle with low RMSE value between frame 1 to 9. The RMSE value has gradually increased between frame 10 to 15 where the target has been fully overlapped which loses the observable vehicle information. This has reduces the accuracy of the observation likelihood computation and hence affected the accuracy of MCMC samples. However, the RMSE values are still within the range of 25 which shows that the algorithm still capable to keep track on the target vehicle. When the target vehicle has finished overlapped in frame 15 and 16, the position sample at the nearby region is proposed and the vehicle model with rear view are selected from the Isomap manifold which enable the algorithm to track the target vehicle accurately. Since both the algorithms have track the target vehicle with adaptable chain length, Fig. 6 has shown that both algorithms have generated similar chain length to track the target vehicle. Longer chain length has been generated during the disturbance occurred and smaller chain length is generated when the samples are accurate and clear from disturbances. Thus, with the similar chain length, Isomap-VR-MCMC has shown better tracking performances compared to the VR-MCMC with lower RMSE value.

VII. Conclusions

The developed Isomap-VR-MCMC has been implemented to track the vehicle undergoes both overlapping and maneuvering disturbances. The implementation results have shown that Isomap-VR-MCMC has accurately tracked maneuvering vehicle that changes from side to rear view. Suitable vehicle model has been selected according to the proposed position sample which enables more accurate observation likelihood computations. Thus, the implementation of Isomap vehicle model into VR-MCMC has successfully improved the tracking robustness by accurately track maneuvering vehicle with lower RMSE value.

Acknowledgement

The authors would like to acknowledge the financial assistance from Ministry of Higher Education of Malaysia (MoHE) under Exploratory Research Grant Scheme (ERGS) No. ERGS0021-TK-1/2012, Universiti Malaysia Sabah (UMS) under UMS Research Grant Scheme (SGPUMS) No. SBK0026-TK-1/2012, and the University Postgraduate Research Scholarship Scheme (PGD) by Ministry of Science, Technology and Innovation of Malaysia (MOSTI).

References

[1] A. Fernandex-Caballero, F. J. Gomez, and J. Lopez-Lopez, “Road- Traffic Monitoring by Knowledge-Driven Static and Dynamic Image Analysis,” Expert Systems with Applications, vol. 35, Issue 2, 2008, pp. 701-709.

[2] A. Dore, A. Beoldo, and C. S. Regazzoni, “Multiple Cue Adaptive Tracking of Deformable Objects with Particle Filter,” Image Processing, 2008, pp. 237-240.

[3] C. Andrieu, N. Freitas, A. Doucet, and M. T. Jordan, “An Introduction to MCMC for Machine Learning,” Machine Learning, vol. 50, Issue 1, 2003, pp. 5-43.

[4] W. Y. Kow, W. L. Khong, H. T. Yew, I. Saad, and K. T. K. Teo, “Adapting Markov Chain Monte Carlo with CUSUM Path Plot for Overlapped Vehicle Tracking,” International Journal of Simulation, Systems, Science & Technology, 2011, vol. 12, no. 3, pp. 36-43.

[5] Y. Jia and C. Zhang, “Front-view Vehicle Detection by Markov Chain Monte Carlo Method,” Pattern Recognition, vol. 42, 2009, pp. 313-321.

[6] V. Venkataraman, G. Fan, L. Yu, X. Zhang, W. Liu, and J. P. Havlicek, “Joint Target Tracking and Recognition using View and Identity Manifold,” Computer Vision and Pattern Recognition, 2011, pp. 33-40.

[7] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai, “Real-Time Probabilistic Covariance Tracking with Efficient Model Update,” Image Processing, vol. 21, Issue 5, 2012, pp. 2824-2837.

[8] N. Rane and S. Birchfield, “Isomap Tracking with Particle Filtering,” Image Processing, vol. 2, 2007, pp. 513-516.

[9] W. L. Khong, W. Y. Kow, Y. K. Chin, I. Saad, and K. T. K. Teo, “Overlapping Vehicle Tracking via Adaptive Particle Filter with Multiple Cues,” Control System, Computing and Engineering, 2011, pp. 460-465.

[10] W. Y. Kow, W. L. Khong, Y. K. Chin, I. Saad, and K. T. K. Teo, “CUSUM-Variance Ratio Based Markov Chain Monte Carlo Algorithm in Overlapped Vehicle Tracking,” Computer Applications & Industrial Electronics, 2011, pp. 50-55.