Adapting MCMC with CUSUM Path Plot for Overlapped Vehicle Tracking

Wei Yeang Kow, Wei Leong Khong, Hoe Tung Yew,
Ismail Saad, Kenneth Tze Kin Teo
Modelling, Simulation and Computing Laboratory,
School of Engineering and Information Technology,
Universiti Malaysia Sabah, Kota Kinabalu, Malaysia.
msclab@ums.edu.my, ktkteo@ieee.org

Abstract **- Traffic surveillance using video sensors has been essential over the recent years and it is capable of obtaining wide range of vehicle information. However tracking overlapped vehicles still remain as a challenging task due to the involvement of high dimensional calculation. Conventional fixed sample size Markov Chain Monte Carlo (MCMC) will encounter tracking error if the sample size is insufficient and will be computationally expensive if the sample size is too large. Therefore cumulative sum (CUSUM) path plot is introduced to aid the difficulties in determining the sample size of MCMC. The adaptive sample size of MCMC has shown significant tracking accuracy especially when the vehicle is overlapped. Furthermore, implementation of observation likelihood by fusing colour and edge distance has further enhanced the tracking performances. Experimental result shows that CUSUM path plot algorithm has overcome the limitation of fixed sample size MCMC with better tracking accuracy and lesser computational time.**

Keywords - Markov Chain Monte Carlo, cumulative sum, vehicle tracking, adaptive sampling

I. INTRODUCTION

The capability of video sensor in obtaining wide range of vehicle information has become essential nowadays and it is widely implemented on vehicle tracking for traffic flow control purposes [1]. This sensor has the capability of obtaining information such as vehicle's motion, size, colour and different types of observable parameters which are applicable for vehicle tracking purposes. As a result, close-circuit television system (CCTV) has been implemented to capture the traffic flow and the traffic behaviour is monitored via the traffic light management system. Various strategies have been carried out in vehicle tracking system to determine the traffic trajectories for applications on sophisticated traffic surveillance [2]. It uses modelled, feature, region and active contour based strategies to track vehicles and has shown promising tracking results. However, tracking the overlapping vehicle still remains as a challenging issue since it requires high dimensional calculation and it is difficult to be solved. MCMC is one of the widely implemented algorithms in object tracking. The sampling efficiency of MCMC is critical in predicting the vehicle position since appropriate sample size is difficult to be determined.

This paper outline consists of seven sections. Section 2 is the literature review of various MCMC approaches in vehicle tracking. Section 3 describes the methodology of the MCMC tracking algorithm. Section 4 discusses the computation of CUSUM path plot in diagnosing the convergence of MCMC and Section 5 is the implementation of CUSUM path plot in MCMC to track the overlapped vehicle. Results and discussions are made at Section 6 and the last section presents the conclusion of this paper.

II. LITERATURE REVIEW

Various studies have been performed on tracking vehicle with MCMC particle filtering algorithm and the implementation results show it is capable of tracking the vehicle in different occluded conditions. However, the performances consist tracking error due to its inappropriate foreground-background segmentation method [3]. Implementation in [4] has shown that detection of overlapped vehicle using shape and width of predefined vehicle model is significant and accurate. Nevertheless, the algorithm will encounter confusion if the vehicle is occluded with another vehicle that has the same size and contour feature. Data Association MCMC is well known on its capability in tracking multiple objects using six moves which are birth, split, death, merge, reduce and extend movement [5, 6]. The tracking performance is accurate but the algorithm is time consuming and will have failure when the vehicle is overlapped for a long period. The research in [7] has developed MCMC tracking algorithm that can track overlapped vehicle from front view without the background segmentation and shadow elimination method. It uses simulated annealing to adaptively sample the MCMC and has successfully tracked the overlapped vehicle. However the developed algorithm can only perform tracking on vehicle that view from the top front which their background is the non complicated traffic lane. It will encounter tracking error if the vehicle is segmented at a complicated background especially from the side view where there are trees and other environmental obstacle that affect the tracking accuracy.

Other than those mentioned approaches, most of the MCMC tracking algorithm are based on fixed sample size which are inefficient since MCMC tracking performance are highly dependent on its sampling efficiency [5, 6, 8]. Small sample size will leads to tracking error due to insufficient information whereas large sample size provides better tracking accuracy with compensation of higher computational time. Therefore appropriate sample size are require to track overlapped vehicle efficiently where higher sample size will be developed when the vehicle is overlapped while smaller sample size will be used if the vehicle are not overlapped and easier to be tracked.

In this paper, CUSUM path plot is proposed to adaptively sample the MCMC for tracking overlapping vehicle. The algorithm is capable of quantitatively determine the convergence rate of MCMC by computing the hairiness of the plot [9, 10]. Furthermore, observation likelihood combining both colour and edge distance enable the algorithm to detect and track overlapped vehicle that has high similarity on the shape and outlook. Fixed sample size MCMC with CUSUM path plot based adaptive MCMC are implemented to track overlapped vehicle and their corresponding performances are discussed and analyzed.

III. MCMC

MCMC is a Bayesian inference computation that is capable of solving high dimensional problem [11]. The algorithm estimates the vehicle state and calculates its prior probability $P(\theta)$, proposal distribution $Q(\theta)$, and observation likelihood $\pi(\theta)$. The new estimated state sample θ^* will be proposed based on the previous accepted state sample θ_t^{i-1} where i is the MCMC sample index and t is the frame index. The estimated sample will be determined to be accepted into the MCMC based on the Metropolis-Hasting acceptance ratio, α as shown in (1).

$$\alpha = \min\left(1, \frac{p(\theta^*)Q(\theta_t^{i-1} \mid \theta^*)\pi(\theta^*)}{P(\theta_{t-1}^{i})Q(\theta^* \mid \theta_t^{i-1})\pi(\theta_t^{i-1})}\right) \tag{1}$$

The proposed sample is accepted with the probability $\alpha = 1$ or else the sample is rejected and the previous accepted sample will be re-generated and accepted as new sample in MCMC. The sample is being estimated and proposed until it reached the stopping criteria that have been set. The developed MCMC will then evaluate its mean value as shown in (2). The variable n is the number of samples in the MCMC which is also defines as the MCMC sample size. θ_t^i is the accepted sample of the MCMC and the expected mean in (2) is the estimated vehicle state which is also indicating the tracked vehicle position.

$$E[\theta] = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_t^i\right) \tag{2}$$

A. Prior Probability Distribution

Prior probability distribution calculates the probability of acceptance of the current estimated vehicle state sample based on the finalized state of previous video frame. The vehicle state space is $\theta = \{x, y\}$ where x and y is the centroid coordinates of the vehicle. Lower probability value will be computed if the estimated sample is far from the distribution range and less likely to be accepted into MCMC. The prior probability is shown in (3).

$$p(\theta) = \frac{1}{2\pi{\sigma_p}^2} e^{-\frac{(\theta-\theta_{t-1})^2}{2{\sigma_p}^2}} \tag{3}$$

Equation (3) is a 2-dimensional Gaussian distribution and θ_{t-1}^i is the mean of vehicle state at the previous tracking frame. Gaussian distribution is used due to its characteristic in providing balance distribution range and is able to determine the vehicle state that will be estimated at unpredictable direction. ${\sigma_p}^2$ is the variance of the distribution which will determine the range of state sample that can be proposed. Larger estimation range indicates that the variance is set to large value and smaller range indicates smaller variance value.

B. Proposal Distribution

Proposal distribution is being used to estimate the new state sample. It is a 2-dimensional Gaussian distribution and the new sample is estimated based on the previous accepted sample. The proposal distribution is defined in (4).

$$Q(\theta \mid \theta_t^{i-1}) = \frac{1}{2\pi{\sigma_q}^2} e^{-\frac{(\theta-\theta_t^{i-1})^2}{2{\sigma_q}^2}} \tag{4}$$

Variable θ_t^{i-1} is the previous accepted vehicle state sample. The distribution will compute lower probability value if the estimated sample is apart from the previous accepted sample. This is necessary to prevent the algorithm from estimating the sample that is out of the desired tracking range. Metropolis-Hasting acceptance ratio will calculate the acceptance probability of the proposed sample along with the observation likelihood and prior probability. If the proposed sample is accepted, it will be used as the reference state sample to compute another proposed state sample for the next iteration.

DOI 10.5013/IJSSST.a.12.03.06

ISSN: 1473-804x online, 1473-8031 print

C. Observation Likelihood

Observation likelihood is the measurement of similarity of the proposed vehicle state sample to the target vehicle model. Edge and color features are common parameters that have been widely implemented in machine vision to determine the outlook characteristics of objects [12]. Therefore the implemented likelihood is based on the edge distance transform and RGB color similarity [7, 13]. The actual vehicle model is extracted and it is used as the reference to calculate the observation likelihood. If the proposed state is near to the target vehicle then the similarity is higher. Higher likelihood value is produced and MCMC will be more probable to accept the proposed sample. Fig. 1 shows the color and edge vehicle model that have been extracted.

RGB color histogram has been constructed to compute the color likelihood. The histogram is built with bin size of $8 \times 8 \times 8$ where the bin combination will increase the likelihood sensitivity. This has enabled the accurate comparison of colour histogram between the proposed sample and the model. Bhattacharyya coefficient is used as the indicator to determine the colour histogram similarity which is defined in (5).

$$\rho(p,q) = \int p(u)q(u)du \tag{5}$$

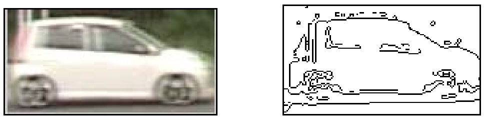
The bin index is indicated by u . $q(u)$ is the constructed colour histogram of vehicle model and $p(u)$ is histogram of the proposed vehicle state. Bhattacharyya distance will then be calculated using (6).

$$d = \sqrt{1-\rho(p,q)} \tag{6}$$

Large Bhattacharyya distance value indicates that the proposed sample is similar to the target vehicle model and low distance value signifies the sample is not similar to the target vehicle. The computation of colour likelihood is defined in (7). If the proposed sample has high similarity or small Bhattacharyya distance value, the colour likelihood value will be increased and the proposed sample is more probable to be accepted into MCMC.

$$\pi(C \mid \theta) = \left(\sqrt{1-\rho(p,q)}\right)^{-1} \tag{7}$$

For the calculation of edge likelihood, edge distance transform has been implemented. Edge distance transform calculates the distance of the pixels from the vehicle edge. Thus the distance value will be larger if the proposed vehicle state is further apart from the actual vehicle's edge. The steps of computing the edge distance using the vehicle



(a) (b)
Figure 1. (a) Colour model (b) Edge distance model

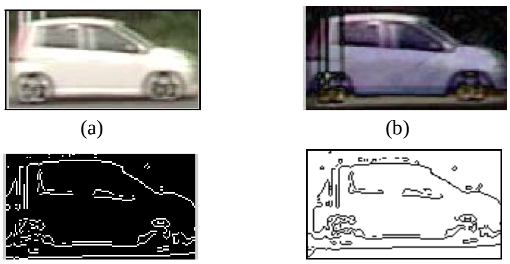
model is shown in Fig. 2. Fig. 2(a) is the extracted target vehicle model. The absolute difference between the background and vehicle model are illustrated in Fig. 2(b). The vehicle edge is then extracted using Sobel edge detection algorithm which is shown in Fig 2(c). Finally the edge distance transform is performed on the extracted edge by computing the Euclidean distance to the nearest neighbour edges as displayed in Fig. 2(d). The edge likelihood is then determined with (8) and (9).

$$d = \frac{1}{n}\sum_{n=1}^{T} EdgeDist(x,y) \tag{8}$$

$$\pi(E \mid \theta) = \frac{1}{2\pi\sigma_d^{\,2}} e^{-\frac{d}{2\sigma_d^{\,2}}} \tag{9}$$

From (8), the variable (x, y) is the coordinate of the edge pixels of the vehicle model. T is the total number of edge pixels and the corresponding coordinate of edge distance of the proposed sample is added up to calculate the distance average. Edge distance likelihood is determined using (9) which is a Gaussian distribution function. Hence if the proposed sample is near to the target vehicle, the distance d is small and the edge distance likelihood value will be increased.

Fusion of colour and edge distance has been implemented to compute the observation likelihood. The combinations of both parameters enable the algorithm to keep track on the overlapped vehicle. When vehicle is fully overlapped at the back of another vehicle, the algorithm can only track the front vehicle based on the edge distance likelihood. After the overlapped vehicle begins to appear, colour likelihood assists to keep tracking on the target



(a) (b) (c) (d)
Figure 2. Edge Distance Transform

vehicle. Thus weighting between these two likelihoods is essential in the tracking algorithm. The defined observation likelihood is shown in (10) where β_θ and γ_θ is the weight of colour and edge distance likelihood respectively. The colour and edge distance likelihood priority can be set by calibrating the weight to suits the overlapping situations.

$$\pi(\theta) = \beta_\theta\left[\pi(C\mid\theta)\right]\cdot\gamma_\theta\left[\pi(E\mid\theta)\right] \tag{10}$$

D. MCMC Tracking Algorithm

The vehicle tracking algorithm is implemented on the image frame of the video data. An image is extracted from the video to track the target vehicle and the first vehicle state θ_t^1 will be initialized. By using the proposal distribution, a new vehicle position θ^* is proposed where computation on prior probability $P(\theta)$ and observation likelihood $\pi(\theta)$ are performed. If the proposed position is near to the target vehicle and met the computed Metropolis-Hasting acceptance ratio then the state will be accepted as a new sample θ_t^i. However the previous accepted sample θ_t^{i-1} will be replicated as the new sample in MCMC if the current proposed state did not achieve the acceptance ratio. The iterative sampling process will be repeated until the stopping criterion is met. Vehicle position is estimated by calculating the expected mean value of the MCMC using (2) and the computed position will then be used for prior probability calculation of the next tracking frame.

Moving to the next frame, the first MCMC sample is initialized by randomly select one of the samples from the previous frame. This can ensure that the initial state are near to the target vehicle and will not vary too far from the estimated position of the previous frame. New sample will be proposed again and the MCMC procedure is repeated until the entire video frames have been processed. TABLE I shows the implemented algorithm which is based on fixed sample size. The sample size is predetermined and the algorithm will stop sampling and extract the vehicle position when the sample size limit is reached.

The conventional MCMC tracking algorithm is lack of computation efficiencies as appropriate sample size are needed to be determined before the tracking is performed. Tracking with large sample size will increase computational time whereas setting sample size that is too small will cause tracking error. Thus, CUSUM path plot is implemented to adaptively determine the appropriate MCMC sample size that suits the tracking situation. The proposed algorithm is capable of enhancing the tracking performance with higher accuracy and lesser processing time.

TABLE I. MCMC TRACKING ALGORITHM

MCMC Tracking Algorithm
Initial first MCMC sample θ_t^1 at time *t* with $P(\theta)$
for frame = 1 to *t*
Initial first sample θ_t^1 at time *t* from θ_{t-1}
for *i* = 1 to *n*
Estimate new state θ^* with proposal distribution Q
Calculate prior probability, $P(\theta_{t-1}^i)$ and $P(\theta^*)$
Calculate observation likelihood, $\pi(\theta_t^{i-1})$ and $\pi(\theta^*)$
Compute the acceptance ratio, α
if $\alpha = 1$ **then** add $\theta_t^i = \theta^*$
else add $\theta_t^i = \theta_t^{i-1}$
end if
end for
Calculate vehicle position $E[\theta]$
end for frame

IV. CUSUM PATH PLOT

CUSUM path plot is capable of quantitatively determine the convergence rate of MCMC. The MCMC will stop sampling or stop proposing new samples when the CUSUM path plot diagnosed MCMC have been converged. The convergence rate is indicated by the hairiness of CUSUM path plot. Hairiness is the mixing rate of MCMC samples and it is calculated based on the maximum and minimum tuning point of the state samples' Euclidean distance [9, 10]. MCMC is diagnosed as converged if the rate of occurrence of turning point increases vastly and reaches CUSUM stopping criteria. The CUSUM path plot algorithm first calculates the mean of MCMC sample until the latest accepted sample θ_t^i as shown in (11). Euclidean distance of each accepted sample is then calculated based on the computed mean on every state acceptance's iteration as defined in (12).

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\theta_t^i \tag{11}$$

$$S_i = \sum_{i=1}^{n}\left(\theta_t^i - \mu\right) \tag{12}$$

The hairiness of CUSUM path plot is determined by plotting the Euclidean distance S_i. If the plot is smooth

DOI 10.5013/IJSSST.a.12.03.06

ISSN: 1473-804x online, 1473-8031 print

then MCMC are diagnosed as undergoing slow mixing rate and still require more sample to reach convergence. In another way, hairy plot indicates that MCMC is in fast mixing rate approximating to convergence [10, 14]. The hairiness of plot is determined by using the hairiness index, D_i as shown in (13).

$$D_i = \begin{cases} 1 & if \quad S_{i-1} > S_i \quad and \quad S_i < S_{i+1} \\ & or \quad S_{i-1} < S_i \quad and \quad S_i > S_{i+1} \\ \frac{1}{2} & if \quad S_{i-1} == S_i \quad and \quad S_i == S_{i+1} \\ 0 & otherwise \end{cases} \tag{13}$$

The hairiness index will be added by 1 if the plot encounter local minimum and local maximum turning point. If the Euclidean distance remains constant and did not undergo changes then hairiness index is incremented by 0.5. If it is undergoing a smooth plot, then increment will not be performed on the hairiness index and MCMC will not reach convergence. The hairiness is calculated as shown in (14).

$$H = k\left(\frac{1}{n}\right)\sum_{i=1}^{n} D_i \tag{14}$$

The constant k is implemented for the calibration of hairiness value which enables the algorithm adaptable to various tracking situations and dynamic environmental effects. As the MCMC sample size n grows larger, the law of large numbers states that the hairiness will approximate to normal distribution with the mean value of 0.5. Hence MCMC can be diagnosed as converged if the hairiness is within the boundary as defined in (15).

$$H = \frac{1}{2} \pm Z_{\frac{\alpha}{2}}\sqrt{k\left(\frac{1}{4n}\right)} \tag{15}$$

The boundary in (15) indicates that MCMC will reach convergence when the computed hairiness is approximated to mean $1/2$ with variance of $k(1/4n)$. Variable $Z_{\alpha/2}$ is the defined confidence interval that determines the amount of samples required within the boundary [14]. To ensure most of the accepted states are accurate, MCMC is identified as converged when 95% of the samples are within the boundary. Equations (16), (17) and (18) are used to determine $Z_{\alpha/2}$.

$$1-\alpha = 0.95 \tag{16}$$

$$\phi\left(Z_{\frac{\alpha}{2}}\right) = 1-\frac{\alpha}{2} = 0.975 \tag{17}$$

$$Z_{\frac{\alpha}{2}} = \phi^{-1}(0.975) = 1.96 \tag{18}$$

Equation (16) calculates the confidence measure where 95% of the MCMC samples fall within the boundary. The cumulative distribution of $Z_{\alpha/2}$ is computed in (17) and will be further evaluated as shown in (18). Equation (15) will then be updated by substituting (18) and the new boundary equation is defined in (19).

$$H = \frac{1}{2} \pm 1.96\sqrt{k\left(\frac{1}{4n}\right)} \tag{19}$$

MCMC is diagnosed as converged when hairiness fulfils the boundary condition and then the algorithm will stop MCMC from generating new samples. The converged MCMC sample size is suitable for further evaluation since the evaluated vehicle position is more accurate compared to the fixed size MCMC.

V. CUSUM PATH PLOT TRACKING

CUSUM path plot is implemented into the conventional MCMC tracking algorithm to adaptively sample MCMC and it has the similar procedure as fixed sample size MCMC. To further enhance the tracking performance, higher-order prior probability distribution is implemented [15]. The proposed distribution is capable of overcoming the limitation of the conventional first order prior probability.

Conventional MCMC encounters tracking error when the vehicle undergoes long overlapping situation. The tracking error occurs when the occluded vehicle is fully covered by the other vehicle. When the overlapped vehicle is partially appear, the prior probability is calculated based on the previous state θ_{t-1} where the vehicle is still fully occluded. The correct information is lost during the overlapping and the prior probability might not be computed based on the correct position of the occluded vehicle. To overcome this problem, *m*-th order prior probability distribution is introduced.

The distribution calculates the prior probability based on the previous *m*-th frame, before the vehicle is overlapped. The multiple frame information will shift the prior probability to a more reliable and accurate position based on the vehicle state from the previous *m*-th frame. It enables the algorithm to track the vehicle that is undergoing long overlapping period. However the selection of the *m*-th order is essential for the tracking performances. Selecting *m*-th order that are too large might lead to tracking error as the vehicle position might change a lot over multiple frames whereas smaller order might not be enough to overcome the information lost in the long period of overlapping. Hence the selection of the *m*-th order must be carried out according to the overlapping situation. The proposed prior probability is defined in (20) and (21).

$$\theta_h = \theta_{t-1} + \frac{1}{m}\sum_{k=1}^{m}\theta_{t-k} \tag{20}$$

$$P(\theta \mid \theta_{t-1:t-m}) = \frac{1}{2\pi\sigma_p{}^2} e^{-\frac{(\theta-\theta_h)^2}{2\sigma_p{}^2}} \tag{21}$$

Equation (20) is used to replace θ_{t-1} in (3) and develop the proposed prior probability in (21). The proposed prior probability will be implemented on both fixed sample size MCMC and CUSUM path plot MCMC and their performances will be compared. TABLE II is the algorithm of the CUSUM path plot MCMC in tracking the overlapped vehicle.

VI. RESULTS AND DISCUSSIONS

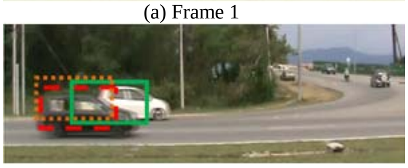
The tracking of overlapped vehicle under fixed sample size MCMC and CUSUM path plot MCMC have been performed. Fixed sample size MCMC is simulated using 30 MCMC sample size and 100 MCMC sample size and their performances are analyzed as compared to CUSUM path plot MCMC. Fig. 3 shows the tracking results of CUSUM path plot MCMC. The tracking accuracy of the algorithm is then determined by using Euclidean distance as shown in Fig. 4. Fig. 5 displays the adaptive sample size of CUSUM path plot MCMC. The Euclidean distance in Fig. 4 is the difference of the tracked vehicle position to the actual vehicle position. The vehicle is determined as lost track if the Euclidean error exceeds the value of 40 whereas error below that indicates the algorithm is still able to keep track on the vehicle.

In Fig. 3, the vehicle tracked by CUSUM path plot MCMC is indicated by the line bracket. The dotted bracket and the line bracket are the tracking results of MCMC with 100 sample size and 30 sample size respectively. Every MCMC based algorithms have shown good tracking results in Frame 1 when the tracking vehicle is free from occlusion. After the vehicle is started to overlap at Frame 4, the CUSUM algorithm is still capable of tracking the vehicle compared to MCMC with 30 and 100 sample size. Both 30 and 100 sample size consist of higher tracking error since the estimated position has been influenced by another vehicle that appears to block the target tracking vehicle. When the target vehicle is occluded, the RGB colour information of the target vehicle is blocked by the occluded vehicle at front and the computed value of colour likelihood will decreased vastly. The edge distance likelihood of the vehicle appears in front of the occluded vehicle has affect the computation of observation likelihood of the proposed sample and become more probable to be accepted near to its area.

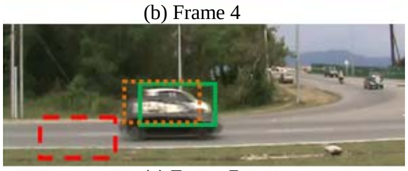
TABLE II. CUSUM PATH PLOT MCMC TRACKING ALGORITHM

CUSUM Path Plot MCMC Tracking Algorithm
1: Initial first MCMC sample θ_t^1 at time *t* with $P(\theta)$
2: **for** frame = 1 to *t*
3: Initial first sample θ_t^1 at time *t* from θ_{t-1}
4: **Loop**
5: Estimate new state θ^* with proposal distribution *Q*
6: Calculate prior probability, $P(\theta_t^{i-1} \mid \theta_{t-1:t-m})$ and $P(\theta^* \mid \theta_{t-1:t-m})$
7: Calculate observation likelihood, $\pi(\theta_t^{i-1})$ and $\pi(\theta^*)$
8: Compute the acceptance ratio, α
9: **if** $\alpha = 1$ **then** add $\theta_t^i = \theta^*$
10: **else** add $\theta_t^i = \theta_t^{i-1}$
11: **end if**
12: Sum up current sample set $\sum_{i=1}^{n} \theta_t^i$
13: Compute μ and S_i
14 **if** found local maximum or minimum point
15: add 1 to hairiness index
16: **else if** 3 consecutive S_i remain constant
17: add ½ to hairiness index
18: **else**
19: add 0 to hairiness index
20: **end if**
21: Compute hairiness *H*
22: **if** *H* lies within $\frac{1}{2} \pm 1.96\sqrt{k\left(\frac{1}{4n}\right)}$
23: go to end Loop
24: **else**
25: go to Loop
26: **end if**
27: **end Loop**
28: Compute vehicle position $E[\theta]$
29: Record computed state θ_t for *m*-th order prior distribution calculation.
30: **end for frame**

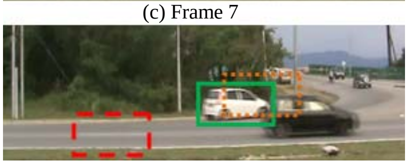
(a) Frame 1



(b) Frame 4



(c) Frame 7



(d) Frame 12

Figure 3. MCMC CUSUM path plot vehicle tracking

Hence it can be observed in Frame 4 that fixed size MCMC tends to track the front blocking vehicle due to the decrease of colour likelihood value. Nevertheless, the capability of CUSUM path plot to keep tracking on the vehicle is due to the implementation of the third order prior probability distribution is shifted to a more accurate position according to the previous frame of the vehicle movement. The shifting of the distribution allows the proposed samples placed near to the target vehicle are more probable to be accepted into MCMC and therefore produces a better tracking position.

The target vehicle is almost completely overlapped at Frame 7 and the CUSUM algorithm is still capable of tracking the target vehicle accurately. This is due to the adaptive sampling of CUSUM path plot where appropriate MCMC sample size has been computed for the computation of vehicle position. MCMC with 30 sample size has totally lost track on the target vehicle whereas MCMC with 100 sample size has shown inaccurate tracking result since its prior probability is poorly distributed. The 100 sample size MCMC is also over sampled where more defected samples are accepted since the complete overlap situation will lead to computation of inaccurate observation likelihood, which lack of color likelihood feature and affected by the edge likelihood of the front blocking vehicle.

At Frame 12, the target vehicle has just completed the overlapping and it is tracked accurately with CUSUM path plot using small MCMC size since there are no disturbances that will affect the observation likelihood calculation. However the dotted tracker has shifted beyond the target vehicle due to the oversize of MCMC where

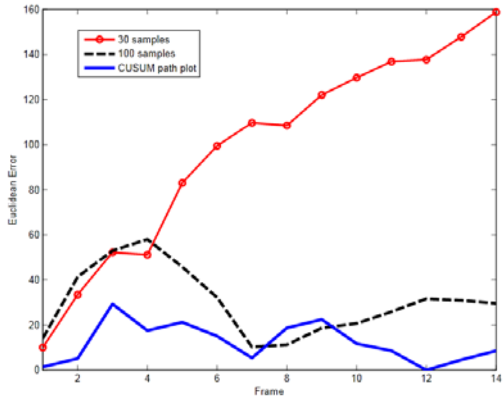


Figure 4. Euclidean distance of vehicle position

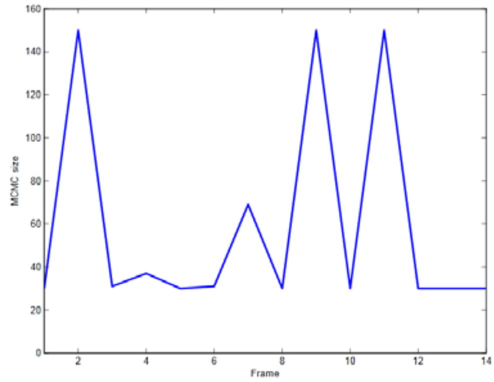


Figure 5. MCMC converged sample size

sample that are defected have been accepted and increased along with the growing sample size.

The overall tracking performance of Fig. 3 is further elaborated and represented with Euclidean distance in Fig. 4. It can be seen that MCMC with 30 sample size has the highest tracking error. The tracking encounters failure at the early frame when the vehicle is overlapped at Frame 2 and the algorithm is not able to track the vehicle since then. The main reason of the failure is due to the insufficient sample size which reduces the possibility of MCMC to accept better likelihood samples and will cause tracking error if the accepted samples are defected at the early stage. The 100 sample size MCMC represented by dash plot has shown better tracking performances compared to the 30 sample size. However, the algorithm encounters higher tracking error when the vehicle is overlapped within Frame 2 and Frame 8. The likelihood information is lost when the target vehicle is occluded and the poorly distributed prior probability has caused the tracking error. Higher sample size MCMC can be implemented to overcome this limitation but it is computationally expensive and bares the risk of increasing defected sample which is not desirable in the tracking algorithm.

Results in Fig. 4 justify CUSUM path plot has better tracking accuracy compared to the 100 sample size MCMC with lower Euclidean error. It can be observed in Fig. 5

that CUSUM path plot MCMC has adapted the MCMC to track the vehicle according to the tracking situation. When the vehicle is not overlapping, CUSUM MCMC tends to generate lesser sample size to track the vehicle. This is because small sample size is sufficient enough to calculate the exact vehicle state and the vehicle can be tracked easily. More sample size is generated when the vehicle begins to overlap and just after the overlapping. During these critical situations, the accepted sample is unstable due to the insufficient information of the target vehicle. The computation of hairiness in (14) shows that the plot is not hairy enough due to the inconsistency of the MCMC samples' mean and is not yet determined as converged where more sample states are required to improve the tracking accuracy.

For instance, the algorithm has stop sampling at the sample size of 150 at Frame 9 and 11 when the vehicle is partially overlapped. The prior vehicle information at this stage is still insufficient because of the long overlapping period. Hence there will be a lot of accepted MCMC samples that are defected and not yet suitable for vehicle position extraction. Therefore additional sample have been proposed to compensate the defected MCMC states until the hairiness has reached within the boundary in (19). Experimental results show that CUSUM path plot MCMC consumes 31% lesser computational time than 100 sample size MCMC. As a result, CUSUM path plot has met a better trade off between accuracy and efficiency of tracking performance.

VII. CONCLUSION

The CUSUM path plot MCMC algorithm produces better performance in tracking the overlapping vehicle with lesser processing time. The proposed algorithm is capable of adaptively determining the sample size in MCMC by generating more samples when the vehicle is occluded and less sample size is utilized to track the vehicle. The ability to cope with environmental disturbances by adaptively alters the sample size in CUSUM path plot has overcome the limitation of the conventional fixed size MCMC. Observation likelihood fusing both color and edge distance likelihood has been introduced and the implementation of *m*-th order prior probability has further enhanced the algorithm tracking accuracy. Thus it can be concluded that the proposed CUSUM path plot MCMC algorithm is efficient in tracking the overlapping vehicle.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial assistance of the Ministry of Higher Education of Malaysia (MoHE) under Fundamental Research Grant Schemes (FRGS) No. FRG0220-TK-1/2010, and the University Postgraduate Research Scholarship Scheme (PGD) by Ministry of Science, Technology and Innovation of Malaysia (MOSTI).

REFERENCES

[1] H.Y. Cheng, P.Y. Liu and Y.J. Lai. "Vehicle tracking in daytime and nightime traffic surveillance videos," in *Education Technology and Computer*, 2010, pp. 122-125.

[2] B. Coifman, D. Beymer, P. McLauchlan and J. Malik. "A real-time computer vision system for vehicle tracking and traffic surveillance," in *Transportation Research Part C: Emerging Technologies*, 1998, pp. 271-288.

[3] F. Bardet and T. Chateau. "MCMC particle filter for real-time visual tracking of vehicles," in *Intelligent Transportation Systems*, 2008, pp. 539-544.

[4] X. Song and R. Nevatia. "A model-based vehicle segmentation method for tracking," in *Computer Vision*, 2005, pp. 1124-1131.

[5] S. Oh, S. Russell and S. Sastry. "Markov Chain Monte Carlo data association for multiple-target tracking." *Automatic Control*, vol. 54, pp. 481-497, 2009.

[6] Q. Yu, I. Cohen, G. Medioni and B. Wu. "Boosted Markov Chain Monte Carlo data association for multiple target detection and tracking," in *Pattern Recognition*, 2006, pp. 675-678.

[7] Y. Jia and C. Zhang. "Front-view vehicle detection by Markov Chain Monte Carlo method," in *Pattern Recognition*, 2009, pp. 313-321.

[8] H.X. Xia, N.W. Yao, Z. Wei, Z. Jiang and F.Y. Xiao. " Multi-object visual tracking based on reversible jump Markov Chain Monte Carlo." *Computer Vision* , vol. 5, pp. 282-290, 2011.

[9] P. Brooks and G. Roberts. "Convergence assessment techniques for Markov Chain Monte Carlo." *Statistics and Computing*, vol. 8, pp. 319-335, 1998.

[10] P. Brooks. "Quantitative convergence diagnosis for MCMC via CUSUMS." *Statistics and Computing*, vol. 8, pp. 267-274, 1998.

[11] C. Andrieu, N. Freitas, A. Doucet and M. Jordan. "An introduction to MCMC for machine learning." *Machine Learning*, vol. 50, pp. 5-43, 2003.

[12] P.N. Trung, W.J. Kang and S.H. Ong. "Fusing color and contour in visual tracking," in *Machine Vision Application*, 2005, pp. 9-12.

[13] W.L. Khong, W.Y. Kow, F. Wong, I. Saad and K.T.K. Teo. "Enhancement of particle filter approach for vehicle tracking via adaptive resampling algorithm," in *Computational Intelligence, Communication Systems and Networks*, 2011, pp. 259-263.

[14] S.E. Adlouni, A.C. Favre and B. Bobee. "Comparison of methodologies to assess the convergence of Markov Chain Monte Carlo methods." *Computational Statistics and Data Analysis*, vol. 50, pp. 2685-2701, 2005.

[15] P. Pan and D. Schonfeld. "Visual tracking using high-order particle filtering," in *Signal Processing Letter*, 2011, pp. 51-54.