Enhancement of Markov Chain Monte Carlo Convergence Speed in Vehicle Tracking Using Genetic Operator

Wei Yeang Kow, Wei Leong Khong, Yit Kwong Chin, Ismail Saad, Kenneth Tze Kin Teo
Modeling, Simulation & Computing Laboratory, Minerals & Materials Research Unit
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Malaysia
msclab@ums.edu.my, ktkteo@ieee.org

Abstract—Markov Chain Monte Carlo (MCMC) has been essential in tracking vehicle undergoing disturbances for traffic surveillance purposes. It is capable of tracking vehicle by estimating the vehicle’s position with the sampling of probability distributions. However the accuracy of the position estimation is highly dependent on the sampling efficiency of MCMC. Therefore the sample size of the MCMC is adapted to track the vehicle according to the disturbances encountered. The adaptive sample size of MCMC is determined by using the CUSUM path plot and variance ratio convergence diagnostic algorithm. To further enhance the convergence speed, genetic crossover and mutation operator is introduced into the adaptive MCMC. The genetic operator (GO) is capable of reduces the variance between samples and hence allowing faster convergence speed on the MCMC samples. Experimental results have shown that the GO adaptive MCMC tracking algorithm have better tracking performances with consumption of lesser sample size.

Keywords—Markov Chain Monte Carlo (MCMC); genetic operator (GO); variance ratio (VR); CUSUM path plot

I. INTRODUCTION

Video sensors have been commonly implemented in vehicle tracking due to their capability in obtaining wide range of vehicle information [1]. Compared to other implemented sensors, the video sensors can provide better vehicle information such as velocity, shape, trajectories or any observable vehicle features. More vehicle information will further enhance the robustness and accuracy of the vehicle tracking algorithm which is desirable in traffic surveillance system [2]. However, the video sensors came with limitations where the vehicle information will be lost if the vehicle is undergoing overlapping disturbances. This will greatly affect the tracking accuracy as the disturbance may cause to insufficient vehicle information for the tracking algorithm computations.

Researches has been performed to track the vehicle undergoes occluded or overlapped situations in front view and side view. MCMC has been implemented as the tracking algorithm and the implementation result are robust and accurate [3, 4]. The performance of MCMC is greatly depends on the efficiency of the sampling procedure. Sample size that is too small will have insufficient information for the tracking whereas large sample size will give better tracking result pairing with higher computational cost. Researches in [5] have tracked multiple targets using MCMC with fixed sample size. The results have shown promising accuracy but the large sample size requires higher computational time. Furthermore, the fixed sample size MCMC also leads to a limitation where suitable sample size is difficult to be determined.

Hence adaptive MCMC has been implemented to track the vehicle according to the disturbances encountered. Small sample size will be generated when vehicle is clear from disturbances and more samples will be generated if the vehicle information is lost due to disturbance [4, 6]. CUSUM path plot is one of the MCMC convergence diagnostic algorithm that is capable of determine the convergence rate of the MCMC sample space [7]. However the hairiness computation of the CUSUM path plot algorithm is sensitive where single defected sample may cause to slower convergences rate. Other than this, VR convergence diagnostic has been implemented to adaptively track overlapped vehicle by using multiple sequences of MCMC [7, 8]. The tracking algorithm has enable better sample exploration rate however the multiple MCMC may cause to more sample generation which escalated the difficulties to reach convergence. Therefore, to further improve the MCMC convergence rate, genetic crossover and mutation operator has been introduced to reduce the sample gap of MCMC and hence allowing the MCMC to reach convergences at smaller sample size [9].

In this paper, MCMC sample size will be adapted by using CUSUM path plot and VR convergence diagnostic algorithm. Genetic crossover operator and mutation operator will be implemented to improve the MCMC convergence speed. The crossover operator is capable of reducing the variance between samples whereas mutation operator has further improved the MCMC sample accuracy. The developed GO-CUSUM-MCMC and GO-VR-MCMC will be implemented to track the vehicle under overlapping disturbances. The proposed tracking algorithm is capable of improving the performance of conventional adaptive MCMC tracking algorithm by tracking the target vehicle accurately with lesser sample size.
II. MCMC TRACKING ALGORITHM

MCMC is a sampling based algorithm that is able to perform prediction on high dimensional probability distributions. Sample state of the target vehicle will be continuously proposed to form a Markov Chain and Monte Carlo integration will be computed on the developed Markov Chain to estimate the target vehicle position. The implemented sample state $\theta$ is the vehicle position coordinate $\{x, y\}$ in the tracking frames. The vehicle position sample cannot be proposed randomly as random proposal will cause to the slow convergence rate of the MCMC. Hence proposal distribution has been implemented to proposed new position sample $\tilde{\theta}$ as indicated in (1).

\[
Q(\theta^* | \tilde{\theta}^{i-1}) = \frac{1}{2\pi \sigma_q} e^{-\frac{(\theta^* - \tilde{\theta}^{i-1})^2}{2\sigma_q^2}}
\]  

(1)

From (1), it can be seen that the proposal distribution is a Gaussian distribution. It will ensure that the proposed position is generated within the variance range of $\sigma_q$ according to the mean of the distribution. Hence the proposed vehicle position will not be distributed away and more focus to the target distribution. The newly proposed vehicle position is then used to compute the prior probability distribution as shown in (2).

\[
p(\theta^*) = \frac{1}{2\pi \sigma_p^2} e^{-\frac{(\theta^* - \theta^i)^2}{2\sigma_p^2}}
\]  

(2)

Prior probability is to compute the probability of acceptance of the current proposed vehicle position sample based on the last computed vehicle position at the previous frame. If the proposed position sample is over the variance range $\sigma_p$, the position sample is less likely to be accepted as the sample space is at an undesirable area and vice versa.

Observation likelihood $\pi(\theta^*)$ will then be computed to determine the similarity of the proposed vehicle position to the actual vehicle outlook. The observation likelihood is computed based on the target vehicle outlook feature. In this implementation, edge distance likelihood and color likelihood are computed. The edge distance likelihood will determine the similarity of the vehicle edge of the proposed position to the target vehicle model. Whereas color likelihood will be computed using the HSV color space to distinguish vehicles that has similar edge likelihood [10]. The edge distance likelihood is computed using equation in (3) and color likelihood is computed using (4).

\[
\pi(E | \theta) = \frac{1}{2\pi \sigma_d^2} e^{-\frac{d}{2\sigma_d^2}}
\]  

(3)

\[
\pi(C | \theta) = \frac{1}{2\pi \sigma_c^2} e^{-\frac{B}{2\sigma_c^2}}
\]  

(4)

The edge distance likelihood in (3) is computed but using the edge distance transforms method. The edge distance value $d$ will become larger if the proposed vehicle position is far from the target vehicle and vice versa. Hence smaller distance value indicates that the proposed vehicle position is more accurate and therefore better edge distance likelihood will be obtained as shown in (3). The color likelihood is computed by using the Bhattacharyya distance $B$. Similar to the edge distance computation, higher color similarity between the proposed vehicle position to the vehicle model will give smaller value of $B$. Hence larger color likelihood value will be obtained in (4) with smaller Bhattacharyya distance. Both of the likelihood is then fused together to compute the observation likelihood as shown in (5).

\[
\pi(\theta) = \beta \pi(C | \theta), \gamma \pi(E | \theta)
\]  

(5)

The variable $\beta$ and $\gamma$ is the weight constant of the color likelihood and edge likelihood respectively. Calibration on the weight will set the computation priority of the observation likelihood. Larger $\beta$ value will ensure that the target vehicle is tracked prior to the color likelihood and larger $\gamma$ value will ensure that the edge of vehicle is prioritized. The acceptance rate of the proposed new position will then be determine by using the Metropolis-Hasting tracking algorithm as shown in (6).

\[
\alpha = \min \left( 1, \frac{P(\theta^*)Q(\tilde{\theta}^{i-1} | \theta^*) \pi(\theta^*)}{P(\tilde{\theta}^{i-1})Q(\tilde{\theta}^i | \theta^{i-1}) \pi(\tilde{\theta}^{i-1})} \right)
\]  

(6)

As indicated in (6), the Metropolis-Hasting algorithm will determine the acceptance rate of the proposed position sample based on the prior probability, transition probability and observation likelihood. The proposed vehicle position will be accepted into the Markov Chain with the acceptance probability $\alpha$ or else the proposed sample will be rejected and the previous accepted position sample will be regenerated as the latest position sample in the MCMC.

New vehicle position sample will be proposed and computed for acceptance probability for the next iteration until the stopping criteria has been met. Therefore the iteration process will form a MCMC with $n$ number of position samples. The final vehicle position will be estimated by using the Monte Carlo integrator as defined in (7).

\[
E(\theta, \frac{1}{n} \sum_{i=1}^{n} \tilde{\theta}^i)
\]  

(7)
III. CUSUM-MCMC VEHICLE TRACKING

The CUSUM path plot convergence diagnostic determines the convergence rate of the MCMC by calculating the occurrence of local maxima in the MCMC samples. Hairiness is the occurrence of local maxima of the MCMC samples where more maximum or minimum point indicates that the MCMC samples are in fast mixing rate. The local maxima is determined by computing the Euclidean distances of the latest accepted samples to the mean of the MCMC as computed in (8).

\[ S_i = \sum_{i=1}^{n} (\theta_i^j - \mu) \]  

(8)

The mean value \( \mu \) is the mean of the MCMC until previous accepted samples position. The difference of the accepted sample position to the mean will be plotted to compute the hairiness index \( D_i \) as shown in (9).

\[ D_i = \begin{cases} 1 & \text{if } S_{i-1} > S_i \text{ and } S_i < S_{i+1} \\ 0.5 & \text{if } S_{i-1} < S_i \text{ and } S_i > S_{i+1} \\ 0 & \text{otherwise} \end{cases} \]  

(9)

The hairiness index will be incremented with value of 1 when the local maxima are detected whereas 0.5 is incremented when the plot is in static state. Increment will be halted if the plot is experiencing smooth plot. Smooth plot indicates that the MCMC sampling is in slow mixing rate and hence smaller hairiness index will be obtained. Therefore more samples are needed to be generated to achieve convergence. Larger index value will be obtained if the plot is hairy which indicates that the samples are in fast mixing rate and closer to convergence. Mean value of the hairiness index is then divided to the number of samples to obtain the average hairiness value \( H \). The MCMC samples are determined as converged if the hairiness value reaches the CUSUM boundary in (10).

\[ \frac{1}{2} - 1.96 \sqrt{k \left( \frac{1}{4n} \right)} \leq H \leq \frac{1}{2} + 1.96 \sqrt{k \left( \frac{1}{4n} \right)} \]  

(10)

Hence the hairiness that reached within the CUSUM boundary has indicated that 95% the MCMC samples are near to the sample means and convergence is achieved. The sample proposal iteration will be stop and the MCMC sample can be evaluated by Monte Carlo integration to obtain the final vehicle position. TABLE I has shown the tracking algorithm of the developed CUSUM-MCMC.

<table>
<thead>
<tr>
<th>TABLE I. CUSUM-MCMC TRACKING ALGORITHM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: for frame ( t = 1 ) to end</td>
</tr>
<tr>
<td>2: Initial first sample ( 1 ) at time ( t )</td>
</tr>
<tr>
<td>3: Loop</td>
</tr>
<tr>
<td>4: Estimate new state ( \bar{\theta}^s ) with proposal distribution.</td>
</tr>
<tr>
<td>5: Calculate prior probability</td>
</tr>
<tr>
<td>6: Calculate observation likelihood</td>
</tr>
<tr>
<td>7: Compute Metropolis-Hasting acceptance ratio, ( \alpha )</td>
</tr>
<tr>
<td>8: Generate random value ( U = rand(0,1) )</td>
</tr>
<tr>
<td>9: if ( U \leq \alpha ), add ( \theta_i^j = \bar{\theta}^s )</td>
</tr>
<tr>
<td>10: else add ( \theta_i^j = \theta_i^{j-1} )</td>
</tr>
<tr>
<td>11: end if</td>
</tr>
<tr>
<td>12: Compute ( \mu ) and ( S_i )</td>
</tr>
<tr>
<td>13: Compute hairiness index, ( D_i )</td>
</tr>
<tr>
<td>14: Compute hairiness ( H )</td>
</tr>
<tr>
<td>15: if ( H ) reached within boundary (10), go to end Loop</td>
</tr>
<tr>
<td>16: else go to Loop</td>
</tr>
<tr>
<td>17: end if</td>
</tr>
<tr>
<td>18: end Loop</td>
</tr>
<tr>
<td>19: Compute vehicle position with Monte Carlo integration</td>
</tr>
<tr>
<td>20: end frame</td>
</tr>
</tbody>
</table>

IV. VR-MCMC VEHICLE TRACKING

The VR convergence diagnostic determines the convergence rate by implementing multiple sequences of MCMC. In this research, two sequences of MCMC will be implemented and the variances within and between the MCMC sequences will be computed. The obtained variances will then be implemented to compute the variance ratio value where ratio value approximate to 1 indicates that the MCMC samples are close to convergence. The variance between MCMC sequences and the variance within single MCMC sequences is defined in (11) and (12) respectively.

\[ B = \frac{1}{n-m} \sum_{j=1}^{m} (\bar{\theta}_j - \bar{\theta})^2 \]  

(11)

\[ W = \frac{1}{m} \sum_{j=1}^{m} \left( \frac{1}{n-1} \sum_{i=1}^{n} (\theta_i^j - \bar{\theta})^2 \right) \]  

(12)

The mean value of a single MCMC sequences is indicated by the variable \( \bar{\theta}_j \) is \( \bar{\theta} \) is the mean value between MCMC sequences. Variable \( m \) is the number of MCMC sequences and \( n \) is the total amount of current accepted MCMC samples. It is notable that smaller between variance value will be obtained if both MCMC sequences are converging to the target vehicle accurately. Similar condition goes to the variance within sequence where smaller value will be obtained when the accepted proposed samples are accurate and near to the target vehicle. The multiple sequences implementation has enabled the MCMC
convergence rate for not being affected greatly by small amount of delected sample in the MCMC sequences. This is because the delected sample can be compensated by the acceptance of good position sample at another sequence of MCMC. The variance ratio $R$ between the variance within sequence and variance between sequences is computed as shown in (13).

$$R = D \left( \frac{n-1}{n} \right) + \left( 1 + \frac{1}{m} \right) \frac{B}{nW}$$  \hspace{1cm} (13)$$

When $R$ approaches the value of 1. The variances between samples are diagnosed as small and hence the MCMC samples are determined as converged. The variable $D$ is to calibrate the inherent approximation for the variance ratio. When convergence is achieved, the final vehicle position will be evaluated by using the Monte Carlo integration. The developed VR-MCMC tracking algorithm is shown in TABLE II.

V. GENETIC OPERATOR IN ADAPTIVE MCMC

Genetic operator is implemented in to the developed adaptive MCMC to increase the MCMC convergence speed. The crossover operator is capable of generate new children sample position by taking one part of description of one point and combine with another part of another point of the parent samples. Arithmetic crossover operator will be implemented to crossover the sample between two latest accepted position sample forming the new children position sample as shown in (14) and (15).

$$\text{Children1} = a \times \text{Parent1} + (1 - a) \times \text{Parent2} \hspace{1cm} (14)$$

$$\text{Children2} = (1 - a) \times \text{Parent1} + a \times \text{Parent2} \hspace{1cm} (15)$$

Hence the new sample position will have the characteristics of both the parent samples. Therefore if one of the parent samples is not in a good quality, the other parent sample that is at better quality will compensate the defect and hence generate children sample solution with acceptable quality. The characteristic is highly desirable in the MCMC as the accepted samples during the initial stage are mainly inaccurate. The continuous acceptance of new proposed position sample which are usually have better quality is capable of improving the quality of the initial accepted samples by using the crossover operation to adapt the better characteristics of the latest accepted position sample with the previous accepted position sample. The crossover operation of the adaptive MCMC is further illustration in Fig. 1.

Fig. 1 has illustrated that the genetic crossover process is implemented during the iterations when new position sample is accepted after the computation of the Metropolis-Hasting algorithm. The latest accepted sample will be crossover with the previous accepted sample using (14) and (15). This has enabled the current better quality sample to compensate the previous accepted sample by generating children solution at better quality. Other than this, the crossover process has reduces the variance between samples which enable the MCMC samples to be close to each other and hence improve the convergence speed with smaller sample size.

Mutation operator is implemented after the crossover process. It is capable of improve the sample accuracy by enabling better sample exploration rate. The mutation operator will temporary mutate the children solution from the crossover process with fairly low probability occurrence. The temporary mutated sample will have larger exploration range than the proposal distribution and observation likelihood will

![Figure 1. Crossover Operation in Adaptive MCMC](image.png)
be computed on the mutated sample. If the mutated sample has better likelihood value, the sample position will be permanently mutated whereas the position sample before mutation will be restored if the likelihood value is poor. Different from the Metropolis-Hasting computation, the mutation process did not compute the prior probability which enable the mutated samples to have wider searching range without restricted by the prior distribution. The genetic operator is implemented after the Metropolis-Hasting algorithm and before the computation of the convergence diagnostic algorithm as shown in TABLE III.

VI. RESULTS AND DISCUSSIONS

The performance of developed GO adaptive MCMC tracking algorithm are evaluated by tracking the target vehicle undergoes overlapping disturbance as shown in Fig. 2. The solid tracker bracket is the tracking result of GO-CUSUM-MCMC and the dotted bracket is the tracking performances of GO-VR-MCMC. It is observable that both the algorithm is capable of keep track on the target vehicle. The target vehicle is tracked accurately when clear from disturbances. However when the target vehicle undergoes partial overlap in frame 7 and frame 9, the tracking brackets are slightly inaccurate. This is because the computation of observation likelihood of the target vehicle has been affected by the disturbing vehicle at the front. Hence the accepted MCMC position sample will be defected and convergence is difficult to be reached which decreases the accuracy of the tracking algorithm. Similar situation has been occurred at frame 10 when the target vehicle just finished overlaps and reappears on the tracking frame. Even though the vehicle is clear from disturbances, the tracking is still slightly inaccurate due to the disturbance of the neighboring vehicle that affected the computation of the observation likelihood.

The RMSE of the developed GO adaptive MCMC tracking algorithm is plotted as shown in Fig. 3 and Fig. 4 shows the adaptive sample size of the developed MCMC tracking algorithm. RMSE is the distance scale of the computed vehicle position to the actual vehicle position. Smaller RMSE value indicates the tracking algorithm has successfully track the target vehicle accurately. It can be seen that the GO-CUSUM-MCMC has lower RMSE value compare to the performance of CUSUM-MCMC. Beside this, the sample size that has been generated for the GO-CUSUM-MCMC is much smaller where maximum of 30 sample size has been generated to track the target vehicle at frame 7 and 10 and sample size between 10 and 15 has been generated for the other tracking frames.

For CUSUM-MCMC tracking algorithm, the maximum of 50 sample size has been generated to track the target vehicle when MCMC samples are difficult to converge due to the overlapped disturbances occurred at frame 6, 8 and 9. This has shown that the genetic operator has successfully reduced the variance between the samples which increased the convergence speed of the MCMC. Therefore smaller sample size is generated for the GO-CUSUM-MCMC to track the target vehicle while preserving the tracking accuracy of the conventional CUSUM-MCMC tracking algorithm.

The tracking performance of VR-MCMC is slightly better than the tracking performances of the GO-VR-MCMC. The GO-VR-MCMC has better tracking accuracy.
at frame 1 to 4 when the target vehicle is free from disturbance but the RMSE value begins to escalate when the target vehicle is overlapped by another vehicle. Nevertheless, the sample size that has been generated for the GO-VR-MCMC is smaller compared to the samples size of VR-MCMC due to the enhancement of convergence rate by the genetic operator. This is the main cause to the higher RMSE value as the sample size is insufficient to compute the vehicle position accurately. However, the average RMSE that are within value of 20 indicates that the target vehicle is still kept tracked accurately. Therefore the GO-VR-MCMC still have better tracking efficiencies as smaller sample size are required to track the target vehicle with slight compensation on then tracking accuracy.

VII. CONCLUSIONS

The proposed GO adaptive MCMC tracking algorithm has successfully tracked the target vehicle undergoing overlapping disturbance. The tracking results have shown that both the GO-CUSUM-MCMC and GO-VR-MCMC are capable to track the target vehicle with lesser sample size. Therefore, the implementation of genetic operator has successfully increased the convergence speed of the CUSUM-MCMC and VR-MCMC tracking algorithm by reducing the variances between the MCMC samples. The proposed algorithm has improved the tracking efficiencies of the conventional adaptive MCMC tracking algorithm which further enhances the robustness of the MCMC to track the target vehicle under different types of disturbances.

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, University Malaysia Sabah (UMS) under UMS Research Grant Scheme (SGPUMS) No. SBBK0026-TK-1/2012, and the University Postgraduate Research Scholarship Scheme (PGD) by Ministry of Science, Technology and Innovation of Malaysia (MOSTI).

REFERENCES


