ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2014.0105
ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING, FEBRUARY 2014, VOLUME: 04, ISSUE: 03

PARTICLE FILTER BASED VEHICLE TRACKING APPROACH WITH IMPROVED RESAMPLING STAGE

Wei Leong Khong[1], Wei Yeang Kow[1], Ismail Saad[1], Chung Fan Liau[1] and Kenneth Tze Kin Teo[2]
Modelling, Simulation & Computing Laboratory, Material & Mineral Research Unit
School of Engineering and Information Technology, Universiti Malaysia Sabah, Malaysia
Email: [1]msclab@ums.edu.my, [2]ktkteo@ieee.org

Abstract
Optical sensors based vehicle tracking can be widely implemented in traffic surveillance and flow control. The vast development of video surveillance infrastructure in recent years has drawn the current research focus towards vehicle tracking using high-end and low cost optical sensors. However, tracking vehicles via such sensors could be challenging due to the high probability of changing vehicle appearance and illumination, besides the occlusion and overlapping incidents. Particle filter has been proven as an approach which can overcome nonlinear and non-Gaussian situations caused by cluttered background and occlusion incidents. Unfortunately, conventional particle filter approach encounters particle degeneracy especially during and after the occlusion. Particle filter with sampling important resampling (SIR) is an important step to overcome the drawback of particle filter, but SIR faced the problem of sample impoverishment when heavy particles are statistically selected many times. In this work, genetic algorithm has been proposed to be implemented in the particle filter resampling stage, where the estimated position can converge faster to hit the real position of target vehicle under various occlusion incidents. The experimental results show that the improved particle filter with genetic algorithm resampling method manages to increase the tracking accuracy and meanwhile reduce the particle sample size in the resampling stage.

Keywords:
Vehicle Tracking, Particle Filter, Genetic Algorithm, Resampling, Occlusion

1. INTRODUCTION

Recently, the number of on-road vehicles has been increased significantly and caused severe traffic congestion problem. Traffic congestion comprises complex dynamics problem and involves many traffic parameters that interact with one another [1]. Moreover, the on-road incidents that are created by the drivers are also elevated. Hence, a wide range of vehicle tracking applications such as traffic surveillance, intelligent driver assistant system (IDAS) and navigation system have incited the researchers to track vehicle with microscopic modelling.

Vehicle tracking system consists of hardware and software. Hardware is the device attached to the vehicle or installed on road to obtain the input information. Software is used to process the input information extracted from the hardware. Generally, hardware implemented in the vehicle tracking system can be categorized into two types, which are active sensors and passive sensors [2]. The active sensors measure the distance through the travel time of a signal emitted by the sensors and reflected from the nearby vehicle. However, the active sensors have drawbacks of low spatial resolution and slow scanning speed. When a huge amount of vehicles are moving simultaneously, the active sensors often obtain wrong signal [3]. On the contrary, the passive sensors such as video cameras can provide a wide range of information to characterize the vehicle. For example, the vehicle features such as colour, edge and shape can be obtained by extracting the information from the video camera via image processing techniques. Due to the low cost of passive sensors, they are more cost-efficient to be implemented in vehicle tracking system as compared to the active sensors.

In many countries, video camera has been mounted on the pole near to roadside for capturing the traffic scene. As the height of pole is constrained, the camera has the low angle view. Hence, occlusion might be frequently occurred. The traffic congestion worsens the occlusion incidents. Vehicle tracking in occlusion scene is a challenging task, because the features used to characterize the target vehicle would influenced by the obstacles, which creates the nonlinear and non-Gaussian situations. Thus, the complexities and difficulties caused by the occlusion incidents become the research's driving force to develop an effective and efficient vehicle tracking algorithm.

Particle filter has been proven as an approach to deal with nonlinear and non-Gaussian situations. Particle filter has been chosen in this study to track the vehicles under various occlusion incidents. Particle filter has faced the particle degeneracy problem, as the variance of the importance weights increases thereby the algorithm failed to evade the weight of degradation. Formerly, particle degeneracy can be mitigated by using huge sample particles or resampling approach. Although the resampling approach is more efficient to solve particle degeneracy compared with the implementation of huge sample particles, the conventional resampling technique such as sampling important resampling (SIR) has created another practical problem known as the sample impoverishment. Therefore, an improved particle filter with genetic algorithm resampling technique has been implemented in this study to track the target vehicle under occlusion incidents. The experimental results show that genetic algorithm based particle filter resampling can estimate the position of the target vehicle more accurately.

2. LITERATURE REVIEW

Vehicle tracking is usually performed in the context where the higher level of applications requires the vehicle position in every consecutive frame. In general, vehicle tracking is a challenging assignment within the field of machine vision. The difficulties in tracking vehicles may arise due to abrupt vehicle motion, occlusion incidents and changing appearance of patterns for both the vehicles and environment.

In the past, different tracking approaches have been implemented such as Kalman filter, optical flow, Markov Chain Monte Carlo, and particle filter. In fact, each tracking approach has its own strengths and weaknesses. For instance, in research [4], the researchers stated that Kalman filter is performed as an estimator that estimates and corrects the state of linear process. Kalman filter is normally used to solve the problem with linear situations because it is a framework that estimates the target state and updates the state estimation directly based on the obtained measurement. However, Kalman filter will face the difficulties when dealing with nonlinear situations. According to research [5], an extended version of Kalman filter will be required in order to solve nonlinear situations. The extended Kalman filter is implemented to change the estimated current measurement from nonlinear to linear. Nevertheless, when the nonlinearity is inaccurately approximated, the estimated tracking results for the extended version of Kalman filter will be diverged and affected the tracking results.

Optical flow is another technique used to track the moving vehicle. It has been used due to its ability to reflect the motion field of the captured image. For instance, the optical flow has been used to detect and track the moving object in research [6]. However, optical flow method faces difficulties when the moving target became static. Optical flow failed to perform when overlapping incidents occurred. This is due to the difficulty level of separating the target object with the obstacles. When overlapping incidents occurred, the optical flow will identify the overlapped objects with the obstacles as a single blob and could easily track to the wrong target.

Markov Chain Monte Carlo is also a common technique that is widely implemented for vehicle tracking purpose. According to research [7], the difficulty of Markov Chain Monte Carlo is how to determine the appropriate amount of sample size for performing good quality estimation. For instance, with the huge number of samples, the complexity of computation will become heavy and lead to unnecessary processing time. Meanwhile, least amount of sample could lead to inadequate tracking results. Although the developed algorithm was able to track the moving target vehicle with adaptive sample size, the algorithm is still unsteady and consists of tracking errors in moving, overlapped target vehicle at the captured frame.

In this study, particle filter has been chosen to track the target vehicle under various occlusion incidents. Normally, overlapping situations will create state uncertainties and lead to non-linear situations. According to research [8, 9], particle filter is a powerful and promising technique that can overcome the non-linear situations. By referring to research [10], the conventional particle filter will face particle degeneracy throughout the tracking process. Particle degeneracy problems occurred when tracking process undergoes several iterations and blocks the further improvement of the tracking algorithm. Research [11] stated that the particle degeneracy problem can be resolved either by implementing a huge amount of sample particles or resampling the particles that were eliminated. Since huge amount of initialize sample particles is always unfeasible due to the computational complexity, resampling the particles becomes the most appropriate solution to overcome the particle degeneracy problem [12].

In research [11, 13], the tracking algorithm was developed by using colour feature. From the results, the tracking algorithm with colour feature can track the target accurately under various occlusion situations but the tracking algorithm has difficulty when the background colour is similar to the colour of the target. In research [14], it had presented a rear view vehicle detection algorithm based on edge feature. The developed algorithm failed to respond when the vehicle edges are unclear or during the occurrence of occlusions. Hence, in order to track the vehicle robustly or to differentiate the target vehicle with the obstacles, a fusion of multiple features will be required because it can provide more information to describe the target vehicle [15]. In research [16], it showed that the tracking algorithm with fusion of multiple features will provide better tracking results even when the target vehicle is undergoing partially and fully occlusion.

3. PARTICLE FILTER FRAMEWORK

Particle filter is also known as sequential Monte Carlo algorithm. It is a mainstream tracking approach used to represent the propagation conditional density distribution when the observation probability density distributions are in nonlinear and non-Gaussian situations throughout the vehicle tracking process. Moreover, it utilizes the sequential estimation of the probability distributions.

The main idea of particle filter is to predict the posterior distribution based on a finite set of random weighted sample particles, N_p. Each weighted particle is drawn to represent the estimated state of the target vehicle based on the posterior distribution as shown in Eq.(1).

$$S_t^i = \left\{x_t^i, w_t^i\right\}_{i=1,2,3,....,N_p} \tag{1}$$

where, x_t^i denotes the state of the target vehicle and w_t^i denotes the weight that associated to the particle. In this study, the weight that associated to the particles will be limited from zero to one, $w_t^i \in [0,1]$. After each particles are associated with the weight, the weight for all the particles can be normalized and summed up to one as shown in Eq.(2).

$$\sum_{i=1}^{N_p} w_i = 1 \tag{2}$$

In general, particle filter approach works out based on three important stages which are the prediction stage, measurement stage and resampling stage. In the prediction stage, the transition state of the vehicle model will be generated randomly and it will be represented by a set of particles. In the measurement stage, the particles will assign weight based on the calculation of the features likelihood. The more similar the reference features with the target features, the more the particles will be assigned with heavy weight. On the other hand, if the likelihood computed is small, the particles will be assigned with low weight. Lastly, in the resampling stage, the low weight particles will be regenerated in order to avoid particle degeneracy.

In this study, particle filter has been developed to track a single vehicle that undergoes various occlusion incidents. The occlusion incidents consist of dynamic changes which will cause the posterior probability density function $p(X_t \mid Z_t)$ and the observation probability density function $p(Z_t \mid X_t)$ computed in the particle filter algorithm are often nonlinear and non-Gaussian. From the posterior probability density function and observation probability density function, X_t denotes the state space of the vehicle being tracked whereas Z_t denotes all the estimation state space.

3.1 PREDICTION STAGE

Prediction stage in the particle filter is the primary stage that initiates the sample particles and randomly estimates the position of the target vehicle. Each particle represents the estimated position of the target vehicle individually. Hence, the tracking accuracy of the algorithm that is initialized with huge sample particles can be improved because of the higher probability to predict the target vehicle real position. However, implementing huge amount of sample particles will cause heavy computation. Thus, the prior probability density function can be computed based on Eq.(3), which is the determined prior probability density function, then the posterior probability density function can be calculated through the updated step by using the Bayes' rule as defined in Eq.(4). After the position of the target vehicle is predicted, the particle filter will move on to the measurement stage in order to compute the weight for each particle.

$$p(X_t \mid Z_{1:t-1}) = \int p(X_t \mid X_{t-1})\, p(X_{t-1} \mid Z_{1:t-1})\, dX_{t-1} \qquad (3)$$

$$p(X_t \mid Z_{1:t}) = \frac{p(Z_t \mid X_t)\, p(X_t \mid Z_{1:t-1})}{p(Z_t \mid Z_{1:t-1})} \qquad (4)$$

3.2 MEASUREMENT STAGE

After each particle has been assigned with the estimated position of the target vehicle, the features of the vehicle will be extracted based on the estimated position. Various features such as colour, edge, shape or texture can be used to characterize the target vehicle. In this study, colour and shape features will be used to compute the likelihood of the target vehicle. Based on the extracted target vehicle features, the weight of each particle is determined by calculating the probability of likelihood.

The colour likelihood can be computed by using Bhattacharyya distance, b_{dist} [17]. Bhattacharyya distance is a famous method that correlates the images using colour histogram. In this study, the HSV colour histogram will be used to represent the colour feature of the vehicle. The Bhattacharyya distance will be used to define a normalized distance among the colour histogram of the target vehicle and the colour histogram of the reference vehicle. Normally, Bhattacharyya coefficient is used to measure the continuous probability distribution, as defined in Eq.(5).

$$\rho[p,q] = \int \sqrt{p_u q_u}\, du \qquad (5)$$

where, p_u represents the colour histogram of the target vehicle whereas q_u represents the colour histogram of the reference vehicle.

Since the colour histogram of the vehicle is formed in discrete density arrangement, the colour likelihood between the target vehicle and the reference vehicle can be measured through Eq.(6).

$$\rho[p,q] = \sum_{u=1}^{N_c} \sqrt{p_u q_u} \qquad (6)$$

After obtaining the Bhattacharyya coefficient, it will be transformed to Bhattacharyya distance by using the Eq.(7).

$$b_{dist} = \sqrt{1 - \rho[p,q]} \qquad (7)$$

Based on the computed Bhattacharyya distance, the colour likelihood can be calculated through Eq.(8).

$$\varphi_c = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{b_{dist}^2}{2\sigma^2}} \qquad (8)$$

The parameters, φ_c and σ in Eq.(8) represent the weight determined from the colour likelihood and the adjustable standard deviation respectively.

Due to the structural outlook of a vehicle, the shape feature will be implemented in the tracking algorithm. The shape likelihood is determined by using Hausdorff distance, H_{dist} [18]. Hausdorff distance is a scalar measurement of the distance value between two sets of points. In practice, the two set of points can be obtained from the shape feature of the reference vehicle, A and the shape feature of the target vehicle, B as shown in Eq.(9) and Eq.(10).

$$A = \{a_i\}_{i=1,\ldots,N_c} \qquad (9)$$

$$B = \{b_i\}_{i=1,\ldots,N_c} \qquad (10)$$

From the two sets of points determined, the Hausdorff distance between set A and set B can be calculated by using Eq.(11).

$$H_{dist}(A,B) = \max[\, h(A,B), h(B,A)] \qquad (11)$$

where, $h(A, B)$ is the distance measure from points in set A to points in set B and $h(B, A)$ is the distance measure from points in set B to points in set A.

After obtaining the Hausdorff distance, the weight of the particles can be generated using Eq.(12).

$$\varphi_s = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{H_{dist}^2}{2\sigma^2}} \qquad (12)$$

The value of Bhattacharyya distance and Hausdorff distance will equal to one if the features of the target vehicle is same with the feature of the reference vehicle. Both the features likelihood will be fused together to provide a more accurate tracking for the particles' weight determination, as shown in Eq.(13).

$$w_t^i = \alpha(\varphi_c) + (1-\alpha)(\varphi_s) \qquad (13)$$

where, α is the weight constant. In this study, α = 0.5 which means 50% of the weight was computed by using colour likelihood and the remaining 50% of the weight was computed based on the shape feature.

The weight of particle will be updated through Eq.(14) when features likelihood is computed.

$$w_t^i = w_{t-1}^i \times \frac{p(Z_t \mid X_t)\, p(X_t \mid X_{t-1})}{q(X_t \mid X_{t-1}, Z_t)} \qquad (14)$$

After the weight to the particles is updated, it will be normalized, by using Eq.(15), before the predictive posterior density function is predicted. The weight of the particles is computed in a discrete form. The posterior density function can be represented by Eq.(16).

$$W_t^i = \frac{w_t^i}{\sum_{i=1}^{N_p} w_t^i} \tag{15}$$

$$p(X_t \mid Z_{1:t}) \approx \sum_{i=1}^{N_p} W_t^i \delta(X_t - X_t(i)) \tag{16}$$

When the predicted posterior density distribution is obtained, the particle filter will determine the final position of the target vehicle by calculating mean state of the vehicle by using Eq.(17).

$$E(X_t) = \frac{1}{N_p}\sum_{i=1}^{N_p} S_t^i \tag{17}$$

3.3 RESAMPLING STAGE

Particle filter faces an inherent problem which is particle degeneracy after a few iteration of tracking algorithm performed. The particle degeneracy occurred because one of the particles will experience negligible weight due to the increasing variance of the particles important weight in every consecutive processing frame. It can be concluded that particle degeneracy is unavoidable but it can be resolved through resampling approach.

In the past, the most common resampling approach was sampling important resampling (SIR) which is performed based on replacement basis. In the SIR approach, the low weight particles will be eliminated meanwhile the heavy weight particles will be preserved for the tracking purpose. This shows that a new set of heavy weights particles will be duplicated to replace the eliminated particles.

The occurrence of particle degeneracy can be determined by calculating the effective sample size, as shown in Eq.(18).

$$N_{eff} = \frac{N_p}{1 + Var(w_t^{*i})} \tag{18}$$

In Eq.(18), the weight w_t^{*i} denotes the true weight which can be calculated through Eq.(19).

$$w_t^{*i} = \frac{p(x_t^i \mid z_{1:t})}{q(x_t^i \mid x_{t-1}^i, z_t)} \tag{19}$$

However, the true weight of the particles is always hard to be computed and hence an estimation of effective sample size can be determined by using the normalized weight, w_t^i as shown in Eq.(20).

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} \left(W_t^i\right)^2} \tag{20}$$

If the computed estimation of effective sample size $\hat{N}_{eff}$ is small or $\hat{N}_{eff} < N_{thres}$, the particle degeneracy problem occurred and resampling stage is required to improve the prediction of the posterior density distribution. The process of estimation will be done when the stopping criteria was fulfilled. The particle filter with SIR resampling stage is illustrated in Table.1.

4. PROPOSED GENETIC ALGORITHM RESAMPLING

Although the SIR approach can be used to reduce the problem of particle degeneracy, it creates another practical problem which is the sample impoverishment. The sample impoverishment occurred because the particles with heavy weight are statistically being selected many times throughout the tracking process. Hence, the state estimated by the algorithm will contain many repeated position and it leads to the loss of diversity among the particles. After a few iterations, the estimated position might collapse to a single position which the computed likelihood is high and causes the algorithm unable to track the target vehicle continuously. By eliminating the particle diversity problem, the genetic algorithm will be implemented in the particle filter resampling stage. Genetic algorithm consists of selection, crossover and mutation stages. The developed genetic algorithm based particle filter resampling algorithm is illustrated in Table.2.

4.1 SELECTION STAGE

Selection stage is important for improving the quality of the population by selecting individuals with higher quality to generate offspring solutions, which are referring to the estimated position of the target vehicle. There are several methods to select the parents for crossover such as Roulette wheel selection, Boltzmann selection, tournament selection, rank selection and steady state selection. In this study, the rank selection will be used to rearrange the position of the vehicles based on the computed particles weight.

In the rank selection approach, the most heavy particles will be assigned with higher rank and meanwhile the light weight particles will be assigned with lower rank. After all the particles are being ranked accordingly, the algorithm will randomly select two particles as the parents. The parents here are referring to the predicted position of the target vehicle. Since the heavy weight particles are being assigned with higher rank, the chance of the heavy particles being selected as the parents will be higher as compared to the particles of lower ranks.

4.2 CROSSOVER STAGE

After the selection stage, genetic algorithm will undergo the crossover stage. Crossover is a process of taking more than one parent solutions and combining their characteristics to produce new offspring solutions. In literature, there are several types of crossover techniques such as one-point crossover, two-point crossover, uniform crossover, heuristic crossover and arithmetic crossover. In this study, arithmetic crossover will be selected as the particle filter resampling approach. The advantage of the arithmetic crossover is to produce a new solution which contains the characteristics of both parents. The offspring solution are determined based on Eq.(21) and Eq.(22).

$$C1 = P1 \times \alpha + P2 \times (1 - \alpha) \tag{21}$$

$$C2 = P2 \times \alpha + P1 \times (1 - \alpha) \tag{22}$$

where, α is the weight factor with a limit of zero to one, $P1$ and $P2$ are the parents and $C1$ and $C2$ are the offspring solution.

Table.1. The algorithm of particle filter with SIR resampling

1:	Initialize model features and sample size
2:	**FOR** FRAME = 1, 2,…,t
3:	**PREDICTION**:
4:	**FOR** i = 1, 2,…, N_p
5:	Draw predicted particles from prior dynamics
6:	Compute the features based on estimated position
7:	**END FOR**
8:	**MEASUREMENT & UPDATE**:
9:	Calculate the likelihood
10:	Compute the weight of the particle
11:	Normalize the weight, $w_t^i = w_t^i (\sum_{i=1}^{N_p} w_t^i)^{-1}$
12:	Calculate $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_t^i)^2}$
13:	$\hat{N}_{eff} = \begin{cases} < N_{thres} & \text{Resampling} \\ \geq N_{thres} & \text{Acceptance} \end{cases}$
14:	**RESAMPLING**:
15:	Eliminate low weight particles
16:	Repeat Step 3 to Step 13
17:	**END IF**
18:	**LOCALIZATION**:
19:	$(x, y) = E(X_t)$

Weight factor is used to determine the fraction or the percentage of the characteristic from the parents' solutions that will contribute to the offspring solutions. In this study, 0.7 was set as the weight factor which means that 70% of the characteristic of first parent and 30% of the characteristic of the second parent to form the first children solution and inversely for the second children solution. In the genetic algorithm resampling approach, all the low weight particles will be eliminated and replaced by the generated children solutions. By using the arithmetic crossover, the position of the target vehicle can be predicted accurately.

4.3 MUTATION STAGE

Mutation stage is a process to maintain the genetic diversity from one generation to the next generation. Mutation stage must be performed after the selection and crossover stage because of its ability as a final checking state to recover the information which might be lost during the selection and crossover process. Mutation is an important stage to prevent the population stagnating at the optimal position. In literature, the mutation rate is suggested to be set fairly low and it is defined by the user. This is to evade the loss of fit solutions which can affect the estimated solution. In this study, the mutation rate was set as 1%. If the mutation rate was hit during the tracking process, a new offspring solution will be generated by adding the determined state of the target vehicle with a random variable which is strictly limited to the values between zero to one.

Table.2. The algorithm of particle filter with genetic algorithm resampling

1:	Initialize model features and sample size
2:	**FOR** FRAME = 1, 2,…,t
3:	**PREDICTION:**
4:	**FOR** i = 1, 2,…, N_p
5:	Draw predicted particles from prior dynamics
6:	Compute the features based on estimated position
7:	**END FOR**
8:	**MEASUREMENT & UPDATE:**
9:	Calculate the likelihood
10:	Compute the weight of the particle
11:	Normalize the weight, $w_t^i = w_t^i (\sum_{i=1}^{N_p} w_t^i)^{-1}$
12:	Calculate $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_t^i)^2}$
13:	$\hat{N}_{eff} = \begin{cases} < N_{thres} & \text{Resampling} \\ \geq N_{thres} & \text{Acceptance} \end{cases}$
14:	**RESAMPLING:**
15:	Eliminate low weight particles
16:	Performed Rank Selection
17:	Arithmetic Crossover
18:	Generate mutation rate
19	**IF** mutation < 1%
20:	$X_t^i = X_t^i + rand(0,1)$
21:	**ELSE**
22:	$X_t^i = X_t^i$
23:	**END IF**
24:	**LOCALIZATION:**
25:	$(x, y) = E(X_t)$

5. RESULTS AND DISCUSSUIONS

In this section, the result of vehicle tracking using SIR (Fig.1) was compared to the results of vehicle tracking using genetic algorithm resampling (Fig.2). The initial amount of particles used for vehicle tracking was set as 200 particles. As shown in Fig.1 and Fig.2, the solid boundary box indicates the boundary of the vehicle meanwhile the cross icon represents the predicted position of the target vehicle. Since particle filter is considering the posterior density distribution, the estimated position of the target vehicle will be determined by calculating the mean value.

By referring to Fig.1 and Fig.2, the tracking process undergoes three stages which are: without occlusion, partially occlusion and fully occluded. When the target vehicle is free from occlusion as shown at Frame 5 in Fig.1 and Fig.2, the target vehicle has been accurately tracked by the particle filter with SIR resampling and the improved particle filter with genetic algorithm. This is due to the colour and shape features being extracted is easily computed into the likelihood. Besides, the information of the target vehicle is not influenced by the obstacle. Hence, heavy weight particles can be produced. The tracking efficiency for target vehicle which is free from occlusion will be always faster and easier.

Comparing Frame 21 at Fig.1 and Fig.2, the target vehicle is partially occluded by another moving vehicle. From the results, the particle filter with SIR resampling merely locates the target vehicle meanwhile the particle filter with genetic algorithm resampling is able to locate the target vehicle. During the partially occlusion, the features that are used to describe the target vehicle is affected by the obstacle vehicle. At this stage, resampling takes an important role to accurately locate the target vehicle. However, the SIR has duplicated the particles of heavy weight to replace the amount of the particles that has been eliminated. Thus, the estimated position is easily trapped in a single location which diverges from the real position of the target vehicle. In particle filter with genetic algorithm resampling, the eliminated particles will be replaced by the recombination of two estimated position. After recombination the particles, the likelihood will be recalculated. After a few iterations of recombination, the tracking will be more accurate due to the convergence of the estimated position to the real position of the target vehicle.

By referring to Frame 33 in Fig.1 and Fig.2, the target vehicle is fully occluded by the moving vehicle. At this moment, the information of the target vehicle is fully lost and is affected by the moving vehicle. From the results, the particle filter with SIR is unable to locate the target vehicle and meanwhile genetic algorithm based particle filter resampling can still predict the position of the target vehicle.

At Frame 41 in Fig.1 and Fig.2, the target vehicle reappeared from a full and partial occlusion. From the results, the particle filter with genetic algorithm has accurately predicted the positions of the target vehicle. However, the particle filter with SIR merely tracks the target vehicle. This is because the particle filter with SIR hardly obtained information of the target vehicle.

Comparing Frame 49 in Fig.1, and Fig.2, the target vehicle reappears after critical full occlusion. The tracking algorithm will try to retain the information of the target vehicle. The particle filter with SIR requires more computational time to track back the target vehicle because the estimated position in SIR is based on the random generated value with a normal distribution. However, particle filter with genetic algorithm will try to recombine the good particles in order to predict the new position of the target vehicle. Based on this recombination and weight recalculation process, the real position of the target vehicle can be obtained.

(a) Frame 5

(b) Frame 21

(c) Frame 33

(d) Frame 41

(e) Frame 49

Fig.1. Result of vehicle tracking via sampling important resampling (SIR)

(a) Frame 5

(b) Frame 21

(c) Frame 33

(d) Frame 41

(e) Frame 49

Fig.2. Result of vehicle tracking via genetic algorithm resampling

In order to investigate the tracking performance of the particle filter with SIR and genetic algorithm, the root mean square error (RMSE) is computed. The RMSE of the particle filter with SIR and the developed genetic algorithm is plotted in

Fig.3. The lower value of RMSE shows a better estimation of the positions of the target vehicle.

From the results shown in Fig.3, it is clearly shown that the RMSE for the particle filter with genetic algorithm resampling is much lower than the SIR approach. Thus, it can be concluded that the particle filter with genetic algorithm resampling have a better estimation results than the particle filter with SIR stage throughout the tracking process.

The comparison of the resampling count and particle size that are required in the resampling stage between SIR and genetic algorithm resampling is plotted in Fig.4. From the results, the number of particles that are required for the genetic algorithm resampling is much lesser than the SIR approach throughout the whole vehicle tracking process. This is because genetic algorithm has the ability to converge the estimated position to the real position of the target vehicle through the recombination process. However, in SIR, the position of the target vehicle is generated based on random prediction. Due to the better estimation results obtained through genetic algorithm, the particles that required resampling are greatly reduced. For instance, when the target vehicle is partially occluded during Frame 21, the number of particles that are required for resampling in SIR approach is 1340 and the particles that are required for genetic algorithm resampling is 775. Hence, the improved particle filter with genetic algorithm resampling has reduced around 42.2% of the particles amount compared to the SIR approach. Besides, the better estimation results obtained by the genetic algorithm also reduce the number of resampling iteration. Hence, it can be concluded that the improved resampling provides a better tracking accuracy under various occlusion incidents with the least amount of particles implemented.

6. CONCLUSION

As discussed earlier, particle degeneracy could diminish the accuracy of the particle filter approach. Thus, resampling stage plays an important part in the particle filter algorithm in order to track vehicle accurately under various occlusion incidents. The most common resampling approach is SIR. However, it will face the sample impoverishment especially during the situations that contains high uncertainty. The implementation of genetic algorithm in the particle filter resampling approach is capable to alleviate the tracking difficulties under various occlusion situations by recombining and recalculating the likelihood of the particles. From the results, the performance and robustness of the developed resampling algorithm is promising.

Based on the better estimation of the genetic algorithm, it can be concluded that the developed vehicle tracking algorithm has improved the accuracy of the tracking results. The number of particles that are required for resampling has also been reduced compared to the SIR approach.

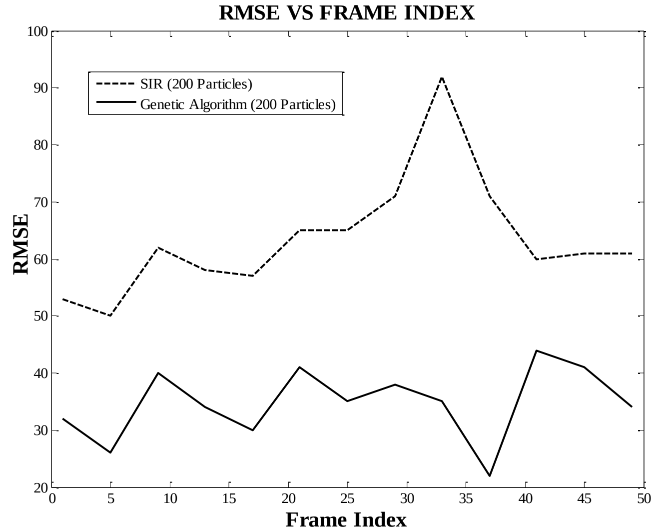

Fig.3. Graph of RMSE vs frame index for SIR resampling and genetic algorithm resampling

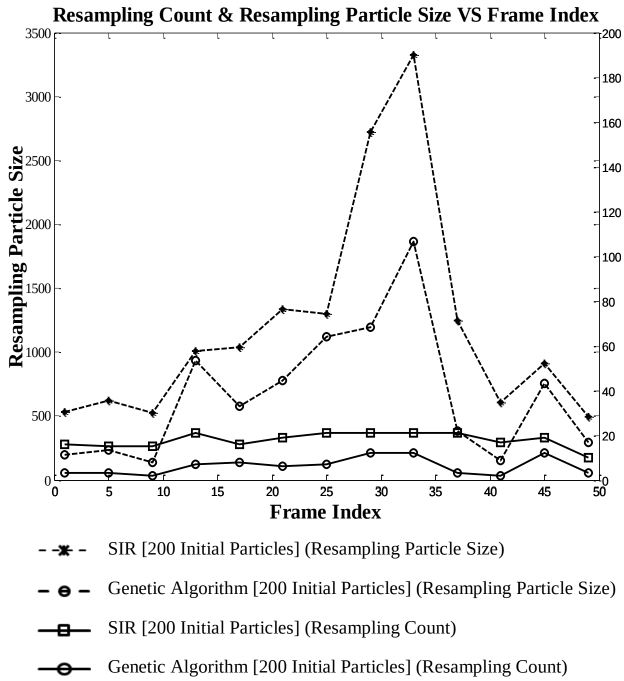

Fig.4. Resampling count and resampling particle size versus frame index for sampling important resampling (SIR) and genetic algorithm based particle filter resampling

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial assistance from Ministry of Higher Education of Malaysia (MOHE) under Exploratory Research Grant Scheme (ERGS), grant no. ERG0021-TK-1/2012, Universiti Malaysia Sabah (UMS) under UMS Research Grant Scheme (SGPUMS), grant 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] G. Papageorgiou, P. Damianou, A. Pitsillides, T. Aphamis, D. Charalambous and P. Ioannou, "Modelling and Simulation of Transportation Systems: a Scenario Planning Approach", *Journal for Control, Measurement, Electronics, Computing and Communications*, Vol. 50, No. (1-2), pp. 39 - 50, 2009.

[2] T. Rabie, A. Shalaby, B. Abdulhai and A. El-Rabbany, "Mobile Vision-based Vehicle Tracking and Traffic Control", *Proceedings of the IEEE Fifth International Conference on Intelligent Transportation Systems*, pp. 13 - 18, 2002.

[3] Z. Sun, G. Bebis and R. Miller, "On-Road Vehicle Detection: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 5, pp. 694 - 711, 2006.

[4] X. Li, K. Wang, W. Wang and Y. Li, "A Multiple Object Tracking Method Using Kalman Filter", *Proceedings of the IEEE International Conference on Information and Automation*, pp. 1862 - 1866, 2012.

[5] M. Nabaee, A. Pooyafard and A. Olfat, "Enhanced Object Tracking with Received Signal Strength using Kalman Filter in Sensor Networks", *Proceedings of International Symposium on Telecommunications*, pp. 318 - 323, 2008.

[6] Y. Fang and B. Dai, "An Improved Moving Target Detecting and Tracking Based on Optical Flow Technique and Kalman Filter", *Proceedings of the Fourth International Conference on Computer Science and Education*, pp. 1197 - 1202, 2009.

[7] 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", *Proceedings of International Conference on Computer Applications and Industrial Electronics*, pp. 50 - 55, 2011.

[8] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filter for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transaction on Signal Processing*, Vol. 50, No. 2, pp. 174 - 188, 2002,

[9] H.P. Liu, F.C. Sun, L.P. Yu and K.Z. He, "Vehicle Tracking using Stochastic Fusion-based Particle Filter", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2735 - 2740, 2007.

[10] H. Li, Y. Wu and H. Lu, "Visual Tracking Using Particle Filters with Gaussian Process Regression", *Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, pp. 261 - 270, 2009.

[11] W.L. Khong, W.Y. Kow, L. Angeline, I. Saad and K.T.K. Teo, "Overlapped Vehicle Tracking via Enhancement of Particle Filter with Adaptive Resampling Algorithm", *International Journal of Simulation, Systems, Science and Technology*, Vol. 12, No. 3, pp. 44 - 51, 2011.

[12] X. Fu and Y. Jia, "An Improvement on Resampling Algorithm of Particle Filter", *IEEE Transaction on Signal Processing*, Vol. 58, No.10, pp. 5414 - 5420, 2010,

[13] K. Nummiaro, E. Koller-meier and L.V. Gool, "Colour Features for Tracking Non-Rigid Objects", *Special Issue on Video Surveillance Chinese Journal of Automation*, Vol. 29, pp. 345 - 355, 2003.

[14] M. Boumediene, A. Quamri and M. Keche, "Vehicle Detection Algorithm Based on Horizontal/Vertical Edges", *7th International Workshop on Systems, Signal Processing and Their Applications*, pp. 396 - 399, 2011.

[15] H. Sheng, Q. Wei, C. Li and Z. Xiong, "Robust Multiple-vehicle Tracking via Adaptive Integration of Multiple Visual Features", *Journal on Image and Video Processing*, Vol. 2, pp. 1-19, 2012.

[16] 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", *Proceedings of International Conference on Control System, Computing and Engineering*, pp. 460 - 465, 2011.

[17] M.S. Khalid, M.U. Ilyas, M.S. Sarfaraz and M.A. Ajaz, "Bhattacharyya Coefficient in Correlation of Gray-Scale Objects", *Journal of Multimedia*, Vol. 1 No. 1, pp. 56 - 61, 2006.

[18] S.C. Park, S.H. Lim, B.K. Sin, and S.W. Lee, "Tracking non-rigid Objects using Probabilistic Hausdorff Distance Matching", *Journal of Pattern Recognition*, Vol. 38, No. 12, pp. 2373 - 2384, 2005.