Vehicle Tracking Using Particle Filter for Parking Management System

Kenneth Tze Kin Teo, Renee Ka Yin Chin, N.S.V. Kameswara Rao, Farrah Wong, Wei Leong Khong
Modelling, Simulation & Computing Laboratory, Material & Mineral Research Unit
Faculty of Engineering, Universiti Malaysia Sabah
Kota Kinabalu, Malaysia
E-mail: msclab@ums.edu.my, ttkteo@ieee.org

Abstract—Increment of on-road vehicles has urged public venues to provide visitors with a larger area of parking space. As the parking area grew larger for example in a hyper mall, a well-organized parking management system is necessary to assist drivers in locating parking position. Besides, it can also help the management team to monitor vehicle flow in the parking lot. Vehicle tracking plays an important role to the parking management system, as accurate tracking result will lead to a more efficient management system. Among commercially available sensors, video sensor has been commonly deployed in the parking area due to its ability in obtaining a wide range of vehicle information. However, images captured using video sensors are limited under situations where vehicles are undergoing occlusion and maneuvering incidents. This will cause tracking error therefore affecting the performance of the parking management system. Particle filter has been proven as one of the promising techniques to track vehicle under disturbances. Therefore, particle filter is proposed to track vehicle under occlusion and maneuvering incidents in this study. Experimental results show that the particle filter is able to track a target vehicle under different disturbances.

Keywords – particle filter; vehicle tracking; occlusion; parking lot; video sensor

I. INTRODUCTION

In this day and age, cars are the dominant mode of transportation in most countries. Thus, ensuring enough vacant space in the parking lot becomes important due to the number of vehicles in an urban area has increased over the recent years. The newer public venues with high flow rate of visitors such as shopping malls, food courts, parks and government administration buildings are constructed with more vehicle parking spaces compared to older venues. With such a huge parking area, a well-equipped and organized parking management system is always essential to monitor and assist the drivers in localizing their vehicles.

Typically, an efficient vehicle tracking system consists of hardware and software. Hardware refers to the device that is installed to obtain the input information. Meanwhile, software is used to analyze the input information obtained from the hardware. The hardware implemented in the vehicle tracking can be categorized into active sensors and passive sensors [1]. Active sensors determine the signals transmitted by the sensor that was reflected or scattered by the surface of an object. For instance, some parking management implements infrared or ultrasonic sensors to detect the occupancy of the parking lot. If the parking lot is vacant, the light-emitting diode (LED) will display green color. Otherwise, it will show a red color. Based on the color displayed by the LED, it will reduce the driver’s searching time in approaching the nearest parking bay. However, the system is cost consuming since it needs to deploy sensors at each parking bay. Besides, the LED emitting the wrong signal due to the unknown object blocking the beam from the sensors will compromise the system accuracy. Thus, it is more effective to detect the occurrence of the vehicle in a parking bay, than to track and localize the vehicle.

On the other hand, passive sensor such as video cameras provides fruitful information regarding the vehicles, which is extractable by software. Since most of the existing parking lots have implemented video sensor for surveillance purposes in order to prevent criminal cases, the video sensors can be used to track the vehicle entry for assisting drivers and monitoring vehicle flow in the parking lot. However, the video information obtained based on the vehicle’s observable outlook will encounter difficulties when the target vehicle undergoing occlusion and maneuvering incidents [2]. The observable information of the target vehicle will be lost or changed during those incidents, resulting in the tracking difficulties [3]. Besides, the dynamic changes of the vehicle flow caused by occlusion and maneuvering incidents will also create a non-linear and non-Gaussian situation. Hence, particle filter (PF) has been proposed in this study to overcome the vehicle undergoing occlusion and maneuvering due to its ability in dealing with the non-linear and non-Gaussian situations.

PF is a set of posterior density estimation algorithms that estimate the posterior density of the state-space by directly implementing the Bayesian recursion equations. Although PF is able to overcome dynamic changes caused by the vehicle flow, it has to deal with particle degeneracy problem. Particle degeneracy happens when the variance of the importance particles weight continues to increase until the algorithm is unable to avoid the degradation of the particles weight. In order to solve the particle degeneracy problem, particle resampling is necessary [4]. Conventional method of resampling might result in particle impoverishment problem by duplicating only heavy weight particles. Thus, a genetic algorithm (GA) resampling approach will be implemented in this study to track the target vehicle more efficiently and effectively.
II. OVERVIEW OF PARTICLE FILTER

The purpose of vehicle tracking is to accomplish the applications goal using vehicle position for surveillance, localization and modelling. The development of video surveillance infrastructure such as the production of high-end computers, the availability of high quality video cameras, and the escalating need for video analysis have incited the interest in visual tracking [5]. Vehicle tracking using video cameras can estimate the position of the interested vehicle correctly in every consecutive frame, but complications may arise because of the vehicle motion, occlusion and maneuvering incidents. Throughout the tracking techniques using information extracted from video cameras, PF can track vehicle under occlusion and maneuvering incidents effectively by estimating the vehicle location through Bayesian probabilistic framework [6]. Unlike other tracking algorithms, PF exploited estimation does not involve linearization with the current estimates. PF approximates the desire posterior distributions by discrete random measure, which is known as the particles with associated weights, or recognized as probability mass [7]. PF is also named as sequential Monte Carlo (SMC) because it implements a recursive Bayesian filter with Monte Carlo simulations. It is an approach that utilizes the sequential estimation of relevant probability distributions using important sampling (IS) techniques. The estimation of distribution will be in discrete random measure [8]. The key idea is to represent the posterior distribution based on a finite set of random weighted particles.

PF is used to represent the propagation conditional density distributions (CDD) when the observation probability density distributions (PDD) involved in the vehicle tracking process are non-linear and non-Gaussian [9]. In addition, PF technique estimates the current state of the target vehicle based on the previous observation states. In video tracking, the observation state is referred to the features that can be used to represent the target vehicle such as color, shape, edge, texture or any combinations of features [10]. In this study, the combination of color and shape features has been chosen as the features to represent the target vehicle’s model.

In estimating the position of target vehicle based on Bayesian probabilistic framework, the posterior probability density function (PDF) and observation PDF can be written as \( p(X_t | Z_t) \) and \( p(Z_t | X_t) \) respectively, where \( X_t \) is the state vector representing the position of the tracked vehicle and \( Z_t \) is the state vector representing all the state space that are being estimated.

As mentioned earlier, PF is used to approximate the posterior distribution of the target vehicle based on a finite set of random weighted particles, \( N_p \). The state estimation of the target vehicle can be written as shown in (1) where \( x_i^t \) represents the location of the target vehicle and \( w_i^t \) is the weight that is allocated to each particle (limited from zero to one). Hence the whole set of particles can be normalized to one.

\[
S_i^t = \{x_i^t, w_i^t\}_{i=1,2,3,...N_p}
\]  

The framework of PF can be summarized in three important steps, which are prediction, measurement and resampling [11]. Fig. 1 shows a general framework of PF. In the prediction step, a set of random particles will be distributed to represent the state transition of the vehicle model based on a Bayesian distribution. In the measurement step, the generated particles will be assigned with weight according to the features’ likelihood. The resampling step is used to avoid particle degeneracy by eliminating and replacing the low weight particles with newly regenerated particles.

A. Prediction Step

Prediction step is the initial step that assigns the particles using Bayesian distribution. As mentioned earlier, the particles represent the estimated target vehicle’s posterior position. The predictions of the posterior position begin when the prior position of the target vehicle is being identified. Based on the prior position, a set of random particles will be distributed to estimate the location of the target vehicle. In terms of tracking accuracy, more initial particles distributed will lead to more promising tracking results or vice versa. However, implementing a huge amount of particles will cause higher computational cost. Since resampling step can take place to regenerate and replace the particles with low weightage, the number of initial particles can be reduced to an optimal value.

The prior position of the target vehicle can be obtained by segmenting the target vehicle from the image sequences. After the prior PDF is obtained, the PF algorithm can compute the posterior PDF by using the Bayer’s rule as shown in (2).

\[
p(X_t | Z_{1:t}) = \frac{p(Z_t | X_t)p(X_t | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})}
\]  

![Figure 1. The general framework of particle filter.](image-url)
B. Measurement Step

Measurement step in the PF algorithm computes the weight of the particles according to the likelihood of shape and color features. The likelihood can be measured by comparing the model features of the target vehicle with the feature extracted from the estimated posterior position. Since vehicle has a solid structural geometry outlook, the shape and color features are used to represent the vehicle in this study. The shape feature likelihood can be measured using Hausdorff distance, \( H_{dist} \) as shown in (3).

\[
\varphi_s = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{H_{dist}^2}{2\sigma^2}} \quad \text{(3)}
\]

The Hausdorff distance has a limit from zero to one. Smaller Hausdorff distance value means the feature likelihood extracted is more similar to the model feature of the target vehicle and vice versa [12]. On the other hand, the color feature likelihood is computed using Bhattacharyya distance, \( B_{dist} \), as shown in (4) [13]. It is identical to the Hausdorff distance, where smaller distance value represents similar features. \( \sigma \) in (3) and (4) is an adjustable standard deviation, which is chosen according to the vehicle tracking cases.

\[
\varphi_c = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{B_{dist}^2}{2\sigma^2}} \quad \text{(4)}
\]

In this study, a fusion of color and shape features has been proposed to compensate the weakness of using only either one of them. Although the shape feature has the ability to track vehicle with a fixed geometry structure, it is weak in dealing with the occlusion incidents. The color feature has the ability to deal with partial occlusion incidents, but it will suffer when the color of the target vehicle is similar to the environment background or other objects. The weightage of the fusion of shape and color features can be obtained through (5), where it has been set to 0.5 in this study.

\[
w_i^f = \alpha(\varphi_s) + (1-\alpha)(\varphi_c) \quad \text{(5)}
\]

After computing the features likelihood, the weight of the particles will be updated accordingly. The set of particles will undergo normalization through (6).

\[
W_i = \frac{w_i^f}{\sum w_i} \quad \text{(6)}
\]

Based on the particles’ weights assigned, the posterior PDF can be obtained with (7) to predict the position of the target vehicle by calculating the mean value of the predicted state.

\[
p(X_t | Z_t) = \sum_{i=1}^{N} W_i \delta(X_t - X_t(i)) \quad \text{(7)}
\]

C. Resampling Step

The resampling step is the most important step in the PF algorithm. It is used to avoid the particle degeneracy problem. Particle degeneracy only becomes apparent after a few iteration of tracking algorithm. This happens because particles experiencing low weight will deal with continuously increasing particle weight variance in every consecutive processing frame, unless the correction step is executed.

In order to determine the occurrence of the particle degeneracy problem, the effective sample size for every iteration needs to be calculated. The effective sample size can be obtained through (8).

\[
\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_i^f)^2} \quad \text{(8)}
\]

If the estimation of effective sample size value is less than the threshold value, the particle degeneracy problem is said to have occurred. Hence, particles need to be resampled in order to improve the prediction of the target vehicle’s posterior position.

III. GENETIC ALGORITHM BASED RESAMPLING APPROACH

The resampling step can assist in producing better posterior PDF. However, without an appropriate resampling method, it will cause issues to the tracking algorithm. For instance, the famous resampling technique, sampling importance resampling (SIR) causes the particle impoverishment problem. SIR resampling takes place by duplicating the heavy weight particles to replace the eliminated low weightage particles. In this case, the particles generated will contain many-repeated posterior position. After a few iterations, all particles might collapse into a single position with the heaviest weightage. Thus, the tracking algorithm enables to track the target vehicle continuously especially when there are disturbances. In order to control the diversity of the particles, GA is implemented in this study to increase the convergence rate of estimating the posterior distribution.

GA is known to be a powerful algorithm for searching through large and complex solution space consisting of multiple local minima. Unlike the deterministic methods, GA works based on randomization [14]. The operation of GA is based on the entire population of positions, instead of searching from a single position. This contributes to the robustness of GA and improves the probability to reach the global optimum, thus reducing the chance of being trapped in a local stationary position. Hence, the characteristics of GA have proven that it has the advantage over the convergence of the posterior PDF estimation. In this study, the fitness function of GA is based on the weight of the particles,
whereas the stopping criterion for GA is based on the effective sample size as shown in (8).

GA can be categorized into three stages, which are selection stage, crossover stage and mutation stage. Selection stage is important in choosing the quality particles to regenerate the children solutions. There are many ways to select the parent particles for crossover. In this study, rank selection is being selected, so that all the particles will be ranked in descending order according to the weightage. After all particles are ranked, the algorithm will randomly select two particles as the parent to generate two new posterior positions. Since particles with heavier weights will be assigned the higher rank, the probabilities for the heavy weight particles to be selected is higher as compared to those particles with lower weight.

After the parents’ selection, the crossover takes place by combining the properties of the parents [15]. In this study, arithmetic crossover will be implemented since it can produce a new solution that contains the characteristics of both parents. The arithmetic crossover is formulated by using (9) and (10).

\[ C_1 = P_1 \times \alpha + P_2 \times (1 - \alpha) \]  
\[ C_2 = P_2 \times \alpha + P_1 \times (1 - \alpha) \]

where \( P_1 \) and \( P_2 \) are the parents selected to perform crossover, \( \alpha \) is the weight factor with the range from 0 to 1, and \( C_1 \) and \( C_2 \) are the children solutions generated from the crossover process.

Weight factor of 0.7 is chosen in this study. It represents that 70% of the first parent particle characteristics will combine with 30% of the second parent particle characteristics to produce a new particle, inversely for the second new particle being generated. Implementing arithmetic crossover in the algorithm can produce children particles that preserve most of the parent characteristics without affecting the speed of convergence. Hence, a more accurate tracking result can be computed.

After the crossover stage, the children particles will undergo mutation stage before being accepted by the tracking algorithm. Acting as a final checkpoint to recover the usable information that might be lost during the selection and crossover stages, the mutation stage prevents the population generated from being stagnated at the local optimal position. Moreover, the mutation stage will only be activated when the children particles hit the defined mutation rate. Although mutation stage can prevent the particles from trapping at local maxima, the rate of mutation needs to be fairly low to avoid the loss of healthy particles, which eventually affects the convergence rate. In this study, the mutation rate is set to be 1%.

IV. RESULT AND DISCUSSION

In this study, the vehicle tracking result using GA resampling approach is shown in Fig. 2 and Fig. 3. Fig. 2 shows the result of the target vehicle undergoing occlusion incidents. Meanwhile, Fig. 3 shows the tracking result whereby the target vehicle is undergoing maneuvering in the parking lot. The video was captured in a parking lot at a shopping mall and the frame rate has been set to 30 frames per second. The initial amount of particles for vehicle tracking was set to be 200 particles. As shown in Fig. 2 and Fig. 3, the solid boundary box denotes the border of the vehicle being tracked. Meanwhile the cross icon indicates the predicted posterior position of the target vehicle. The predicted position of the target vehicle is estimated based on the mean value calculated from the posterior PDF.

From Fig. 2, the vehicle tracking process consists of three stages, which are the target vehicle without being occluded, partially occluded and fully occluded. At Frame 48, the target vehicle is moving without any occlusion. From the result, the algorithm can locate and predict the location of the target vehicle. When it comes to Frame 72 and Frame 88, the target vehicle is partially occluded by another moving vehicle from the opposite direction. Normally, the conventional tracking algorithm will track the wrong target at this moment because the features of the target vehicle will be influenced by the other vehicle, which is nearer to the video camera. However, the proposed GA based PF algorithm is able to track the target vehicle accurately.

Figure 2. Result of vehicle tracking under occlusion incidents.
Another crucial moment takes place when the target vehicle reappeared after being occluded by the moving vehicle as shown in Frame 104 and Frame 120. From the results, the tracking algorithm is able to robustly track the target vehicle immediately after the occlusion is completed. Besides, the algorithm can also recover the information of the target vehicle and resume the tracking process without any delay. Hence, it shows that GA based PF algorithm is suitable to be implemented in the parking lot that consists of two-way directions.

As shown in Fig. 3, the target vehicle undergoes maneuvering and partially occluded by the column structures of the building. Generally, tracking a target vehicle with maneuvering is a challenging task due to the outlook features of the target vehicle that is always changing. In order to robustly track the target vehicle, the features must be updated to the developed algorithm before estimation of posterior position is performed. Throughout the vehicle tracking results shown in Fig. 3, the PF technique has the ability to deal with the maneuvering incidents.

The performance of the vehicle tracking result can be calculated using root mean square error (RMSE) as shown in (11), where $x_e$ and $y_e$ are the estimated coordinate for the target vehicle in unit of pixel. Meanwhile, $x_o$ and $y_o$ are the original coordinate of the vehicle in unit of pixel.

$$\text{RMSE} = \sqrt{(x_e - x_o)^2 + (y_e - y_o)^2} \quad (11)$$

Fig. 4 shows the RMSE versus frame index for vehicle tracking under occlusion incident. RMSE calculates the difference between the estimated posterior positions with the original position of the target vehicle. From the result, the RMSE from Frame 72 to Frame 112 is higher because the obstacle vehicle influences the target vehicle’s features when undergoing occlusion. However, the RMSE is considered acceptable because the highest difference is only 45 pixels away from the original position. When the target vehicle reappears from the occlusion, the algorithm is able to track the target vehicle with just 6 pixels away from the original position. It can be concluded that the proposed GA based PF algorithm is able to track the target vehicle under occlusion incidents.

Fig. 5 shows the tracking performance for target vehicle undergoing maneuvering and partial occlusion by the column structures of the building. By comparing Fig. 4 and Fig. 5, the average RMSE from Fig. 5 is higher than Fig. 4 because the target vehicle undergoes scale invariant and shape changing when it is maneuvering. This increases the difficulties in tracking the target vehicle. However, PF manages to track the target vehicle with the maximum of 60 pixels away from the original position.
V. CONCLUSION

A well-organized and efficient vehicle parking management is needed to manage the vehicles surveillance and localization in huge parking area. Video cameras that are commonly installed in the parking area can be useful resources to perform vehicle tracking, but they are normally fixed at a static position, resulting in occlusion and maneuvering incidents that affects the tracking accuracy. The proposed GA resampling based PF algorithm has successfully tracked the target vehicle undergoing occlusion and maneuvering incidents in a parking lot because GA has the ability to converge the estimated posterior position instead of diverge to the wrong position. Thus, it can be concluded that the proposed vehicle-tracking algorithm is capable of robustly tracking the target vehicle under different types of disturbances.

ACKNOWLEDGMENT

The authors would like to acknowledge the Ministry of Education Malaysia (KPM) for supporting this research under Exploratory Research Grant Scheme (ERGS), grant no. ERG0042-ICT-1/2013.

REFERENCES


