Research Article
Vehicle Tracking Algorithm Based on Observation Feedback and Block Symmetry Particle Filter

Yanshuang Hao, Yixin Yin, and Jinhui Lan
School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Yanshuang Hao; haoyanshuang@sina.com

Received 15 August 2013; Revised 5 December 2013; Accepted 31 December 2013; Published 13 February 2014

Academic Editor: Vesa Valimaki

Copyright © 2014 Yanshuang Hao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a novel particle filter algorithm for vehicle tracking, which feeds observation information back to state model and integrates block symmetry into observation model. In view of the proposal distribution in traditional particle filter without considering the observation data, a new state transition model which takes the observation into account is presented, so that the allocation of particles is more familiar with the posterior distribution. To track the vehicles in background with similar colors or under partial occlusion, block symmetry is proposed and introduced into the observation model. Experimental results show that the proposed algorithm can improve the accuracy and robustness of vehicle tracking compared with traditional particle filter and Kernel Particle Filter.

1. Introduction

In recent years, the traffic pressure in some cities has become higher and higher, and the role of the intelligent transportation system (ITS) is more and more important. Video sensors are particularly important in traffic applications due to their fast response, easy installation, operation, and maintenance, and their ability to provide wide area monitoring allowing analysis of traffic flow, speed measurement, vehicle classification, and incident detection [1]. Moving vehicle tracking is the key technology in the video-based traffic monitoring system [2].

Particle filter is a kind of popular and robust approach for vehicle tracking. It is also called Sequential Monte Carlo method [3] or Condensation algorithm [4]. Different from the Kalman filter, which can only solve the linear, Gaussian system state estimation problems, the particle filter does not suppose system model to be linear and can deal with nonlinear non-Gaussian distribution state model. It is a method based on Monte Carlo theory and Recursive Bayesian Estimation [3]. The particle filtering predicts a new set of particles using a proposal distribution, updates the weight for every particle using an observation function, and estimates the posterior probability density of the states by computing the weighted sum of the set of particles. The propagation of particles and the updating of weights are very important steps during the particle filtering.

In order to get a preferable proposal distribution, Chang and Ansari [5] proposed the Kernel Particle Filter (KPF) for visual tracking, in which particles are allocated based on the gradient of the kernel density to form a continuous estimate of the posterior density function. The gradient estimation and particle allocation are implemented by the mean shift algorithm. Rui and Chen [6] introduced Unscented Particle Filter (UPF) into audio and visual tracking, which used Unscented Kalman Filter to generate sophisticated proposal distribute that seamlessly integrates the current observation. KPF and UPF can improve the tracking performance but increase the computation complexity. Zhu et al. [7] combined particle filter and grey prediction model and proposed a novel visual tracking algorithm, called GMPF. GMPF utilized the history of state estimation sequence to set up grey prediction model for generating proposal distribution. Compared with UPF, the GMPF consumed less time per frame but presented worse results [7]. Wei et al. [8] proposed a layered particle filter (LPF) architecture for vehicle tracking. The strategy of LPF is dividing the particle set into two subsets randomly first, then predicting a new subset via state transition equation, selecting the particle with largest weight in another subset and predicting another new subset at the new position.
obtained via Camshift algorithm, and finally merging the two subsets.

In process of weights updating, the observation model combining multiple features was often used. Wei et al. [8] built a fusion observation model with color and local integral orientation descriptor. Tian et al. [9] used the weighted color histogram and rotated complex wavelet filter to describe color and texture features, respectively, and fused the two features under particle filter framework to improve the robustness and accuracy of face tracking. Rezaee et al. [10] fused color, edge, texture, and motion cues in particle filter algorithm for vehicle tracking. Maggio et al. [11] proposed a multifeature tracking algorithm that adaptively weights the reliability of each feature in particle filtering framework. The target representation is based on color and orientation histograms. Gao et al. [12] constructed proposal distribution by use of state partition technique and parallel extended kalman filter and fused adaptively color model and shape model in particle filter framework to enhance the performance of object tracking. Chan et al. [13] proposed a vision-based automatic system to detect multiple vehicles under various lighting conditions with a single particle filter. The system fuses four cues including underneath shadow, vertical edge, symmetry, and taillight. With BSAS clustering technique and initial data-driven sampling, multiple vehicles can be tracked with only one set of particles. Niknejad et al. [14] proposed a method for multiple vehicles tracking combining a deformable object model with particle filter. The likelihood of particles is updated by integrating the detection scores map and intensity correlations for vehicle’s root and parts between two sequential frames.

In this paper, the novelty of the proposed algorithm lies in a new state model fed back observation information and an observation model based on block symmetry. The rest of the paper is organized as follows. Section 2 introduces the normal particle filter framework. Section 3 proposes a novel particle state transition model. In Section 4, the block symmetry observation model based on color histogram and block symmetry are given. Section 5 presents the algorithmic description of the proposed particle filter. The experimental results are discussed in Section 6. Finally, Section 7 concludes the paper.

2. Review of Particle Filter Algorithm

A nonlinear, non-Gaus system can be represented by the state equation and the observation equation as follows [15]:

\[
x_t = f(x_{t-1}, u_{t-1}),
\]

\[
z_t = h(x_t, n_t),
\]

where \( x_t, u_t \) denote the system state vector and the state noise vector at time \( t \), respectively, and \( z_t, n_t \) denote the system observation vector and the observation noise vector. Particle filtering is a method to approximate the state posterior distribution \( p(x_t | z_{1:t}) \) by finite weighted particles.

At the beginning, the initial set of particles is generated from the priori density \( p(x_0) \), and every particle is assigned an equal weight.

At time \( t \), the new set of particles \( \{x^i_t\}_{i=1}^N \) is derived from the proposal distribution \( q(x^i_t | x^i_{0:t-1}, z_{1:t}) \), which is also called the importance function. Supposing that the states of particles comply with a first order Markov process and the observations are independent of each other, the weight of the particle \( i \) at time \( t \) can be updated by [3]

\[
w^i_t = w^i_{t-1} \frac{p(z_t | x^i_t) p(x^i_t | x^i_{t-1})}{q(x^i_t | x^i_{0:t-1}, z_{1:t})},
\]

where \( p(z_t | x^i_t) \) is the likelihood function defined by the observation equation (2) and \( p(x^i_t | x^i_{t-1}) \) is the state transition density distribution described by the state equation (1). Normalize the weights:

\[
\bar{w}^i_t = \frac{w^i_t}{\sum_{i=1}^N w^i_t}.
\]

According to the theory of Monte Carlo Integration, the unknown particle state at time \( t \) can be estimated as follows:

\[
\hat{x}_t \approx \sum_{i=1}^N \bar{w}^i_t x^i_t.
\]

The resampling step is implemented to eliminate the particles with weak weights and reduce particle degeneracy problem [3].

3. State Model

The normal particle filters select the state transition density distribution \( p(x^i_t | x^i_{t-1}) \) as the proposal distribution [15], that is, predict the new particles according to the state equation (1). Equation (3) can be simplified as

\[
w^i_t = w^i_{t-1} p\left(z_t | x^i_t\right).
\]

This method is simple and effective but requires an accurate model-of-state transition. Due to not considering the observation information, the proposal distribution cannot represent the true distribution effectively when the state equation is not exact enough or the observation noises become larger [12]. To solve the problem, this paper proposes a new model-of-state transition which takes the observation information into account during the process of particle propagation.

Let us represent the target vehicle area with a rectangle centered in \( C = (cx, cy) \), with the width of \( h_x \) and the height of \( h_y \). Then, we define the state vector of the target as \( x = (cx, cy, h_x, h_y)^T \).

The state equation is designed as follows:

\[
x_t = x_{t-1} + A\delta_t + u_t,
\]

\[
\delta_t = (1 - \rho (\delta_{t-1})) \delta_{t-1} + \rho (\delta_{t-1}) \left( \tilde{C}_{t-1} - \tilde{C}_{t-2} \right).
\]

In (7), \( \hat{\delta}_t = (\hat{\delta}_x, \hat{\delta}_y)^T \) means the estimate of target center displacement from the frame \( (t - 1) \) to the frame \( t \); \( A \) is
the coefficient matrix of $\delta_t$, and $v_i$ is a Gaussian noise vector with zero mean. The displacement of target $\tilde{\delta}_t$ is updated adaptively by (8), in which $\tilde{C}_{t-1}$ is the estimate of target center at frame $(t-1)$ and $\rho(\tilde{x}_{t-1})$ is the similarity between the target color histogram and the color histogram of the estimated state $\tilde{x}_{t-1}$.

Karavasilis et al. [16] updated the translation of target using the Earth Mover’s Distance (EMD) between the source and target GMM. In (8), the displacement is evaluated by employing the similarity $\rho(\tilde{x}_{t-1})$. A similarity $\rho(\tilde{x}_{t-1})$ close to 1 indicates that the estimated state $\tilde{x}_{t-1}$ has a small error with the true state at the frame $(t-1)$, while a similarity close to 0 indicates a large error. Therefore, the displacement of target center at the th frame $\tilde{\delta}_t$ will be mainly determined by the difference between $\tilde{C}_{t-1}$ and $\tilde{C}_{t-2}$ if $\rho(\tilde{x}_{t-1})$ is close to 1. On the other hand, the displacement $\tilde{\delta}_t$ will be mainly decided by $\tilde{\delta}_{t-1}$ if $\rho(\tilde{x}_{t-1})$ is close to 0. Obviously, the estimated state $\tilde{x}_{t-1}$ includes the observation information, so the displacement $\tilde{\delta}_t$ is a function of the observation. Consequently, our proposed model-of-state transition also considers the observation information.

The size of vehicle in a video sequence increases when the vehicle moves towards the camera, and it decreases while the vehicle gets far away. The real distance between a vehicle gets far away, apparently. The real distance between a vehicle moves toward the camera, and it decreases while the vehicle gets far away. Consequently, our proposed model-of-state transition also considers the observation information.

The size of vehicle in a video sequence increases when the vehicle moves towards the camera, and it decreases while the vehicle gets far away. The real distance between a vehicle and the camera is reflected in the vertical coordinate of the vehicle in the video image. The size of rectangle area can be corrected by the displacement of target in vertical direction $\delta y_t$ and then the coefficient matrix $A$ can be defined as

$$A = \begin{bmatrix} 10 \\ 01 \\ 0a \\ 0b \end{bmatrix},$$

where $a, b$ are constants, whose values vary with the angle between the camera and road.

### 4. Observation Model

The color histogram is invariant to the scale and orientation changes of the target and is robust to partial occlusions but cannot provide geometry information of the target [11], which makes it difficult to track the target in the background with similar colors. Considering that most of vehicles keep a good symmetry, this paper integrates the color histogram and the symmetry measurements to design observation model for vehicle tracking.

#### 4.1. Color Histogram

The RGB color space can be quantized into $8 \times 8 \times 8 = 512$ bins, and the color histograms of the reference target region is represented by $q^0 = \{q^0_{\langle n \rangle} \}_{n=1:512}$. At time $t$, the color histograms of the candidate target region defined by particle $x^i_{t}$ can be denoted by $q(x^i_{t}) = \{q^i_{\langle n \rangle} \}_{n=1:512}$. The Bhattacharyya coefficient [17] is used to measure the similarity between target color histogram and candidate histogram as follows:

$$\rho(x^i_{t}) = \sum_{n=1}^{512} \sqrt{q^i_{n} \cdot q_{n}(x^i_{t})}. \quad (10)$$

The distance between the two histograms can be calculated on the basis of the Bhattacharyya coefficient [17]:

$$d(q^0, q(x^i_{t})) = \sqrt{1 - \rho(x^i_{t})}. \quad (11)$$

The observation model based on color feature is defined as

$$P_c(z_t | x^i_{t}) \propto \exp \left[ -\frac{d^2(q^0, q(x^i_{t}))}{2\sigma^2} \right], \quad (12)$$

where $\sigma$ denotes the normal standard deviation set by experience.

#### 4.2. Block Symmetry

Assume that the state of the particle $i$ at time $t$ is $x^i_t = (cx^i_t, cy^i_t, hx^i_t, hy^i_t)^T$ and $q(l,c)$ denotes the gray value of the pixel on row $l$ and column $c$ in the image of frame $t$. We compute the symmetry within the rectangle region defined by particle state $x^i_t$ using the symmetry model proposed in [18].

The mean gray value of the pixels on row $l$ in the rectangle region can be expressed by $M^i_{t,l}$:

$$M^i_{t,l} = \frac{1}{hx^i_t} \sum_{m=-hx^i_t/2}^{hx^i_t/2} g(l, cx^i_t + m). \quad (13)$$

Then we calculate the following two functions for $k, k \in [0,hx^i_t/2]$:

$$E^i_{t,l}(k) = \frac{g(l, cx^i_t + k) + g(l, cx^i_t - k)}{2} - M^i_{t,l}, \quad (14)$$

$$O^i_{t,l}(k) = \frac{g(l, cx^i_t - k) - g(l, cx^i_t + k)}{2}.$$

The symmetry property of the row $l$ in the rectangle region can be computed as follows [18]:

$$S^i_{t,l} = \frac{\sum_{k=0}^{hx^i_t/2} \left( |E^i_{t,l}(k)| - |O^i_{t,l}(k)| \right)}{\sum_{k=0}^{hx^i_t/2} \left( |E^i_{t,l}(k)| + |O^i_{t,l}(k)| \right)}. \quad (15)$$

The range of $S^i_{t,l}$ is the interval $[-1, 1]$. If the gray scale image on the row $l$ has a perfect symmetry, $S^i_{t,l}$ should be equal to 1. On the other hand, $S^i_{t,l} = -1$ shows an ideal asymmetry. Figure 1 gives the example of symmetry computing, and Figure 1(a) shows two rectangle regions to be extracting the symmetry, the red one is symmetry and the blue one is asymmetry. Figure 1(b) gives the symmetry of every row in the two rectangle regions.
Different from [18], we propose the block symmetry-based observation model to accurately track the vehicles under partial occlusion. When the vehicle tracked is under partial occlusion, it can be divided into two parts, the occluded part and the unoccluded part. The former no longer has symmetry, while the latter still keeps the symmetry. Therefore, we split the rectangle region defined by particle $x_i^t$ into the upper half and lower half and calculate the symmetries of the upper half and lower half, respectively, as follows:

$$
S_{i,\text{up}}^t = \frac{2}{h y_i^t} \cdot \sum_{l=c y_i^t-h y_i^t/2}^{c y_i^t} S_{i,l}^t \\
S_{i,\text{low}}^t = \frac{2}{h y_i^t} \cdot \sum_{l=c y_i^t+1}^{c y_i^t+h y_i^t/2} S_{i,l}^t
$$

(16)

If both $S_{i,\text{up}}^t$ and $S_{i,\text{low}}^t$ are less than a given threshold $s_{\text{th}}$, the rectangle area is considered asymmetry. If either $S_{i,\text{up}}^t$ or $S_{i,\text{low}}^t$ is greater than or equal to the threshold $s_{\text{th}}$, the rectangle area is regarded as a region with mirror symmetry about a vertical axis. Thus, we define the global symmetry of the rectangle region as

$$
S_i^t = \begin{cases} 
\max (S_{i,\text{up}}^t, S_{i,\text{low}}^t) + 1, & \max (S_{i,\text{up}}^t, S_{i,\text{low}}^t) \geq s_{\text{th}}, \\
0, & \max (S_{i,\text{up}}^t, S_{i,\text{low}}^t) < s_{\text{th}},
\end{cases}
$$

(17)

where the threshold $s_{\text{th}}$ is a value approximate to but less than 0 (e.g., −0.05), which is selected by experiment. The range of $S_i^t$ is the interval $[1 + s_{\text{th}}, 2]$ and 0.

The observation model based on symmetry feature is defined as

$$
P_i (z_t | x_i^t) = S_i^t.
$$

(18)

4.3. Combination of Color and Block Symmetry. The observation model which fused color histogram and symmetry feature is defined as follows:

$$
P (z_t | x_i^t) = P_c (z_t | x_i^t) P_s (z_t | x_i^t) \propto \exp \left( -\frac{d^2 (q_0^t, q (x_i^t))}{2\sigma^2} \right) \cdot S_i^t.
$$

(19)

If the candidate target region defined by particle $x_i^t$ has no symmetry, $S_i^t = 0$ and the observation density $p(z_t | x_i^t) = 0$. If the symmetry of the candidate target region is acceptable, the observation density is modified by multiplying the symmetry measurement $S_i^t$.

5. Proposed Algorithm

The complete vehicle tracking algorithm proposed is presented below.

Given the target color histogram $q_0^t$, consider particle set at time $(t − 1)$ $\{x_i^t\}_{i=1}^N$ and its weights $\{w_i^t\}_{i=1}^N$, estimation of state at time $(t − 1)$ $\bar{x}_{t-1}$, and the similarity $\rho_i (\bar{x}_{t-1})$ and the estimate of target displacement at time $t \delta_i$.

Step 1. Generate a predicted set at time $t$ $\{x_i^t\}_{i=1}^N$ according to (7).
Step 2. For \( i \)th particle, compute the likelihood function \( p(z_t \mid x^i_t) \) according to Section 4, update the weight of particle \( w^i_t = w^i_{t-1} p(z_t \mid x^i_t) \), and normalize the weights by (4).

Step 3. Estimate the state of target at time \( t \) by \( \hat{x}_t = \sum_{i=1}^{N} \tilde{w}^i_t x^i_t \).

Step 4. Calculate the similarity \( \rho(\hat{x}_t) \) and displacement at time \( (t+1) \) \( \delta_{t+1} \) by (8).

Step 5. Compute the number of effective sample by \( \tilde{N}_{\text{eff}} = 1/ \sum_{i=1}^{N} (\tilde{w}^i_t)^2 \) and resample the set of particles using the importance sampling method if \( \tilde{N}_{\text{eff}} \) is less than a threshold.

Step 6. If the frame \( t \) is not the last one, return to Step 1; otherwise end the algorithm.

6. Experimental Results

In order to test the tracking performance of the algorithm proposed above, video sequences obtained from real roads are used. The camera is fixed on pedestrian overpasses. The experiments are performed by MATLAB on PC with a 2.1 GHz Intel(R) Core(TM)2 Duo CPU and a 2 GB memory. In all of the following experiments, we locate the tracked vehicle in first frame manually. The standard deviations of the Gaussian noise vector of (7) are set to \( [\sigma_{cx}, \sigma_{cy}, \sigma_{hx}, \sigma_{hy}] = [3, 3, 0.3, 0.3] \), and the standard deviation of (12) is set to \( \sigma = 0.2 \).

6.1. Tracking with Different State Models. Under particle filter framework, the state models in common use are the random walking model [18, 19] and the second-order linear regression model [8, 20]. The former can be represented as \( x_t = x_{t-1} + v_t \), where \( v_t \) is a multivariate zero-mean Gaussian random variable. The latter can be expressed as \( x_t - x_{t-1} = x_{t-1} - x_{t-2} + v_t \).

Figure 2 shows the tracking results of three particle filter algorithms with random walking model, 2nd order linear model, and proposed state model, respectively. The observation models of three methods are based on color histogram only. The image sizes of the video sequence are \( 640 \times 480 \). In this video, the road condition is simple, but the target vehicle moves fast. The number of particles is 60 in each method, and the tracking results are marked with red rectangle bounding boxes.

Figure 2(a) shows that the method using a random walking model takes on an obvious deviation of the estimated vehicle position at frame 11 and loses the target vehicle at frame 26. Figure 2(b) gives the results using second-order linear regression model, in which errors of target size appear from frame 21 and deflection of target position appears at frame 26. The tracking results of proposed state model are presented in Figure 2(c). Because the displacement of target between adjacent frames and the observation information are considered during the particle propagation, the tracking results can keep up with the true target in time. The sizes of the tracking windows are changed adaptively due to a suitable coefficient matrix \( A \), as shown in Figure 2(c).

6.2. Tracking with Different Observation Models. To verify the effect of the observation model proposed, we compare it with the observation model based on color histogram only by using a video sequence, in which two vehicles with similar colors are close to each other. Figure 3(a) shows the tracking results with the observation model based on color histogram, and Figure 3(b) presents the results with the proposed observation model. The state models of two approaches are both the model proposed in Section 3.

From Figure 3(a), we can see that the color-histogram based approach cannot locate the target accurately, and the tracking results wander back and forth between the target vehicle and adjacent vehicle because of similar color interference. In contrast, the proposed approach can track the target vehicle stably owing to taking advantage of symmetry, as shown in Figure 3(b).

6.3. Tracking under Partial Occlusion. In this experience, we select a video sequence, in which a car is partially occluded by a white minibus when the minibus moves from the middle lane to the inside lane. Figure 4 presents the tracking results of two approaches with same state models and different observation models. Figure 4(a) employs the symmetry particle filter, which fuses the symmetry feature, not block symmetry, and color histogram. Figure 4(b) applies...
the proposed method. The number of particles is 60 in both methods.

From Figure 4, we can see that the symmetry particle filter gives the results with distinct deflection compared to the proposed approach at frame 10 and frame 30. To evaluate the tracking accuracy, the deviation from the vehicle real position is calculated by [14]

$$DE_t = \sqrt{(\tilde{x}_t - x_t)^2 + (\tilde{y}_t - y_t)^2}, \quad (20)$$

where $(\tilde{x}_t, \tilde{y}_t)$ and $(x_t, y_t)$ mean the estimated center position of tracked vehicle and the true center of the vehicle at frame $t$, respectively.

The deviations in frames 1 to 90 by symmetry particle filter and the proposed approach are computed and presented in Figure 5. The 90 frames can be divided into two periods, frame 1–40 (asymmetry occlusion), frame 41–90 (symmetry occlusion). Figure 5 indicates that the deviations by symmetry particle filter are larger than those by the proposed approach during the period of asymmetry occlusion and become comparable during the period of symmetry occlusion. The results of Figures 4 and 5 show that the using of block symmetry can improve the tracking accuracy.

6.4. Compare with Kernel Particle Filter. In this example, the particle filter algorithm proposed is compared with Kernel Particle Filter (KPF) [5] and the traditional particle filter. In the approach of Kernel Particle Filter, we choose the random walking model as the state model, the color histogram as the observation model and quantize the RGB color space into $8 \times 8 \times 8$ bins. In the procedure of gradient estimation, each particle is moved to its mean using the mean shift method.

The image sizes of the video sequence are $320 \times 240$; the number of particles is 60 in three methods, traditional particle filter, KPF, and proposed algorithm. The results shown in Figure 6 indicate that both the proposed algorithm and KPF can track vehicles in the video, while the traditional PF fails to track the target at frame 90. In addition, the sizes of tracking windows obtained by the proposed algorithm are more close to the true sizes of target vehicle than those obtained by KPF.

The deviations in each frame by the proposed algorithm and KPF are calculated according to (20), and the results are given in Figure 7.

We also measure the average time consumptions per frame of the two algorithms with different numbers of particles. The results are presented in Figure 8. Obviously, the KPF spends more time than our approach because of the procedure of gradient estimation for every particle.

Figures 7 and 8 imply that our approach can obtain higher tracking accuracy using less time than the Kernel Particle Filter.

Figure 8 demonstrates that the more particles are allocated the more time is consumed. On the other hand, the less particles are allocated, the worse tracking accuracy will be. Thus we compute and compare the deviation in the different number of particles by the proposed approach to determine a suitable number of particles. The results are shown in Figure 9. In this example, the proposed particle filter with 30 particles only can accurately track vehicles in real time.
7. Conclusions

This paper proposes a novel state transition model based on observation feedback and an observation model which fused block symmetry and color features in the particle filter framework. During the process of particles propagation, the latest observation information is employed to update the displacement of particles, and size of target region defined by particle is corrected by the particle displacement on vertical direction. In order to improve the tracking performance when the tracked vehicle is under occlusion, the block symmetry is put forward. The observation model takes color histogram and block symmetry measurements as observation features. Experimental results illustrate the effectiveness of the observation feedback and block symmetry particle filter.


Submit your manuscripts at http://www.hindawi.com