Hybrid Particle Filter Algorithm Using Multi Features for Video Target Tracking

Hui Li*, Shengwu Xiong, Pengfei Duan
School of Computer Science and Technology,
Wuhan University of Technology, Wuhan 430070, China,
lipielin1984xyz@163.com

Abstract

Video target tracking is an essential problem in the field of computer vision. Particle filters have been proved to be very useful in target tracking for non-linear and non-Gaussian estimation problems. However, for the target tracking in complex background, it is often difficult to achieve robust tracking by using single target feature information. To solve this problem, this paper presents a hybrid particle filter algorithm using multi features for video target tracking. The algorithm integrates multiple features into particle filter to get better observation results, and then automatically adjusts the weight value of each feature according to the current tracking environment. In order to describe the target movement well, the method automatically adjusts the transfer range of particles according to the target speed changes, thus the particles can reach high likelihood region, which reduces the target lost phenomenon caused by speed changes. Experimental results demonstrate that the proposed algorithm improves the tracking performance in complicated real scenarios.

Keywords: Estimation Problems, Multi Features, Particle Filter, Video Target Tracking

1. Introduction

Video target tracking is an important research field in computer vision for its wide range of applications requirements and prospects in many industries, such as military guidance, visual surveillance, visual navigation of robots, human-computer interaction and medical diagnose [1-3]. The main task of target tracking is to track one or more mobile targets in video sequences so that we can get the position, velocity, trajectory and other parameters of the target. However, automated tracking target is still an open problem in many settings [4]. Video target tracking has been studied intensively and a number of elegant algorithms have been established.

One popular tracking method is mean shift procedure [5], which finds the local maximum of probability distribution in the direction of gradient. Comaniciu et al. [6] gave a strict proof of the convergence of the algorithm and proposed a mean shift based on tracking method. As a deterministic method, mean shift keeps single hypothesis and is thus computationally efficient. But it may run into trouble when similar targets are presented in background or occlusion occurs. Another common approach is the use of the kalman filter [7]. This method is based on the assumption that the probability distribution of the target state is Gaussian, and therefore the mean and covariance, computed recursively by the Kalman filter equations, can fully characterize the behavior of the tracked target. However, in video target tracking, tracking targets in real world rarely satisfy Gaussian assumptions required by the Kalman filter in that background clutter may resemble a part of foreground features. One promising category is sequential Monte Carlo method, also known as particle filter [8], which recursively estimates target posterior with discrete sample-weight pairs in a dynamic Bayesian framework. Due to particle filters’ non-Gaussian, non-linear assumption and multiple hypothesis property, they have been successfully applied to video target tracking [8,10].

Various researchers have attempted to extend particle filters to target tracking. Among others, one of the most successful features used in target tracking is color. Nummiaro et al. [11] proposed a tracking algorithm that considered color histograms, as features, that were tracked using the particle filters algorithm. Despite the algorithm is more robust to the partial blocked target and the target shape changes, the algorithm exhibits high sensitivity to illumination changes that may cause the tracker to fail. Vignon et al. [12] proposed a tracking algorithm based on Hausdorff distance, the algorithm uses the target shape information to track the target, it’s more robust to the changes of the target color, but...
this algorithm is easy to fail in tracking when the background image is more complex. For the target tracking in complex background image, it is often difficult to achieve robust tracking by using a single target feature information. Spengler et al. [13] proposed a tracking algorithm which can automatically adapt and fuse all feature information. They use the algorithm which is similar to Expectation Maximization, EM, to adjust online the weight of each characteristic information, but the tracking is the matching algorithm which is based on the local area searching. The greatest weakness of this algorithm is prone to incorrect tracking when the target is partially or completely obscured. Therefore, a more profound and systematic investigation of the interaction of multiple features modalities is desirable.

In this paper, we extend the particle filter to allow the fusion of multiple features which lead to a better target tracking. The proposed tracking algorithm is an effective integration of color, texture and shape feature in observation model. And the observation model automatically adjusts the weight value of each feature for the current tracking situation. In the state-space model, the algorithm can automatically adjust the transfer range of particles according to the target speed changes, thus the particles can reach high likelihood region, which reduces the target lost phenomenon caused by speed changes and increases the robustness of target tracking. Experimental results demonstrate the effectiveness and robustness of our proposed algorithm.

2. Hybrid particle filter algorithm

Video target tracking can be considered as the corresponding matching of relevant characteristics of target in sequence image frames. The more commonly used algorithm is based on target characteristics, the core idea of which is identifying region that has the most similar features with the target template being tracked and considering it as the region where the target locates. Particle filters have been proved to be powerful tools for video target tracking [14].

2.1 State-space model

In order to describe the target’s movement well, the tracking system is modeled in a nonlinear state-space approach. Particle state \( x \) is defined as \([x, y, v_x, v_y, w, h]\), which represents the center coordinates of particle, two-axis velocity, width and height respectively. At each time step \( t \), the state vector \( X_t \) is defined as: \([x_t, y_t, v_{x_t}, v_{y_t}, w_t, h_t]\). Target state equation is defined as follows:

\[
X_t = F \cdot X_{t-1} + G \cdot w_{t-1}
\]  

(4)

Where \( w_t \) represents the process noise. The process noise caters for any mismodeling effects or unforeseen disturbances in the motion model. Matrices F and G are defined as follows in Eq.(5), \( \Delta T \) denotes the sampling period.

\[
F = \begin{bmatrix}
1 & 0 & \Delta T & 0 & 0 & 0 \\
0 & 1 & 0 & \Delta T & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(5)

\[
G = \begin{bmatrix}
\frac{\Delta T^2}{2} & 0 & 0 & 0 \\
0 & \frac{\Delta T^2}{2} & 0 & 0 \\
\Delta T & 0 & 0 & 0 \\
0 & \Delta T & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

In practical applications, the target’s movement speed usually changes, and there is a certain correlation between the target movement trend of the moment and that of the period before, so two-axis velocity changes:

\[
v_x = \lambda (x_{t-1} - x_{t-2}) + (1 - \lambda)(x_{t-2} - x_{t-3}) \\
v_y = \lambda (y_{t-1} - y_{t-2}) + (1 - \lambda)(y_{t-2} - y_{t-3})
\]

(6)

The method can automatically adjust the transfer range of particles according to the target speed changes, thus better prediction results can be achieved: the particles can reach high likelihood region,
which reduces the target lost phenomenon caused by speed changes and increases the robustness of target tracking.

2.2. Observation model

1) Color feature

Color distributions are used as target models for they can achieve robustness against non-rigidity, rotation and partial occlusion. In our experiments, the histograms are typically calculated in the RGB space using $8 \times 8 \times 8$ bins. We only consider target region color distribution condition, suppose the state of target is defined as $X_i$, $X_i = (x_i, y_i)$ denotes the center of the target region and $d = (h, h)$ denotes the radius, the location of the pixels in the target region is described as $X_i = (x_i, y_i), i = 1, \cdots, n_i$, where $n_i$ is the number of pixels in the region. So the color distribution in the target region is calculated as:

$$p^{\text{color}}(X_i) = \frac{\sum x_i \delta[h(x_i) - u]}{\sum x_i}, \quad u = 1, \cdots, m$$

(7)

Where $\delta$ is the delta function, the parameter $d = \sqrt{h^2 + h^2}$ is used to adapt the size of the region, the histograms are produced with the function $h(x_i)$, that assigns the color at location $x_i$ to the corresponding bin. $u$ is the color level index. $k()$ is the weighting function

$$k(p) = \begin{cases} 1 - \frac{1}{\| p \|}, & \| p \| < 1 \\ 0, & \text{others} \end{cases}$$

(8)

Where $r$ is the distance from the region center. The introduction of $k(\cdot)$ is in view of the fact that the farther away from the target center, the smaller possibility of the pixels’ belongingness to the target area. Therefore the closer to the target center, the larger weight value the pixels will be given by the weight value function.

The measurement likelihood is computed based on the distance between the measured target histogram and the reference histogram of the target. One common approach to compute the distance between the color distributions is to use the Bhattacharyya distance:

$$d_{\text{Bh}}[p(X), q] = [1 - \rho(p(X), q)]^{\frac{1}{2}}$$

(9)

where

$$\rho(p(X), q) = \sum_{i=1}^N (p^{\text{color}}(X_i)q^{\text{color}})^{\frac{1}{2}}$$

Where $d_{\text{Bh}}[p(X), q]$ is the Bhattacharyya coefficient between the measured color distribution $p(x_i)$ and the reference color distribution $q$. The measurement likelihood is then given by a function of the Bhattacharyya distance:

$$p_{\text{color}}(Z_i | X_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{d_{\text{Bh}}^2[p(X_i), q]}{2\sigma^2})$$

(10)

The greater value of the formula shows there are more similarities in color between the candidate goals and the target template; as a result there will be greater possibility that the candidate target is the real target.

2) Texture feature

Texture features is an innate property of virtually all surfaces. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. Local binary pattern, LBP [15], is vastly used for texture description which has good performance in
texture classification, fabric defect detection and moving region detection. The LBP operator is defined as follows:

$$LBP(P) = \sum_{x=0}^{n-1} s(g_x - g_c)2^x$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

(11)

Where $g_c$ is the intensity value of center pixel $p_c$ and $g_n$ is the intensity of neighboring pixels.

The image histogram obtained from the computation of the LBP is defined as follows:

$$H_i = \sum_{x,y} I(f(x,y) = i), i = 0, \ldots, n-1$$

$$I(f(x,y) = i) = \begin{cases} 1, & f(x,y) = i \\ 0, & f(x,y) \neq i \end{cases}$$

(12)

Where $n = 2^p$ represents the length of the encode bit generated by the LBP operator, $p$ represents the number of pixels in the neighborhood, $f(x,y)$ is the LBP value at $(x,y)$, in this way, $H_i$ represents the number of pixels which have the LBP value of $i$, the histogram can reflect the distribution of the LBP values.

After we get the images’ LBP histogram, texture observation likelihood function is defined as follows:

$$p_{\text{texture}}(Z_t | X_t) = \frac{1}{\sqrt{2\pi\sigma^2_{\text{texture}}}} \exp\left(-\frac{d_t^2[S,M]}{2\sigma^2_{\text{texture}}}\right)$$

(13)

Where $S$ denotes the eigenvector of the test sample, $M$ denotes the eigenvector of the template, $d_t[S,M]$ denotes the histogram intersection distance of the two histogram images.

3) Shape feature

Chamfer distance \[^{[16]}\] is a very effective method when comparing the degree of similarity of the evaluation of both target shapes. In chamfer matching, all the edge points of an image are detected firstly by an edge detection algorithm such as the canny method which will not be addressed in this paper. Secondly, the binary edge images are preprocessed by eigen analysis to rectify their principle directions. Thirdly, a distance transformation is employed on the binary edge image to obtain a distance image. During this step, the values of edge pixels are set to zero. The value of each non-edge pixel is set to the distance to its nearest edge pixel.

Assuming that the binary image of the shape template shows $T$, the binary image of the current frame picture shows $I_t$, and its range image shows $DI_t$, then the Chamfer distance between two shapes is calculated as follows:

$$d_t[T(X),I_t] = \frac{1}{\sqrt{P}} \sum_{k=1}^{P} D_{I_t}(k)$$

(14)

Where $T(X)$ denotes the center position and radius of the shape template in the current frame, $|T|$ represents the total number of pixels which have the value of “1” in image $T$, $k$ denotes the k-th pixel which has the value of “1” in image $T$, $D_{I_t}(K)$ denotes the gray value in $DI_t$ which is at the location where the $k$-th pixel has the value of “1” in image $T$ when the image $T$ is put on the distance image $DI_t$. After getting the Chamfer distance, we define the shape of the observation likelihood function:

$$p_{\text{shape}}(Z_t | X_t) = \frac{1}{\sqrt{2\pi\sigma_s}} \exp\left(-\frac{d_t^2[T(X),I_t]}{2\sigma_s^2}\right)$$

(15)
Where $\sigma^2$ denotes Gaussian variance, which is taken as 0.5 in the test. The greater the value in forum (3.4) shows there are more similarities in shape template between the candidate goals and objectives. Thus the candidate target is more likely to be the real target.

2.3. Multi features fusion

Using multi-feature information to track the target, the complementarities of the various characteristics information can be achieved; thereby the robustness of tracking is improved. The algorithm uses the color, LBP and chamfer distance information to describe the observation information of the goals. In the case that $X_t$ is the given target state, the total observation likelihood function of the target is as following:

$$p(Z_t|X_t) = a \cdot p_{\text{color}}(Z_t|X_t) + b \cdot p_{\text{texture}}(Z_t|X_t) + c \cdot p_{\text{shape}}(Z_t|X_t), \quad a + b + c = 1$$  

(16)

Where $p_{\text{color}}(Z_t|X_t)$, $p_{\text{texture}}(Z_t|X_t)$, $p_{\text{shape}}(Z_t|X_t)$ are the observation likelihood functions of color, texture and shape features, $0 \leq a, b, c \leq 1$ are the weights of the three characteristics information in the fusion. The feature weights can be dynamically calculated through the weight distribution of the particle set. The method is as following:

$$r_i^f = \max(w_i^f) - \min(w_i^f)$$  

(17)

$$\pi_i^f = \frac{r_i^f}{\sum_{i=1}^{N} r_i^f}$$  

(18)

Where $\pi_i^f$ is the weight of feature $f$, $r_i^f$ is the divergence of particle observation likelihood distribution of feature $f$.

Using this method, we can get the observed value after the integration of target features.

2.4. Algorithm process

The entire algorithmic flowchart can be summarized as follows:

Hybrid Particle Filter Algorithm Using Multi Features for Video Target Tracking
Hui Li, Shengwu Xiong, Pengfei Duan

1. Initial template observation model in the first frame
   1.1 Specifying a rectangle region.
   1.2 Compute the reference color histogram, LBP histogram and chamfer distance.
   1.3 Initiate particles in reference region.
   1.4 Initialize particle state distribution $\{X_i^{(0)}\}_{i=1}^N$ using the center of specifying region.
   1.5 Set initial weight value of feature information $a=b=c=0$.

2. For frame=2,3,…
   2.1 Important sampling step
      Propagate $\{X_i^{(t-1)}\}_{i=1}^N$ and get new particles $\{X_i^{(t)}\}_{i=1}^N$ using Eq.(5), (6).
   2.2 Update the weights
      Compute the observation likelihood function $p_{\text{color}}(Z_t|X_i^{(t)})$ for each particle state’s corresponding color using Eq.(10).
      Compute the observation likelihood function $p_{\text{texture}}(Z_t|X_i^{(t)})$ for each particle state’s corresponding texture using Eq.(13).
      Compute the observation likelihood function $p_{\text{shape}}(Z_t|X_i^{(t)})$ for each particle state’s corresponding edge using Eq.(15).
Update weights value of features information using Eq.(16).

Compute every particle weight using Eq.(16) and normalize the weight \( w_i^{(t)} = \frac{w_i^{(t)}}{\sum_{i=1}^{N} w_i^{(t)}} \).

2.3 Resampling

Compute \( N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w_i^{(t)})^2} \)

If \( N_{\text{eff}} < N_{th} \) // \( N_{th} \) is a pre-defined threshold.

Calculate the cumulative probability \( C_i, C_i' = C_{i-1} + w_i^{(t)}, t = 1, \ldots, N \)

For \( i = 1, \ldots, N \)

Generate a uniformly distributed random number \( r \) between \([0, 1] \).

If \( C_i > r \), \( X_i^{(t)} = X_i^{(t)}, \{w_i^{(t)}\}_{i=1}^{N} \approx \frac{1}{N} \)

End For

Else \( \{X_i^{(t)}, w_i^{(t)}\}_{i=1}^{N} = \{X_i^{(t)}, w_i^{(t)}\}_{i=1}^{N} \)

End If

2.4 State estimation

Estimate the state \( X_i = E(X_i \mid Z_{1:t}) \approx \sum_{i=1}^{N} \tilde{w}_i^{(t)} X_i^{(t)} \)

3. Experimental verification and analysis

To evaluate performance of the proposed hybrid particle filter algorithm, we collected three videos recorded in campus environment where the target pedestrian moves in different conditions. The video parameters in the evaluation are shown below in table 1.

**Table 1. Video parameters in simulation**

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Frame size</th>
<th>Total frames</th>
<th>fps(frames per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Person in zebra crossing</td>
<td>640×480</td>
<td>183</td>
<td>30</td>
</tr>
<tr>
<td>2. Two person on the way</td>
<td>640×480</td>
<td>103</td>
<td>30</td>
</tr>
<tr>
<td>3. Pedestrian on the grass</td>
<td>320×240</td>
<td>166</td>
<td>30</td>
</tr>
</tbody>
</table>

For comparison purposes, color-based, LBP-based, chamfer distance-based particle filter and hybrid particle filter algorithm are utilized. In our experiment, we only focus on target tracking algorithm, and does not concern the target detection algorithm, so a single target is selected by user.

First of all, sequence “person in zebra crossing” is tested, in which the lady carrying a bag is walking through zebra crossing on campus. This sequence is utilized to evaluate the performance of color-based, LBP-based, chamfer distance-based particle filter and the proposed hybrid particle filter algorithm under the situation of occlusion. Exemplar tracking results are illustrated in Fig.1, Fig.2, Fig.3 and Fig.4 respectively. Fig.1 represents the outcomes of the color-based particle filter. We can see from these frames that this algorithm leads to drifts under the situation of the cyclist’s occlusion and the pedestrian’s distraction. This is due to the fact that the background color is approximate to that of the cyclist and the pedestrian. In Fig.2, the LBP-based particle filter is used to track pedestrian target. The algorithm uses only the texture features, if the background image is relatively simple, it can succeed in tracking. But when the background is more complex, especially when the texture of the background image is similar with that of the target, it will fail in tracking. Fig.3 represents chamfer distance-based particle filter. The tracking error also appeared in the process of tracking such as the 162nd frame because of another pedestrian’s distraction. The fourth group represents hybrid particle filter algorithm using multi features. The tracking results show that the algorithm has better tracking.
accuracy and robustness. As a consequence, Fig.4 demonstrates that the proposed algorithm outperforms each of the preceding three.

Fig.1 Sequences 1: Color-based particle filter

Fig.2 Sequences 1: LBP-based particle filter
Fig. 3 Sequences 1: chamfer distance-based particle filter

Fig. 4 Sequences 1: hybrid particle filter

Fig. 5 shows the variation curve of a, b and c in the process of tracking. In this graph, weight value a begins to decline after the 117th frame in which the cyclist’s occlusion emerges and the credibility of the color information becomes low, but the texture and shape features are still reliable, therefore the values of the weights b and c increase. The weight value c continues to increase after the 150th frame, this results from the emergence of the pedestrian’s distraction, and the pedestrian is highly similar with the color and texture features of the target, thus the weight values of b and c decrease relatively. As the target moves, the interference of the target moves away after the 179th frame, therefore weight value a starts to increase.
Fig. 5 Sequences 1: weight value of each feature

Fig. 6 and fig. 7 show x coordinate and y coordinate respectively in the process of target tracking. From the two figures, we can see that proposed hybrid algorithm has better stability than any of the other three in that using the color-based, the LBP-based and chamfer-based particle filter algorithm, two interferences occur in the process of tracking and both x and y have varying degrees of volatility. It is can be seen that the robustness of tracking is improved by using the hybrid algorithm.

Fig. 6 x coordinate changes of each frame
Fig. 7 y coordinate changes of each frame

Fig. 8 shows target motion trajectory from the first frame to the last by using proposed algorithm. The blue points in the graph constitute target motion trajectory, and each point represents the target location of each frame.

Secondly, sequence “two person on the way” is tested, in which a girl walked rapidly and was completely obscured by a boy at a time. It differs from the first group of videos in that the tracking pedestrian’s walking speed is faster. Fig. 9 represents the proposed algorithm has better tracking accuracy and robustness. Although the girl’s walking speed changes rapidly and she was completely obscured by a boy at a time, she was tracked out with accurate location.
Fig. 9 Sequences 2: hybrid particle filter

Fig. 10 shows the variation curve of a, b and c in the process of tracking. In the second sequences, weight value a begins to decline after the 35th frame in which the boy’s occlusion emerges and the color information becomes unstable, but we can also use the texture and shape features, so the values of the weights b and c increase. As the target moves, the interference of the target moves away after the 55th frame, therefore weight value a starts to increase.

Fig. 10 Sequences 2: weight value of each feature

Thirdly, sequence “pedestrian on the grass” is tested. In this group of video, a boy walked on the grass and he was completely obscured by a tree. It can be seen from the whole process that the proposed algorithm can reliably track the pedestrian and has better tracking results.

Fig. 11 Sequences 3: hybrid particle filter
It can be seen from the test results of the above three groups of videos, the proposed algorithm has better tracking performances in dealing with complex situations such as the target’s translation, occlusion, speed changes, as well as analogue interference etc..

4. Conclusion

In this paper, we propose a hybrid particle filter algorithm using multi feature for video target tracking. The contribution of our work can be listed as following: 1) we extend the particle filter to allow the fusion of multiple features which lead to a better target tracking. 2) Proposed algorithm can automatically adjust the transfer range of particles according to the target speed changes, thus the particles can reach high likelihood region, which reduces the target lost phenomenon caused by speed changes. 3) According to the current tracking environment, we get particle weight value distribution to automatically adjust the weight value of each feature. The method was tested on a pedestrian tracking application in campus environment. In that case the algorithm can reliably track the target and target’s trajectory in difficult sequences with dramatic color changes, some partial occlusion, and background clutter edges. Experimental results demonstrate the effectiveness and robustness of our proposed algorithm.

Acknowledgement

This work was supported in part by the National Science Foundation of China under Grant No. 61170202 and Wuhan Municipality Programs for Science and Technology Development under Grant No. 201210121029).

5. References


