International Journal of Security and Its Applications
Vol.8, No.3 (2014), pp. 309-324
http://dx.doi.org/10.14257/ijsia.2014.8.3.32

Multi-cue Integration Object Tracking Based on Blocking

Lichuan Gu[1],Chengji Wang[1] , Jinqin zhong[2] , Jianxiao Liu[1] and Juan Wang[1]

[1] School of Computer and Information, Anhui Agriculture University
No.130 West Changjiang Road, Hefei Anhui, China,230036
[2]School of International Business, Anhui University
No.8 Shihe Road, Hefei ,Anhui, China,230036
[1]glc@ahau.edu.cn, [2] jinqinzhong@163.com

Abstract

Aiming at object occlusions and appearance changes and other issues under complex environment, a new tracking method of multi-cue integration particle filter based on blocking is proposed. Based on fragment the method using color, texture and edge cues integration constructs particle weights updating scheme, and according to the probabilities of which varied information belongs to the object or to the background region, distinguishes appearance changes and occlusions, then the local template updating is applied and the stability of tracking is improved. In order to verify the performance and effectiveness of this method, the experiments use the indoor and outdoor test videos which contained the light changes, occlusions, and the appearance changes of objects. The experimental results showed that the method had strong robustness and high tracking accuracy.

Keywords: *Object Tracking, Particle Filter, Multi-cue Integration, Blocking-based Tracking, Template Updating.*

1. Introduction

Video object tracking is to find the object location in the next frame through the known object locations in the image frames. It is one of the computer vision research hot spots. In the actual process of tracking, object appearance changes (as shown in Figure 1 (a)) and occlusions (as shown in Figure 1 (b)) will affect the stability of the tracking algorithm; and can lead to inaccurate object tracking and even lost object. As a result, how to effectively deal with object appearance changes and occlusions has always been the difficulty in object tracking problems.

A common way to solve the above problems is tracking through the integration of multiple visual cues, namely each cue provides a likelihood value for possible positions of the object in the next frame, and these cues are integrated according to the likelihood values to locate the final output. In recent ten years, plenty of tracking algorithms using multi-cues integration have been proposed. Such as in the early Birchfield proposed using intensity gradient and color histogram to track people's head [1]. The main problem of this method is that each cue's weight of the object is equal and remains the same in the process of tracking, so it can't provide contribution to each cue according to the actual situation. In order to solve this problem, Triesch and Vonder Malsburg proposed a new integration framework [2] which can take advantage of the uncertainty of each cue to adaptively adjust the contribution of different cues to the tracking results. Following this framework appeared many adaptive cues integration algorithms, as shown in [3-6]. These algorithms integrated a variety of cues, complemented each other

ISSN: 1738-9976 IJSIA

and improved the accuracy of the tracking process to some extent. But cues were assumed independent for each other, every cue weights updated according to the tracking response of the current frame, ignoring the information of consecutive frames.

Another way to solve the above problems is to track moving objects based on fragments. In the tracking of human, for example, human is divided into the head, torso and limbs [7, 8]. This approach generally required object model is known or a priori premise and is not suitable for generic object tracking. In order to solve this problem, some scholars put forward the general fragment tracking algorithms [9, 10]. This kind of method divided the object into different parts, but the division was arbitrary without considering any reference object model. In the process of tracking, this method weighted each fragment, combined the contributions of different fragments through statistical method to get the final output location of object. If there were object appearance changes or occlusions, the weight of corresponding fragment would be smaller and the impact on the overall goal would also be small, thus the resolution of object would be improved.

There is also a object tracking method which based on motion detection has attracted increasing attention [11-13]. Previous original image template matching based on tracking method had been paid a lot of attention because of its simple operation. This method extracted some cues as a template, and then looked for a region whose cues were the most similar to the template in current frame, so the template is the only representation of the object. However, the object may have been occluded during motion, may also have made appearance change by its own motion, and then the template needs to update online to track the object accurately.

(a) object's variable appearance.(b) Occlusions

Figure 1. Difficulties in Tracking Conditions

Table 1. Comparison of Current Tracking Algorithms

Tracking algorithms	Cue(s) for tracking	Integration of cues	Object fragment	Online updating template
Birchfield[1]	Color, edge	no	no	no
Maggio et al[3]	Color,orientation	yes	no	no
Adam et al[9]	Intensity	no	yes	no
Erdem et al[10]	Intensity	yes	yes	no
Porikli [11]	Subspace model	yes	no	yes
Ours	Color, LBP, edge	yes	yes	yes

This paper's contribution lies in: 1) it put forward a kind of fragment based adaptive multi-cues integration particle filter tracking framework. Different from existing methods, considering consecutive information of video frames when the cues were integrated, this method proposed a new weight evaluation method to measure the significance of each cue: the weight of each cue was looked as its state, and the weight update of each frame was understood as a kind of state tracking problem. 2) It can update the fragment template online during the tracking process. Fragment template was the combination of the fragment cues after the object had been divided into fragments. Fragment template improved the traditional template. The traditional one modeled the whole object without the spatial information. Fragment template method used all the small fragments for tracking, detected the possible fragments where appearance changes or occlusions occurred according to the matching of the small fragments and then took corresponding update strategies. Experimental results showed that the method will solved the problem of object appearance changes or occlusion, improved tracking accuracy and robustness.

The result of this paper is demonstrated as follows. The subsequent section describes the fragment tracking algorithm. Then an adaptive multi-cue integration model for particle filter is given in Section 3. Section 4 presents a particle filter algorithm of multi-cue integration based on fragment, which is the main contribution of this paper. Section 5 is devoted to algorithm analysis, including experimental results on videos with different tracking conditions, give qualitative and quantitative analysis. The final section concludes this paper and indicates the future direction of our research.

2. Fragment Tracking Algorithm [9]

Fragment tracking is a kind of object tracking algorithm based on cues proposed by Adam. In the FragTrack algorithm, tracking is treated as a sequential detection process. The target object is described by a template patch T, and in each image frame I, the detection is carried out by matching the template T to the image I. The output is a rectangular region enclosing the target object. As we discussed in the introduction, the novelty of the FragTrack comes from the arbitrary image patches and the corresponding robust estimation scheme for cue integration used in the template matching process. Detailed description is as follows:

The template patch T is subdivided into multiple image patches {PT} with each of them describing a different section of the target object. The important point is that these multiple patches are chosen arbitrarily and are not based on any predetermined object model. The authors suggested to use the patch layouts presented in Figure 2.

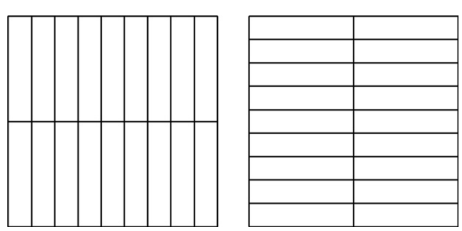

Figure 2. Vertical and Horizontal Fragment Template

Fragment tracking is to find the most similar region in current frame with the template. The tracking principle is shown in Figure 3, every coordinate point in the

$(x + dx, y + dy)$

searching window is a candidate for the best position of object in current frame, the centered the candidate point neighborhood is a candidate object.

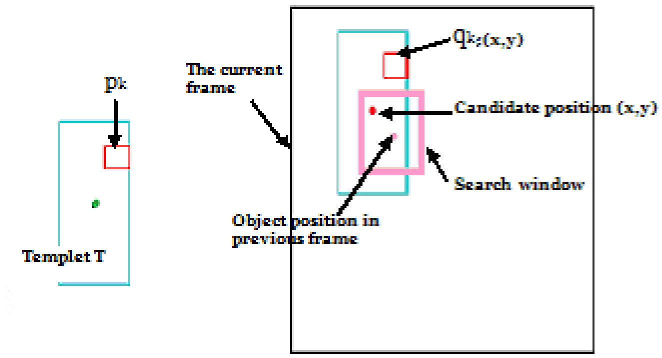

Figure 3. Fragment Tracking Principle

Location information (dx, dy) of a fragment is the difference between the center point coordinate of the fragment and the one of the object template. As shown in Figure 3, assuming a candidate coordinate of current frame as (x, y), fragment $q_{k:(x,y)}$ is then defined as a rectangle fragment with the center. Calculating the histogram matching distances of each $q_{k:(x,y)}$ in the candidate region and the corresponding fragment in the template, we can get Formula (1) using the linear weighting of all the matching distances to represent the similarity between candidates and object template.

$$S(x, y) = \sum_{p_k = p_1}^{p_n} d(p_k, q_{k:(x,y)}) \lambda_{p_k} \qquad (1)$$

$S(x, y)$ is the similarity between the candidate at the location (x, y) and the template, $d(p_k, q_{k:(\hat{x},\hat{y})})$ is the histogram matching distance between fragment p_k and fragment $q_{k:(x,y)}$ and its definition is

$$d(p_k, q_{k:(x,y)}) = \sum_{b=1}^{B} \frac{(h_{p_k}(b) - h_{q_{k:(x,y)}}(b))^2}{h_{p_k}(b) + h_{q_{k:(x,y)}}(b)} \qquad (2)$$

The smaller of $d(p_k, q_{k:(x,y)})$, means more similarity between the histograms. After (x, y) is in all candidate point traversing, we obtain similarity between each candidate object and the current template. Then the current object location $(\hat{x}, \hat{y})$ is defined as:

$$(\hat{x}, \hat{y}) = \arg\min_{(x,y)\in\Theta}(S(x, y)) \qquad (3)$$

Θ is the set of all the candidate locations.

To immediately reflect the variation of each fragment relative to the template, so that we can distinguish easily between occlusions and appearance changes, after we find the object location in current frame, according to the matching distance between fragment $q_{k:(\hat{x},\hat{y})}$ in the object region and the corresponding fragment p_k, we update the weight of p_k as

$$\lambda_{p_k} = \exp(\frac{-(d(p_k, q_{k:(\hat{x},\hat{y})}))}{\sigma_d^2}) \quad (4)$$

σ_d is the variance of $d(p_k, q_{k:(\hat{x},\hat{y})})$. For convenient representation, $q_{k:(\hat{x},\hat{y})}$ is represented as q_k below.

3. Adaptive Multi-Cues Integration Particle Filter Tracking

3.1. Basic Particle Filter

Particle filter algorithm [14] is a kind of filtering method based on Bayesian estimation and Monte Carlo sampling [15], its main idea is that look for a series of random samples of approximate posterior probability density distribution in state space, replace integral operation by a sample mean, finally estimate the state. The algorithm is applicable to handle arbitrary nonlinear non-Gaussian problems.

Assume that X_t is the state of the dynamic system at time t, observing value is Z_t and the observing sequence is $Z_{1:t} = \{Z_1, \dots, Z_t\}$, then the post probability distribution of X_t is $p(X_t | Z_{1:t})$. Based on Bayesian filter theory, state posterior distribution can be iteratively calculated through state prediction and observing update.

State Prediction： $p(X_t | Z_{1:t-1}) = \int p(X_t | X_{t-1}) p(X_{t-1} | Z_{1:t-1}) dX_{t-1}$ (5)

Observing Update： $$p(X_t | Z_{1:t}) = \frac{p(Z_t | X_t) p(X \mid Z_{1:t-1})}{\int p(Z_t | X_t) p(X \mid Z_{1:t-1}) dX_t} \quad (6)$$

For nonlinear, non-Gaussian dynamic systems, the integral calculation of formula (5) and (6) cannot be analytical calculation. Particle filter converts integral calculation to limited sample points' sum calculation. Then $p(X_t | Z_{1:t})$ can be described approximately as

$$p(X_t | Z_{1:t}) \approx \sum_{j=1}^{N} \omega_{t,j} \delta(X_t - X_{t,j}) \quad (7)$$

N is the number of particles, $\omega_{t,j}$ is the weight of the j th particle at time t, $\delta(\cdot)$ is Dirac function. If the particles are obtained through the sampling of importance distribution $q(X_{t,j} \mid X_{t-1,j}, Z_{1:t})$, then recursive Bayesian estimation process can be treated as the following weights update process:

$$\omega_{t,j} \propto \omega_{t-1,j} \frac{p(Z_t \mid X_{t,j}) p(X_{t,j} \mid X_{t-1,j})}{q(X_{t,j} \mid X_{t-1,j}, Z_{1:t})} \quad (8)$$

If the prior distribution $p(X_t \mid X_{t-1})$ is selected as the importance sampling function, then the weights update process can be simplified as

$$\begin{cases} \omega_{t,j} \propto \omega_{t-1,j} p(Z_t \mid X_{t,j}) \\ \sum_{j=1}^{N} \omega_{t,j} = 1 \end{cases} \quad (9)$$

$p(Z_t \mid X_{t,j})$ represents the j th particle' state observation probability distribution, it can be built using the similarity of the cue distribution of the j th particle with the object particle. So current state $\hat{X}$ can be obtained by the estimation of posterior distribution $p(X_t \mid Z_{1:t})$, that is $\hat{X}_t \approx \sum_{j=1}^{N} X_{t,j}\omega_{t,j}$. The details are in references [16].

3.2. Adaptive Multi-Cues Integration Particle Filter Tracking

In complicated scene, it is difficult to obtain good tracking performance by single visual cue. To further improve tracking robustness, we need integrate other cues. Given the object state X_t at the time t, observation value Z_t and observation likelihood $p(Z_t \mid X_t)$, assuming that the n cues are independent, Z_t can be represented as $Z_t = \{Z_t^1, Z_t^2 \ldots, Z_t^n\}$. The overall observation likelihood is the joint similarity of multiple cues:

$$p(Z_t \mid X_t) = \prod_{i=1}^{n} p(Z_t^i \mid X_t) \quad (10)$$

$$p(Z_t^i \mid X_t) \propto e^{-d_i^2(Z_t^i, T_i)/\sigma^2} \quad (11)$$

T_i is the template for cue i, $d_i(Z_t^i, T_i)$ is the distance between observation Z_t^i and the template T_i, $p(Z_t^i \mid X_t)$ is the observation similarity of cue i. Putting formula (11) into formula (10), we get

$$p(Z_t \mid X_t) = e^{-\sum_{i=1}^{m} \pi_t^i d_i^2(Z_t^i, T_i)/\sigma^2} \quad (12)$$

It can be seen that each cue is distributed equal weight. But in real applications, different cues have different contributions to the ability of identifying the whole object due to the impact of the environment, so different cues should have different weights.

$$p(Z_t \mid X_t) = e^{-\sum_{i=1}^{m} \pi_i d_i^2 (Z_t^i, T_i)/\sigma^2} \qquad (13)$$

Compared with single cue tracking, the multi-cues integration tracking in the formula above is more effective. But the tracking accuracy is not high when there are object's appearance changes or occlusions, because the cues weights remain the same in the whole video sequence without online update. Formula (13) can be revised as:

$$p(Z_t \mid X_t) = e^{-\sum_{i=1}^{m} \pi_t^i d_i^2 (Z_t^i, T_i)/\sigma^2} \qquad (14)$$

In this way, the weight of each cue in the each frame of video sequence is updated dynamically to adapt to the tracking environment and object appearance of the changing model. There are two problems to be solved: one is how to evaluate the weights according to different contribution of each cue in different tracking environment, the other is how to update the weights in a steady and real-time way.

4. Algorithm Proposed by This Paper

This paper proposes a new tracking method of adaptive multi-cue integration particle filter based on fragment. The basic idea is that the initial template is established and is divided into fragments in the initial frame, each corresponding to a set of particles. In current frame fragments are got according to the template division rules, then every fragment is searched, the most similar one to the template is searched out by multi-cue integration particle filter, and then the best position of the object in current frame can be got. After that whether there are occlusions or object appearance changes are determined, template is updated accordingly. The tracking schedule is shown in Figure 4.

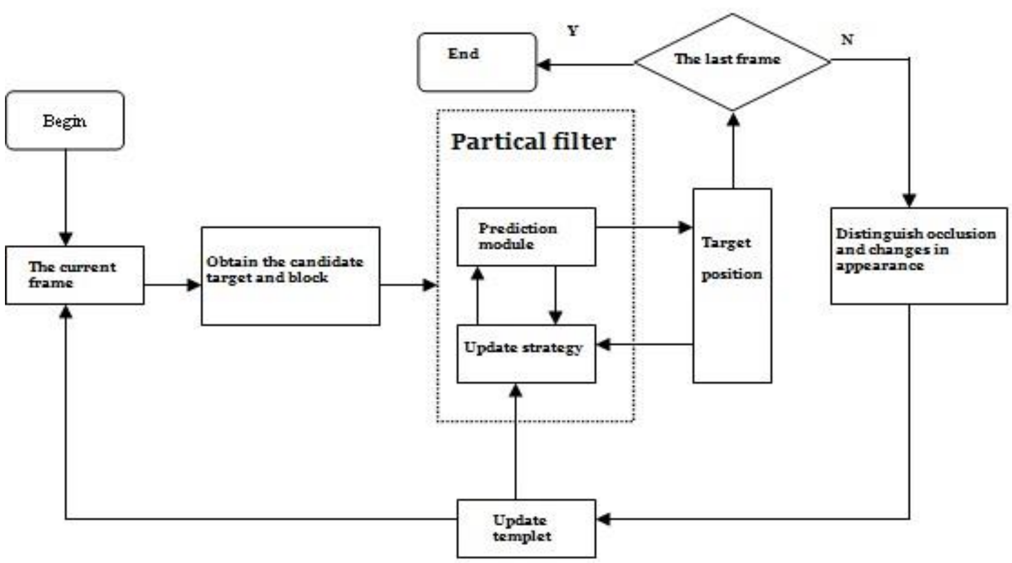

Figure 4. Tracking Flow Charts of the Algorithm

The algorithm proposed by this paper divides the tracking object into 36 fragments randomly. Each fragment corresponds to a particle set $q_k (k = 1, 2, \ldots, 36)$. $\gamma_{t,k}^{i} = \{ S_{t,k}^{i}, \pi_{t,k}^{i} \} (i = 1, \ldots, N)$ are the N particles in set q_k at time t, $S_{t,k}^{i}$ is the particle state, $\pi_{t,k}^{i}$ is the particle weight. By doing so, the particle weights are updated after each time step of particle filtering. Then the fragment position is got by calculating the average of the particles in the particle set. The particle filtering procedure of every fragment is:

1) Selection Procedure: Because the particle number of each particle set is relatively small $(N = 25)$, the sampling process is easily degraded. We do re-sampling at every time step;

2) Prediction Procedure: We use CONDENSATION algorithm for prediction. At time t, we use the tracking results at time $t-1$ as mean to generate a Gaussian distribution of N particles. $\sigma = \lambda \times W_0$ is the variance of Gaussian distribution, λ is a regulatory factor, by regulating the value of λ we can change the size of the distribution region of the new particle;

3) Observation and Evaluation Procedure: We construct the likelihood function through the integration of color, edge and texture cues to fulfill the observation procedure in each fragment. At the same time we measure the occlusions and appearance changes; calculate the similarity between the candidates and the object template by formula (1) to get the best position of the object in the current frame;

4) Template Updating Procedure: We judge whether there are occlusions or appearance changes according to their measurements . If there is occlusions, then go to step 1) directly without template update and start the next frame tracking; otherwise update the template and then go to step 1).

4.1. Multi-cues Integration

In complicated situation, it is difficult to obtain good tracking performance by one single visual cue. To further improve tracking robustness, we need integrate other cues. It is very important to select the strong discriminate cues. This paper uses the integration of color, edge and texture cues to construct likelihood function for the purpose of scattering the impacts of all kinds of changes.

Color is a major cue to describe the object. There have been many studies about the description of object color cues before. Ref. [17] proposed a color histogram to describe the object color cues; it had simple calculation and fast processing, was relatively robust in solving problems such as partial occlusion, rotation, *etc*. This paper uses HSV histogram to describe colors. The expression is

$$p_y^{(u)} = c \sum_{i=1}^{N} k\left(\frac{\| y - x_i \|}{a}\right) \delta [h(x_i) - u] , \quad u = 1, \ldots, m \quad (15)$$

N is the number of pixel, δ is Dirac impulse function, $\| y - x_i \|$ is the distance from pixel x_i to the center, $h(x_i)$ assigns the color of x_i to its corresponding bin in the histogram, c is the normalized constant, u is the histogram index and m is histogram order number.

LBP cue is a kind of nonparametric operator cue for describing texture changes of image local region [18]. Because of its simple calculation, strong discriminate ability and the invariance of scale, rotation and illumination, it caused wide public concern. The formula to calculate the LBP value of each pixel is

$$LBP(x_0) = \sum_{i=0}^{7} s(x_i - x_0)2^i \qquad (16)$$

x_0 is the grey level of the center point, x_i is the gray value of the around sample point and $s(\bullet)$ is step response function.

$$s(x) = \begin{cases} 1 & x >= 0 \\ 0 & x < 0 \end{cases} \qquad (17)$$

In order to keep the rotation invariance, it constantly rotated circular neighborhood to get a series of initial definition of LBP values and took the minimum value as the LBP value of the neighborhood. After getting each pixel decimal LBP code, using the histogram statistics for all pixels it got the following LBP histogram description of the image:

$$t_u = \sum_{i=1}^{N} \delta[h(x_i) - u], \; u = 1, \ldots, m \qquad (18)$$

Edge is another commonly used cue to describe the object. This paper chooses weighted gradient direction histogram under the image gray level information to describe the edge cues of the object [19]. First, we do Gaussian smoothing for image processing regions, uniformly sample pixels $I(x, y)$ within the ellipse and calculate the edge strength G and direction angle θ for each point.

$$G(x, y) = \sqrt{I_x^2 + I_y^2}, \; \theta(x, y) = \tan^{-1}\left(\frac{I_y}{I_x}\right) \qquad (19)$$

Bounded by two axes, the ellipse is divided into four parts. For each part, edge direction angle is quantified to B_e level direction histogram and the histograms are aligned. Second, the edge intensity information for each point is fused to get weighted gradient direction histogram of each part. Finally, the histograms of the four parts are combined and normalized.

For histogram description, Bhattacharyya distance is a commonly used measurement for similarity [20]. $Z_{t,k}$ and $T_{t,k}$ represent the histograms of the k th region after the normalization at current time t and the template respectively. The similarity of the two histograms is

$$D(Z_{t,k}, T_{t,k}) = \sqrt{1 - \sum_{u=1}^{m} Z_{t,k}(u) T_{t,k}(u)} \qquad (20)$$

On this basis, the observation probabilities of the three cues are defined by formula (13). $i \in \{1,2,3\}$, color is the first cue, LBP texture is the second and edge is the third, σ is the function variance.

4.2. Cue Weight Update

In order to adapt to all kinds of environments and object appearance changes, the weight $\{\pi_t^i\}_{i=1}^n$ of each cue in formula (13) in each frame must be updated in real time. The current method is to estimate the weights of the cues according to their contributions in each frame, it has some robustness to real-time changes. But visual tracking is a problem on video sequence; this method will lose the information contained in the consecutive frames. This paper proposes a cue weight updating method based on consecutive frame information; this method can overcome the weight change caused by noise or calculation errors.

In most frames in a video sequence, adjacent consecutive frames have small weight changes. If the cue weight is treated as a state, its changes as the change of the state, then the weight update of each frame can be understood as a tracking problem of states. And for dealing with the nonlinear non-Gaussian motion, particle filter is the most appropriate. At time t for the k th fragment, the weight of cue $j = 1,\ldots,n$ is represented as $\pi_{t,k}^j$. At time $t-1$, assuming that the priori probability distribution $p(\pi_{t-1,k}^j \mid y_{t-1,k}^j)$ is approximately represented as the N weighted particles $p(\pi_{t-1,k}^j \mid y_{t-1,k}^j) \approx \{\pi_{t-1,k}^{j,i}, \tau_{t-1,k}^{j,i}\}_{i=1}^N$, $\tau_{t-1,k}^{j,i}$ is the weight of the *i*th particle, then the posteriori probability for the object can approximately be:

$$p(\pi_{t,k}^j \mid y_{t,k}^j) \approx cp(y_{t,k}^j \mid \pi_{t,k}^j)\sum_i \tau_{t-1,k}^{j,i}\, p(\pi_{t-1,k}^j \mid \pi_{t-1,k}^{j,i}) \qquad (21)$$

Particle filter can be treated as importance sampling function $f(\pi_{t,k}^j)$

$$\pi_{t,k}^{j,s} \sim f(\pi_{t,k}^j) = \sum_r \tau_{t-1,k}^{j,i}\, p(\pi_{t,k}^j \mid \pi_{t-1,k}^{j,i}) \qquad (22)$$

Then the posteriori probability can be expressed as $p(\pi_{t,k}^j \mid y_{1:t,k}^j) \approx \{\pi_{t,k}^{j,s}, \tau_{t,k}^{j,s}\}_{s=1}^N$, *i* and *s* are particle index at time t-1 and t. In this way, we can estimate the weight of each cue in each frame. Weight update is converted into state tracking problem, the weight of each cue update depends not only on the observation of the current frame, and is also associated with the information of adjacent frames. This method avoids the error of the update method based on the current frame information.

4.3. Template Online Update

In order to ensure real-time tracking effect, this paper tracks the object by dividing the object into fragments, judges states of fragments, adjusts updating strategy according to whether there is occlusion or appearance change or not, and tracks moving objects adaptively. It will increase the distance fragment from the template because of occlusion or appearance change, if occlusion the template can't be updated, if appearance change the template need to be updated.

Fragment weights in the template were got in Section 2. After the experiments we found that if $\lambda_{p_t} < 0.5$, the gray scale distribution of q_t changed a lot compared to corresponding p_t, that means there might be occlusions or appearance changes. We call such q_t as invalid fragment. Through the analysis we found that, if the invalid fragment is caused by occlusion, the probability is larger for the fragment gray distribution appearing in the previous frame on the background region or lost region; if it is caused by object's own changes, the probability is larger for the fragment gray distribution appearing in the previous frame on the object region. Because obstacles will appear around the object only, we select a "ring" background region around the object as an aid to determine whether the object is occluded, the area of the annular region is about 2 times of object area. This paper proposes the method of projecting the gray level distribution of the invalid fragment to the object and annular background in the previous frame, to obtain the region in the previous frame where gray levels of the invalid block appeared in current frame and then calculating respectively the probabilities of the invalid fragment belonging to the object region and the background region.

First, in inverse image, we count the number of the foreground pixels in the object area S_0 and the number of the foreground pixels in the whole image S_t. If the probability of the gray of the invalid fragment belonging to the object p_0 satisfies

$$p_0 = \log \frac{S_0}{S_t - S_0} > th \qquad (23)$$

Then we think the object appearance changes, or the object is occluded. th is set to 0.8 after many experiments. If the invalid block is determined as occlusion, the template will not be updated and the location by the particle filter prediction is the location of the fragment; if the invalid block is determined as appearance change, the information of the current frame will be updated to the template. So that we can rapidly get the object change information and suppress background disturbance to the template at the same time, obviously improve the tracking performance.

5. Experiments and Analysis

We implemented the proposed method in Microsoft visual studio 2008 and Opencv2.0, and analyzed the results in MATLAB2008. In order to test the effectiveness of the proposed method, we compare our results with those of FragTracker, the dynamic appearance model (DamTracker) and Partical Filter (PFTracker) algorithms [4, 22]. The robustness of the proposed model is tested on video sequences with various tracking conditions, such as illumination change, occlusions, and non-rigid object's appearance changes. When comparing different methods in tracking errors, the ground truth data are obtained by manually labeling in each frame. In the figures, the red tracking window is our method; the blue one is DamTrackier; the pink one is PFTracker; and the green one is FragTracker. The qualitative analysis is done through observing test images and the quantitative analysis is done through measuring location errors. Location error is defined as

$$err(x_{obj}, y_{obj}) = \sqrt{\frac{(x_{obj} - x_{true})^2 + (y_{obj} - y_{true})^2}{2}} \qquad (24)$$

$err(x_{obj}, y_{obj})$ is the location error of the object location (x_{obj}, y_{obj}) , (x_{true}, y_{true}) is the true location by manual calibration.

5.1. The Experiments and Analysis on the Videos with Occlusions

Test Sequence 1: Figure 5 is the simulation result of a video in a campus named "Face". The video sequence contains 270 384×288-pixel color images. The main difficulty with this sequence is that the target is occluded by another people. When the target is occluded at the video frame 33-39, Shown in Figure 5, the tracking window of method1 had lost the object; method2 didn't lose the object, the estimation location of the object had already deviated from the actual location. In frame 47, the tracking window with frame-by-frame update template lost the object too. The tracking window of the method proposed by this paper tracked the object accurately all the time. It can be seen that the proposed template update method had more robustness to illumination changes.

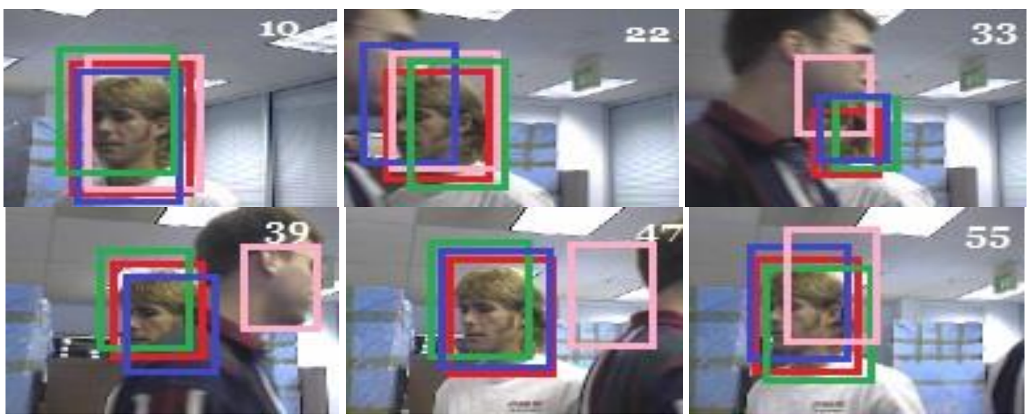

Figure 5. Tracking Result of A "Face1" Video(Frames：10,22,33,39,47,55)

Test Sequence 2: Figure 6 is the simulation result of video "girl" which comes from CAVIAR database. The Occluded Face 2 and the Girl sequences show the target objects in occlusion and in different poses. For these sequences, our method generally gives better results than the others. In the Girl sequence, our method with the variable scale model handles the changes in the target's scale as well. The video sequence contains 600 384×288-pixel color images, where the girl's head in the sequence has irregular translation and rotation movements.. In this video, object was almost completely occluded for a time. You can see the object begin to be occluded from the video frame 89-105, before that, the method proposed by this paper more accurately tracked the object than the other two methods; when occlusion appeared, method1 lost the object at frame 436, to frame 441. The method2 lost the object too, while the metod1 found the missing object again. The method proposed by this paper can track the object accurately when there is no occlusion, will not lose the object after occlusion appears and can re-track the object accurately when occlusion disappears.

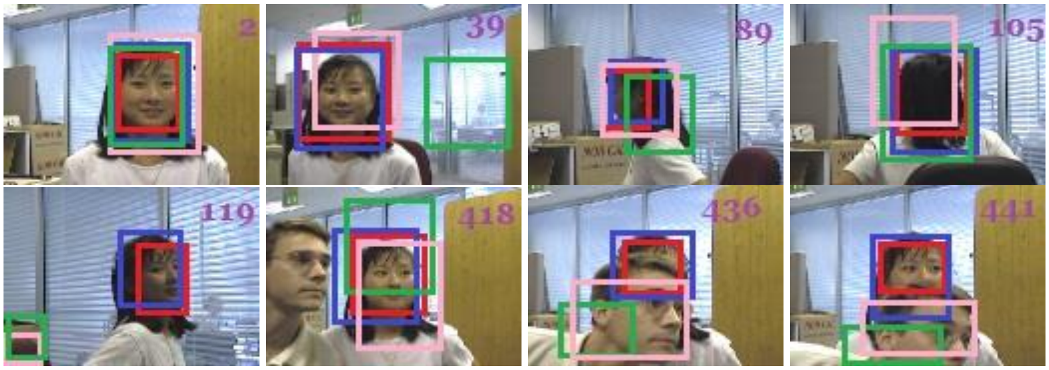

Figure 6. Tracking Result of Video “girl”(Frames：2,39,89,105,119,418,436,441)

Test Sequence 3: Figure 7 is the simulation result of video “car” which comes from SPEVI database. The video sequence contains 250 320×240-pixel color images. In the video, from frame 99 to frame 183, the target enters into dark region from bright region and later returns to the bright region. The illumination has a great change. At frame 119, the tracking window of the fragment-based tracker loses the target; the color-texture based mean-shift tracker doesn’t lose the target, but the estimation location of the target has already deviated from the actual location. At frame 183, the tracking window of the color-texture based mean-shift tracker loses the target too while the tracking windows of our method and the decentralized template tracker never lose the target.

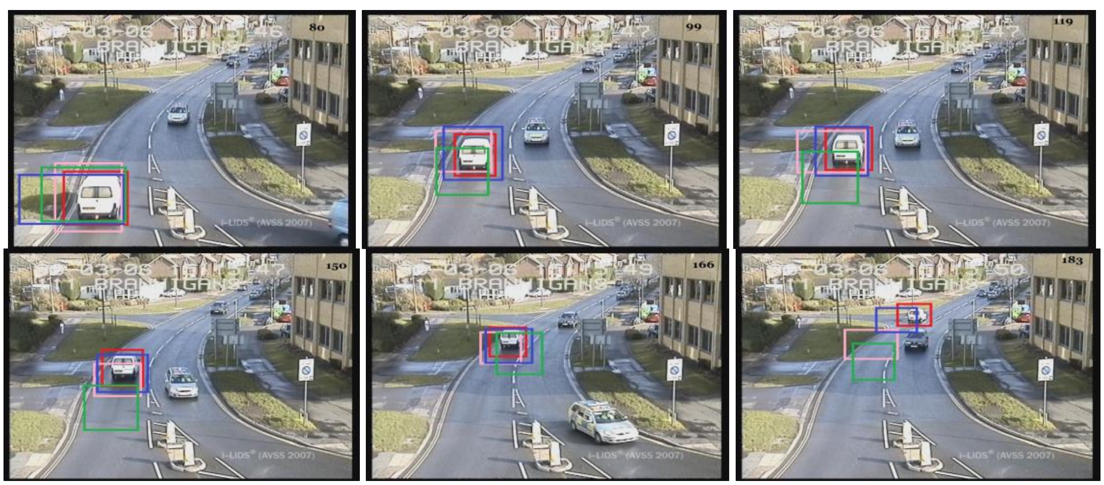

Figure 7. Tracking Result of Video “car”(Frames：80,99,119,150,166,183)

Test Sequence 4: Figure 8 is the simulation result of video “man2” which comes from SPEVI database. The video sequence contains 800 320×240-pixel color images. the target person undergoes significant pose, expression and appearance changes. Our method successfully deals with these changes during tracking. The FragTrack algorithm, on the other hand, gives poor results. The Girl with Many Eyes sequence shows the target object in severe occlusion. From the given results, we see that the proposed algorithm outperforms the FragTrack algorithm. Accurate tracking of the target under a high degree of potential occlusion is achieved as a result of our adaptive formulation for the multi-cue integration.

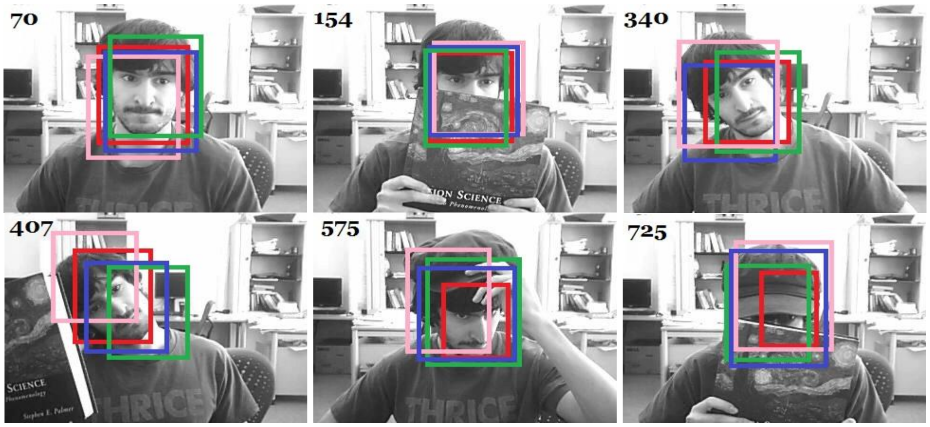

Figure 8. Tracking Result of Video "Face2"(Frames: 70,154,340,407,575,725)

5.2. The Quantitative Analysis

Figure 5-8 show the qualitative analysis result. Figure 9 shows the error plots of each sequence for quantitative analysis. It can be seen from the error curves of "girl" sequence in figure 9, error values of method1 and method2 are significantly greater than our method. Especially in the frame 55, the error value of method1 has reached 150, error value of method2 is nearly 50. But the error value of our method is only 35 or so. As for frame 500, the error value of method1 is still very high while the error of the other two methods is significantly lower especially error of method2 because these two methods adopt adaptive multi-cues integration strategy.

It can be seen from error curves of "face 1" sequence, tracking error were similar of the three methods in absence of occlusion and partial occlusion and the error is small, 20 or so. But at frame 47 with full occlusion, the tracking error of method1is very big, 55, and the tracking error of the other two methods is between 5 and 30, because these two methods adopt multi-cues integration strategy to keep not losing the object. After occlusion, the error of method1 decreases while that of method2 has been rising owing to template drift. The error of our method is smaller always.

As what can be seen from error curves of "car" sequence, at frame 183, owing to the light changes much, method1 using fixed template has no effect, the error is large about 150, while the error of method2 is 50 and the error of our method is 60. From the error curves of "car" we can see the error of the three methods have big fluctuation, the error at frame 150 and 183 are larger, because there are illumination changes, occlusions and appearance changes. But in comparison, the error of our method is much smaller than the other two, because this algorithm adopts the fragment and real-time update template strategy, and can distinguish from the occlusion to appearance change to update template.

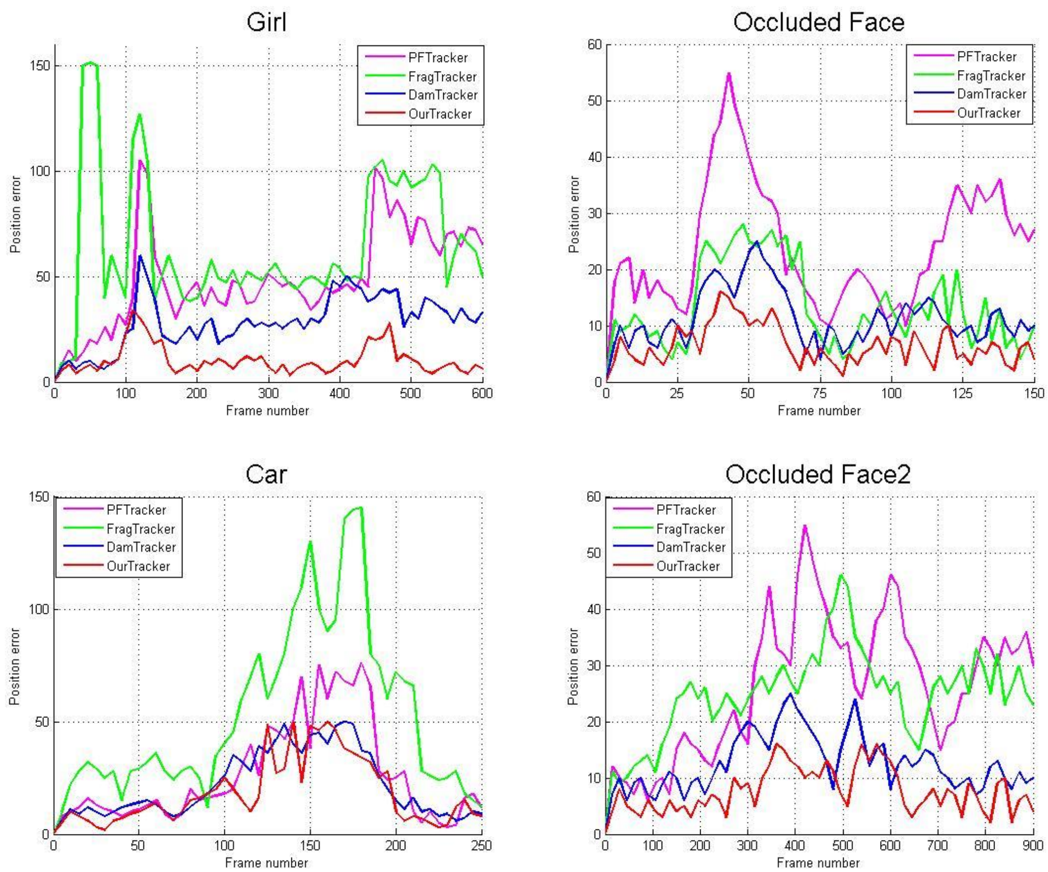

Figure 9. Error Plots for the Video Sequences used in the Quantitative Analysis

6. Conclusions

Video object tracking algorithm has encountered many problems, such as illumination change, occlusions, appearance changes, etc. An adaptive cues integration algorithm has been proposed in this paper. The color, texture and edge characteristics organically unify in together. Combined with fragment particle filter method, the particle filter is designed based on dynamic weight updating. Furthermore, the algorithm uses the template updating for object tracking, designs the corresponding template update algorithm, and successfully implements the object tracking of the complex background. This algorithm not only effectively reduces the tracking number of particles which are needed to improve the operation efficiency, but also largely improves the tracking accuracy, especially for occlusions and appearance changes to keep the good tracking effect. As what can be seen from the results of the experiments, this algorithm has overcome the shortcomings of the existing multi-cues integration fragment tracking algorithm, and has got a satisfactory tracking result as to the video of the test in the complex background of illumination, occlusion, and the appearance change of object tracking.

Particle filter tracking algorithm is a popular one, and methods of this article is based on the development. But the dynamic multi-cues integration framework proposed in this paper can also be extended to other tracking algorithms; therefore, the core thought can be improved. To design a more robust dynamic model suitable for more tracking algorithms is a challenging task in the future work. Also this algorithm does not take changes into account in the object dimension, which is subject to our further research.

Acknowledgements

This work was supported by the National Natural Science Foundation of china(Grant No.31371533),National Science and Technology Support Program(Grant No. 2013BAJ10B12), the Key Technologies R & D Program of AnHui Province (Grant No.1301032169) and by the Natural Science Foundation of AnHui Province(Grant No.1308085MF89).

References

[1] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms", Proc. Conf. Computer Vision and Pattern Recognition (CVPR), **(1998)**, pp. 232–237.
[2] J. Triesch and C. von der Malsburg, "Democratic integration: self-organized integration of adaptive cues", Neural Comput., vol. 13, no. 9, **(2001)**, pp. 2049-2074.
[3] E. Maggio, F. Smeraldi and A. Cavallaro, "Combining colour and orientation for adaptive particle filter-based tracking", Proc. British Machine Vision Conf., **(2005)**, pp. 659–668.
[4] "DAMTracker: Yuru Wang, Xianglong Tang, Qing Cui, Dynamic Appearance Model for Particle Filter Based Visual Tracking", Pattern Recognition, vol. 45, no. 12, **(2012)**, pp. 4510-4523.
[5] J. Q. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking", IEEE Transaction on Image Processing, vol. 17, no. 2, **(2008)**, pp. 235-240.
[6] G. Jia, Y. Tian and Y. Wang, "Dynamic multi-cue tracking with detection responses association", Proceedings of the ACM Multimedia, Firenze, Italy, **(2010)**, pp. 1171–1174.
[7] S. M. S. Nejhum, J. Ho and M.-H. Yang, "Visual tracking with histograms and articulating blocks", Proc. Conf. Computer Vision and Pattern Recognition, **(2008)**, pp. 1–8.
[8] K. Nickel and R. Stiefelhagen, "Dynamic integration of generalized cues for person tracking", Proc. Eur. Conf. Computer Vision, **(2008)**, pp. 514-526.
[9] A. Adam, E. Rivlin, I. Shimshoni, "Robust fragments-based tracking using the integral histogram", Proc. Conf. Computer Vision and Pattern Recognition, **(2006)**, pp. 798–805.
[10] E. Erdem and S. Dubuisson, "Fragments based tracking with adaptive cue integration", Computer vision and image understanding, **(2012)**, pp. 827-841.
[11] F. Porikli, O. Tuzel and P. Meer, "Covariance tracking using model update based on Lie algebra", Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York, USA, IEEE, **(2006)**, pp. 728-735.
[12] R. T. Collins, Y. X. Liu and M. Leordeanu, "Online selection of discriminative tracking features", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, **(2005)**, pp. 1631-1643.
[13] W. Yong-Zhong, L. Yan, Z. Chun-Hui and P. Quan, "Kernel based tracking based on adaptive fusion of multiple cues", Acta Automatica Sinica, vol. 34, no. 4, **(2008)**, pp. 393-399.
[14] J. Carpernter, P. Clifford and P. Fearnbead, "Improved particle filter for nonlinear problems", IEEE Proceeding, vol. 17, no. 4, **(1999)**, pp. 439-447.
[15] G. Dong and W. Xiaodong, "Quasi- Monte Carlo filtering in nonlinear dynamic systems", IEEE Transactions on Signal Processing, vol. 54, no. 6, **(2006)**, pp. 2087-2098.
[16] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking", International Journal of Computer Vision, vol. 29, no. 1, **(1998)**, pp. 5-28.
[17] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, **(2003)**, pp. 564–577.
[18] O. Timo, P. Matti and M. Top, "Multi-resolution gray scale and rotation invariant texture classification with local binary patterns", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, **(2002)**, pp. 971-987.
[19] C. Yang, R. Duraiswami and L. Davis, "Fast multiple object tracking via a hierarchical particle filter", Proceedings of the Tenth IEEE International Conference on Computer Vision, Beijing, China, **(2005)**, pp. 212–219.
[20] P. Perez, J. Vermaak and A. Blake, "Data fusion for visual tracking with particles", Proceedings of the IEEE, vol. 92, no. 3, **(2004)**, pp. 495-513.
[21] Z. Wang, X. Yang, Y. Xu, S. Yu, "CamShift guided particle filter for visual tracking", Pattern Recognition Letters, vol. 30, no. 4, **(2009)**, pp. 407-413.

