Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Object Tracking Based on the Combination of Learning and Cascade Particle Filter

Hanjie Gong, Cuihua Li, Pingyang Dai and Yi Xie
Department of Computer Science
Xiamen University
Xiamen, China
dxawicso@gmail.com chli@xmu.edu.cn pydai@xmu.edu.cn csyxie@xmu.edu.cn

***Abstract*—The problem of object tracking in dense clutter is a challenge in computer vision. This paper proposes a method for tracking object robustly by combining the online selection of discriminative color features and the offline selection of discriminative Haar features. Furthermore, the cascade particle filter which has four stages of importance sampling is used to fuse two kinds of features efficiently. When the illumination changes dramatically, the Haar features selected offline play a major role. When the object is occluded, or its rotation angle is very large, the color features selected online play a major role. The experimental results show that the proposed method performs well under the conditions of illumination change, occlusion, object scale change and abrupt motion of object or camera.**

***Keywords*—cascade particle filter, online selecting, offline learning, object tracking**

I. Introduction

Many powerful algorithms have been proposed for object tracking. They can be classified into four classes: Region-based methods[1][2][3], Feature-based methods[4][5][6][7], Deformable-template-based methods[8][9][10][11] and Model-based methods[12][13][14]. However, most of these traditional tracking approaches depend on some expensive assumptions. For example, they assume that motion and appearance are continuous and that the representative fixed features can always distinguish the interested objects from background well. The representative fixed features are selected before the tracking task starts. Unfortunately, motion and appearance continuity is often not satisfied because of an abrupt motion of the object or the camera. An object detector trained offline can be integrated into the tracker to solve this problem. Moreover, the fixed features cannot always distinguish the objects from the background well. There are two reasons for this: the object appearance will change when the illumination changes, occlusion happens or viewpoint varies; and the background will change as the target object moves from place to place. The remedy for the drawback of fixed features is using online selection of discriminative features for object tracking. For example, Collins et al[15] proposed a method in which a feature evaluation mechanism is embedded in a mean-shift tracking system that adaptively selects the top-ranked discriminative features for tracking. Jianyu Wang et al[16] online selected discriminative features from a set of Haar features into the appearance model for tracking. However, the problem of an online learning system is obvious: setting the online features updating ratio may be very difficult because the over updating may even ruin the original model. It can be solved by combining offline learning and online learning. For example, Yuan Li et al[17] proposed a cascade particle filter with discriminative observers of different Lifespans. The features selected online can represent object appearance more specifically, while the features selected offline can produce more accurate result.

In order to keep the tracker robust to the background clutters and the abrupt motion of both object and camera, this paper has proposed a method based on combining learning and cascade particle filter. According to the type of the object, we use the pictures containing many postures of the object as the positive samples, and the pictures without containing the object as the negative samples to train a cascade AdaBoost classifier [18] before the tracking task starts. Then we online select the discriminative color features to represent the object appearance. Color features are chosen since they are relatively insensitive to variations in object appearance due to viewpoint, occlusion, and nonrigidity. Finally we use a cascade particle filter which has four stages of importance sampling to efficiently fuse the features which are obtained by offline learning and online learning. Each stage of importance sampling uses an observation to amend prior probability. The three color features selected online and the classifier cascade trained offline correspond to the four observations. Although this paper only considers color features, the proposed approach can be extended easily to other features such as texture, shape and motion.

The rest of the paper is organized as follows: The strategy of selecting color features online is described in section Ⅱ. Section Ⅲ describes the algorithm of the cascade particle filter. The experimental results are presented in section Ⅳ and the conclusion is drawn in section Ⅴ.

II. Online Selection of Discriminative Color Features

In this section, we employ a method to evaluate the discriminative ability of color features. Top-three-ranked discriminative color features are selected as three observations in cascade particle filter for tracking.

A. The Color Feature Set

The linear combinations of R, G, B pixel values compose the set of candidate color features[15], which can enlarge the differences between the color distribution of object and the color distribution of background.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

Denote the feature set as

$$F = \{w_1 R + w_2 G + w_3 B \mid w_1, w_2, w_3 \in \{-2, -1, 0, 1, 2\}\} \quad (1)$$

where w_1, w_2, w_3 are the coefficients.

B. Online selection of discriminative color featurens

Recently learning techniques have been introduced into tracking problems. The tracking is viewed as a classification problem for separating object from background. The current background patches are the only negative examples. As we know that particle filter [20] results in many particles corresponding to background areas, which can be treated as the negative examples in this object/background classification task.

We use the object window gained from the previous frame as a positive sample. We regard the $K-1$ particle-representative windows also sampled from the previous frame as $K-1$ negative samples. The $K-1$ particle-representative windows do not intersect the object image window (see Figure 1).

We denote each feature in F as f_j. For f_j, $K-1$ negative examples and a positive example are obtained. Suppose the center coordinate of positive sample is $X_{pos} = (x_{pos}, y_{pos})$, the radius is $H = (h_x, h_y)$, the pixels location in the window of positive sample are $X_m = (x_m, y_m), m = 1 \ldots n_h$, where n_h is the total number of pixels in the window of positive sample. Then the color distribution in the window of the positive sample under f_j is given by:

$$p_u^j(X_{pos}) = \frac{\sum_{m=1}^{n_h} k_e\left(\left\|\frac{X_{pos} - X_m}{H}\right\|\right)\delta[b_j(X_m) - u]}{\sum_{m=1}^{n_h} k_e\left(\left\|\frac{X_{pos} - X_m}{H}\right\|\right)}, u = 1..l \quad (2)$$

where δ is the Kronecker delta function. The $k_e(\bullet)$ is defined as $k_e(\|X\|) = \begin{cases} 1 - \|X\|^2 & \text{if } \|X\| < 1 \\ 0 & \text{otherwise} \end{cases}$. The function $b_j : R^2 \to \{1 \ldots l\}$ associates to the pixel at location X_m the index $b_j(X_m)$ of the histogram bin corresponding to the color of that pixel under f_j.

Similarly, we can get the color distribution in the window of the k^{th} negative sample under f_j: $p_u^j(X_k)$, $u = 1 \ldots l$, where X_k is the center coordinate of the k^{th} negative sample. For each f_j described in subsection Ⅱ.A , we can evaluate its discrimination ability as below:

$$R(f_j) = \min(d^j_{pos,k}), k = 1 \ldots K-1 \quad (3)$$

where $d^j_{pos,k}$ is the Bhattacharyya distance[19] between the color distribution of the k^{th} negative sample and the color distribution of positive sample under f_j. $d^j_{pos,k}$ is given by:

$$d^j_{pos,k} = \sqrt{1 - \sum_{u=1}^{l} \sqrt{p_u^j(X_{pos}) p_u^j(X_k)}} \quad (4)$$

In descending order of $R(f_j)$, the top-three-ranked features are selected as three observations in the cascade particle filter for tracking.

III. CASCADE PARTICLE FILTER

Particle filter [20] is very successful for non-linear and non-Gaussian estimation problems. The key idea is to represent the required posterior by a set of random samples with associated weights $\{X_t^i, w_t^i \mid i = 1, \ldots, N\}$. Particle Filter uses the dynamical models and the likelihood model to propagate the random set over time. In this section we describe the cascade particle filter[17] which can efficiently fuse the top-three-ranked features chosen in section Ⅱ and the cascade AdaBoost classifier trained offline.

A. Dynamic Model

Let $X_t^i = [x_t^i, y_t^i, width_t^i, height_t^i]^T$ be the state of particle i in the t^{th} frame. In each frame, we calculate the average speed of α previous frames as the speed of current frame. The dynamic model in our experiments is given by:

$$x_t^i = x_{t-1}^i + \frac{1}{\alpha}(x_{t-1}^i - x_{t-m}^i) + bx * N(0,1) \quad (5)$$

$$y_t^i = y_{t-1}^i + \frac{1}{\alpha}(y_{t-1}^i - y_{t-m}^i) + by * N(0,1) \quad (6)$$

$$width_t^i = width_{t-1}^i + N(0,1) \quad (7)$$

$$height_t^i = \frac{width_t^i}{width_{t-1}^i} height_{t-1}^i \quad (8)$$

where bx and by are constants known as the spread radius of particles (We set bx =15, by =15 in our experiments). Formula (8) is obtained when we restrict that the aspect ratio of the object remains unchanged during the tracking.

B. Likelihood Model

The selected three features described in section Ⅱ correspond to three observations: $z_{1,t}$, $z_{2,t}$, $z_{3,t}$. According to the type of object, we use the pictures containing many postures of the object as the positive samples, and the pictures without the object as the negative samples to train a cascade AdaBoost classifier [18] beforehand. Then we use the cascade AdaBoost classifier as the fourth observation $z_{4,t}$.

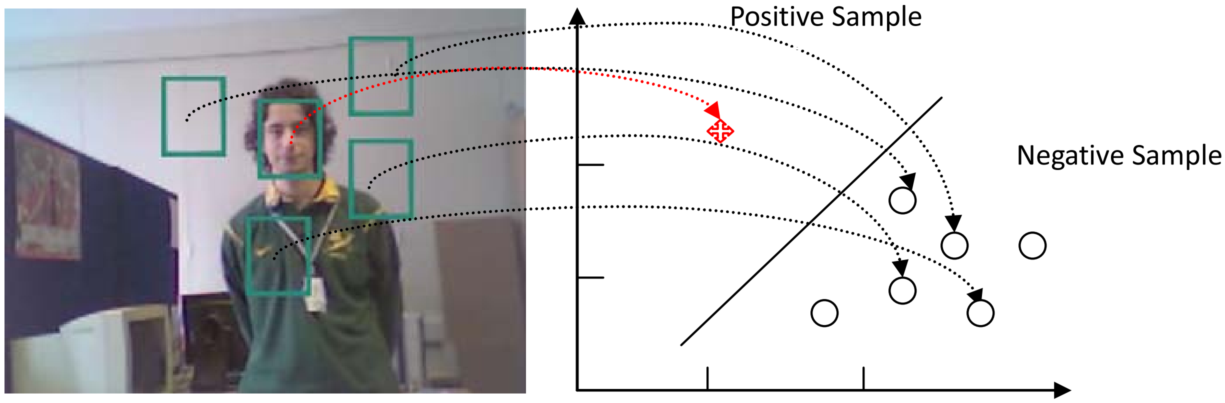


Figure 1. Demonstrating the process of sampling from object and background(image from [22])

The likelihood of observations 1~3 is defined as:

$$p(z_{m,t} \mid X_t^i) \propto \frac{1}{1+\exp(-d_{object,i}^{j})}, \quad m=1,2,3 \tag{9}$$

where $d_{object,i}^{j}$ is the Bhattachayrya distance between the color distribution of the candidate represented by the particle i and the color distribution of object under f_j (see section II).

Suppose the cascade AdaBoost classifier trained offline has h stages. The likelihood of observation 4 is defined as:

$$p(z_{4,t} \mid X_t^i) \propto \frac{h'/h}{1+\exp(-\sum_{k=1}^{K}\alpha_k h_k(X_t^i)/\sum_{k=1}^{K}\alpha_k)} \tag{10}$$

where h' denotes the number of stages that the candidate window passes. The candidate window is represented by the particle i. And $\sum_{k=1}^{K}\alpha_k h_k(X_t^i)$ denotes the final strong classifier which the candidate window passes.

C. Four Stages Of Importance Sampling

We use four stages of importance sampling to fuse the four observations efficiently. The first stage is that we amend the priori probability by applying the observation $z_{1,t}$. Similarly, the m^{th} stage is that we amend the posterior probability amended by $z_{1,t}$ … $z_{m-1,t}$ by applying the observation $z_{m,t}$ (m=2,3,4). The method can be described as follows.

The priori probability represented by the particle set $\{X_{t-1}^i, w_{t-1}^i \mid i=1,\dots,N\}$ is obtained in the $t-1^{th}$ frame. In the t^{th} frame, the algorithm of the cascade particle filter is described in Figure 2.

Re-sampling:

Simulate $X_{t-1}^i \sim \{X_{t-1}^i, w_{t-1}^i \mid i=1,\dots,N\}$, and replace $\{X_{t-1}^i, w_{t-1}^i \mid i=1,\dots,N\}$ with $\{X_{t-1}^i, 1/N \mid i=1,\dots,N\}$.

Prediction:

Use formula (5), (6), (7), (8) to propagate particles to generate predicted states.

Stage 1:

We denote $\{X_t^i, w_t^i \mid i=1,\dots,N\}$ as $\{X_{1,t}^i, w_{1,t}^i \mid i=1,\dots,N\}$

For $i=1\dots N$, let $w_{1,t}^i = w_{1,t}^i p(z_{1,t} \mid X_{1,t}^i)$.

Normalize weight so that $\sum_{i=1}^{N} w_{1,t}^i = 1$.

For stage m=2,3,4:

Re − sampling : simulate $X_{m,t}^i \sim \{X_{m-1,t}^i, w_{m-1,t}^i \mid i=1\dots N\}$,

and replace $\{X_{m-1,t}^i, w_{m-1,t}^i \mid i=1\dots N\}$

with $\{X_{m,t}^i, w_{m,t}^i \mid i=1\dots N\}$, where $w_{m,t}^i = 1/N$.

For $i=1\dots N$, let $w_{m,t}^i = w_{m,t}^i p(z_{m,t} \mid X_{m,t}^i)$.

Normalize weight so that $\sum_{i=1}^{N} w_{m,t}^i = 1$.

Output object status:

The ultimate status of object in current frame is given by

$$\hat{X}_t = \sum_{i=1}^{N} w_{4,t}^i X_{4,t}^i \tag{11}$$

We denote $\{X_{4,t}^i, w_{4,t}^i \mid i=1,\dots,N\}$ as $\{X_t^i, w_t^i \mid i=1,\dots,N\}$.

Figure 2. Algorithm of cascade particle filter

D. Steps of the Our Algorithm

The steps of our method can be described as follows:

Step 1: Offline classifier training

According to the type of object, we train a cascade Ada-boost classifier to distinguish object from nonobject.

Step 2: Initialization

First we use the classifier obtained from **Step 1** to detect the object in the first frame. Then we get the size and the location of object according to which we can initialize the status of particles. Then we chose the three color features according to section Ⅱ.

Step 3: Algorithm of cascade particle filter

As described in subsection Ⅲ.C, we use cascade particle filter to efficiently fuse the four observations.

Step 4: Online selection of discriminative color features

Every p frames top-three-ranked features are selected according to section Ⅱ.

Go to step3

IV. Experimental Results

The algorithm is implemented in C++ and runs on a PC with Conroe2 2.0GHz CPU. We use OpenMP to parallelize the code. The tracker has been evaluated on some video sequences. Two results of tracking are shown as follows.

A. Tracking Boat

We prepared a video named "boat.avi". From Figure 4, we can see that when the ship in the 662nd frame has been occluded, Camshift algorithm converged to rock bottom. Because Camshift algorithm is based on the color feature and the color distribution of rock bottom is similar to the ship. And it can be seen from Figure 5 that the online selection of color features which enlarges the difference between the color distribution of the background and the color distribution of object performs good.

B. Tracking Face

The video named "toni.avi" is available at [22]. The main challenges of the video arise from that the face in the video rotates 360 degrees, the background has similar color with the face, and light changes significantly from darkness to light. From Figure 6 we can see that because the background has similar color with the object and there is a strong illumination change during tracking, Camshift algorithm basically cannot keep up with the object. Figure 7 shows that the color-based particle filter fails to track the object due to dramatic changes in light intensity from # 210 to # 230 and from # 285 to # 405.

From Figure 8 , Figure 3 and TABLE 1, we can see that our approach has better performance than Camshift algorithm and particle filter based on color. Except that from # 248 to # 256 and from # 362 to # 365 object has 180 degrees rotation and is completely occluded, our approach keeps up with the object with high-precision. The success rate reaches 95.8025%.

V. Conclusion and Future Work

The success of object tracking depends on the degree of discrimination between object and background, which is related to the features used in the tracker. The features selected online can represent object appearance more specifically, while the features selected offline can produce more accurate result. This paper has proposed a method to track object robustly by combining the online selecting discriminative color features and offline selecting discriminative Haar features. And the cascade particle filter is used to integrate the features efficiently. The experimental results show that the proposed approach has better performance than Camshift algorithm and particle filter based on color.

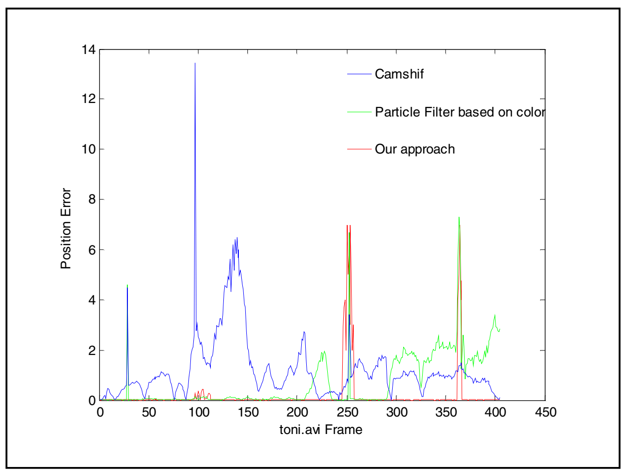
In future work, we will extend the proposed approach to other features such as texture, shape and motion.

TABLE I. The effect of tracking by different algorithms on toni.avi.

Algorithm	*Mean Pos Err*	*Success Rate*
Camshift	1.216439	24.4444%
Particle Filter based on color	0.687926	66.9136%
Our approach	0.195788	95.8025%

a. The criterion of successful tracking is that the position error is smaller than 0.5

Figure 3. Position error curves of different methods



Acknowledgment

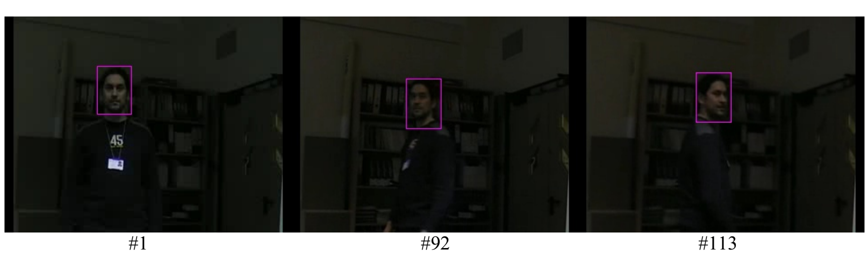
This paper is supported by the National Grand Fundamental Research 973 Program of China （Grant No.2007CB311005）, Natural Science Foundation of Fujian Province of China (Grant No. A0710020) and the 985 Inovation Project on Information Technique of Xiamen University (2004-2009).

Correspondence：Prof. Li Cuihua，chli@xmu.edu.cn

Figure 4. Tracking by Camshift implemented by Opencv[21]

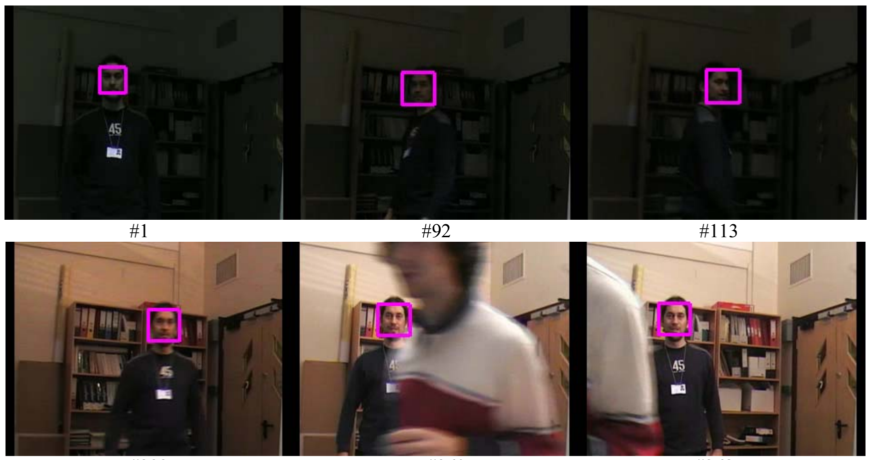
Figure 5. Tracking by our approach

Figure 6. Tracking by Camshift implemented by Opencv[21]



#292 #360 #368

Figure 7. Tracking by Traditional Particle Filter based on color



#292 #360 #368

Figure 8. Tracking by our approach

REFERENCES

[1] Aggarwal J K, Nandhakumar N, "On the computation of motion from sequences of images - A review," Proceedings of the IEEE, 1988, 76(8): 917~935.

[2] Meyer F, Bouthemy P, "Region-based tracking using affine motion models in long image sequences," CVGIP: Image understanding, 1994, 60(2): 119~140.

[3] Hager G, Belhumeur P, " Efficient region tracking with parametric models of geometry and illumination," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(10): 1025~1039.

[4] Kaneko T, Hori O, "Feature selection for reliable tracking using template matching," Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'03. Madison, Wisconsin, USA: IEEE Press, 2003. 1: 796~802.

[5] MItra P, Murthy C, Pal S, "Unsupervised feature selection using feature similarity," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(3): 301~312.

[6] Tissainayagam P, Suter D, "Object tracking in image sequences using point feature," Pattern Recognition, 2005, 38(1): 105~113.

[7] Nickels K, Hutchinson S, " Estimating uncertainty in SSD-based feature tracking," Image and Vision Computing,2002, 20(1): 47~58.

[8] Yu Z, Jain A K, Dubuisson-Jolly M, "Object tracking using deformable templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(5): 544~549.

[9] Kass M, Witkin A, Terzopoulos D, "Snakes: Active contour models," International Journal of Computer Vision,1988, 1(4): 321~331.

[10] Caselles V, Kimmel R, Sapiro G, "Geodesic active contours ,"International Journal of Computer Vision, 1997,22(1): 61~79.

[11] Paragios N, Deriche R, "Geodesic active contours and level set for the detection and tracking of moving objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(3): 266~280.

[12] Drummond T, Cipolla R, " Real-time visual tracking of complex structures," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 932~946.

[13] Gerard P, Gagalowicz A, "Three dimensional model-based tracking using texture learning and matching," Pattern Recognition Letters, 2000, 21(13-14): 1095~1102.

[14] Jung S K, Wohn K Y, "A model-based 3-D tracking of rigid objects from a sequence of multiple perspective views,"Pattern Recognition Letters, 1998, 19(5-6): 499~512.

[15] R.T. Collins, Y. Liu, and M. Leordeanu, " Online Selection of Discriminative Tracking Features," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1631-1643, Oct. 2005.

[16] J. Wang, X. Chen, and W. Gao, " Online Selecting Discriminative Tracking Features Using Particle Filter," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2005.

[17] Yuan Li, Haizhou Ai, Takayoshi Yamashita, Shihong Lao , and Masato Kawade, "Tracking in Low Frame Rate Video:A Cascade Particle Filter with Discriminative Observers of Different Life Spans," IEEE Trans. Pattern Analysis and Machine Intelligence, VOL. 34, NO. 10, OCTOBER 2008.

[18] P. Viola and M. Jones. Robust real-time object detection. In IEEE Workshop on Statistical and Theories of Computer Vision, 2001.

[19] T.Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," IEEE Trans. Commun. Tech., COM_15:52-60,1967.

[20] M. Isard, A. Blake.CONDENSATION-Conditional Density Propagation for Visual Tracking.Int. Journal of Computer Vision, 29(1), pp.5--28, 1998.

[21] Intel opencv library. http://www.sourceforge.net/projects/opencvlibrary.

[22] http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html