MEAN-SHIFT BLOB TRACKING WITH ADAPTIVE FEATURE SELECTION AND SCALE ADAPTATION

Dawei Liang[1, 2], Qingming Huang[2, 3], Shuqiang Jiang[3], Hongxun Yao[1], Wen Gao[4]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China
[2]Graduate School of Chinese Academy of Sciences, and [3]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[4]Institute of Digital Media, Peking University, Beijing, 100871, China

ABSTRACT

When the appearances of the tracked object and surrounding background change during tracking, fixed feature space tends to cause tracking failure. To address this problem, we propose a method to embed adaptive feature selection into mean shift tracking framework. From a feature set, the most discriminative features are selected after ranking these features based on their Bayes error rates, which are estimated from object and background samples. For the selected features, a criterion is proposed to evaluate their stability for tracking and to guide feature reselection. The selected features are used to generate a weight image, in which mean shift is employed to locate the object. Moreover, a simple yet effective scale adaptation method is proposed to deal with object changing in size. Experiments on several video sequences show the effectiveness of the proposed method.

Index Terms— Visual Tracking, Mean Shift, Feature Selection, Bayes Error Rate, Scale Adaptation

1. INTRODUCTION

Visual tracking has been a hot research topic in the past decade, since it is a core component in many computer vision applications ranging from video surveillance, human computer interaction, traffic monitoring to robotics. Among a large body of tracking algorithms, mean shift algorithm has gained much attention due to its computational efficiency and its robustness to non-rigid deformation. The algorithm was first introduced in the seminal work of Fukunaga and Hostetler [1] in 1975, and was almost neglected until Cheng's paper [2] rekindled interest in it. Mean shift is a nonparametric density gradient estimation approach to local mode seeking. Through iteratively shifting kernel window towards current mean location, local mode can be sought at last. In blob tracking scenario, tracking is performed by running mean shift on the weight image which is generated either explicitly (e.g. Bradski [3]) or implicitly (e.g. Comaniciu et al. [4]). Both methods derive the weight image from color histogram in a fixed color space. However, a color space working well in one condition may not perform properly in other conditions, especially when surrounding background continuously changes during tracking.

Recently, Collins et al. [5] proposed to online select discriminative tracking features from linear combinations of RGB values. In the previous frame, the features are ranked according to two-class (i.e. foreground vs. background) variance ratio and the top *N* features are selected. In the current frame, each selected feature produces a weight image, in which mean shift is employed to locate the object. The median location is selected as the final object location. By treating tracking as a foreground/background discrimination problem, the method can adapt to appearance changes of the tracked object and surrounding background. Therefore, the weight image more suitable for tracking can be generated. However, the scale problem is not taken into account in this approach. When the object changes in size, some background samples will be involved in the object sample set and vice versa. This may cause tracking failure. Moreover, performing feature selection in every frame is inefficient for a large feature set. Though they deal with this problem by selecting features every tenth frame, this is also dangerous when the appearance of the tracked object or surrounding background changes remarkably in between two adjacent feature-selection frames.

In this paper, we extend the work of Collins et al. by introducing adaptive feature selection and scale adaptation. Moreover, a new feature selection method based on Bayes error rate is proposed.

2. THE PROPOSED APPROACH

First, Adaptive feature selection is introduced in section 2.1. Then, Weight image generation is provided in section 2.2. Finally, scale adaptation and tracking algorithm are detailed in section 2.3.

2.1. Adaptive Feature Selection

This work is supported by National Hi-Tech R&D Program (863 Program) of China under grant No. 2006AA01Z117.

The feature set [5] which consists of linear combinations of RGB values is adopted. The coefficients of RGB are given by 3-tuple set $\{(c_1,c_2,c_3)^T \mid c_1,c_2,c_3 \in \{-2,-1,0,1,2\}\}$ .After discarding redundant features; we are left with a set of 49 features. We denote it as F . All features are scaled into the range from 0 to 255 and further uniformly discretized into histograms of m ($m = 32$ in our experiments) bins.

The center-surround approach is used to sample pixels from the object and background. A $w \times h$ rectangular set of pixels covering the object is selected as object pixels, while a larger surrounding ring of pixels with the width of $0.5 \times \max(w,h)$ is selected as background pixels. To eliminate the influence of background pixels when the object can not be accurately represented as a rectangle, only those pixels whose weights (see section 2.2 for details) are above some threshold (0.5 in our experiments) are selected as object pixels. Given a feature $f \in F$, denote p_f^t as the normalized feature histogram of the object and q_f^t as the normalized feature histogram of the background in frame t . When the object is not accurately located, sampling in this manner will introduce model drift problem and will cause tracking failure. To avoid this problem, we simply average object feature histogram in the current frame and the one in the first frame like [5].

To evaluate the discriminating power of each feature, Bayes error rate is employed. Intuitively, smaller Bayes error rate demonstrates better discriminating power. By using histogram to approximate the likelihood function and by assuming that the two classes are equally likely, Bayes error rate can be estimated by equation (1) as pointed out in [6].

$$e_f^t = \frac{1}{2}\sum_{i=1}^{m} \min(\mathrm{p}_f^t(i), \mathrm{q}_f^t(i)) \tag{1}$$

Bayes error rate for feature selection has several advantages. First, Bayes error rate can deal with multimodal feature distributions, while variance ratio fails to work when feature distributions are multimodal. To tackle this problem, Collins et al. [5] apply variance ratio to the log likelihood ratio of feature distributions of the object and background. Second, given feature distributions, Bayes error rate is easy to compute. Only several comparison and addition operations are needed. Third, Bayes error rate has a good theoretical foundation.

After ranking features based on their Bayes error rates, the top *N* features are selected. We denote the selected feature set as F_s . Since Bayes error rate of each feature is a function of the appearances of the object and background, we can believe that it changes not too much when the appearances of the object and surrounding background change slowly. Here we model $\{e_f^t \mid f \in F_s\}$ as Gaussians and use equations (2) and (3) to incrementally estimate their means and variances. t_0 denotes the frame number of the last feature selection. $t' = t - t_0 + 1$, $\mu_f^0 = 0$ and $\sigma_f^0 = 0$. If one of e_f^t satisfies $e_f^t > \mu_f^{t'-1} + \alpha\sigma_f^{t'-1}$, perform feature reselection. α is set to be 2.5 in our experiments.

$$\mu_f^{t'} = \mu_f^{t'-1} + \frac{1}{t'}(e_f^t - \mu_f^{t'-1}) \tag{2}$$

$$(\sigma_f^{t'})^2 = \frac{t'-1}{t'}((\sigma_f^{t'-1})^2 + \frac{1}{t'}(e_f^t - \mu_f^{t'-1})^2) \tag{3}$$

2.2. Weight Image Generation

Each feature $f \in F_s$ produces a weight image W_f^t with the same size as current search window. We denote $b_f^t(\mathrm{u}) \in \{1,\ldots,m\}$ as the bin index associated with feature f at pixel location $\mathrm{u} = (x,y)$ in frame t . Each pixel value $W_f^t(\mathrm{u})$ is computed as follows

$$W_f^t(\mathrm{u}) = \frac{\mathrm{p}_f^{t-1}(b_f^t(\mathrm{u}))}{\mathrm{p}_f^{t-1}(b_f^t(\mathrm{u})) + \mathrm{q}_f^{t-1}(b_f^t(\mathrm{u}))} \tag{4}$$

Equation (4) is actually a Bayesian classifier with the assumption of equal priors of the two classes and with feature histogram to approximate the likelihood function. The final weight image is computed as the weighted sum of weight images corresponding to the top *N* selected features. Denote $\tilde{\omega}_f^t = 0.5 - e_f^t$ and $\omega_f^t = \tilde{\omega}_f^t / \sum_f \tilde{\omega}_f^t$ (hence, $\sum_f \omega_f^t = 1$), the final weight image is given as follows.

$$W_t(\mathrm{u}) = \sum_f \omega_f^t W_f^t(\mathrm{u}) \tag{5}$$

To eliminate the influence of background pixels, pixel value below 0.5 is set to be 0 as shown in equation (6).

$$\hat{W}_t(\mathrm{u}) = \begin{cases} 0 & W_t(\mathrm{u}) \le 0.5 \\ W_t(\mathrm{u}) & otherwise \end{cases} \tag{6}$$

2.3. Scale Adaptation

When the object changes in size, the scale should be adapted. Collins [7] dealt with this problem in scale space when the aspect ratio of the object's bounding rectangle is fixed. When this condition is violated, the method may not work well. To tackle this problem, a simple yet effective scale adaptation method is presented. The basic idea is based on the observation that pixel values in weight image change sharply across object's boundaries. We introduce four kinds of correlation templates C_L and C_R with the size of $3 \times h$, C_T and C_B with the size of $w \times 3$ to locate left, right, top and bottom boundaries, respectively, as shown in Fig.1 (a)-(d), where h and w are the height and width of the object's bounding rectangle in the previous frame. Each template can be viewed as a summation of two templates, just taking (a) as an example as shown in Fig.1 (e). Ideally, these correlation templates will achieve locally maximal responses on the object's boundaries. Denote (x_c, y_c) as the object's location found by mean shift in the current frame,

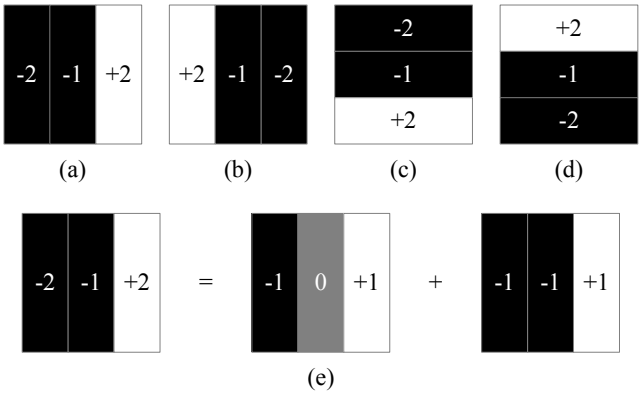

Fig.1. Correlation templates are used to locate the left (a), right (b), top (c), and bottom (d) boundaries of the object's bounding rectangle. Each template can be viewed as a summation of two templates, just taking (a) as an example as shown in (e).

then initial locations of left, right, top and bottom boundaries x_L, x_R, y_T, and y_B can be obtained. The final locations of these boundaries are achieved by equations (7)-(10), where $\lambda = 0.1$ in our experiments.

$$x_L = \arg\max_{|x-x_L|\le\lambda w, y=y_c} C_L \otimes \hat{W}_t(x,y) \quad (7)$$

$$x_R = \arg\max_{|x-x_R|\le\lambda w, y=y_c} C_R \otimes \hat{W}_t(x,y) \quad (8)$$

$$y_T = \arg\max_{x=x_c, |y-y_T|\le\lambda h} C_T \otimes \hat{W}_t(x,y) \quad (9)$$

$$y_B = \arg\max_{x=x_c, |y-y_B|\le\lambda h} C_B \otimes \hat{W}_t(x,y) \quad (10)$$

$\otimes$ denotes the cross correlation operator, and is defined in equation (11), where $I \in \{L, R, T, B\}$, w' and h' are the width and height of the correlation template, respectively.

$$C_I \otimes \hat{W}_t(x,y) = \sum_{i=-w'/2}^{w'/2} \sum_{j=-h'/2}^{h'/2} C_I(i,j)\hat{W}_t(x+i, y+j) \quad (11)$$

The proposed tracking algorithm is summarized in Fig.2.

3. EXPERIMENTS

We perform experiments on public data set [8] and our own collected data set. The tracker is initialized manually, and the top three features are selected. Actually using either three or five features has little difference in the tracking results in our experiments, which is also verified in [5]. Moreover, video frames are down-sampled by a factor of two to eliminate noises in video frames and for computational efficiency. Three challenging tracking examples are presented in this section.

In the first video, the car being tracked loops around on a runway, then drives straight, speeds up and overtakes others. The car changes in size remarkably during tracking. Without feature selection (here, we mean that features are selected in the first frame only and are kept unchanged during tracking) and scale adaptation, the tracker drifts away from the car in frame 533, for surrounding background shows

Algorithm 1 The proposed tracking algorithm

Input: Video frames $I_1, I_2, \ldots, I_T$ and initial minimal bounding rectangle (MBR) of the tracked object.
Output: MBRs of the tracked object in $I_2, \ldots, I_T$.

Initialization: Generate feature histogram p_f^1 and q_f^1 for each feature $f \in F$, set $t = 1$.

1. Perform feature selection to obtain F_s as shown in section 2.1, set $t_0 = t$ and $t' = 1$, set $\mu_f^{t'} = e_f^{t_0}$ and $\sigma_f^{t'} = 0$ for each $f \in F_s$;
2. Set $t = t+1$ and $t' = t'+1$. If $t > T$ then exit, or else generate weight image $\hat{W}_t$ for local search window in I_t as shown in section 2.2;
3. Run mean shift algorithm in $\hat{W}_t$ initialized by MBR in frame I_{t-1};
4. Perform scale adaptation to find the MBR in frame I_t as shown in section 2.3;
5. Sample object and background pixels to estimate p_f^t, q_f^t and e_f^t for each $f \in F_s$ as shown in section 2.1;
6. If none of e_f^t is above $\mu_f^{t'-1} + \alpha\sigma_f^{t'-1}$ or $t' \le 2$, then update $\mu_f^{t'}$ and $\sigma_f^{t'}$ as shown in section 2.1 and go to step 2. Otherwise, go to step 1.

Fig.2. The proposed tracking algorithm.

Fig.3. Sample frames (80, 245, 306, 460, 762, 1025, 1373, 1625, and 1820) of *egtest01* [8] sequence are shown.

Fig.4. Sample frames (40, 180, 181, 182, 485, and 727) of *egtest04* [8] sequence are shown.

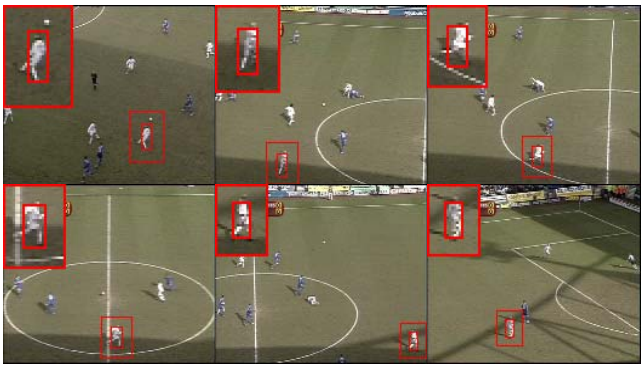

Fig.5. Sample frames (2, 141, 177, 201, 254, and 432) of *player* sequence are shown.

similar appearance to the car. With adaptive feature selection and scale adaptation, our tracker successfully tracks the car in the whole sequence. The sequence has 1820 frames, and the times of feature selection is only 47. Some sample frames are shown in Fig.3. Note though there is a strong reflection of sunlight in frame 306, our tracker accurately obtains the location and scale of the car through adaptive feature selection and scale adaptation.

The second video has many challenges as shown in Fig. 4. There is some defocusing at times (e.g. frame 485) and short occlusion by trees. Sensor recording dropped some frames, which is manifested in the sequence as duplicated frames followed by a sudden discontinuity (e.g. frame 181). Our tracker successfully tracks the car until it is occluded by trees in frame 740. The times of feature selection is 46. Without scale adaptation, the tracker drifts away from the car after frame 181, because the mean-shift window and the basin of attraction of the car do not overlap at all in frame 181. However, due to scale adaptation our tracker successfully re-locks on the car in frame 182.

In the third video, a player is successfully tracked from shadow through sunshine as shown in Fig.5, until he totally disappears in frame 473. The times of feature selection is 27. Note the non-rigid motion of the player and the changes of the illumination add many challenges to the tracker. Thanks to adaptive feature selection and scale adaptation, our tracker successfully tracks the player. Without feature selection the tracker drifts away from the player when he moves from shadow to sunshine.

4. CONCLUSIONS

In this paper, adaptive feature selection is embedded into mean shift tracking framework. From a feature set, the most discriminative features are selected based on evaluating their Bayes error rates, which are estimated from object and background samples. Hence, a weight image more suitable for tracking can be generated, in which mean shift is employed to locate the object. We model the Bayes error rate of each selected feature as a Gaussian, and perform feature reselection when any one of them does not match Gaussian well. This adaptive feature selection scheme can greatly decrease the times of feature reselection. Furthermore, based on the observation that pixel values in weight image change sharply across object's boundaries, we introduce four kinds of correlation templates to locate boundaries of the object's bounding rectangle. Experimental results show that it can work well when surrounding background is not very cluttered.

5. ACKNOWLEDGEMENTS

This work is partly supported by "Science 100 Plan" of Chinese Academy of Sciences (m2041), Beijing Natural Science Foundation (4063041), and the research start-up fund of GUCAS.

6. REFERENCES

[1] K. Fukunaga, L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. on Information Theory,* vol.21, no.1, pp.32-40, Jan 1975

[2] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol.17, no.8, pp.790-799, Aug 1995

[3] G.R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," *Fourth IEEE Workshop on Applications of Computer Vision,* pp.214-219, 19-21 Oct 1998

[4] D. Comaniciu, V. Ramesh, P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* vol.2, pp.142-149, *Hilton Head Island, SC*, June 13-15, 2000

[5] R.T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.27, no.10, pp.1631-1643, Oct. 2005

[6] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, 2nd edition, New York: Wiley-Interscience, 2001.

[7] R.T. Collins, "Mean-shift blob tracking through scale space," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol.2, pp.234-240, 18-20 June 2003

[8] R.T. Collins, X. Zhou, and S.K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site," IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, January, 2005