Advances in Engineering Research (AER), volume 61
International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2017)

Fast Compressive Tracking based on Adaptively Learning Scheme

Ling Gan [1, 2, a] and Jian Ding [1, 2, b]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2]Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]ganling@cqupt.edu.cn, [b]1552182091@qq.com

Keywords: Compressed Sensing, Target Tracking, Real-Time, Adaptively Learning.

Abstract. In this paper, we proposed a fast compressive tracking algorithm based on adaptively learning scheme (FCTAL). First, we designed a special nonlinear model for updating the learning parameter of naïve Bayes classifier. Second, we improved the target position decision strategy from FCT for getting it refrain from the single maximum classifier response value. Experimental results demonstrated that FCTAL can not only achieve a greater tracking accuracy than FCT and other three compared tracking algorithms on video frame sequences from Background Clutters & Low Resolution (BC&LR) and Fast Motion & Motion Blur (FM&MB) but also meet the requirements of real-time applications.

1. Introduction

Traditional tracking algorithms need to deal with lots of image samples to get much more information in order to improve the tracking accuracy, but that will increase the computational burden and slow down the tracking speed. Along with the theory of compressed sensing was proposed by Candes and Donoho, the computation of signal sampling and processing will be more easy and effective [1]. Based on this signal sampling theory, Kaihua Zhang proposed a simple but yet effective tracking algorithm FCT [2]. But when the tracking object moving quickly or the background can hardly be distinguished with the tracking object, FCT will perform weakly and tracking errors will lead to tracking drift.

In order to deal with the difficulties mentioned above, we proposed a new fast compressive tracking algorithm: FCTAL. First, a special sigmoid function model was designed for updating the learning parameter of naïve Bayes classifier in every frame tracking procedure. The adaptively learning scheme was based on the comparison of image features of current video frame and previous frames. Moreover, we adopted a new position determination strategy of the tracking target according to classification results from classifier. Experimental results on video frame sequences from Background Clutters & Low Resolution (BC&LR) and Fast Motion & Motion Blur (FM&MB) have demonstrated that FCTAL can achieve a greater tracking accuracy than FCT and other three compared tracking algorithms, CT [3], Frag [4], VTD [5] under the premise of real-time tracking speed.

2. Fast Compressive Tracking

2.1 Theory of Compressed Sensing.

For a high-dimension signal $X \in R^m$, we can translate it to low-dimension signal $v \in R^n$ via random matric $R \in R^{n \times m}$:

$$v = RX \tag{1}$$

where $n << m$.We can treat the low-dimension signal v as a projection of the high-dimension signal X on R. The low-dimension signal v contains enough information which can be used for reconstructed the high-dimension signal X perfectly if the random matric R satisfy the Restricted Isometry Property (RIP) [6].

2.2 Image Features Extraction and Compression.

Each sample $Z \in R^{w \times h}$ in every frame image will be extracted for multi-scale image feature by convolving it with a set of rectangle filters $\{F_{1,1}, \ldots, F_{w,h}\}$ defined as

$$F_{w,h}(x,y) = \frac{1}{wh} \times \begin{cases} 1, & 1 \le x \le w, 1 \le y \le h \\ 0, & otherwise \end{cases} \tag{2}$$

Where w and h respectively stand for the width and height of rectangle filter. These multi-scale image features will be represented as column vectors in order to be concatenated with each other as a high-dimension column vector. So, the image feature of every sample can be represented as just one column vector $X \in R^m$, $m = (w \times h)^2$.

Using the theory of compressed sensing, this high-dimension image feature vector X can be compressed as a low-dimension vector $v = (v_1, ..., v_n)^T$, $v \in R^n$ via sparse random measurement matrix $R \in R^{n \times m}$ and it defined as:

$$r_{ij} = \sqrt{s} \times \begin{cases} 1, & p = 1/2s \\ 0, & p = 1 - 1/s \\ -1, & p = 1/2s \end{cases} \tag{3}$$

A measurement matrix that satisfies RIP is random Gaussian matrix. Li et al. have proved that when $s = o(m)$, the measurement matrix can be a large extent approximate to random Gaussian matrix [7]. After compression of this measurement matrix, the low-dimension image feature vector can be thought of as computing the relative intensity difference just like Haar-Like feature [8]. And then the integral image method [9] can be used for accelerating the computational process of image features.

2.3 Fast Search Strategy.

A coarse-to-fine sliding window search strategy can be adopted to avoid calculating too much unnecessary redundant samples. First in the coarse search step, the position of the target object in last frame would be the coarse search center of samples in current frame. A large search radius r_c and search step Δ_c are implemented in this step. Second in the fine search step, the result of the coarse search step will be the fine search center of samples. The search radius and step will be both replaced by a smaller one, r_f and Δ_f. The fine search result is the tracking position of target object in current frame.

2.4 Classifier Construction and Update.

For the low-dimension image feature vector $v = (v_1, ..., v_n)^T$ of each sample z, all elements are assumed as independently distributed. Constructing a naïve Bayes classifier for classifying candidate samples and it defined as:

$$H(z) = \sum_{i=1}^{n} \log \frac{p(v_i | y=1) p(y=1)}{p(v_i | y=0) p(y=0)} \tag{4}$$

Where y=1 and y=0 respectively stand for positive and negative sample, $p(y=1) = p(y=0)$. Diaconis and Freedman have proved that the random projection of high-dimension vector satisfies Gaussian distribution [10]. According to this demonstration, those two conditional probabilities $p(v_i | y=1)$ and $p(v_i | y=0)$ are Gaussian distributed. So they can be represented by four Gaussian distribution parameters $\{\mu_i^+, \sigma_i^+, \mu_i^-, \sigma_i^-\}$.

$$p(v_i | y=1) \sim N(\mu_i^+, \sigma_i^+) \qquad p(v_i | y=0) \sim N(\mu_i^-, \sigma_i^-) \tag{5}$$

Using maximum likelihood estimation for online updating these four parameters and defined as:

$$\begin{aligned} &\mu_i^+ = \lambda \mu_i^+ + (1-\lambda)\mu_{i+} \qquad \mu_i^- = \lambda \mu_i^- + (1-\lambda)\mu_{i-} \\ &\sigma_i^+ = \sqrt{\lambda(\sigma_i^+)^2 + (1-\lambda)(\sigma_{i+})^2 + \lambda(1-\lambda)(\mu_i^+ - \mu_{i+})^2} \\ &\sigma_i^- = \sqrt{\lambda(\sigma_i^-)^2 + (1-\lambda)(\sigma_{i-})^2 + \lambda(1-\lambda)(\mu_i^- - \mu_{i-})^2} \end{aligned} \tag{6}$$

Where λ is the learning speed of classifier. The calculating formulas of $\mu_{i+}, \mu_{i-}, \sigma_{i+}, \sigma_{i-}$ are defined as:

$$\mu_{i+} = \frac{1}{N}\sum_{k=0|y=1}^{N-1} v_i(k) \qquad \mu_{i-} = \frac{1}{M}\sum_{k=0|y=0}^{M-1} v_i(k)$$
$$\sigma_{i+} = \sqrt{\frac{1}{N}\sum_{k=0|y=1}^{N-1} (v_i(k)-\mu_{i+})^2} \qquad \sigma_{i-} = \sqrt{\frac{1}{M}\sum_{k=0|y=0}^{M-1} (v_i(k)-\mu_{i-})^2} \tag{7}$$

In the formula 7 above, N and M respectively stand for number of positive and negative samples.

3. Adaptively Learning Scheme

Whether FCT can track the target object successfully or not depends on the performance of naïve Bayes classifier in every frame. But the performance of classifier is closely related with the capability of classifier that whether it can learn enough image features information of positive and negative samples for distinguishing the target object from the tracking background.

FCT holds a relatively high learning speed λ=0.85 throughout the whole tracking process. Consequently, the naïve Bayes classifier of FCT cannot learn much more variation of feature information of samples in the smooth motion but neither can keep more original feature information of previous frames in fast motion. To crown it all, the classifier is less resistant to noise pollution, so the precise tracking result cannot be obtained further. Even worse, once the tracking drift will lead to the classifier error accumulation, until the tracking failure. In order to improve FCT, we need to establish an adaptively learning scheme for the learning speed of classifier. We use a special sigmoid function model for updating the learning speed of classifier in every frame and this model is defined as figure 1 and formula 8:

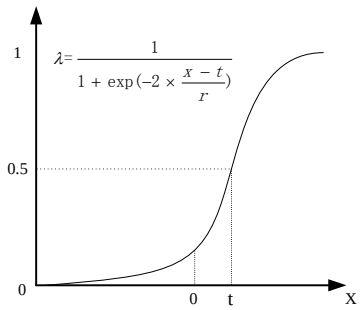


Fig. 1 Adaptively learning model

$$\lambda = \frac{1}{1+\exp(-2\cdot\frac{x-t}{r})} \tag{8}$$

In formula 8, x represents the difference between the feature parameters of samples in current frame and the current classifier parameters, which means the sampled image feature information that classifier should be learned. This variable t is set to dynamic changes and its value is the mean of four parameters for each sample of the current frame image. The variable r represents the learning speed variation parameter. Calculations of those three variables are defined as formula 9:

$$t=\begin{cases} \sum_{i=1}^{n}\mu_i^+ / n, & update \quad \mu_i^+ \\ \sum_{i=1}^{n}\mu_i^- / n, & update \quad \mu_i^- \\ \sum_{i=1}^{n}(\sigma_i^+)^2 / n, & update \quad \sigma_i^+ \\ \sum_{i=1}^{n}(\sigma_i^-)^2 / n, & update \quad \sigma_i^- \end{cases} \qquad \begin{cases} r=t/2, & x \ge t \\ r=2t, & x < t \end{cases} \qquad \begin{cases} x=|\mu_{i+}-t|, & update \quad \mu_i^+ \\ x=|\mu_{i-}-t|, & update \quad \mu_i^- \\ x=|(\sigma_i^+)^2-t|, & update \quad \sigma_i^+ \\ x=|(\sigma_i^-)^2-t|, & update \quad \sigma_i^- \end{cases} \tag{9}$$

Observing figure 1 we can know that the change of learning speed is divided into two states: When $x \geq t$ which represents the feature information of sampled images in current frame meets a tremendous change compare with the feature information that classifier has learned. We can assume that the target object moves quickly at this moment. In order to prevent the classifier to be polluted by the noise or the classification accumulating error, the learning speed of classifier should change grow drastically. For the sake of fast motion that target position may exceed the maximum sampling range, the coarse search radius should be expanded and we set it extended as 1.5 times. Conversely when $x < t$, variations of the mean and variance of each sampled images feature in adjacent frames are less than the overall difference. It declares that the tracking target tends to move smoothly and the learning speed of classifier should stay less than 0.5.

4. Target Position Decision Strategy

In every frame of video frame sequence, we need to choose one of tested samples to fit the true target position. Usually, which sample will be decided as target position depends on the Bayes classifier response value of test sample. The higher the response value represents an even more definite difference between sample and background and closer to true target position. In consideration of FCT adopts a coarse-to-fine search strategy which uses a large search radius and step in the coarse search. FCT constantly choose the sample which has the biggest classifier response value may cause problems in some special situations, such as background clutter or similarity between target and background caused by low image resolution. In these situations, there will be more than one sample stay closely to the true target position and the sample has the maximum classifier response value maybe not the closest one. Using the target position decision strategy of FCT will probably happen some tracking drift and even worse lead to tracking failure.

This paper proposed a new target position decision strategy as expressed in formula 11. After classifier does response to test samples, we choose the biggest three samples for further judgment. Those three samples are marked as S_1, S_2, S_3, so corresponding classifier response value and position are defined as $H_{S_1}(v) > H_{S_2}(v) > H_{S_3}(v)$ and $L_{S_1}, L_{S_2}, L_{S_3}$.

$$L_s = \begin{cases} L_{s_1}, & H_{s_1}(v) - H_{s_2}(v) > H_{s_2}(v) - H_{s_3}(v) \\ \sum_{i=1}^{2} L_{s_i}, & H_{s_1}(v) - H_{s_2}(v) < (H_{s_2}(v) - H_{s_3}(v))/2 \\ \sum_{i=1}^{3} L_{s_i}, & else \end{cases} \tag{10}$$

When $H_{S_1}(v) - H_{S_2}(v) > H_{S_2}(v) - H_{S_3}(v)$, it represents that the sample S_1 locates at an obvious shorter distance from the true target position than other two samples S_2 and S_3 . So we choose the sample S_1 as the target position; When $H_{S_1}(v) - H_{S_2}(v) < (H_{S_2}(v) - H_{S_3}(v))/2$, it represents that there are two samples S_1, S_2 both get nearly same closer to the true target position than S_3 . We set the target position in this step as the mean of these two samples position. This setting can avoid choosing a fake target sample due to the similarity clutter; In other circumstances, we do believe that there three samples all get closely to the true target position. According to the strategy mentioned above, we choose the mean position of these three samples position as the target position.

5. Experimental Results and Analysis

5.1 Experimental Settings.

Six test video frame sequences and corresponding target ground truth both come from the Visual Tracker Benchmark (VTB) standard test sequences [11]. The initial tracking target position of each tested sequence is set by the target ground truth from VTB, format as $[x, y, w, h]$, where (x, y) represents the position of the upper left corner of the target window, w and h represent the width and height of the target window. In the classifier updating stage, we set the search radius of positive

samples as 4 pixels, 8 pixels and 30 pixels for the inner and outer radius of the annular negative samples searching region. The classifier learning speed of the first frame is set as 0.85 as FCT does. In the target searching stage, we set the coarse searching radius and step as 25 pixels and 4 pixels, 10 pixels and 1 pixel for the fine searching radius and step. Other four compared tracking algorithms run under the setting by original author and do not make any changes.

5.2 Results and Analysis.

For objectively evaluating the merits of these five tracking algorithms, we adopt Center Location Error (CLE) for determining quantitative Euclidean distance between centers of the tracked target area and the real target area. The real target area derives from the manually labeled target ground truth data in VTB. We list the average CLE results of these video frame sequences in Table 1 and Table 2. The red font and the underlined numbers respectively represent the smallest and the second smallest average CLE result of the video frame sequence in each column among these five tracking algorithms, namely the best overall tracking performance.

Observing Table 1 and Table 2 we can find out that the proposed algorithm FCTAL is significantly better than the original algorithm FCT on the overall tracking performance. More importantly, FCTAL achieves the best or second best tracking result in these six video frame sequences.

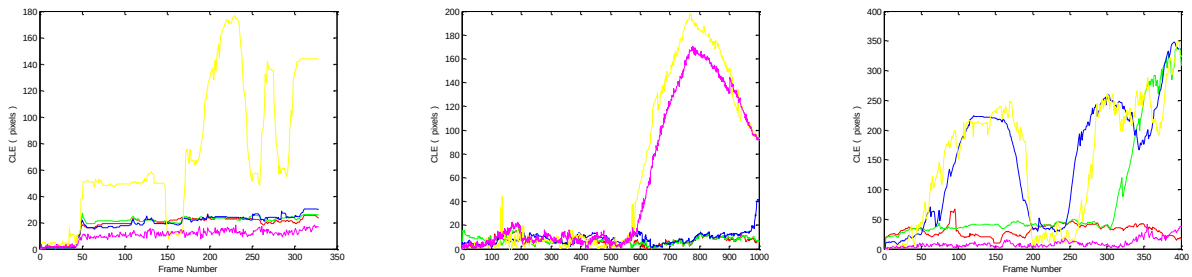
Table 1. BC&LR Average CLE (pixels)

Sequences	FCT-AL	FCT	CT	Frag	VTD
Coupon	18.58	19.51	18.98	71.57	10.65
Panda	6.60	7.47	9.00	61.66	54.60
Skating1	31.35	81.65	150.46	149.35	9.35

Table 2. FM&MB Average CLE (pixels)

Sequences	FCT-AL	FCT	CT	Frag	VTD
BlurBody	40.00	47.17	176.55	37.80	146.90
Boy	5.72	7.02	9.03	40.52	7.57
Deer	9.84	9.21	246.42	105.09	134.85

We can easily recognize the tracking performance difference between the five algorithms from the whole point of view in these two tables above. But they cannot exhibit the specific tracking difference between 5 algorithms in every frame. To make up for the lack of this, we calculate CLE of 5 algorithms in every video frame and draw them as Figure 2-5, where Figure 2 and 3 are the results on BC&LR video frame sequences, Figure 4 and 5 are the results on FM&MB video frame sequences.



(a) Coupon (b) Panda (c) Skating1

FCTAL FCT CT Frag VTD

Fig. 2 CLE curves of 5 tracking algorithms (BC&LR)

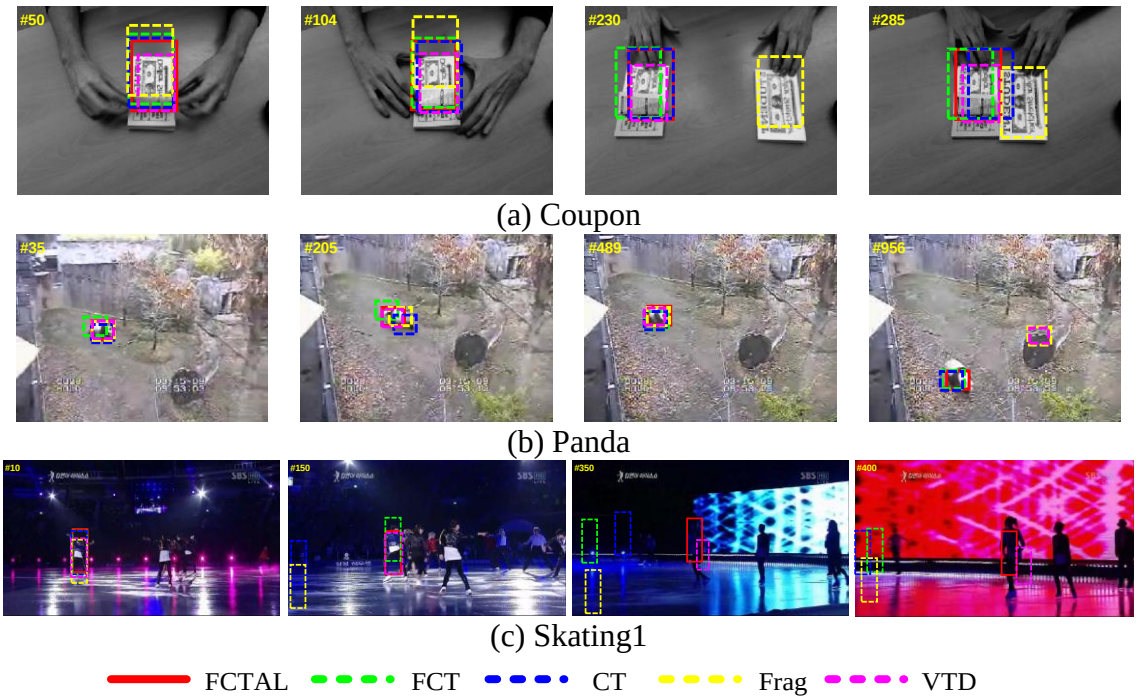


Fig. 3 Tracking results of 5 algorithms (BC&LR)

For these three BC&LR video frame sequences, the difficulty of tracking lies in that the feature information of the target object closely approximate the tracking background and not easy to distinguish the target object from the background because of the low image resolution. It is for these reasons that the target object will easily drift or even be lost.

For the Coupon video frame sequence, there was another coupon with exactly the same as the target. Both two objects moved in two opposite directions and in the instant of contact, Frag lost the object coupon but turned into tracking the interfering coupon. Observing Figure 2(a), 3(a), we can find that although FCTAL happened some drift due to the deformation of the target object, the whole tracking process was relatively stable, and the tracking window was always close to the true location.

For the Panda video frame sequences, the low image resolution did contribution to the incompetent of distinguishing the target object from the background and the vulnerable anti-disturbance performance by objects in the background, such as rocks, tree and other objects with similar color or size. Comparing with FCTAL and as seen in the Figure 2(b), 3(b), FCT met a huge drift in the initial stage of Panda. Even worse for Frag and VTD, they both had happened the loss of target. The proposed algorithm FCTAL had always maintained CLE value under 20 pixels per frame and that performance was more outstanding than other four algorithms.

For the Skating1 video frame sequence, Figure 2(c), 3(c) has shown that only FCTAL and VTD can keep tracking the skater closely during the bright-dark-bright background transition.

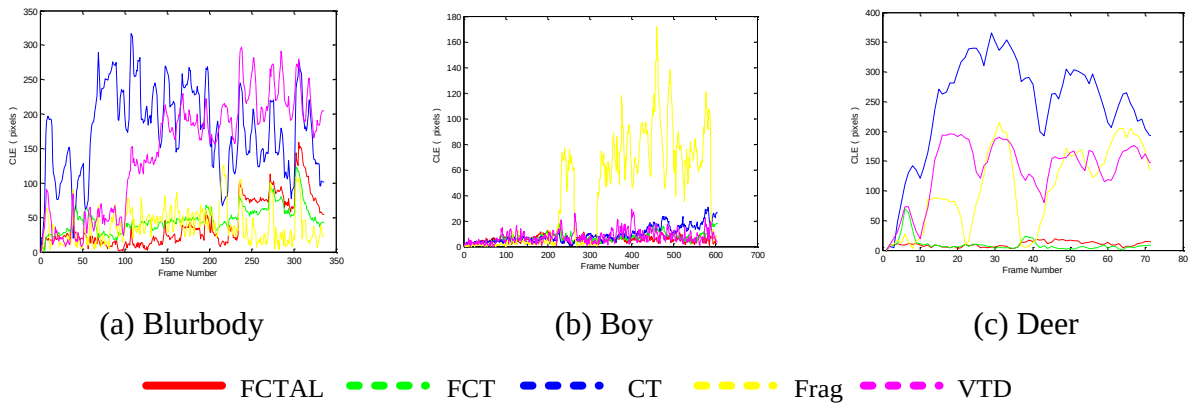


Fig. 4 CLE curves of 5 tracking algorithms (FM&MB)

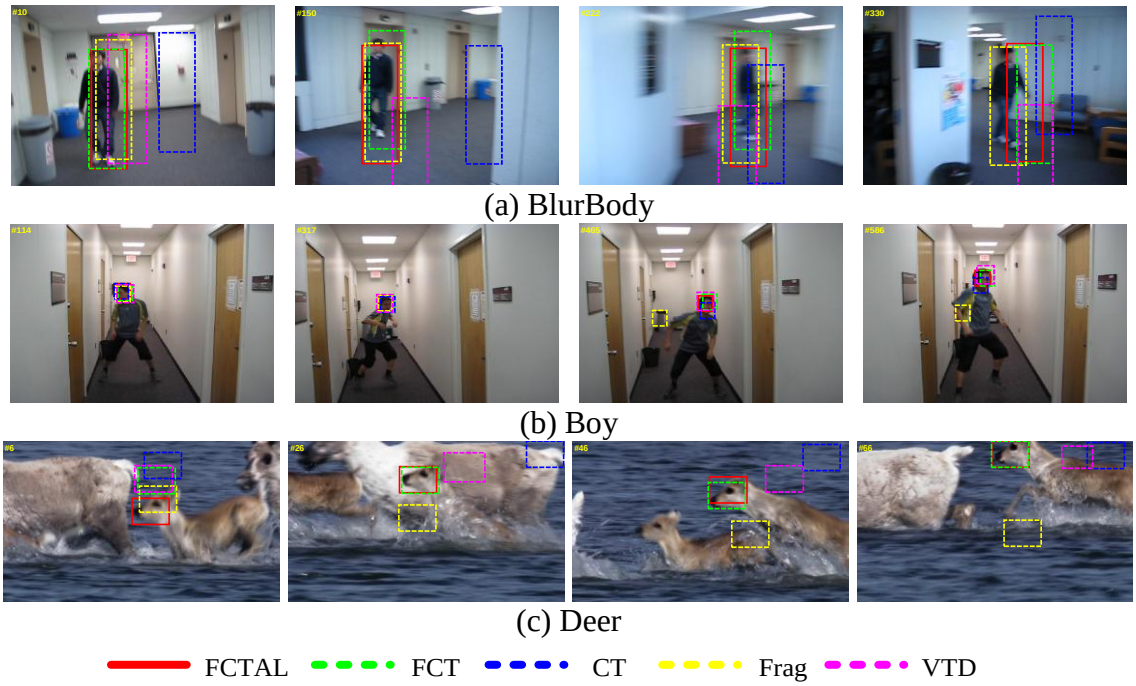


Fig. 5 Tracking results of 5 algorithms (FM&MB)

For FM&MB video frame sequences, the difficulty of tracking lies in that the classifier lacks enough effective feature information to distinguish the target object from background because of motion blur and huge variation of feature information which are caused by the fast movement of the tracking target. The proposed FCTAL can cope with these situations above well and keep the classifier constantly in an effective condition.

For the video frame sequence BlurBody, camera shook severely with respect to the target human body. For watching Figure 4(a), 5(a), CT and VTD kept a high CLE value over 100 pixels throughout most of the whole tracking process. Figure 4(a) obviously told us these two algorithms had already lost the tracking ability in that situation. Although FCTAL happened some drift caused by scale of the target human overflow tracking window at the final stage, it was better than other algorithms from the overall point of view of the whole tracking process.

For the video frame sequence Boy, the tracking target was the non-stop shaking head. For watching Figure 4(b), 5(b), the proposed FCTAL received a great tracking performance and the maximum CLE just as 18 pixels. And the variation curve of CLE kept at a much stabler and lower level than other four algorithms.

For the video frame sequences Deer, the tracking targets deer and boy moved violently comparing with something similar to these targets from background interfere the tracking process. Analyzing the CLE variation curve in Figure 4(c), 5(c) we can find out that CT, Frag, VTD lost the target most of the time. Although FCT can almost correctly track the deer through the whole process but it still happened some huge drift at the beginning stage. The proposed FCTAL did not have these defects and kept tracking not only effectively but also stably.

6. Conclusion

In order to deal with the defects of existing tracking algorithms that they cannot well adapt the fast movement of target and the interference from similar background, we proposed a new tracking algorithm FCTAL to establish an adaptively learning scheme for classifier and improve the tracking target position decision strategy. Experiments ran on several video frame sequences from both two classifications BC&LR and FM&MB and compared with four existing outstanding tracking algorithms, the result had proved that our algorithm FCTAL can do a more accurate tracking performance overall. Meanwhile, our algorithm received a maximum tracking frame speed at 36FPS on video frame sequence Panda and a minimum speed at 24FPS on Skating1. These experimental

data demonstrated that FCTAL can satisfy the real-time requirement of practical application in the experimental conditions.

References

[1]. D.L. Donoho. Compressed Sensing. Information Theory. Vol. 52 (2006) No.4, p. 1289-1306.
[2]. K. Zhang, L. Zhang, M.H. Yang. Fast Compressive Tracking. Pattern Analysis and Machine Intelligence. Vol. 36(2014) No.10, p.2002-2015.
[3]. K. Zhang, L. Zhang, M.H. Yang. Real-time Compressive Tracking. European Conference on Computer Vision. Florence, Italy, 2012, p. 864-877.
[4]. A. Adam, E. Rivlin, I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. Computer Vision and Pattern Recognition. New York, NY, USA, June 2006, p. 798-805.
[5]. J. Kwon, K. M. Lee. Visual Tracking Decomposition. Computer Vision and Pattern Recognition. San Francisco, CA, USA, June 2010, p. 1269-1276.
[6]. E.J. Candes, T. Tao. Decoding by Linear Programming. Information Theory. Vol. 34 (2010) No.4, p. 435-443.
[7]. P. Li, T.J. Hastie, K.W. Church. Very Sparse Random Projections. International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, June 2006, p. 287-296,.
[8]. B. Babenko, M. H. Yang, S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. Pattern Analysis and Machine Intelligence. Vol. 33 (2011) No. 8, p.1619-1632.
[9]. P. Viola, M. Jones. Rapid Object Detection using A Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition. Kauai, HI, USA, December 2001, p. 511-518.
[10]. P. Diaconis, D. Freedman. Asymptotics of Graphical Projection Pursuit. Annals of Statistics. Vol. 12 (2010) No. 3, p.793-815.
[11]. Y. Wu, J. Lim, M.H. Yang. Object Tracking Benchmark. Pattern Analysis and Machine Intelligence. Vol. 37 (2015) No.9, p. 1834-1848.