Proceedings of the 34th Chinese Control Conference
July 28-30, 2015, Hangzhou, China

Object Tracking Algorithm Based on HSV Color Histogram and Block-Sparse Representation

Chi Xiao[1,2], Wenjie Chen[1,2], Huilin Gao[1,2]

1. School of Automation, Beijing Institute of Technology, Beijing 100081
E-mail: bitxc129@126.com

2. Beijing Key Laboratory of Automatic Control System (Beijing Institute of Technology), Beijing 100081
E-mail: chen.wenjie@163.com

Abstract: Sparse representation has been applied in tracking rapidly. However, most of these methods use the holistic model and grayscale image which result in insensitiveness to color information and ineffectiveness to deformation of non-rigid object. In this paper, we propose a robust and effective tracking method (SRH) based on the block-sparse representation (local information) and HSV color histogram (spatial information). This method not only keeps the merits of block-sparse representation in handling illumination changes and occlusions, but also adds the object color resolution and is not subjected to similar colors interference. It improves the accuracy and efficiency of tracking result. In addition, we calculate the weight of block-sparse representation and HSV color histogram and introduce a fusion method to fuse these two data. Furthermore, the occlusion handling reduces the influence of heavy block, the update strategy guarantees the tracker to adapt to the complex background and morphological changes of target which enhances the reliability of this method. Experimental results compared with several state-of-the-art trackers on challenging sequences demonstrate the robustness and accuracy of the proposed tracking algorithm.

Key Words: block-sparse representation, HSV color histogram, feature fusion, occlusion handling

1 Introduction

Visual tracking is an extensively applicable technology in computer vision. It plays a critical role in surveillance, human computer interface and robotics, etc. A lot of tracking algorithm has been proposed in last decades, however there are still plenty of challenging problems need to be solved such as heavy occlusions, illumination changes and deformations.

There are four dominant methods of visual tracking. The first method is based on the center weighted matching, which's typical method is Mean-shift [1]. This method overcomes the rotation and distortion of the target, but it's easy to be influenced by similar color that results in an unsuccessful tracking. The second one is based on patch matching [2], which divides the target region into several patches, and then tracks these patches respectively. This tracker can handle the partial occlusion well but is not stable as it depends on division method too much. The third one is based on track prediction, such as Kalman filtering [3]. This method uses target movement information, such as position, velocity and acceleration to predict the location of the target in the next frame. It has a good tracking effect for linear motion, but not for nonlinear motion. The last one is based on Bayesian theory, such as particle filter [4], this method can apply to all non-linear and non-Gaussian systems, but as for long time tracking, the degradation of particle swarm reduces the stability and reliability of the algorithm.

In recent years, tracking methods based on sparse representation have developed rapidly as its simple representation, robustness and effectiveness to illumination changes and occlusions. The key idea of sparse representation is to find the best candidate with minimal reconstruction error by using sparse dictionary and templates.

Yang et al. [5] propose a tracking algorithm based on LBP texture features of sparse representation. Gao et al. [6] adopt the Gabor texture representation of the object as the sparse dictionary and occlusion dictionary, and then track the object by solving the l_1 minimization problem. Zhang et al. [7] develop a tracking method named TOD, using occlusion detection via structured sparse learning. Liu et al. [8] propose a tracking algorithm based on local sparse appearance model and k-selection, which employs histograms of sparse coefficients, voting map and the mean shift method for tracking. Zhong et al. [9] propose a robust object tracking algorithm using a collaborative model which consists of sparsity-based generative model (SGM) and sparsity-based discriminative classifier (SDC). Wright et al. [10] present the concept of over-complete dictionary and a new algorithm based on image recognition. Jia et al. [11] propose a tracking method based on adaptive structural local sparse appearance model by using alignment-pooling method to obtain the partial information and spatial information of the target. Mei et al. [12] propose a visual tracking method by casting tracking as a sparse approximation problem in a particle filter framework. However, most of these methods use the holistic model and grayscale image which result in insensitiveness to color information and ineffectiveness to deformation of non-rigid object.

In this paper, we propose an efficient tracking algorithm based on block-sparse representation and HSV color histogram (SRH). It retains the advantages of block-sparse

[*]This work is supported by Projects of Major International (Regional) Joint Research Program NSFC (61120106010).

representation in handling illumination changes and occlusions, moreover, it adds the object color resolution and is not subjected to similar colors interference which improves the accuracy and efficiency of tracking result. Furthermore, we can handle the occlusion effectively, meanwhile, we adopt a practical template update method. Massive experiments on various challenging sequences show that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

2 Proposed Algorithm

In this section, we present the proposed algorithm in details. Our method is for tracking moving targets in dynamic scenes. On the basis of SGM [9], we improve its poor performance on deformation of non-rigid object. In first frame, we get the target area and template manually. Then we use particle filter to get the information of N candidates in current frame. In order to track the target better, we use the affine transformation method to model object motion. Through comparing candidate and template, we calculate the sparse similarity and HSV similarity of each candidate. The results are weighted and multiplied to get the final observed value of each candidate. And our current tracking target is the candidate which has the maximum value. At last, we handle the heavy occlusion and update templates.

2.1 Block-Sparse representation model

2.1.1 Target Block

In first frame, we get the target area manually and convert it into a 32*32 block by bilinear interpolation. Then we use 8*8 sliding block to sample it every four pixels. Thus, the 32*32 block is divided into 49 patches and each patch can be represented by vector $y_i \in R^{64\times 1}$, 64 denotes the dimension of the patch. The template matrix can be represented as $Y_0 \in R^{64*49}$, similarly, we can use $Y_\text{i} \in R^{64*49}$ to represent the matrix of each candidate.

2.1.2 Dictionary Learning

Through section 2.1.1 we get the temple matrix $Y_0 = \{\text{y}_i\}$. Since the initial dictionary D is unknown, we use the theory of sparse dictionaries [13] to construct the dictionary $\text{D} \in R^{64*50}$ to assure the sparsest Y_0. The optimization problem can be expressed as formula (1).

$$\min_{D\in C,\beta\in R^{k*n}} \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\|y_i - D\beta_i\|_2^2 + \lambda\|\beta_i\|_1) \tag{1}$$

2.1.3 Sparse Observation Values

According to the theory of sparse, we can get the sparse coefficient $\beta_0 \in R^{50*49}$ of temple vector Y_0 by solving the l_1 optimization problem. The definition of l_1 optimization problem can be expressed as formula (2).

$$\min_{\beta_0}\|Y_0 - D\beta_0\|_2^2 + \lambda\|\beta_0\|_1 \quad st\text{:}\ \beta_0 \geq 0 \tag{2}$$

In formula (2), D denotes dictionary. The sparse coefficient β_0 is expanded into one-dimensional vector $\eta_0 \in R^{2450*1}$, which is called the template sparse histogram.

Similarly, in the current frame, we get the sparse coefficient β_i and sparse histogram η_i of the i-th candidate by solving the l_1 optimization problem. It can be expressed as formula (3).

$$\min_{\beta_i}\|Y_i - D\beta_i\|_2^2 + \lambda\|\beta_i\|_1 \quad st\text{:}\ \beta_i \geq 0 \tag{3}$$

Next, we use formula (4) to calculate the sparse similarity.

$$\text{sim}_i = \exp(-\frac{d_c(\eta_0,\eta_i)}{2\sigma_c^2}) \tag{4}$$

In this work, σ_c is the standard deviation, sim_i denotes the sparse similarity of the i -th candidate in the current frame. $d_c(\cdot,\cdot)$ expresses Bhattacharyya distance, as formula (5).

$$d_c(\eta_i,\eta_0) = \sqrt{1 - \frac{1}{\sqrt{\bar{\eta}_0\bar{\eta}_i\text{N}^2}}\sum_I\sqrt{\eta_0(I)\eta_i(I)}} \tag{5}$$

2.2 HSV Color Histogram Model

In order to improve tracking accuracy, it is necessary to use the HSV color histogram model.

In first frame, we convert target area from GRB image to HSV image. There are so many colors in an RGB image that the corresponding dimension of HSV image will become massive inevitably which results in ineffectiveness of the algorithm. In order to solve this problem, we quantify the HSV space appropriately. According to the color awareness of human, we quantify H, S, V spaces unequally [15].

According to the visual resolution of human, the tonal space H is divided into 8 parts, both the saturation space V and brightness space S are divided into 3 parts. Meanwhile, on the basis of color range and subjective color perception, the quantization formula can be expressed as formulas (6), (7), (8).

$$\text{H}=\begin{cases} 0 & \text{H}\in[316,360]\cup[0,20) \\ 1 & \text{H}\in[20,40) \\ 2 & \text{H}\in[40,75) \\ 3 & \text{H}\in[75,155) \\ 4 & \text{H}\in[155,190) \\ 5 & \text{H}\in[190,270) \\ 6 & \text{H}\in[270,295) \\ 7 & \text{H}\in[295,316) \end{cases} \tag{6}$$

$$\text{S}=\begin{cases} 0 & \text{S}\in[0,0.2) \\ 1 & \text{S}\in[0.2,0.7) \\ 2 & \text{S}\in[0.7,1] \end{cases} \tag{7}$$

$$\text{V}=\begin{cases} 0 & \text{V}\in[0,0.2) \\ 1 & \text{V}\in[0.2,0.7) \\ 2 & \text{V}\in[0.7,1] \end{cases} \tag{8}$$

Next, we construct one-dimensional feature vector. According to formulas (6), (7), (8), the image of target area is converted into one dimensional HSV color histogram L_0 which contains 72 bin, as formula (9).

$$\text{L}_0 = 9H_0 + 3S_0 + V_0 \tag{9}$$

Similarly, we get the HSV color histogram L_i of the i-th candidate in the current frame by converting the candidate area i into one dimensional histogram.

For simplicity, we use the intersection method to match the template. The definition of HSV similarity function of the i-th candidate in the current frame can be expressed as formula (10).

$$\mathrm{cor}_i = \sum_{\mathrm{I}} \min(L_0(I), L_i(I)) \tag{10}$$

In this work, cor_i denotes the HSV similarity of the i -th candidate.

2.3 Feature Fusion

We propose a method to fuse the features of block-sparse representation and HSV color histogram, which is fast and easy to implement. Our tracking algorithm is based on the complementary superiority of block-sparse representation and HSV color histogram each other. It keeps the merits of block-sparse representation in handling illumination changes and occlusions, moreover it adds the color resolution of object and improves the accuracy and efficiency of this algorithm. Through section 2.1, we get the sparse similarity sim_i and HSV similarity cor_i from i-th candidate in the current frame. Due to the magnitude of sim_i and cor_i are different, we adopt a fast and universal linear normalization method to assure that the values of sim_i and cor_i are within the range of [0, 1]. The linear normalization formula can be expressed as formulas (11), (12).

$$\mathrm{sim}_i{*} = \frac{\mathrm{sim}_i - \mathrm{sim}^{\min}}{\mathrm{sim}^{\max} - \mathrm{sim}^{\min}} \tag{11}$$

$$\mathrm{cor}_i{*} = \frac{\mathrm{cor}_i - \mathrm{cor}^{\min}}{\mathrm{cor}^{\max} - \mathrm{cor}^{\min}} \tag{12}$$

In this work, $\mathrm{sim}_i{*}$ and $\mathrm{cor}_i{*}$ are values after linear normalized. $\mathrm{sim}^{\min}$, $\mathrm{sim}^{\max}$ denote the minimum and maximum of the sparse similarity, $\mathrm{cor}^{\min}$, $\mathrm{cor}^{\max}$ represent the minimum and maximum of the HSV similarity respectively.

Since SRH model relies mainly on block-sparse representation not HSV color histogram, the weight value of HSV similarity is smaller than sparse similarity. Because $\mathrm{cor}_i{*}$ and $\mathrm{sim}_i{*}$ are with the range of [0, 1], we square $\mathrm{cor}_i{*}$ to decrease its value. Thus, we not only add the color resolution of object, but also avoid the similar color interference. The likelihood function of the i-th candidate can be constructed by formula (13).

$$\mathrm{l}ikehood_i = \mathrm{sim}_i{*} \times (\mathrm{cor}_i{*})^2 \tag{13}$$

In formula (13), $\mathrm{l}ikehood_i$ denotes the final observation value of i-th candidate. The tracking result is the candidate with the largest probability.

2.4 Occlusion Handling

Due to the current sparse histogram of target region is composed of each overlapping patches, the SRH model has a good effect in partial occlusion. But when there is full occlusion or long time occlusion, it is prone to fail.

Since block-sparse representation is anti-occluding and robust, we use it to deal with occlusion problem and propose a judging and handling procedure. In this work, we set up a threshold ε to distinguish occlusion. If $\mathrm{sim}_i{*} \le \varepsilon$, we consider the occlusion of target is serious. In this condition, the sampling and updated template are not proceeded, while the search distance of particles is increased over the distance in the current frame. Under this way, the tracking result will not change until $\mathrm{sim}_i{*} > \varepsilon$, which prevents the drift caused by full occlusion. On the other hand, if $\mathrm{sim}_i{*} > \varepsilon$, there is no occlusion or partial occlusion, then the particle filter works as usual.

2.5 Template Update

In order to save time and increase efficiency, the dictionary D is fixed for the same sequence, but both sparse histogram and HSV color histogram of template will be updated continuously to adapt to the changes of background and target. In general, the target template will be updated every 5 frames, the formulas can be expressed as formulas (14), (15).

$$\eta_{\mathrm{n}} = \mu\eta_0 + (1-\mu)\eta_l \quad if \quad \mathrm{sim}_i{*} \le \varepsilon \tag{14}$$

$$L_{\mathrm{n}} = \mu L_0 + (1-\mu)L_l \quad if \quad \mathrm{sim}_i{*} \le \varepsilon \tag{15}$$

In formula (14), the new histogram η_{n} is composed of the histogram η_0 at the first frame and the histogram η_l at current frame, μ denotes the weight value. These are also suitable for formula (15).

3 Experimental Results

In order to prove the superiority of our tracking method, six challenging image sequences are adopted as experimental samples. The challenges of these sequences are summarized in Table 1, including OCC (occlusion), IV (illumination), SC (scale change) and DEF (deformation). For comparison, we run five outstanding tracking methods with same initial position of the target. These algorithms are STC [14], CPF [15], Frag [16], CXT [17], SGM [9]. The proposed algorithm is implemented in MATLAB and runs at 2 frames per second on an Intel 3.40 GHz CPU with 4 GB memory.

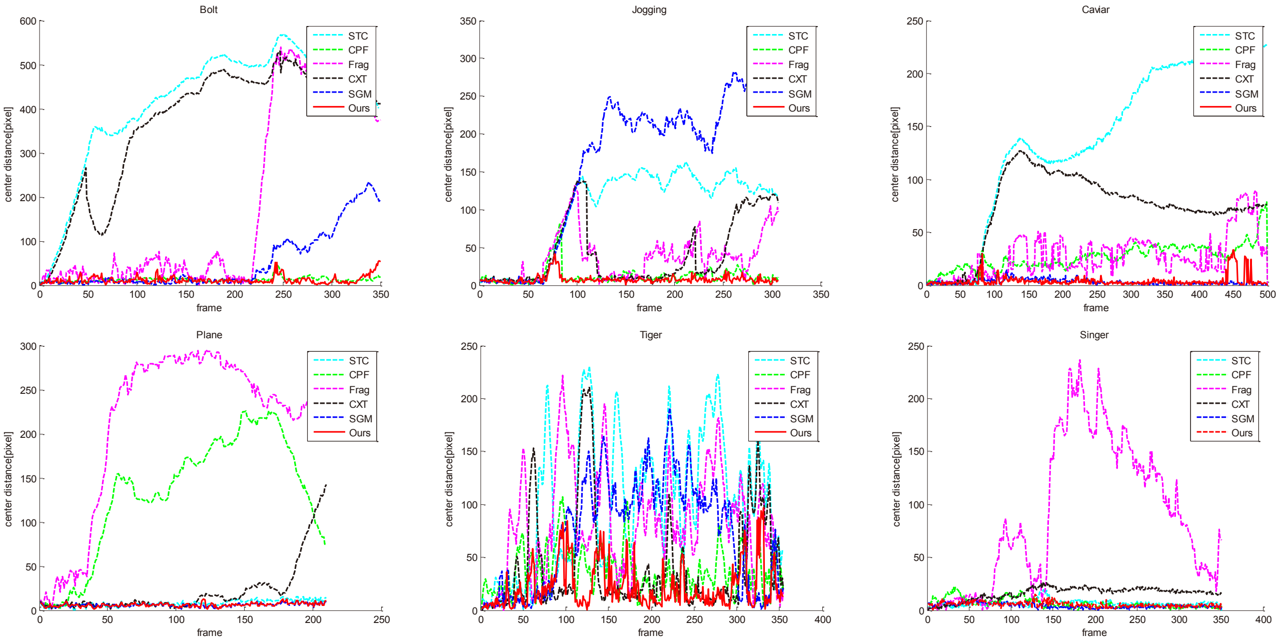


Fig.1: Quantitative comparison of the trackers in terms of position errors (in pixels)

Table 1. The challenges of the sequences

Sequence	IV	OCC	SC	DEF
Bolt	×	√	×	√++
Jogging	×	√++	×	√
Caviar	×	√++	√+	√
Plane	√	√+	×	×
Tiger	√	√+	√	√+
Singer	√++	×	√++	√

++ Heavy occlusion or variation
+ Partial occlusion or illumination changes

The parameters are presented as follows. The number of particles is with the range of [50,100], the threshold ε in Eq.14 and Eq.15 is 0.2, the update rate μ in Eq.14 and Eq.15 is 0.8.

3.1 Quantitative Comparison

The tracking result is evaluated by average center location errors and average overlap rate (see Table 2 and Table 3). In Fig.1, the average center distance of each algorithm on the 6 sequences is plotted. STC and Frag are not stable on several sequences. CPF performs well expect under heavy illumination and is easy to be disturbed by similar colors. CXT is just mediocre and not good at dealing with deformation and occlusion. SGM handles partial occlusion and illumination well, but performs poorly under deformation and heavy occlusion. Our SRH model consistently produces a higher average overlap rate and smaller average center distance than others. Overall, in most of sequences, the performance of our algorithms is better than other trackers.

Table 2. Average center location error (in pixel). The best and second best results are shown in red and green fronts.

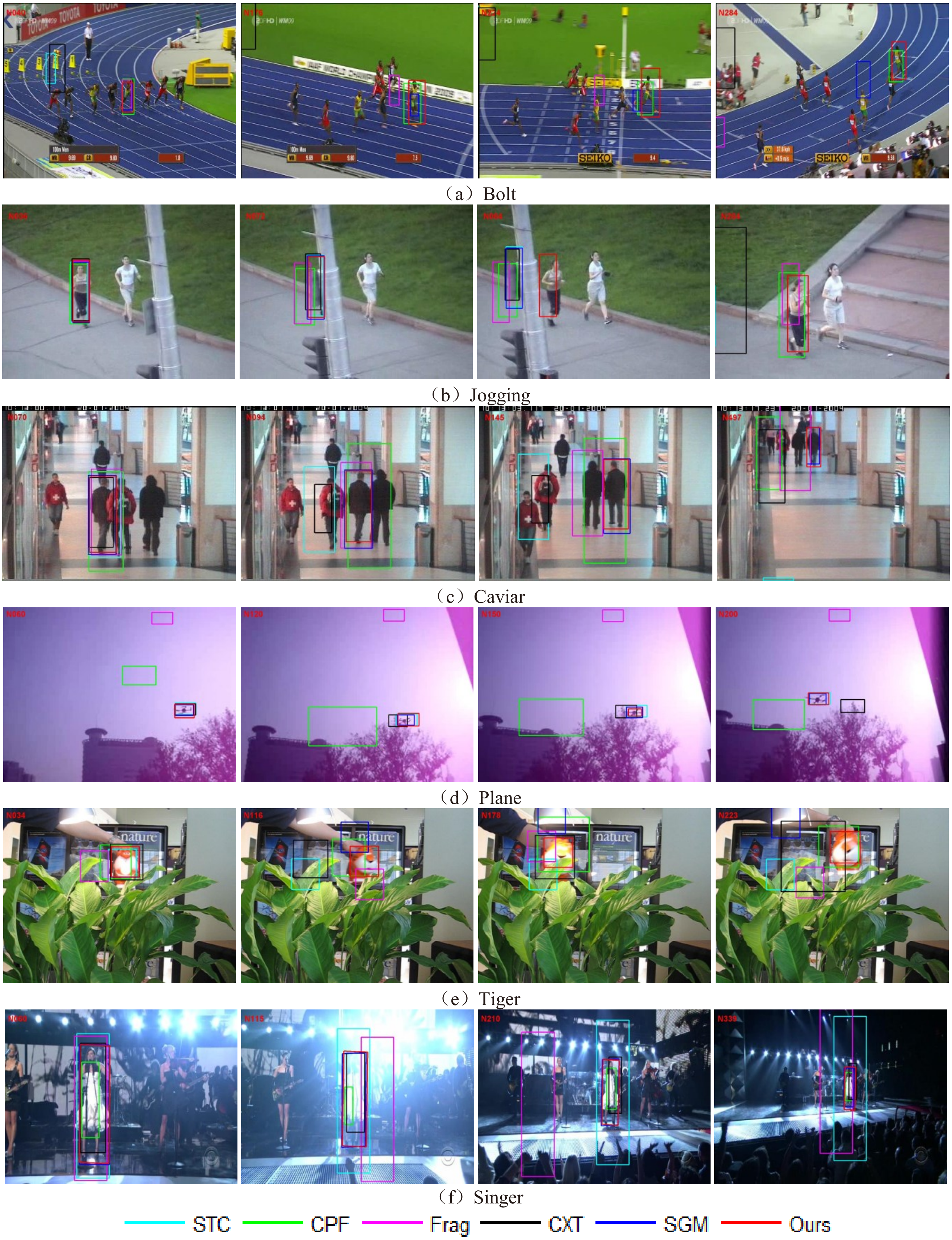
	STC	CPF	Frag	CXT	SGM	Ours
Bolt	412.8	12.8	183.4	367.8	50.5	11.9
Jogging	100.6	12.1	36.2	41.1	162.5	7.8
Caviar	138.8	26.4	27.7	72.9	3.1	3.9
Plane	8.2	131.8	211.6	21.5	5.9	5.7
Tiger	101.3	38.1	81.5	41.8	71.3	21.5
Singer	6.6	6.6	88.9	15.9	4.3	5.3

Table3. Average overlap rate based on [18]. The best and second best results are shown in red and green fronts.

	STC	CPF	Frag	CXT	SGM	Ours
Bolt	0.01	0.48	0.13	0.02	0.38	0.45
Jogging	0.11	0.58	0.28	0.38	0.18	0.72
Caviar	0.13	0.21	0.28	0.14	0.82	0.79
Plane	0.59	0.10	0.05	0.45	0.60	0.61
Tiger	0.15	0.35	0.17	0.41	0.28	0.58
Singer	0.36	0.45	0.21	0.50	0.85	0.81

3.2 Qualitative Comparison

In Fig.2, we show the tracking results of 6 algorithms on these challenging sequences. The details are presented as follows.



（a）Bolt

（b）Jogging

（c）Caviar

（d）Plane

（e）Tiger

（f）Singer

Fig.2: Tracking results of selected frames from 6 methods on 6 sequences

In the Bolt sequence, Bolt is the tracking target. While Bolt tries his best to run, his legs and arms wave dramatically and the appearance of the target is changed from the front side to the back side as time goes by. This sequence contains heavy deformation which requires of tracking algorithm a good update strategy and discriminant ability. The results are shown in Fig.2 (a). CXT and STC fail after frame 40. Frag is easy to fail around frame 176 because of its poor update strategy, and SGM fail to track since the deformation of target is too great. Only CPF and SRH can track the target during the entire sequence, for overlap rate and center location errors, they are neck and neck.

In the Jogging sequence, the woman who dresses in brown jogs in street is the tracking target. This woman is completely blocked by a telegraph pole, the tracking results are presented in Fig.2 (b). The target is invisible in frame 72 and all trackers are stay in the location where the target disappears. When the target appears again in frame 84, only SRH succeeds in tracking the woman accurately by its efficient disposal ability of heavy occlusion. Although CXT and CPF track the woman in frame 284, these two trackers are not accurate and robust enough.

Fig.2(c) shows the tracking results of the Caviar sequence. At the beginning of this sequence, the man in black on the left is the tracking target. He encounters serious block for twice in the entire sequence. Only Frag, SGM and SRH are able to track the target correctly in the first block around frame 94. Due to the interference of similar colors, Frag starts to drift in frame 145. SGM and SRH track the man quite well by handling the heavy occlusion and scale change effectively.

Fig.2 (d) shows the tracking results of the Plane sequence. In this sequence, partial illumination and occlusion occurs. Some of the trackers start to drift at the beginning because they rely on color information too much, such as CPF and Frag. While CXT fail to track because of block (the plane is blocked by branches). Although the rest trackers can track the plane through the whole sequence, SRH is the most accurate of all.

In the Tiger sequence (See Fig.2 (e)), due to a variety of challenging characteristics, such as occlusion, large pose change, fast move and illumination, most of the trackers can't track the target correctly in the entire sequence. However, the proposed SRH tracker shows the best performance.

Fig.2 (f) shows the tracking results of the Singer sequence. This sequence contains scale change (the scale of the target decreases as the camera zooms out) and heavy illumination (dazzling light). Because SGM and SRH algorithms update the target template continuously and handle heavy illumination, they can track the singer well. But CPF and CXT track the target inaccurately. Both STC and Frag fail to track the target because of scale change and illumination respectively.

4 Conclusion

In this paper, we propose a robust and effective tracking method based on the block-sparse representation and HSV color histogram. In our tracker, block-sparse representation reflects the local information of the target area which can be used to handle the partial occlusion and illumination problem. HSV color histogram reflects the spatial information of the target area which enables our tracker to handle heavy deformation better. Furthermore, the occlusion handling avoids the influence of heavy block. The update strategy reduces the possibility of drifts and enhances our tracker to adapt to appearance changes better in dynamic scenes. We also analyze the experiment results of our tracker on challenging sequences and prove that it performs favorably against several state-of-the-art trackers.

References

[1] COMANICIU.D, RAMESH.V, MEER.P. Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5): 564-577, 2003.

[2] MAGGIO E, CAVALLARO. A Multi-part target representation for color tracking, IEEE International Conference on Image Processing, 2005: 1113-1116.

[3] KALMAN R E.A new approach to linear filtering and prediction problems, Trans ASME-J of Basic Eng, 1960 (Series D).

[4] ISARD M, BLAKE A.CONDENSATION-Conditional density propagation for visual tracking, International Journal of Computer Vision, 29(1): 5-28, 1998.

[5] D-W.Yang，C.Yang，Y-D.Tang. Object Tracking Method Based on Particle Filter and Sparse Representation, Pattern Recognition and Artificial Intelligence, 26(7): 680-687,2013.

[6] L.Gao，Y.Fan，N-N.Chen. Object Tracking Algorithm Under Occlusion Based on Sparse Representation, Computer Engineering，38 (15): 5-8, 2012.

[7] T-Z Zhang, Bernard Ghanem, C-S.Xu, Narendra Ahuja. Object Tracking by Occlusion Detection via Structured Sparse Learning, in 26th IEEE Conference on Computer Vision and Pattern Recognition, 2013: 1033-1040.

[8] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust Tracking using Local Sparse Appearance Model and K-Selection, IEEE Conference on Computer Vision and Pattern Recognition, 2011: 1313-1320.

[9] W. Zhong, H. Lu, and M.-H. Yang. Robust Object Tracking via Sparsity-based Collaborative Model, IEEE Conference on Computer Vision and Pattern Recognition, 2012: 1838-1845.

[10] John Wright, Allen Y. Yang，Arvind Ganesh, S. Shankar Sastry, Yi Ma. Robust Face Recognition via Sparse Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2): 210-227, 2009.

[11] X..Jia,, M-H Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model, IEEE Conference on Computer Vision and Pattern Recognition, 2012: 1822-1829.

[12] Xue Mei, Haibin Ling. Robust Visual Tracking and Vehicle Classification via Sparse Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(11): 2259-2272, 2011.

[13] Lanchi Jiang, Guoqiang Shen, Guoxuan Zhang. An image retrieval algorithm based on HSV color segment histograms, Mechanical & Electrical Engineering Magazine, 26(11): 54-57, 2009.

[14] Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang,and David Zhang, Fast Tracking via Spatio-Temporal Context Learning, in 13th European Conference on Computer Vision, 2014: 127-141

[15] P. P´erez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking, in 7th European Conference on Computer Vision, 2002: 661-675.

[16] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments based tracking using the Integral Histogram, IEEE Conference on Computer Vision and Pattern Recognition, 2006: 798-805.

[17] T. B. Dinh, N. Vo, and G. Medioni. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments, IEEE Conference on Computer Vision and Pattern Recognition, 2011: 1177-1184.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge, International Journal of Computer Vision. 88(2):303-338, 2010.