 CMC, vol.55, no.3, pp.509-521, 2018

Real-Time Visual Tracking with Compact Shape and Color Feature

Zhenguo Gao[1], Shixiong Xia[1] , Yikun Zhang[1] , Rui Yao[1, *], Jiaqi Zhao[1] , Qiang Niu[1] and Haifeng Jiang[2]

Abstract: The colour feature is often used in the object tracking. The tracking methods extract the colour features of the object and the background, and distinguish them by a classifier. However, these existing methods simply use the colour information of the target pixels and do not consider the shape feature of the target, so that the description capability of the feature is weak. Moreover, incorporating shape information often leads to large feature dimension, which is not conducive to real-time object tracking. Recently, the emergence of visual tracking methods based on deep learning has also greatly increased the demand for computing resources of the algorithm. In this paper, we propose a real-time visual tracking method with compact shape and colour feature, which forms low dimensional compact shape and colour feature by fusing the shape and colour characteristics of the candidate object region, and reduces the dimensionality of the combined feature through the Hash function. The structural classification function is trained and updated online with dynamic data flow for adapting to the new frames. Further, the classification and prediction of the object are carried out with structured classification function. The experimental results demonstrate that the proposed tracker performs superiorly against several state-of-the-art algorithms on the challenging benchmark dataset OTB-100 and OTB-13.

Keywords: Visual tracking, compact feature, colour feature, structural learning.

1 Introduction

As one of the basic topics in the field of computer vision, visual tracking aims to find and mark the position of the tracked object in each frame of video sequences. Visual tracking has important applications and very promising prospects in military guidance, video surveillance, medical diagnosis, product testing, virtual reality and many other fields [Ross, Lim, Lin et al. (2008); Mei and Ling (2010); Kwon and Lee (2010)]. Recently, great progress has been made in the research of visual tracking, and some achievements have been put into practical application. However, under the influence of factors such as deformation, light, fast motion, occlusion and complicated background, it is still a

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China.

[2] School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia.

[*] Corresponding Author: Rui Yao. Email: ruiyao@cumt.edu.cn.

challenge to track the target efficiently in real-time.

A lot of work has been done on the task of visual tracking. Existing methods of tracking can be divided into three categories, namely, generating, discriminative and deep learning based methods. The generation method treats the tracking task as a template matching problem. The generate tracker searches for potential target locations that most closely resemble the appearance of the generated model. An object is usually represented as a set of base vectors in a series of templates or subspaces. In recent years, there are many generation-based visual tracking algorithms, frameworks and solutions proposed [Ross, Lim, Lin et al. (2008); Mei and Ling (2010); Kwon and Lee (2010); Li, Hu, Zhang et al. (2008); Liu, Huang, Yang et al. (2011); Xing, Gao, Li et al. (2013); Zhang, Zhang, Yang et al. (2012)]. A tracking algorithm using a local sparse appearance model and K-selection was proposed in Liu et al. [Liu, Huang, Yang et al. (2011)], which was robust to changes in appearance and drift. Ross et al. [Ross, Lim, Lin et al. (2008)] presented an appearance-based tracker that incrementally learn a low dimensional eigenbasis representation for robust object tracking. Xing et al. [Xing, Gao, Li et al. (2013)] developed the template update problem as online dictionary learning and proposed a robust object tracking method with online multi-lifespan dictionary learning. While the discriminative method model visual tracking task as a classification problem, this method is also commonly known as tracking-by-detection methods. What differs from the generative model is that tracking the maximum classification score between object and background is the goal of the discriminative model. Hare et al. [Hare, Golodetz, Saffari et al. (2016); Zhang, Zhang, Liu et al. (2014); Kala, Matas and Mikolajczyk (2010); Babenko, Yang and Belongie (2011); Grabner, Leistner and Bischof (2008); Avidan (2004); Avidan (2007); Yao, Shi, Shen et al. (2012, 2013); Yao (2015); Yao, Xia, Zhang et al. (2017); Collins, Liu and Leordeanu (2005)] are some attempts and achievements in recent years to solve the visual tracking task with the discriminative method. Convolutional neural networks (CNNs) perform well in many areas of computer vision by means of their powerful feature representation capabilities, such as medical image processing, biometric identification and visual tracking. Hong et al. [Hong, You, Kwak et al. (2015); Ma, Yang, Zhang et al. (2015); Qi, Zhang, Qin et al. (2016); Wang, Ouyang, Wang et al. (2015, 2016)] show the state-of-the-art results of deep-learning-based visual tracking methods. A novel visual tracking algorithm based on pre-trained CNN was proposed in Hong et al. [Hong, You, Kwak et al. (2015)]. With the CNN features and the learning recognition model, Hong et al. calculated the target-specific saliency map by back-projection, highlighting the differentiating target regions in the spatial domain. Wang et al. [Wang, Ouyang, Wang et al. (2016)] proposed a tracking algorithm using a full convolutional network that is pre-trained on image classification tasks after studying some important properties of CNN features in the perspective of visual tracking. And then, they regarded the online training process for CNNs as sequentially learning an optimal ensemble of base learners and proposed a sequential training method for CNNs to effectively transfer pre-trained deep features for online applications in Wang et al. [Wang, Ouyang, Wang et al. (2016)]. Although these methods have achieved considerable performance, they usually come at the cost of time and computational resources.

Tracking-by-detection is currently the most popular and effective framework for visual tracking tasks, and it obtains information about the target from each detection online.

Collins (DLSSVM) algorithm which approximates non-linear kernels with explicit feature maps. Avidan [Avidan (2004)] proposes a tracker based on offline Support Vector Machine (SVM). Then, Avidan [Avidan (2007)] uses an online boosting method to classify object and background. Babenko et al. [Babenko, Yang and Belongie (2011)] propose a tracker based on Multiple Instance Learning (MIL). The MIL is used to handle ambiguously labeled positive and negative data obtained online to alleviate visual drift. Hare et al. [Hare, Golodetz, Saffari et al. (2016)] propose a structural learning based tracking algorithm. Motivated by the successful of Struck and learned some tricks form [Atluri (2004)], Yao et al. [Yao, Shi, Shen et al. (2012)] propose weighted online structural learning for visual tracking to deal with the unbalanced weight problem of samples during tracking.

Recently, a group of correlation-filter (CF) based tracker [Bolme, Beveridge, Draper et al. (2010); Bertinetto, Valmadre, Golodetz et al. (2015); Danelljan, Khan, Felsberg et al. (2014); Zhang, Ma and Sclaroff (2014) (2014); Henriques, Rui, Martins et al. (2014) (2014)] has drawn much attention due to its significant computational efficiency. CF achieves real-time training and detection of densely sampled instances and high-dimensional features by using Fast Fourier Transform (FFT). Bolme et al. [Bolme, Beveridge, Draper et al. (2010)] describe their pioneering work and, adopt CF to visual tracking for the first time. Later, in order to improve the tracking performance, researchers also propose some extensions. Henriques et al. [Henriques, Rui, Martins et al. (2014)] propose a CSK method based on illumination characteristics. In addition, Ma et al. [Ma, Yang, Zhang et al. (2015)] propose a long-term tracker to learn the discriminant-dependent filters used to estimate the object's translation and scale variations. Danelljan et al. [Danelljan, Häger, Khan et al. (2014)] calculate the fast scale estimation problem by learning the discriminant CF based on the scale pyramid representation. Subsequently, in order to solve the unwanted boundary effect introduced by the periodic hypothesis of all cyclic shifts, Danelljan et al. [Danelljan, Häger, Khan et al. (2015)] introduce spatial regularization components into learning, and punish the CF coefficients according to spatial locations to achieve excellent tracking accuracy.

The colour feature is often used in the visual tracking [Rez, Hue, Vermaak et al. (2002); Possegger, Mauthner and Bischof (2015)]. The method [Danelljan, Khan, Felsberg et al. (2014)] extracts the colour features of the object and the background area, and distinguishes them by correlation filter based on the kernel function; then the object tracking of the video sequence is achieved. However, this method simply uses the colour information of the object pixel and does not consider the shape feature of the object, so that the description capability is limited. Moreover, incorporating shape information often leads to large feature dimension, which is not conducive to real-time object tracking. In addition, the construction of least squares classifier based on correlation filter can only be used for binary classification tasks and cannot accurately describe occlusion information of the object. The above problems will cause drift of visual tracking in complicated actual scenarios.

To handle the aforementioned problem, this paper presents a real-time visual tracking method with the compact shape and colour feature. In this paper, compact shape and colour features are formed by fusing shape and colour features of candidate object

regions, which reduced the dimension through the Hash function. Then, a structural classification function is presented for object classification and prediction. The proposed method is able to enhance the description ability of object appearance model, while the structural classification can improve the accuracy of the object classification, which can effectively avoid the visual tracking drift and improve the tracking performance.

2 The proposed method

In this section, we first introduce the structural classifier for visual tracking. Next, we present the shape and colour feature. Thirdly, we describe how to learn the compact representation of features. Finally, we use the proposed online learning method to build our tracking approach. The overall framework of the proposed tracker is presented in Fig. 1.

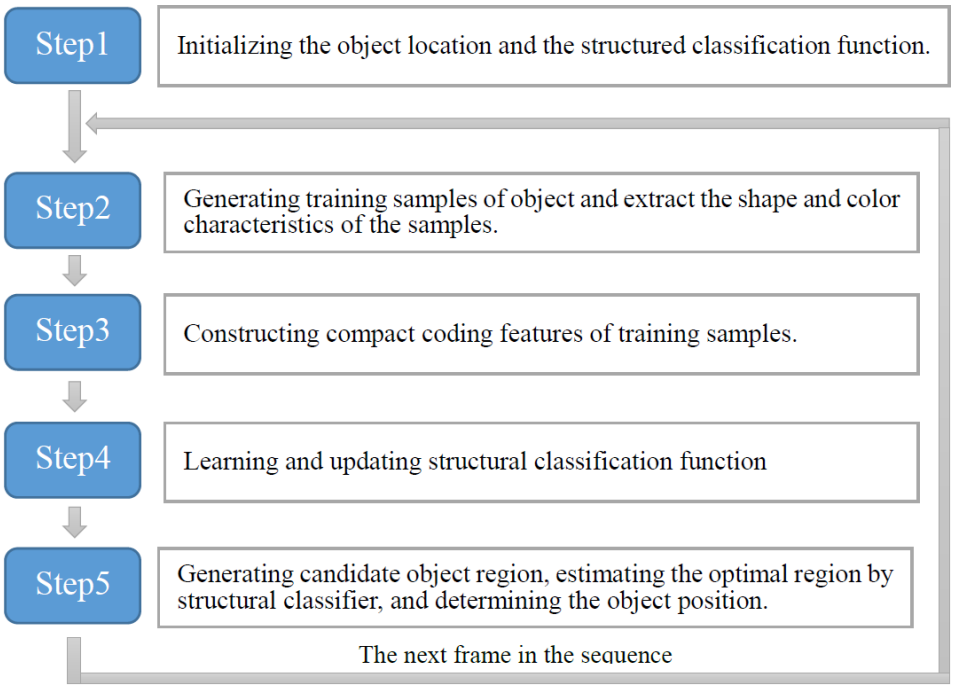

Figure 1: Overview of real-time visual tracking with compact shape and colour feature

2.1 Structural classifier for visual tracking

At the first frame of the sequence, the bounding box $B_1 = (c_1, r_1, w_1, h_1)$ of the object is given manually, where B_1 is the position of the object, c_1, r_1, w_1 and h_1 are the column coordinates, row coordinates, width and height of the upper left corner. The bounding box B_t represents the position of the object at frame t, and then we describe the displacement of the target using offset $y_t = (\Delta c_t, \Delta r_t, \Delta w_t, \Delta h_t) \in Y$. The tracking procedure starts from the second frame, the accurate bounding box is estimated for the

object location. The boundary box B_t of the object in frame *t* can be obtained by:

$$B_t = B_{t-1} + y_t^*, y_t^* = \arg\max_{y \in Y} f(x_t, y), \quad (1)$$

where $f(x_t, y) = \langle w, \psi(x_t, y) \rangle$ denotes a structured classification function, and x_t represents the frame *t* in the video sequence. $\psi(x_t, y)$ is a vector of *k* dimension, which represents the compact shape and colour feature of the candidate object region and will be constructed by Step 3 in the Fig. 1. The parameter w is a k-dimensional vector which initialized with k random real numbers between 0 and 1, and it will be updated online at Step 4 by learning samples of each frame.

2.2 Shape and color feature

An intensive sampling method is used to get samples close to the real object bounding box, and the corresponding image regions are cropped as training sample to extract the shape and colour features of these samples.

The dense sampling method is designed as follows: The real object bounding box of the current frame *t* is B_t, therefore, the true object offset is $y_t = (0, 0, \Delta w_t, \Delta h_t)$, in this paper, the fixed object size is set as $\Delta w_t = 0, \Delta h_t = 0$. Taking the current object offset in a circle which $(0, 0)$ as the centre and S as the radius ($S = 30$ in this paper), we sample M offsets in this circle, which is $Y = \{ y = (\Delta c, \Delta r, 0, 0) : \Delta c^2 + \Delta r^2 < s^2 \}$. According to the definition of Eq. 1, the object bounding box can be obtained by adding M offsets y of the samples to the B_t, and we take M image regions obtained by cropping these object bounding boxes in frame *t* of image x_t as training samples.

Next, we will extract the shape features for each training samples. In this paper, we use Haar-like features to describe the shape information of the object. The Haar-like feature is a commonly used feature description operator in the field of visual tracking. This paper uses three basic types of features, which are divided into two rectangular features, three rectangular features and diagonal features. The result of the sum of all the pixel values of a class of rectangular part in the three types of matrix image regions is subtracted from the sum of all the pixel values of the other type of rectangular part is a single eigenvalue. In this paper, the integral graph is used to speed up the calculation of this eigenvalue. Finally, we combine the eigenvalues of all three types of features into a vector, and build the Haar-like feature of the image region.

The colour information of each training sample is extracted and merged with the shape feature into a new feature vector. The colour information is extracted as follows: The colours are divided into 11 categories (black, blue, brown, grey, green, orange, pink, violet, red, white and yellow). For the three types of rectangular of Haar-like feature obtained from the previous step, we count the probability of the RGB values of all pixels in each rectangle, and then we put the 11 probability values into a colour vector. Finally, we put this colour vector after the Haar-like feature, thus we can get the new features containing the shape and colour information. The colour vector CN and all the

probabilities of mapping to the 11 colours from the RGB values of all the pixels in a rectangle $p\left(cn_i \mid I\right)$ is defined by:

$$CN = \left\{ p\left(cn_i \mid I\right) \right\}_{i=1}^{11}, p\left(cn_i \mid I\right) = \frac{1}{N} \Sigma_{c \in I} p\left(cn_i \mid g\left(c\right)\right), \tag{2}$$

where cn_i is the i-th colour of the 11 categories, c is the coordinates of the pixels in the rectangle I, N is the total number of pixels in the rectangle I, and $g\left(c\right)$ is the Lab colour space value of the pixel c, and we can get $p\left(cn_i \mid g\left(c\right)\right)$ from the common colour name mapping.

2.3 Compact representation of feature

The feature vector extracted from the second step, which contains the shape and colour information of the sample, has a high dimension. Using this feature directly will increase the computational complexity of the object tracking, which is not conducive to real-time tracking. In this paper, the local sensitive hash is used to map the high-dimensional features obtained in step two to generate compact colour-coded feature vectors $\psi\left(x_t, y\right)$. The is described as follows: Suppose the dimension of the eigenvector obtained in Step 2 of Fig. 1 is d, that is, the characteristic of each sample is a *d*-dimension vector. In order to map high-dimensional *d*-dimension vectors into *m*-dimension ($m \ll d$ in this paper $m = 100$) compact binary coding features. We define a hash function family H composed of *m* Hash functions $ha\left(\cdot\right)$. More specifically, a random vector $\upsilon \in \mathbb{R}^d$ is generated as a hyperplane from the *d*-dimension Gaussian distribution $N\left(0,1\right)$, then the hash function $ha_\upsilon\left(\cdot\right)$ is defined as:

$$ha_v\left(q\right) = \begin{cases} 1 & v \cdot q > 0 \\ 0 & v \cdot q < 0 \end{cases}, \tag{3}$$

where $q \in \mathbb{R}^d$ is the eigenvector of the single sample obtained in Step 2 of Fig. 1. Constructing m hash functions by the above method and substituting *q* into these Hash functions, a binary coded string of m dimensions can be obtained. That is, a compact coded feature vector is constructed. Note that the above m Hash functions are generated only in the first frame of the video sequence, and will continue to be used in the following frames.

2.4 Online learning

In this step, we will learn and update structural classification functions. Object tracking is an online update process for dynamic data flow. The object tracking method needs to learn and update the parameters from the training samples to adapt to the new frame. This step updates the parameters *w* of the structured classification function $f\left(x_t, y\right)$ in Eq. 1 using the compact colour-coded features of the samples generated in Step 3, and then we use the updated *w* to estimate the optimal object position in the new video frame.

The method of updating the parameter *w* is described in detail below. Substituting *M*

samples represented by a compact coding feature into Eq. (4), new parameter *w* is obtained by optimizing Eq. (4):

$$\begin{aligned} &\min_{w} \frac{\lambda}{2} \| w \|^2 + \Sigma_{t=1}^{T} \xi_t \\ &s.t. \forall t : \xi_t \geq 0; \forall t, y \neq y_t : \langle w, \psi(x_t, y_t) \rangle - \langle w, \psi(x_t, y) \rangle \geq \Delta(y_t, y) - \xi_t, \end{aligned} \tag{4}$$

where λ is the regularization coefficient, in this case, the ξ_t is the relaxation of variables, marking cost Δ is used to measure the coverage of the bounding box, defined as:

$$\Delta(y_t, y) = 1 - \frac{(B_{t-1} + y_t) \cap (B_{t-1} + y)}{(B_{t-1} + y_t) \cup (B_{t-1} + y)} \tag{5}$$

The sub-gradient descent method is used to iteratively optimize the Eq. (4) to determine the final value of the new parameter *w*. Assuming that the current frame is the frame *t*, the sub-gradient of the Eq. (4) with respect to the parameter w_t is:

$$\nabla_t = \lambda w_t - \mathrm{II}\left[\Delta(y_t, y) + \left\langle w_t, \psi(x_t, y_t) \right\rangle - \left\langle \psi(x_t, y) \right\rangle > 0 \right] \delta\psi_t, \tag{6}$$

where $\delta\psi_t = \psi(x_t, y) - \psi(x_t, y_t)$, $\mathrm{II}(\cdot)$ is an indicator function that returns 1 if the condition is met and returns 0 otherwise. In this way, the structural classification function parameters of the *t+1* frame $w_{t+1} \leftarrow w_t - \eta_t \nabla_t$, $\eta_t = 1/(\lambda_t)$ is updated step distance, the above equation can be written as:

$$w_{t+1} \leftarrow (1 - \eta_t \lambda) w_t + \frac{1}{M} \mathrm{II}\left[\Delta(y_t, y) + \left\langle w_t, \psi(x_t, y_t) \right\rangle - \left\langle w_t, \psi(x_t, y) \right\rangle > 0 \right] \delta\psi_t. \tag{7}$$

The eigenvector of *M* samples calculated in Step 3 are respectively substituted into the formula (7), and the recalculated one w_{t+1} is the updated structural classification parameter. The stochastic sub-gradient method [Shwartz, Singer and Pegasos (2011)] is guaranteed to converge to the optimal SVM solution.

2.5 Real-time visual tracking

In this section, we will generate the candidate object region, and use structural classification function to estimate the optimal object region, and finally determine the object location. When the frame *t+1* image arrives, the tracking method requires sampling close to the object location of the last frame, and estimates the highest classification scores in the samples by using the structured classification function which has been updated parameters w_{t+1}, and then the region that corresponding to this sample is the optimal target location. After getting the new object position, we turn to Step 2 until the video sequence ends. The detailed processing is described as follows:

First of all, we assume that B_t is the object boundary box of the last frame in the sequence, offset $y_{t+1} = \left\{ y = (\Delta c, \Delta r, 0, 0) : \Delta c^2 + \Delta r^2 < S^2 \right\}$ is sampled in a circle with

$(0,0)$ as the centre and S as the radius ($S=60$ in this paper). P bounding boxes of candidate object $(B_t + y)$ are obtained by adding B_t to the sampled P offsets y. Then we take the corresponding P image region cropped from the current frame *t+1* image x_{t+1} as the candidate object region.

Second, the compact shape and the colour feature vector of P candidate target regions $\psi(x_t, y)$ is calculated by using the feature generation method described in Steps 2 and 3. Eq. 8 is used to calculate the optimal offset:

$$y_{t+1}^{*} = \arg\max_{y \in Y_{t+1}} f(x_{t+1}, y) = \arg\max_{y \in Y_{t+1}} f(w_{t+1}, \psi(x_{t+1}, y)). \quad (8)$$

The position of the current frame target $B_{t+1} = B_t + y_{t+1}^{*}$ is obtained according to the target boundary box B_t and the calculated optimal migration of Eq. (8).

3 Experiments

We conducted our experiments with the most extensive and authoritative datasets OTB-13 [Wu, Lim and Yang (2013)] and OTB-100 [Wu, Lim and Yang (2015)]. All these sequences are annotated with 11 attributes which cover various challenging factors: Such as scale variation (SV), occlusion (OCC), illumination variation (IV), motion blur (MB), deformation (DEF), fast motion (FM), out-of-plane rotation (OPR), background clutters (BC), out-of-view (OV), in-plane rotation (IPR) and low resolution (LR). We follow the evaluation protocol provided by the benchmark [Wu, Lim and Yang (2015)].

Four metrics with one-pass evaluation (OPE) are used to evaluate all the compared trackers: 1) bounding box overlap, which is measured by VOC overlap ratio (VOR); 2) centre location error (CLE), which is computed as the average Euclidean distance between the ground truth and the estimated centre location of the target; 3) distance precision (DP), which indicates the relative number of frames in the sequences where the centre location error is within a given threshold; and 4) overlap precision (OP), which is defined as the percentage of frames where VOR is larger than a certain threshold.

In order to evaluate the proposed method, we use the two evaluation indicators mentioned in Wu et al. [Wu, Lim and Yang (2013)]. We use the precision plot to measure the overall tracking performance, which shows the percentage of frames whose position is within a given true threshold distance. As a representative precision score for each tracker, we use a score of threshold=20 pixels. The success plot is defined as the area under the curve (AUC) for each success graph, which is the average of the success rates corresponding to the overlap threshold of the samples. Our visual tracking method is implemented in MATLAB (R2017a) on a PC with a 3.60 GHz CPU and 12 GB of RAM. The average running speed of our tracker is 0.15 second per frame.

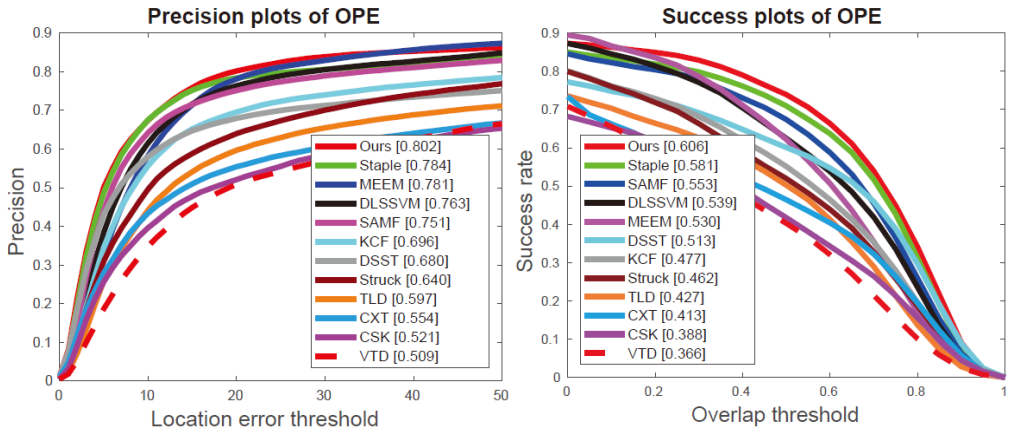

Figure 2: Overall distance precision plot (left) and overlap success plot (right) with one-pass evaluation (OPE) over 100 sequences (OTB-100). The legends show the precision scores and AUC scores for each tracker. The top performer in each measure is shown in red, and the second and third best are shown in blue and green, respectively

We evaluate our tracker with 11 state-of-the-art trackers designed with conventional handcrafted features including DLSSVM [Ning, Yang, Jiang et al. (2016)], DSST [Danelljan, Häger, Khan et al. (2014)], STAPLE [Bertinetto, Valmadre, Golodetz et al. (2015)], SAMF SAMF [Li and Zhu (2014)], KCF [Henriques, Rui, Martins et al. (2014)], MEEM [Zhang, Ma and Sclaroff (2014)], STRUCK [Hare, Golodetz, Saffari et al. (2016)], TLD [Kalal, Matas and Krystian (2010)], VTD [Kwon and Lee (2010)], CXT [Dinh, Vo and Medioni (2011)], CSK [Rui, Martins and Batista (2012)]. Among them, Struck and DLSSVM are structured SVM based methods, Staple, KCF, DSST, CSK and SAMF are CF based tracers, MEEM is developed based on regression and multiple trackers. Fig. 2 shows the overall precision and success plots on OTB-100.

Table 1: Comparison with 11 state-of-the-art trackers on 100 sequences in terms of DP, OP, CLE, and VOR

	Ours	DLSSVM	DSST	VTD	STAPLE	SAMF	TLD	KCF	MEEM	STRUCK	CXT	CSK
DP	**0.80**	0.76	0.68	0.51	0.78	0.75	0.60	0.70	0.78	0.64	0.55	0.52
OP	**0.74**	0.62	0.60	0.40	0.71	0.67	0.50	0.55	0.62	0.52	0.46	0.42
CLE	34.48	32.85	50.34	67.41	31.42	36.39	60.70	44.75	**27.71**	47.03	67.41	304.02
VOR	**0.61**	0.54	0.52	0.37	0.59	0.56	0.43	0.48	0.53	0.47	0.42	0.39

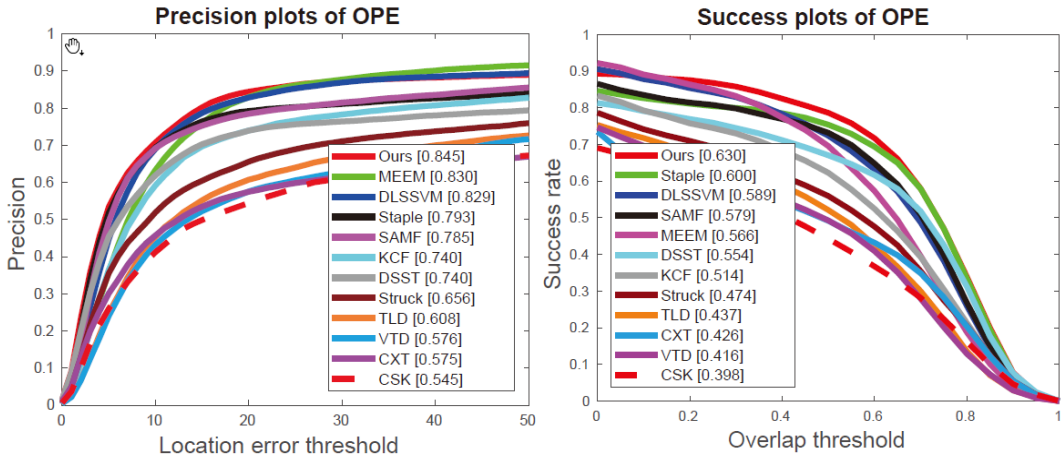

Figure 3: Overall distance precision plot (left) and overlap success plot (right) with one-pass evaluation (OPE) over 51 sequences (OTB-13). The top performer in each measure is shown in red, and the second and third best are shown in blue and green, respectively

Due to the lack of benchmark datasets and evaluation methods, the results of algorithms such as TLD, VTD, CXT, and CSK are not satisfactory with the highest DP indicator is 0.60. With the solution to the key issues of the benchmark and the presentation of some advanced methods (correlation filtering and deep learning), the performance of the algorithm has been greatly improved and the lowest index of DP has reached 0.64. Without comparing to our method, DLSSVM, STAPLE, MEEM, and SAMF have relatively competitive performance among the 11 methods. What is worth noting is the performance of MEEM in CLE, which is mainly owing to the multi-expert restoration program to solve the problem of model drift in online tracking. Through comparison, we can find that the tracker we proposed performs better than all the methods. As shown in Tab. 1, our algorithm consistently performs better than 11 recently proposed methods in DP, OP, CLE, and VOR on OTB-100. In addition, we also report our results on the OTB-13 dataset, as shown in Fig. 3. The proposed method performs better than the competing methods.

4 Conclusion

This paper proposed a structured object tracking method with compact shape and colour feature. In order to enhance the descriptive ability of the features, we added the shape features of the tracking object and fused them with the colour features to form new features. However, the direct use of such high-dimensional features will increase the computational complexity of object tracking. To alleviate computational cost, we used hashing method to reduce the dimensions of the new features and generate a compact representation for shape and colour feature of the object. Then we used the structured classification function to learn and update online to estimate the optimal object region. Through experimental verification, we can find that the proposed method achieves promising performance compared with the state-of-the-art visual tracking methods.

Acknowledgement: This work was supported by the National Key Research and Development Plan (No. 2016YFC0600908), the National Natural Science Foundation of China (No. 61772530, U1610124), Natural Science Foundation of Jiangsu Province of China (No. BK20171192), and China Postdoctoral Science Foundation (No. 2016T90524, No. 2014M551696).

References

Atluri, S. N. (2004): *The Meshless Local Petrov-Galerkin (MLPG) Method.* Tech Science Press, USA.

Avidan, S. (2004): Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064-1072.

Avidan, S. (2007): Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271.

Babenko, B.; Yang, M. H.; Belongie, S. (2011): Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619-1632.

Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P. H. S. (2015): Staple: Complementary learners for real-time tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1401-1409.

Bolme, D. S.; Beveridge, J. R.; Draper, B. A.; Lui, Y. M. (2010): Visual object tracking using adaptive correlation filters. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2544-2550.

Collins, R. T.; Liu, Y.; Leordeanu, M. (2005): Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643.

Danelljan, M.; Häger, G.; Khan, F. S.; Felsberg, M. (2014): Accurate scale estimation for robust visual tracking. *British Machine Vision Conference*, pp. 1-11.

Danelljan, M.; Häger, G.; Khan, F. S.; Felsberg, M. (2015): Learning spatially regularized correlation filters for visual tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 4310-4318.

Danelljan, M.; Khan, F. S.; Felsberg, M.; Weijer, J. V. D. (2014): Adaptive color attributes for real-time visual tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1090-1097.

Dinh, T. B.; Vo, N.; Medioni, G. (2011): Context tracker: Exploring supporters and distracters in unconstrained environments. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1177-1184.

Grabner, H.; Leistner, C.; Bischof, H. (2008): Semi-supervised on-line boosting for robust tracking. *European Conference on Computer Vision*, pp. 234-247.

Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M. M. et al. (2016): Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096-2109.

Henriques, J. F.; Rui, C.; Martins, P.; Batista, J. (2014): High-speed tracking with

kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583-596.

Hong, S.; You, T.; Kwak, S.; Han, B. (2015): Online tracking by learning discriminative saliency map with convolutional neural network. *International Conference on Machine Learning*, pp. 597-606.

Kalal, Z.; Matas, J.; Mikolajczyk, K. (2010): Pn learning: Bootstrapping binary classifiers by structural constraints. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 49-56.

Kwon, J.; Lee，K. M. (2010): Visual tracking decomposition. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1269-1276.

Li, X.; Hu, **W.; Zhang, Z.; Zhang, X. Q.; Zhu, M. L. et al.** (2008): Visual tracking via incremental log-Euclidean Riemannian subspace learning. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Li, Y.; Zhu, J. (2014): A scale adaptive kernel correlation filter tracker with feature integration. *European Conference on Computer Vision*, pp. 254-265.

Liu, B.; Huang, J.; Yang, L.; Kulikowsk, C. (2011): Robust tracking using local sparse appearance model and k-selection. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1313-1320.

Ma, C.; Huang, J. B.; Yang, X.; Yang, M. H. (2015): Hierarchical convolutional features for visual tracking. *2015 IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3074-3082.

Ma, C.; Yang, X.; Zhang, C.; Yang, M. H. (2015): Long-term correlation tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 5388-5396.

Mei, X.; Ling, H. (2011): Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259-2272.

Ning, J.; Yang, J.; Jiang, S.; Zhang, L.; Yang, M. H. (2016): Object tracking via dual linear structured SVM and explicit feature map. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 4266-4274.

Possegger, H.; Mauthner, T.; Bischof, H. (2015): In defense of color-based model-free tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2113-2120.

Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q. et al. (2016): Hedged deep tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 4303-4311.

Rez, P.; Hue, C.; Vermaak, J.; Gangnet, M. (2002): Color-based probabilistic tracking. *European Conference on Computer Vision*, pp. 661-675.

Ross, D. A.; Lim, J.; Lin, R. S.; Yang, M. H. (2008): Incremental learning for robust visual tracking. *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125-141.

Rui, C.; Martins, P.; Batista, J. (2012): Exploiting the circulant structure of tracking-by-detection with kernels. *European Conference on Computer Vision*, pp. 702-715.

Shwartz, S.; Singer, Y.; Pegasos, N. (2011): Primal estimated subgradient solver for

SVM. *Mathematical Programming*, vol. 27, no. 1, pp. 807-814.

Wang, L.; Ouyang, W.; Wang, X.; Lu, H. (2015): Visual tracking with fully convolutional networks. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3119-3127.

Wang, L.; Ouyang, W.; Wang, X.; Lu, H. (2016): Stct: Sequentially training convolutional networks for visual tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1373-1381.

Wu, Y.; Lim, J.; Yang, M. H. (2013): Online object tracking: A benchmark. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2411-2418.

Wu, Y.; Lim, J.; Yang, M. H. (2015): Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834-1848.

Xing, J.; Gao, J.; Li, B.; Hu, W.; Yan, S. (2013): Robust object tracking with online multi-lifespan dictionary learning. *IEEE International Conference on Computer Vision*, pp. 665-672.

Yao, R. (2015): Robust model-free multi-object tracking with online kernelized structural learning. *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2401-2405.

Yao, R.; Shi, Q.; Shen, C.; Zhang, Y.; Hengel, A. V. D. (2017): Part-based robust tracking using online latent structured learning. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 6, pp. 1235-1248.

Yao, R.; Shi, Q.; Shen, C.; Zhang, Y.; van den Hengel, A. (2012): Robust tracking with weighted online structured learning. *European Conference on Computer Vision*, pp. 158-172.

Yao, R.; Shi, Q.; Shen, C.; Zhang, Y.; Van Den Hengel, A. (2013): Part-based visual tracking with online latent structural learning. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2363-2370.

Yao, R.; Xia, S.; Zhang, Z.; Zhang, Y. (2017): Real-time correlation filter tracking by efficient dense belief propagation with structure preserving. *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 772-784.

Zhang, J.; Ma, S.; Sclaroff, S. (2014): Meem: Robust tracking via multiple experts using entropy minimization. *European Conference on Computer Vision*, pp. 188-203.

Zhang, K.; Zhang, L.; Liu, Q.; Zhang, D.; Yang, M. H. (2014): Fast visual tracking via dense spatio-temporal context learning. *European Conference on Computer Vision*, pp. 127-141.

Zhang, K.; Zhang, L.; Yang, M. H. (2012): Real-time compressive tracking. *European Conference on Computer Vision*, pp. 864-877.

Zhang, L.; Van Der Maaten, L. (2014): Preserving structure in model-free tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756-769.