Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5531222, 9 pages
https://doi.org/10.1155/2021/5531222

Research Article

Robust Visual Tracking Based on Convolutional Sparse Coding

Yun Liang, Dong Wang, Yijin Chen, Lei Xiao, and Caixing Liu

College of Mathematics and Informatics, South China Agricultural University, Guangzhou 501642, China

Correspondence should be addressed to Dong Wang; wdngng@163.com

Received 6 January 2021; Revised 2 February 2021; Accepted 28 February 2021; Published 24 March 2021

Academic Editor: Xiaojie Wang

This paper proposes a new visual tracking method by constructing the robust appearance model of the target with convolutional sparse coding. First, our method uses convolutional sparse coding to divide the interest region of the target into a smooth image and four detail images with different fitting degrees. Second, we compute the initial target region by tracking the smooth image with the kernel correlation filtering. We define an appearance model to describe the details of the target based on the initial target region and the combination of four detail images. Third, we propose a matching method by the overlap rate and Euclidean distance to evaluate candidates and the appearance model to compute the tracking results based on detail images. Finally, the two tracking results are separately computed by the smooth image, and the detail images are combined to produce the final target rectangle. Many experiments on videos from Tracking Benchmark 2015 demonstrate that our method produces much better results than most of the present visual tracking methods.

1. Introduction

Visual tracking is a hot topic in computer vision and graphics. The changes in background and object bring many tracking challenges such as deformation, occlusion, rotation, and so on. Now, it is still under well solved to produce accurate tracking results. Many tracking methods have been proposed recently. They are divided into two categories: generative tracking methods and discriminative tracking methods. The generative methods usually describe and identify the target with the maximum likelihood probability or posterior probability. The discriminative tracking methods often train a classification model to separate target and background.

The generative tracking methods find the candidate most similar to the target object as the tracking result. For example, Black and Jepson [1] proposed a subspace-based method to calculate the radiation transformation between the current frame and the image reconstructed with the feature vector of target. Later, Ross et al. [2] improved it by updating the basis of feature space online. Mei and Ling [3] solved the sparsity between the target template and the subspace composed of positive and negative trivial templates through the l_1 regularized least squares, which performed well in dealing with illumination variations and occlusions. Subsequently, many tracking methods [4, 5] have proposed to improve the tracking results by optimizing the l_1 algorithm. Although the generative tracking methods made a great breakthrough, they are limited by how to accurately separate the background and target, especially when dealing with clutter background and great deformation.

The discriminative methods often learn to distinguish target and background based on the cues from previous frames. For example, the tracker based on support vector machine (SVM) [6] distinguishes the target and background by learning positive and negative samples. Following the SVM, Hare et al. [7] proposed the structured support vector machine (SSVM) to further enhance its discriminating ability in dealing with deformation and occlusion challenges. Later, Ning et al. [8] proposed the dual linear structured support vector machine (DLSSVM) based on SSVM to produce efficient high-dimensional features of target and candidates.

The recent discriminative trackers are often defined by correlation filtering [9–15]. For example, Bolme et al. [9] proposed minimum output sum of squared error (MOSSE) tracking algorithm, which first applied correlation filtering in visual tracking. The MOSSE performs a convolution calculation on the Fourier domain between the target template and

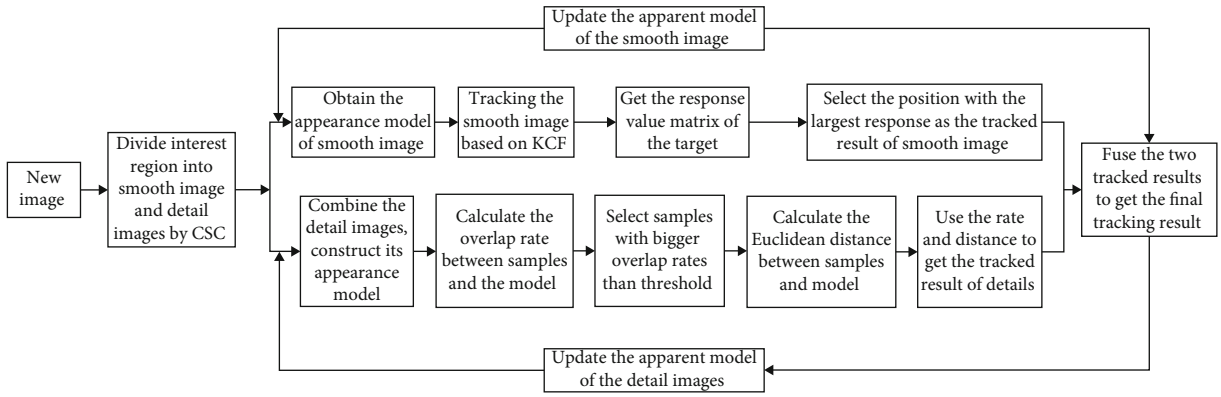

Figure 1: The flowchart of the proposed tracking method.

the interest region of target. Based on MOSSE, Henriques et al. [10] introduced the cyclic matrix and kernel method of tracking and convolving dense samples with the cyclic matrix formed by the target template in the Fourier domain. Then, they proposed the circulant structure kernel (CSK) tracking algorithm. Based on the CSK, Henriques et al. [11] introduced histogram of oriented gradient (HOG) feature to convert a single channel to multiple channels to improve the tracking results without increasing the time cost. Bertinetto et al. [12] integrated the features from both HOG and color cues to further improve tracking accuracy. Danelljan et al. [13] used the depth feature of single-layer convolution in CNN to replace the HOG feature in spatially regularized correlation filters to deal with tracking challenges. Danelljan et al. [14] improved the speed and stability of the algorithm based on the continuous convolution operators by reducing the model parameters and adopting a sparse update strategy. Valmadre et al. [15] used an end-to-end learning method to treat correlation filtering as a layer in CNN to reduce tracking drift and failure.

Recently, the tracking methods with deep features [16–19] have become very popular for their good performance in describing target and background. Li et al. [16] proposed a method to learn target perception features and integrate these features with Siamese matching network. Wang et al. [17] proposed a SiamFC-based tracker using "rough matching" and "fine matching." They enhanced tracking robustness through training in rough matching and improved discrimination through distance learning network in fine matching. Du et al. [18] proposed a tracker to detect target corners. It first uses the Siamese network to roughly distinguish the foreground and background to get the interest region of target. Then, they used the relationship between the target template and interest region to highlight the corner regions and enhance the features of the corner to produce a more accurate bounding box. Guo et al. [19] proposed a fully convolutional Siamese network for tracking. Chen et al. [20] proposed a Siamese box adaptive network structure named SiamBAN. The network views tracking as a parallel classification and regression problem, then classifies the objects, and regresses their bounding boxes in a fully convolutional network. Danelljan et al. [21] proposed a probabilistic regression model for tracking. Yang et al. [22] defined a tracking model by an offline recurrent neural optimizer to update the tracking model in a meta-learning setting. Li et al. [23] improved the tracking by integrating alignment data with deep features, then used ConditionNet to bridge the gap between the preconditioning and learning process.

The above tracking methods are mostly defined by establishing an appearance model of target. Therefore, when the target appearance model becomes not robust or inaccurate, it is very difficult to correct the model to improve the performance in tracking the following frames. Especially, for the updated target appearance model, if the online update ability of the model is too strong, it is easy to take the surrounding background as target information and introduce overfitting. However, if the online update ability of the appearance model is too weak, it leads to underfitting and tracking drift. This paper divides the target into a smooth image and four detail images based on convolutional sparse coding (CSC) and separately establishes target appearance models for them to track targets. The proposed tracker is achieved by combining the tracking results of the two parts to cope with challenges and improve the tracking performance.

2. Our Tracking Framework

This paper first extracts the interest region by expanding the target rectangle area by 2.5 times. Then, we divide the interest region into a smooth image and four detail images with different fitting degrees by CSC. For the smooth image, the model is initialized and tracked based on the KCF. For the detail images, we first combine the four detail images with different fitting degrees and construct the appearance model of the combined image to represent the target details. Then, we evaluate the candidates by measuring the overlap rate and Euclidean distance between the candidates and the appearance model. This evaluation describes that how much a candidate matches the appearance model, and the best-matched candidate is the tracked result based on detail images. Finally, the tracked results of the details and the smooth image are combined to produce the final tracking result. To suit the changes of the target, we update the appearance model with the tracked result frame by frame. The flowchart of our method is shown in Figure 1. It includes the target model initialization phase, target tracking, and model update.

2.1. Initialize the Appearance Model. As shown in Figure 1, we first extract the interest region based on the target rectangle of the last frame. Then, we divide the interest region into a smooth image and four detail images with different fitting degrees by the CSC. For the smooth image, we initialize its appearance model based on the KCF method. For the detail images, we combine the detail images with different fitting degrees and establish an appearance model for it to describe the target details.

2.2. Tracking Target Object. After producing the smooth image and detail images of the target, we track the two parts separately with different approaches. As described in the top row of Figure 1, for the smooth image, we use the KCF method to construct its appearance model and get the response value matrix. The tracking result based on a smooth image is selected as the candidate with the largest response value. Similarly, as shown in the bottom row of Figure 1, for the detail images, we first combine the four details and construct the appearance model. Then, we compute the overlap rates between samples and the appearance model and select the samples with bigger rate values. Later, we compute the Euclidean distance values between the selected samples and the appearance model. Then, we select the sample with the biggest overlap rate and minimum distance as the tracking result based on detail images. Finally, we fuse the two tracked results based on the smooth image and detail images to get the final tracking result.

2.3. Update the Appearance Model. The model update includes the appearance model update of smooth image and detail images. For the smooth image, an appearance model is first established according to the tracked result. Then, it is combined with the old one to form the updated appearance model. For the detail images, according to the new tracked result, four new detail models are extracted and then combined with different fitting degrees to replace the appearance model about target details to achieve model update.

3. Target Tracking Based on the CSC

In this section, we first describe how to divide an interest region into a smooth image and four detail images based on the CSC and how to establish the appearance models of smooth image and detail images, respectively, as shown in Figure 2. Second, we separately describe how to track the smooth image and the detail images in Section 3.2. Finally, we describe the details of the appearance model update in Section 3.3.

3.1. Initialize the Appearance Model for Smooth Image. Following the method proposed in [24], we divide the interest region of the target into the smooth part and the detail parts using N filters as shown in Figure 2. The smooth part contains the cues about the color and shape features of target. The detail parts describe the features about the image edge and texture structure. Equation (1) describes the separation

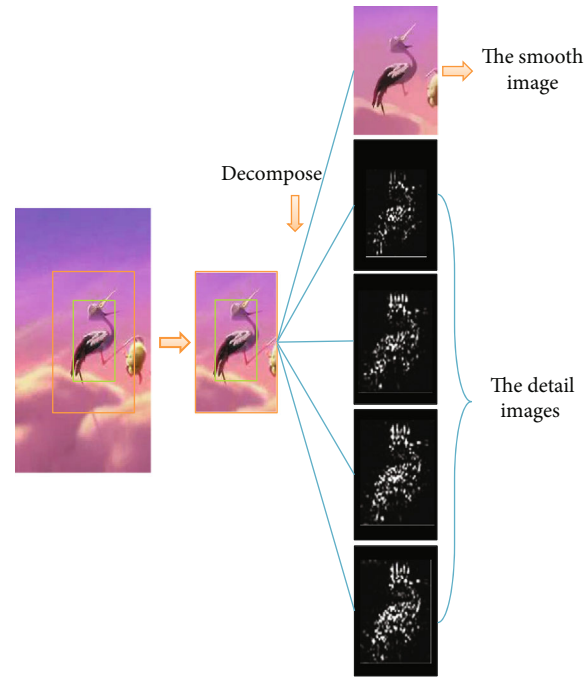

Figure 2: Divide the interest region into a smooth image and four detail images.

of smooth part and detail parts:

$$y = f^s \otimes Z_y^s + Y, \tag{1}$$

where y represents the original image, $f^s \otimes Z_y^s$ describes the smooth part, and Y describes the detail part. f^s is a low-pass filter and Z_y^s is a characteristic diagram.

As shown in Figure 2, the green rectangle describes the target region, and the red rectangle shows the interest region of target. As shown in the third column, it is divided into a smooth image and four detail images with different fitting degrees based on the CSC.

3.1.1. Initialize the Target Appearance Model of the Smooth Image. We construct the appearance model about the smooth image based on the KCF method. Therefore, the first step to initialize the tracking target is to construct a cyclic matrix $C(x)$ by extracting the target features. Then we diagonalize the cyclic matrix to obtain the diagonal cyclic feature matrix X using the Discrete Fourier transform, as shown in Equation (2).

$$X = F \operatorname{diag}\left(\widehat{x}\right) F^H, \tag{2}$$

where F describes the Constant Fourier matrix and satisfies $F F^H = 1$. $\widehat{x}$ is the generated vector after Fourier transform. We solve the least squares in the Fourier domain based on X to train the target detector $\widehat{\alpha}$, as described in Equation (3).

$$\widehat{\alpha} = \frac{\widehat{y}}{k\wedge^{xx} + \lambda}, \tag{3}$$

where $k\wedge^{xx}$ is the first row of the kernel matrix in the Fourier

domain, and $\hat{y}$ is the regression target training based on the Gaussian kernel function.

3.1.2. Initialize the Appearance Model for the Detail Images. As shown in Figure 3, this paper constructs the appearance model for the details of the target based on the detail images. This paper employs 400 filters to implement CSC. Therefore, it can get 400 detailed feature maps about tracking targets $(r_1 \sim r_{400})$. To prevent underfitting or overfitting, this paper combines every 100 feature maps to form a detail image. Therefore, we obtain four detail images by $R_1 = \sum_{i=1}^{100} r_i, R_2 = \sum_{i=101}^{200} r_i, R_3 = \sum_{i=301}^{300} r_i, R_4 = \sum_{i=401}^{400} r_i$, then four detailed models $(R_1 \sim R_4)$ with different fitting degrees are constructed to describe the details of target. Finally, we combine $R_1 R_1$, R_2, R_3, and R_4 to get the final appearance model of detail images named as R, as shown in Equation (4) where $\sum_{i=1}^{4} \lambda_i = 1$.

$$R = \sum_{i=1}^{4} \lambda_i R_i. \tag{4}$$

3.2. Tracking Target Based on CSC. This section introduces our tracking method based on the CSC in detail. For each frame of a video, we first use the CSC to divide it into the smooth image and four detail images. Second, we use KCF to track the target region based on the smooth image. Then, we predict the target region based on the appearance model of image details. Finally, we compute the final tracking result by combining the target regions based on both the smooth image and detail images.

3.2.1. Tracking Target Based on Smooth Image. We expand the target region from the last frame by 2.5 times to form the sampling area Z. We extract the features of the sampling area to form a feature matrix Z. Then, we calculate the kernel correlation values of the cyclic feature matrix X and Z using the kernel functions. It means that $K^x = C(k^{xz})$, where k^{xz} is a row vector from X and Z through the kernel function operation, and C is a function that constructs a cyclic matrix. K^{xz} is a circled function. After that, we use target detectors $\widehat{\alpha}$ and K^{xz} to do dot multiplication to get the response value matrix $\widehat{f}(z)$ of each position in the sampling area by Equation (5).

$$\widehat{f}(z) = \widehat{\alpha} \odot k\wedge^{xz}. \tag{5}$$

The sample with a larger response value matrix has more possibility to be used as the target. The position of the max response value is taken as the target center based on the smooth image.

3.2.2. Tracking Target Based on Detail Images. To predict the target based on detail images, we first randomly select 400 samples $(s_1 \sim s_{400})$ which has the same size as target in the interest region Z. Then, we calculate the overlap rate (OR) between each sample and the appearance model of detail images by Equation (6).

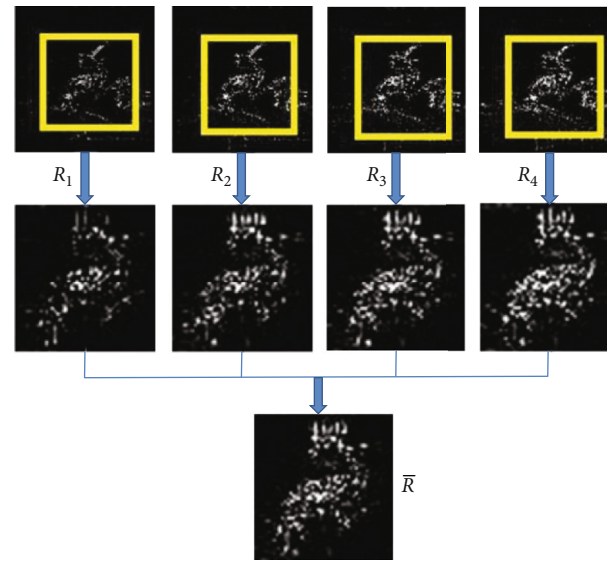

Figure 3: Constructs the appearance model for the detail images to describe the target details.

$$\begin{cases} \text{Width} = (W_s + W_t) - (\max(x_{sr}, x_{tr}) - \min(x_{sl}, x_{tl})), \\ \text{High} = (H_s + H_t) - (\max(y_{su}, y_{tu}) - \min(y_{sd}, y_{td})), \\ \text{OR} = \dfrac{\text{Width} \bullet \text{High}}{W_t * H_t} \times 100\%. \end{cases} \tag{6}$$

The Width is the width of the overlapping part, and the High is the high of the overlapping part. W_t and H_t are the width and height of the target model. W_s and H_s are the width and height of samples. (x_{tr}, y_{td}) and (x_{tl}, y_{tu}) represent the coordinate values of the down right point and the top left point of the target model. (x_{sr}, y_{sd}) and (x_{sl}, y_{su}) represent the coordinate values of the down right point and the top left point of the sample. When the overlap rate is greater than the setting threshold, the sample is retained; otherwise, the sample is rejected directly.

For the retained samples, we calculate the Euclidean distance between them and the appearance model of the detail images. First, for each sample, we extract four detail images with different fitting degrees accumulated by every 100 detail features. Then, we combine the four detail images to obtain the detail description of the sample, denoted by S. Finally, we calculate the Euclidean distance as EU between S and the appearance model of the detail part R by Equation (7).

$$\text{EU} = \sqrt{(R - S)^2}. \tag{7}$$

The sample with the minimum distance is taken as the final tracked result based on detail images.

3.2.3. Computing the Final Tracking Result. We use pos_s to describe the center of the tracked result based on the smooth image and use pos_r to describe the center of the tracked result

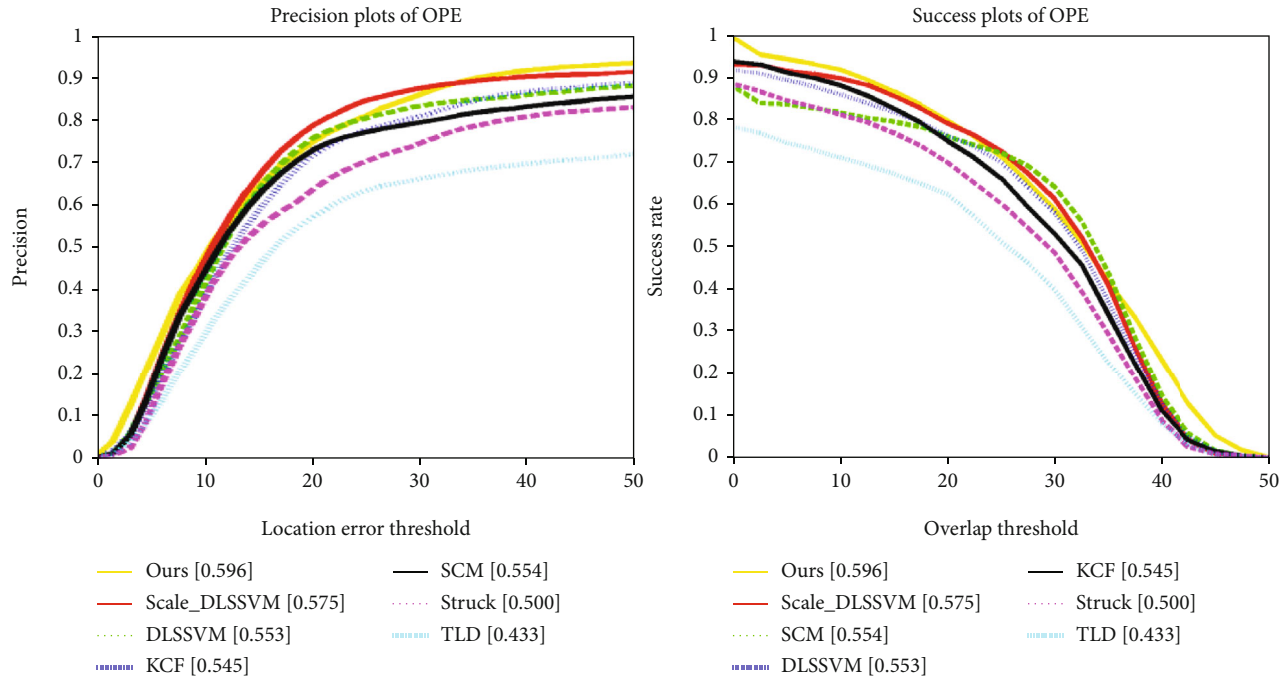

Figure 4: The overall rate of precision for center position error and success (for success rate).

Table 1: Tracking accuracy of 11 challenges where * is the first, ** is the second, and *** is the third.

Trackers	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Ours	0.658	0.781*	0.652	0.681**	0.712***	0.745**	0.636***	0.677**	0.713**	0.648**	0.708***
KCF	0.665***	0.757**	0.662***	0.676***	0.726*	0.710***	0.610	0.657***	0.700***	0.606***	0.712**
DLSSVM	0.705*	0.708***	0.752*	0.703*	0.714**	0.747*	0.747**	0.749*	0.755*	0.714*	0.736*
Struck	0.684**	0.596	0.719**	0.574	0.626	0.640	0.841*	0.639	0.649	0.606	0.672
TLD	0.558	0.537	0.616	0.469	0.581	0.604	0.515	0.542	0.590	0.537	0.608
VTD	0.346	0.449	0.306	0.451	0.482	0.583	0.559	0.509	0.594	0.434	0.584
CT	0.345	0.411	0.324	0.443	0.394	0.448	0.327	0.400	0.430	0.260	0.459

Table 2: The coverage of the success rate value for 11 challenges.

Trackers	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Ours	0.520***	0.605*	0.523	0.494***	0.523*	0.555*	0.369***	0.504**	0.520**	0.566**	0.462**
KCF	0.520***	0.588**	0.527***	0.497**	0.509***	0.525***	0.369***	0.486***	0.501***	0.530***	0.461***
DLSSVM	0.569*	0.560***	0.592*	0.526*	0.518**	0.545**	0.452**	0.563*	0.545*	0.609*	0.484*
Struck	0.549**	0.483	0.566**	0.430	0.458	0.474	0.482*	0.479	0.468	0.522	0.442
TLD	0.481	0.429	0.524	0.345	0.427	0.458	0.335	0.424	0.440	0.486	0.439
VTD	0.294	0.341	0.252	0.345	0.361	0.428	0.350	0.381	0.443	0.417	0.410
CT	0.287	0.316	0.233	0.352	0.282	0.323	0.126	0.292	0.308	0.280	0.310

based on detail images. The final tracked result is computed by combining pos_s and pos_r based on Equation (8).

$$pos_{tar} = \eta_1 pos_s + \eta_2 pos_r, \tag{8}$$

where $\eta_1 + \eta_2 = 1$. In our experiments, we set η_1 and η_2 both to be 0.5. The size of the tracked result also follows the same process of the center points of tracking results.

3.3. Update the Appearance Model of Target. Obtaining the tracking result on the current frame, we need to update the appearance model of the target to adapt to the changes of the target. This update is achieved by separately updating

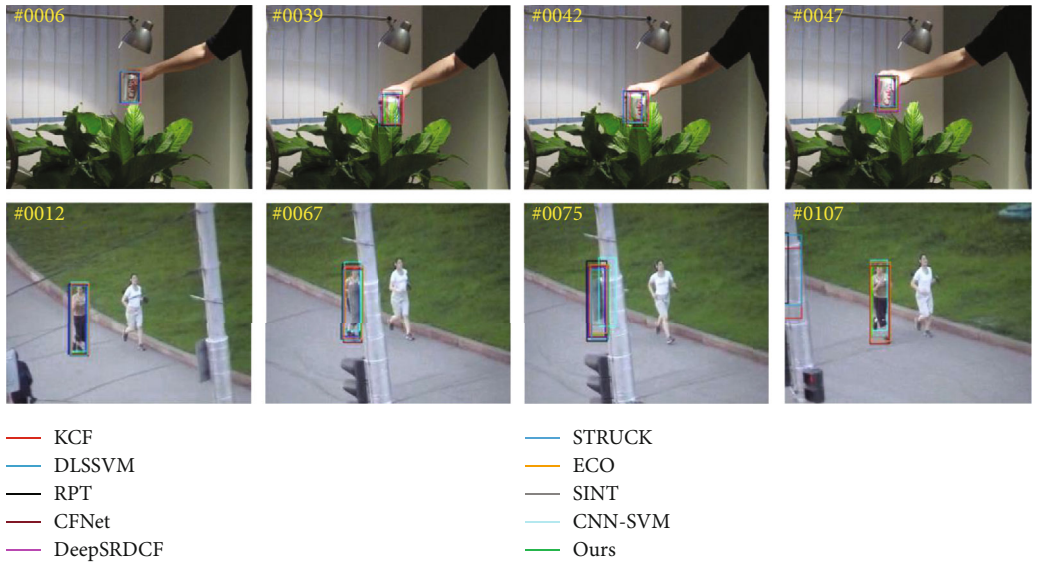

Figure 5: Comparison with occlusion challenge.

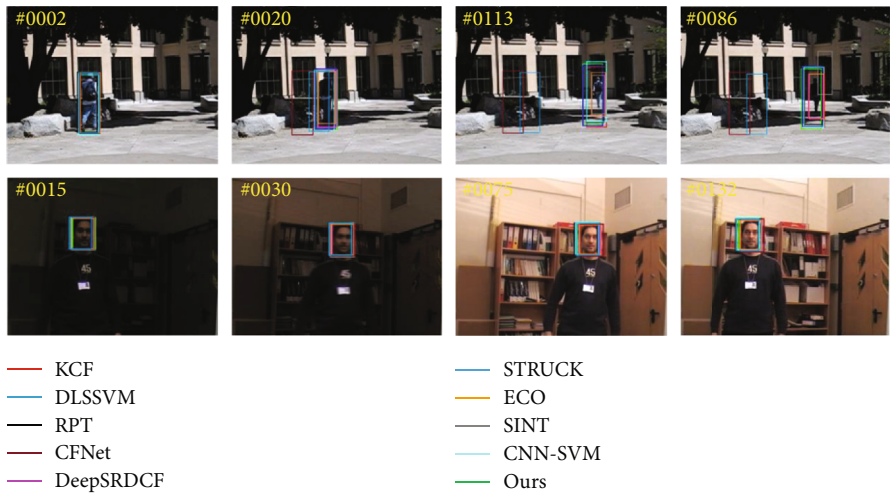

Figure 6: Comparison with illumination variation challenge.

the appearance model of the smooth image and the appearance model of the detail images.

3.3.1. Update the Appearance Model of the Smooth Image. The update of the model about the smooth image is achieved by updating the cyclic feature matrix X and the target detector $\widehat{\alpha}$. After getting the center position of target, the matrix X_t and the detector $\widehat{\alpha}_t$ on frame t are obtained. Then, we combine them with the cyclic feature matrix X_{T-1} and the target detector $\widehat{\alpha}_{T-1}$ at frame $t-1$ to get the updated X_t and $\widehat{\alpha}_t$. The equation to achieve the update is defined by:

$$\begin{cases} X_T = (1-\eta)X_{T-1} + \eta X_t, \\ \widehat{\alpha}_T = (1-\eta)\widehat{\alpha}_{T-1} + \eta\widehat{\alpha}_t, \end{cases} \tag{9}$$

where X_T and $\widehat{\alpha}_T$ are the updated cyclic feature matrix and the target detector on frame t, and η is the learning parameter. X_{T-1} and $\widehat{\alpha}_{T-1}$ are the cyclic feature matrix and target detector from the last frame, which greatly preserve the stable target feature from previous frames.

3.3.2. Update the Appearance Model of the Detail Images. The update of the appearance model of the detail images is achieved by updating the four models of detail images with different fitting degrees. First, we extract four new detail images with different fitting degrees based on the interest region of the tracked result. Then, we use the method to initialize the appearance model of detail images proposed in Section 3.1 to construct the new appearance model of detail images for the current tracked result. Finally, we update the appearance model of detail images by replacing the present model with the new one.

4. Results

Our method is implemented with MATLAB 2014b on the PC with Windows 7 system, Intel i7-6700 3.4GHz processor,

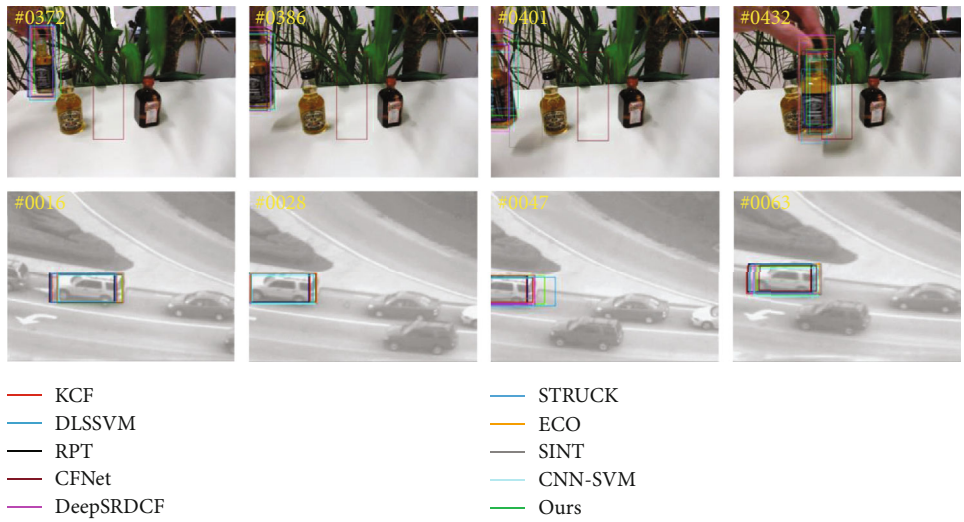

Figure 7: Comparison with out-of-plane rotation challenge.

12G video memory, and 12G memory. 61 video sequences from the Visual tracking Benchmark 2015 were used for experiments. They include several challenges such as complex background, illumination variation, and rotation. We do quantitatively and qualitatively evaluations on the famous present 13 trackers such as DLSSVM [8], KCF [11], Staple [12], ECO [24], CNN-SVM [25], CFNet [15], SINT [26], Struct [7], DeepSRDCF [14], RPT [27], TLD [28], VTD [29], and CT [30]. The results show that our method is more effective for in-plane rotation, complex background, and illumination variations.

4.1. Quantitative Evaluation. We evaluate the tracking results based on the accuracy of center position error and the success rate. The center accuracy is obtained by the center distance between the tracked result and the ideal target region, and the success rate is computed by the overlap rate of them. Compared with six target trackers, the center accuracy and the success rate of our method perform better, as shown in Figure 4. Tables 1 and 2 separately describe the evaluations for dealing with different challenges. For complex backgrounds, the center accuracy and the success rate value are ranked first with 0.781 and 0.605. For in-plane rotation, the center accuracy is 0.745 and the success rate value is ranked first with 0.555. For illumination variations, the center accuracy is 0.712 and ranks as the third, but relative to the first place KCF and the second DLSSVM differs only by 0.014 and 0.002. In addition, our success rate value is ranked first with 0.523. More details about the comparisons can be reviewed in Tables 1 and 2.

4.2. Qualitative Evaluation. This section qualitatively evaluates our method for occlusion, illumination variations, out-of-plane rotation, and so on. We compare the proposed method with nine famous algorithms.

4.2.1. Occlusion. Figure 5 shows the comparisons about target occlusion by video Coke and video jogging.1. The target is occluded by the surroundings, such as the leaves in Coke from frame 39 and the pole in Jogging.1 from frame 75. Many present trackers easily lose targets, such as Struct [7] and DLSSVM [8]. If the target is occluded completely, the background is taken as the target and finally leads to tracking drift or failure. We construct the appearance models of smooth image and detail images to describe the target, which greatly improve our performance in occlusion.

4.2.2. Illumination Variation. Figure 6 shows the comparisons among several trackers with illumination variation. We use the video Human8 (top row in Figure 6) and the video Man (bottom row in Figure 6) as examples. For video Human8, the illumination undergoes great changes. When people pass the shadow, the present trackers such as CFNet [15] and Struck [7] fail in identifying the blurred target. Using the proposed appearance model update scheme, our method performs much better than the present trackers by efficiently adapting to the changes of target appearance.

4.2.3. Out-of-Plane Rotation (OPR). Figure 7 shows the tracking results of various trackers under the OPR challenge. We use the video Liquor (top row in Figure 7) and video SUV (bottom row in Figure 7) as examples. From frame 386 to frame 401 in video Liquor, the target bottle rotates out of the plane for several times. From frame 28 and frame 47 in the video SUV, the fast movement of the car makes the camera unable to keep up, so part of target exceeds the image range. As shown in frame 386 on the top row and frame 47 on the bottom row, our method accurately detects the target region. However, some present trackers such as ECO [24] introduce track drifting. The main reason is our method using much detail information of the target to effectively separate the target and its surroundings.

5. Conclusion

This paper defines a new visual tracking method based on convolutional sparse coding. First, it extracts an interest region of the target in the current frame. Then by the

convolutional sparse coding, we divide the interest region into a smooth image and four detail images with different fitting degrees. For the smooth image, we initialise its appearance model and compute the general tracking result by kernel correlation filter. For the detail images, we first extract four detail images and combine them to initialize the appearance model for target details. We randomly sample 400 candidates in the interest region and calculate the overlap rate and Euclidean distance between each candidate and the appearance model of details to determine the tracking result based on the target details. By combining the tracking results of the smooth image and the detail images, we get the final tracking result of target. By introducing the appearance models of both the smooth image and detail images, the proposed method performs favourably in dealing with the tracking drift and failure introduced by the deformation and occlusion of target. We do quantitative and qualitative evaluations on some famous trackers on the Tracking Benchmark 2015. Many experiments demonstrate that our method produces better results in dealing with many challenges such as illumination variations, occlusion, and complex background.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (61772209), Science and technology planning project of Guangdong province (2019A050510034, 2019B020219001), the Production Project of Ministry Education China (201901240030), and the College Students Innovations Special Project of China under Grant 201910564037, 202010564026.

References

[1] M. J. Black and A. D. Jepson, "EigenTracking: robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[2] D. A. Ross, J. Lim, R. S. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[3] X. Mei and H. Ling, "Robust visual tracking using ℓ1 minimization," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1436–1443, Kyoto, Japan, September-October 2009.

[4] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient ℓ1 tracker with occlusion detection," in *CVPR 2011*, pp. 1257–1264, Colorado Springs, CO, USA, June 2011.

[5] D. Wang, H. Lu, and M. H. Yang, "Online object tracking with sparse prototypes," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 314–325, 2013.

[6] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[7] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[8] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4266–4274, Las Vegas, NV, USA, June 2016.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and L. Yui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, June 2010.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7575*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 702–715, Springer, Berlin, Heidelberg, 2012.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with Kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[12] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.

[13] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 621–629, Santiago, Chile, December 2015.

[14] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9909*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 472–488, Springer, Cham, 2016.

[15] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008, Honolulu, HI, USA, July 2017.

[16] X. Li, C. Ma, B. Wu, Z. He, and M. Yang, "Target-aware deep tracking," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1369–1378, Long Beach, CA, USA, June 2019.

[17] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: series-parallel matching for real-time visual object tracking," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3638–3647, Long Beach, CA, USA, June 2019.

[18] F. Du, P. Liu, W. Zhao, and X. Tang, "Correlation-guided attention for corner detection based visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6836–6845, Seattle, WA, USA, June 2020.

[19] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for

visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6269–6277, Seattle, WA, USA, June 2020.

[20] Z. Chen, B. Zhong, G. Li, and S. Zhang, "Siamese box adaptive network for visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6668–6677, Seattle, WA, USA, June 2020.

[21] M. Danelljan, L. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7183–7192, Seattle, WA, USA, June 2020.

[22] T. Yang, P. Xu, R. Hu, H. Chai, and A. B. Chan, "ROAM: recurrently optimizing tracking model," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6718–6727, Seattle, WA, USA, June 2020.

[23] Y. Li, A. Bozic, T. Zhang, Y. Ji, T. Harada, and M. NieBner, "Learning to optimize non-rigid tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4910–4918, Seattle, WA, USA, June 2020.

[24] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: efficient convolution operators for tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6931–6939, Honolulu, HI, USA, July 2017.

[25] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 597–606, Lille, France, July 2015.

[26] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1420–1429, Las Vegas, NV, USA, July 2016.

[27] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 353–361, Boston, MA, USA, June 2015.

[28] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 49–56, San Francisco, CA, USA, June 2010.

[29] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, San Francisco, CA, USA, June 2010.

[30] K. Zhang, L. Zhang, and M. Yang, "Real-time compressive tracking," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7574*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 864–877, Springer, Berlin, Heidelberg, 2012.