Visual Tracking via Structure Rearrangement and Multi-scale Block Appearance Model

Guang Han, Xingyue Wang[✉], and Jimuyang Zhang

Nanjing University of Posts and Telecommunications, Nanjing, China
wangxingyue2009@163.com

Abstract. In this paper, we propose a tracking method via structure rearrangement and multi-scale block based appearance model, plus a dynamic template mechanism within a two-stage search framework. Firstly, a wide range sampling is performed then confidence value of each candidate is calculated via a discriminative reverse sparse coefficient vote method, a part of candidates with large confidence value are selected to join in next stage. After a small range resampling in stage two, the candidates and target templates are divided into multi-scale patches, in addition, the background information is used to model the error term. Furthermore, a labeled template pool is maintained in the tracking process to dynamically generate an appropriate template set for next frame according to the occlusion map of current tracking result. Both qualitative and quantitative evaluations on challenging image sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

Keywords: Visual tracking · Template structure rearrangement · Multi-scale patch · Dynamic template · Sparse representation

1 Introduction

Visual tracking [1] [2] is one of the classic computer vision problems, and has a wide application in numerous scenarios [3] [4] such as video surveillance, human-computer interaction, etc. Despite the success, to achieve a reliable object tracking still needs to overcome many difficulties, for example, the target partially occlusion, illumination variation, background clutter, appearance change, etc.

In general, a visual tracking algorithm [11][12][23]-[28] can be divided into three key components: a motion model, an appearance model and a tracking strategy. The motion model [5][14] is used to describe the state of the target motion and forecast the likely target position in next frame, the appearance model[10][19]-[22] is a description method of target information including intrinsic and extrinsic characteristics of the target object, and the tracking strategy[15]-[18] is used to build the main framework of the algorithm with the motion model and appearance model.

Since the first time sparse representation is introduced into object tracking by Mei and Ling [32], it has been used in various tracking algorithms [13][23][26][28][30]-[33]. Mei use a target template set and a trivial template set to linear represent each candidate

H. Zha et al. (Eds.): CCCV 2015, Part II, CCIS 547, pp. 106–119, 2015.
DOI: 10.1007/978-3-662-48570-5_11

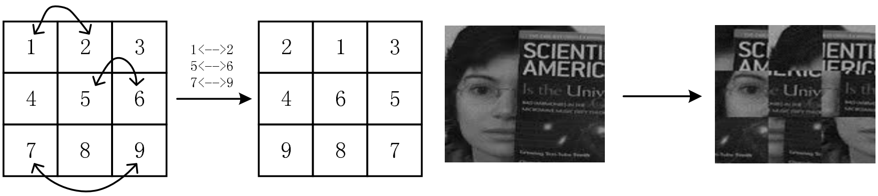
within particle filter framework. In spite of demonstrated success, this method consumes a large amount of computation and is sensitive to partial occlusion. Bai et al [34] apply a structured sparse representation model to visual tracking algorithm, and propose a block orthogonal matching pursuit (BOMP) algorithm based on orthogonal matching pursuit algorithm and mutual relations of the target on spatial structure. This algorithm fully combines the structural characteristics of the target to reduce the amount of calculation and has good robustness on the target occlusion. In addition, Jia et al. [28]proposed an align method to extract the target local sparse and spatial information, and also robust to partial occlusion, but it did not take the particle resampling process into consideration, once tracking result deviates from the actual location of the target, the tracker may completely lose the target. Zhong et al. [33] propose a sparse collaborative model that exploits both local and holistic templates, the algorithm is able to deal with partial occlusion. However, different sequences require a separate debug and different parameters, the performance is sensitive to parameters and needs high computational cost to solve L1 problem for so many local patches.

In view of above analysis, we propose the following method to deal with challenges during tracking: Firstly, to take full advantage of the holistic spatial structure information of target, we propose a template structure rearrangement method to rearrange the spatial structure of the template, this method can effectively deal with occlusion without any loss of target information. Secondly, we propose a multi-scale patch based method to resist partial occlusion with the horizontal and vertical multi-scale patches. Thirdly, we propose a dynamic target template set generated from the labeled template pool to deal with long-term occlusion or appearance change. Fourth, we use the background information to construct negative templates and model the error term, which can improve tracking robustness. Fifth, we use a two-stage search strategy to handle the situation that the target object moves fast and randomly. In addition, we use the APG [30] method to solve optimum problem to improve the processing efficiency.

2 Proposed Tracking Algorithm

2.1 Template Structure Rearrangement

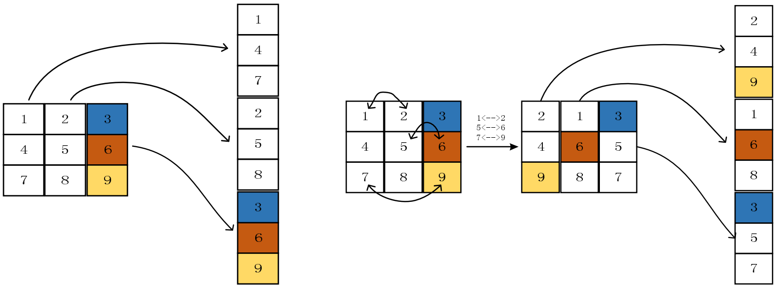
We find that the traditional L1tracker is robust to Gaussian noise, however not to partial occlusion. Inspired by this phenomenon and analysis of existing methods, we propose a template structure rearrangement method to rearrange the spatial structure of template, this method has the ability to fragment the partial occlusion appeared anywhere and as close as possible to make partial occlusion obeys the Gaussian distribution. The template is divided into 9 patches shown in Figure 1(a) and numbered from 1 to 9, and then the positions of patch 1 and 2, 5 and 6, 7 and 9 are swapped respectively. Finally, all patches are re-assembled into a holistic template as shown in the right side in Figure 1(a), the Figure 1(b) is a demonstration with actual template image.



(a) (b)

Fig. 1. A demonstration of structure rearrangement method

Usually the templates will be reshaped into a vector using the method demonstrates in Figure 2(a) when solving the L1 regularizations problem, if the target object is partial occluded (the colored patch represents occlusion), the occluded patch is still continuous distribution, which is seriously deteriorate the tracking performance. If the template structure is rearranged before reshaped into a vector shown in Figure 2(b), the occluded patch is divided into several smaller sub-patches so that the original continuous partial occlusion becomes close to trivial Gaussian occlusion.



(a) (b)

Fig. 2. A demonstration of handling partial occlusion

The proposed structure rearrangement method can effectively divide the partial occlusion into trivial occlusion and maintain spatial structure information of the target object itself without increasing the computational cost.

2.2 Discriminative Reverse Sparse Representation Based Vote Method

The traditional sparse representation based trackers use the target templates to linear represent the candidates and perform computationally expensive L1 regularizations problem at each frame for each candidate. To make sure the robust of L1 tracker, the candidates always amounts to hundreds or even thousands, which results in the tracker is unsuitable for an application with a real-time requirement. In this paper, motivated by a reverse thought of traditional sparse representation, we construct the dictionary with the candidate set Y to represent each target template as in Eq.1.

$$T = Y_a + e \tag{1}$$

Where T denotes the target template, Y donates the candidate set, a is the sparse coefficient vector, e is the error term, then the L1 regularizations problem can be rewrite as Eq.2.

$$\arg\min_a \|T_i - Ya_i\|_2^2 + \lambda\|a_i\|_1,\ s.t.\ a_i \geq 0 \tag{2}$$

With sparsity constraint, only a few candidates which are similarly to the template would be involved in representing the template, and with non-negativity constraints the coefficient vector denotes the similarity between candidate and target template, therefore, a larger element of a means the corresponding candidate is more similar with the target template.

According to the characteristics of the sparse coefficient, each template contributes a vote for the candidate set, the vote value are a_i, combining all the voting values of the target template to arrive at a more accurate voting results a.

$$\mathrm{a} = \sum_i a_i \tag{3}$$

To further increase the accuracy of the voting, we introduce a weight W_{ij} for each voting value, W_{ij} represents the similarity between i-th template and j-th candidate:

$$W_{ij} \propto \cos < t_i, y_j > \tag{4}$$

Thus, the vote value of j-th candidate is W_j.

$$W_j \propto \sum_i \cos < t_i, y_j > \tag{5}$$

And the confidence value of each candidate is S_j^f.

$$\mathrm{S}_j^f = W_j \odot a_j \tag{6}$$

Where $\odot$ is the element-wise product, $W_j \propto \sum_i W_{ij}$,is the vote value of j-th candidate.

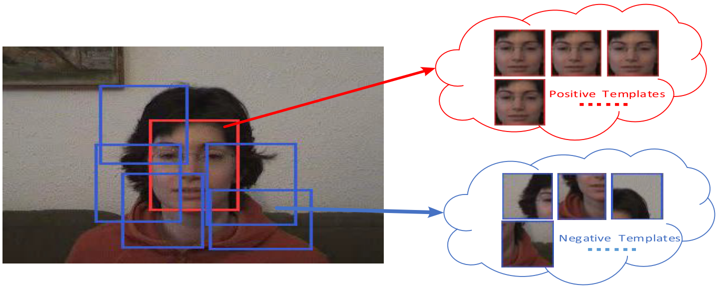


Fig. 3. Positive and negative templates

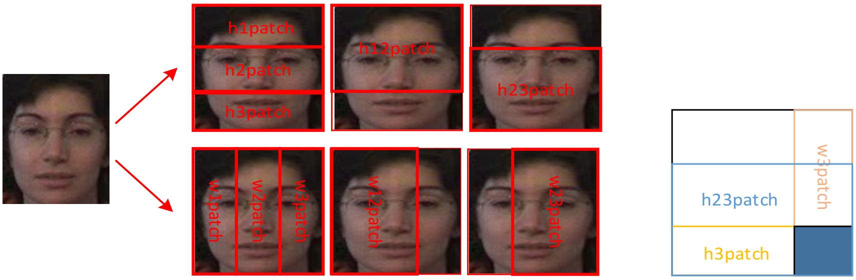
In order to further improve the tracking robustness, a negative template set (shown is Figure 3) is used to vote for each candidate in accordance with the above steps, then we can get the confidence value S_j^b generated from negative templates, thus, each candidate has a final confidence value S_j.

$$S_j = S_j^f - S_j^b \tag{7}$$

In this paper, the solution process is reduced to only dozen times by using the method of reverse sparse representation instead of the traditional sparse representation, and take full advantage of the inherent characteristics of the sparse coefficient to calculating a confidence value for each candidate.

2.3 Multi-scale Block Appearance Model

Multi-scale blocks are the patches divided from target template as shown in Figure 4(a). Each template has four scales of 10 sub-patches, and these sub-patches divided as following: trisect the template in the vertical direction and denoted by h1patch, h2patch, h3patch, respectively, then combine h1patch and h2patch as h12patch, combine h2patch and h3patch as h23patch; similarly, obtain w1patch, w2patch, w3patch, w12patch, w23patch in the horizontal direction. The benefits of this division method are: Firstly, the number of needed patches is less than the overlap patches based method, therefore the additional amount of calculation is small; Secondly, some patches contain other patches which maintains the structural information of the target object itself; Finally, the use of vertical and horizontal direction patch method having a similar function with coordinate axis which can be more quickly locate a smaller portion occlusion. As shown in Figure 4(b), when h3patch and w3patch are occluded, but h23patch or w23patch are not, then we can know that the colored patch in bottom right corner is occluded.



(a) (b)

Fig. 4. A demonstration of multi-scale target region division

2.4 Occlusion Map Creation

Firstly, we obtain the multi-scale patches of target template set and current tracking result, then calculate the cosine similarity between the patch $r_{i,\mathrm{i}=\{1,2,\ldots,10\}}$ of tracking result and the corresponding position patch t_j of target templates, the confidence value con_i of each patch is proportional to the cosine similarity.

$$con_i \propto \sum_j \cos < r_i, t_j > \tag{8}$$

Thus, the occlusion map of current tracking result is written as $O = \{o_1, o_2, \ldots, o_{10}\}$,

$$o_i = \begin{cases} 1 & con_i < thr \\ 0 & con_i > thr \end{cases} \tag{9}$$

Where the thr is a predefined threshold. The occlusion map indicate whether 10 patches are occluded, depending on each element of O we can infer more accurate occlusion position and occlusion size.

2.5 Construction and Update of Dynamic Template Set

Due to partial occlusion occurs randomly, we consider not only the size but also the duration of occlusion. As shown in Figure 5, the dynamic template set of each frame is generated from the labeled template pool by the template factory, and the tracking result is used to update the labeled template pool and the template factory.

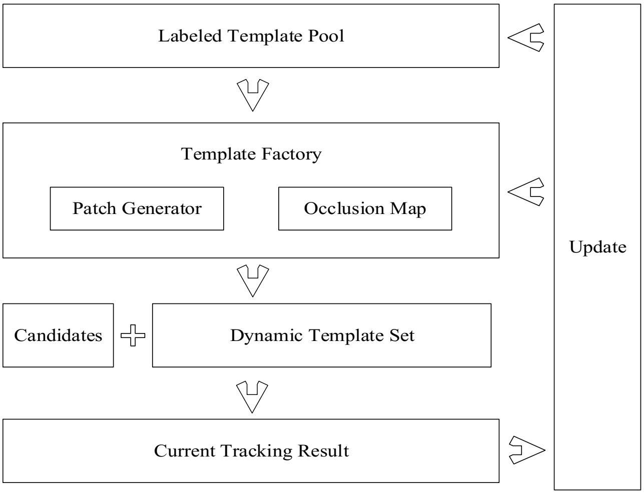


Fig. 5. Construction and update of dynamic template set

Labeled template pool, denoted by LTP. LTP contains 3 kinds of target template set with different labels: a template set without occlusion denoted by NT; a template set with small occlusion area (the occlusion area less than one-third of the area of the template), denoted by GT; a template set with large occlusion area (the occlusion area less than two-third of the area of the template), denoted by HT. Thus, LTP = {NT, GT, HT}.

Template factory consists of two parts: Patch generator and occlusion map. Patch generator is used to generate multi-scale template patches, and occlusion map used to indicate the occlusion position of the target.

Dynamic template set changes in a new frame according to the occlusion map of current tracking result, as shown in Eq.10.

$$T = \begin{cases} \{NT\} & sum(o) = 0 \\ \{NT, GT\} & 0 < sum(o) \leq 6 \\ \{NT, GT, HT\} & sum(o) > 6 \end{cases} \tag{10}$$

If current tracking result is not occluded, then it will be labeled as NT and only update the NT set, of course the template set is T={NT} in next frame. If current tracking result is partial occluded and satisfy $sum(o) > 6$, then it will be labeled as HT and only update the HT set, the template set is $T = \{NT, GT, HT\}$ in next frame, other cases handled according to the same principle. The proposed dynamic template set can effectively deal with a long term occlusion and reduce the negative impact of inappropriate update strategy for tracking.

In addition to the above holistic based dynamic template set, in this paper we also construct a multi-scale patch based dynamic template set, once the template T is obtained, T is divided into multi-scale patches, thus we can obtain a patch based dynamic template set $T_p = \{T_{h1}, T_{h2}, T_{h3}, T_{h12}, T_{h23}, T_{w1}, T_{w2}, T_{w3}, T_{w12}, T_{w23}\}$. The multi-scale patch based method can be more flexible to handle partial occlusion, pose change, etc.

2.6 Error Term Modeling

Motivated by that the occlusion usually comes from the background in tracking scenario, in this paper we use the background information to model the error term. As shown in Figure 6, we randomly select N patches with size 2×2 in our experiments near the target object in the red box, then assign to the trivial templates with the pixel values of these N patches, the other elements of trivial templates is set to zero. This background-patch templates take full advantage of the characteristics of occlusion itself, not only reduce the computational complexity by decreasing the number of dictionaries, but also improve the tracking accuracy.

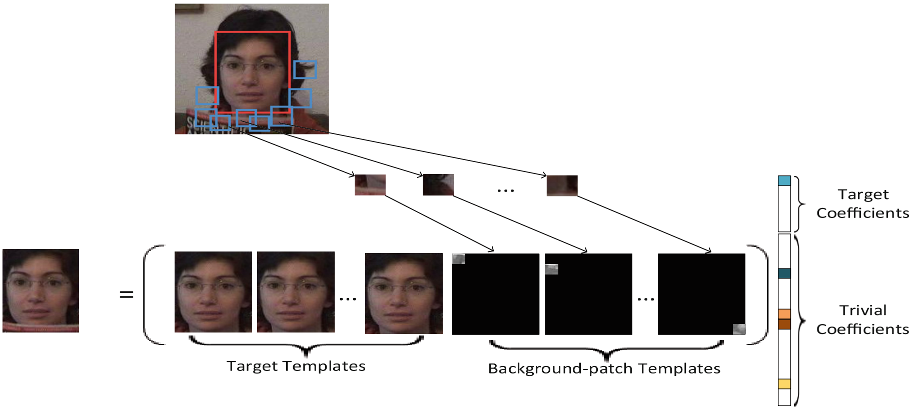


Fig. 6. Model error term with background-patch templates

2.7 Proposed Tracking Algorithm

After manually selecting target to be tracked, NT, the template set without occlusion, is generated by random tiny disturbance within the target region, and the negative templates are drawn further away from the marked location as shown in Figure 3, upon completion of the initialization process, the proposed tracking algorithm executed from the second frame. We uses a two-step search method to locate the tracking target, the specific process of this method is as follows: Firstly, we carry a large-scale search based on holistic template in the vicinity of the target location determined in last frame and sample a candidate set Y_1, then calculate the confidence values of each candidate using the discriminative reverse sparse representation method(see section 2.2) with the structure rearrangement template set Y_1 and dynamic template set T, the candidate with largest confidence value denoted by $\mathrm{r}_{\mathrm{s1}}$and the candidates whose confidence value are top k are denoted by Y_{s1}. In second stage, we draw a candidate set $\mathrm{Y}_{\mathrm{s2}}$with a small-scale particle sampling in the vicinity of r_{s1}, thus the candidate set of second stage is $\mathrm{Y}_2 = \{\mathrm{Y}_{\mathrm{s1}}, \mathrm{Y}_{\mathrm{s2}}\}$ (those candidates are raw images without structure rearrangement), then divide the candidate set Y_2 and dynamic template set T into multi-scale patches, now we can construct ten traditional sparse representation problems(different from the reverse sparse representation, in second stage we use the dynamic template set T to linear represent the candidates Y_2) as shown in Eq.11.

$$\begin{cases} Y_{h1} = T_{h1} * A_{h1} + e \\ Y_{h2} = T_{h2} * A_{h2} + e \\ \quad\cdots \\ Y_{w23} = T_{w23} * A_{w23} + e \end{cases} \tag{11}$$

Where Y_{h1} and T_{h1} are composed of the h1patches of Y_2 and dynamic template set T, A_{h1} is the coefficient vector, e is the error term, other formulas are constructed in the same way. Then we can calculate the reconstruction error for each patch of each candidate,

$$\xi_i = \|y_i - T_i * A_i\|_2^2 \tag{12}$$

Where $\mathrm{i} = \{h1, h2, h3, h12, h23, w1, w2, w3, w12, w23\}$, then we can obtain the confidence value for each candidate in Eq.13,

$$conf_j \propto \sum_i e^{-\xi_i/w} \tag{13}$$

Where w is the weight parameter for reconstruction error, it is a predefined constant. We choose the candidate with largest confidence value as the tracking result in current frame.

2.8 Update Scheme

After locating the target position, the labeled template pool is updated firstly, we calculate the occlusion map of current tracking result, then determine its type by Eq.14.

$$\text{type(result)} = \begin{cases} NT & sum(o) = 0 \\ GT & 0 < sum(o) \leq 6 \\ HT & sum(o) > 6 \end{cases} \tag{14}$$

For example, if current tracking result belongs to NT set, we only update NT set by replacing the template with the smallest weight, the weight of template equals the corresponding confidence value. Similarly, GT set and HT set will be filled before updated. After updating the labeled template pool, we construct the dynamic template set T for next frame as shown in Eq.10, and the details of the update process will be found in section 2.5.

3 Experiments

The proposed algorithm is implemented in MATLAB and runs at 0.8 frames per second on an Intel 3.2 GHz i5-4570 Core PC with 4GB memory. In order to better evaluate the performance of our tracker, we conduct experiments on eleven challenging image sequences, these sequences focus cover the most challenging situations: heavy occlusion, illumination variation, fast motion, background clutter, in-plane and out-of-plane rotations and scale variation (See Figure 7). All the sequences are available online for public download. In this paper we compare with six state-of-the-art tracking methods including IVT[23]、FCT[29]、STC[27]、L1APG[30]、SCM[33] and ASLA[28]. For fair evaluation, all trackers run with the same initial positions of the targets. In our experiments, the target image patch is normalized to 32×32 pixels, the threshold thr in Eq.9 is fixed to be 9.5, the weight parameter w in Eq.12 is fixed to be 0.04, the labeled template pool is updated in every 3 frames, and all the parameters are fixed in all the experiments.

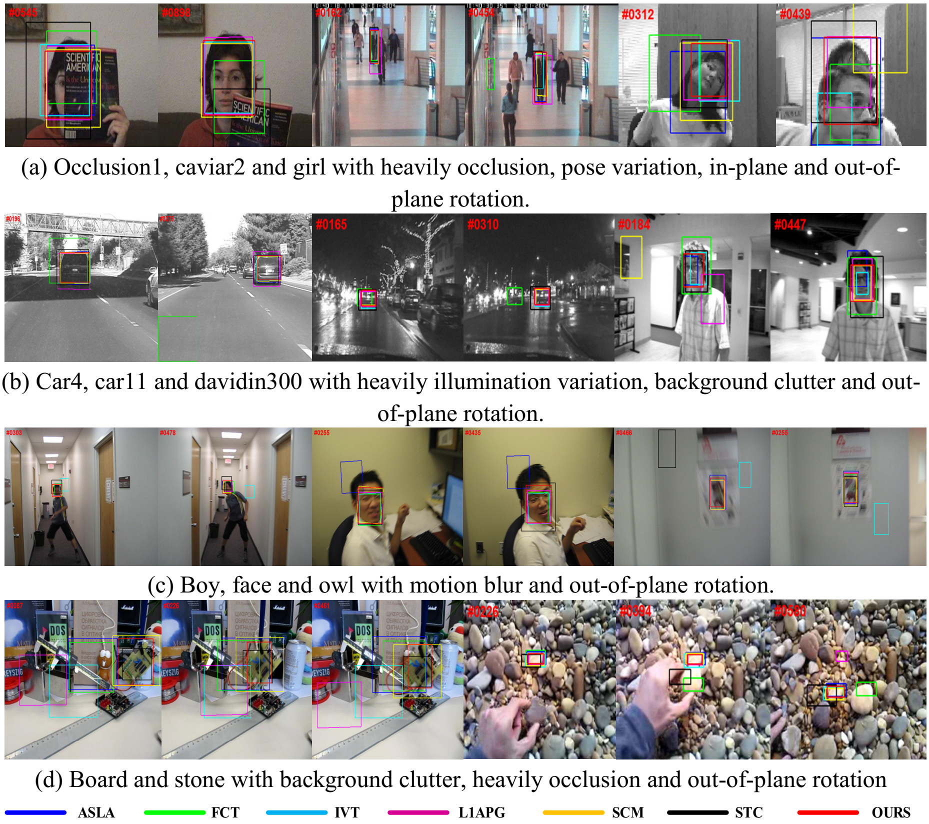
3.1 Qualitative Evaluation

Heavy Occlusion: We test three datasets occlusion1, caviar2 and girl, there are heavy occlusion and in-plane and out-of-plane rotations in these datasets shown in Figure 7(a). The proposed algorithm achieves better tracking performance judging from the experiments mainly because of the advantages of template structure rearrangement and the dynamic template mechanism, another importance reason is the two-stage search mechanism. The STC, IVT and FCT methods undergo some degree of drift, because their update mechanisms do not consider how to deal with partial occlusion. In contrast, L1APG、SCM and ASLA achieve better performance because of their update schemes deal with occluding patches.

Illumination Variation: We test three datasets car4, car11 and davidin300 with large illumination variation as shown in Figure 7(b), in addition, there are a certain scale variation and pose variation in these datasets. The FCT method does not track the target well when large illumination variation occur and has drift in all three sequences. In the davidin300 sequence, L1APG method tracks a wrong target which can be attributed to the fact that its trivial template mechanism mistakenly target pose and facial expression variation as occlusion. As a whole, the proposed tracker, SCM and ALSA perform well because they all use the local patches strategy based on sparse representation.

Fast Motion: We test three datasets boy, Owl and face with motion blur caused by target object fast motion as shown in Figure 7(c). The results show that, most tracking algorithms fail to follow the target right even lose target object. Some algorithm can continue to track the target object due to the fast moving target is in its place, but may be completely lose the target object if the fast moving target is not in the same place. Compared to other tracking algorithm, the proposed algorithm achieves the best tracking results, because the two-step search mechanism is applied.

Background Clutter: We test two datasets board and stone with complex background as shown in Figure 7(d), there are multiple objects in the background including some region which is similar to the target in terms of appearance. In addition, there are serious out-of-plane rotation and heavily occlusion. In the board sequence, the L1APG and IVT tracker almost both lose the target in all the frames, and the SCM, ALSA, FCT, STC tracker all drift to the background when the target object undergoes serious out-of-plane rotations. In contrast, our tracker performs well throughout this long sequence. In the stone sequence, the L1APG, FCT and STC all lose the target, but our tracker, SCM, ASLA, IVT perform well.



(a) Occlusion1, caviar2 and girl with heavily occlusion, pose variation, in-plane and out-of-plane rotation.

(b) Car4, car11 and davidin300 with heavily illumination variation, background clutter and out-of-plane rotation.

(c) Boy, face and owl with motion blur and out-of-plane rotation.

(d) Board and stone with background clutter, heavily occlusion and out-of-plane rotation

Fig. 7. Sample tracking results on eleven challenging sequences.

3.2 Quantitative Evaluation

We evaluate the above-mentioned algorithms using the center location error and overlap ratio, center location error is the center distance (in pixels) of the tracking results and the manually labeled location, the smaller center location error represents the better performance; and the overlap ratio is computed by intersection over union based on the tracking result R_T and the ground truth R_G, i.e., $\frac{R_T \cap R_G}{R_T \cup R_G}$, the larger overlap ratio represents the better performance.

Table 1. Comparison results in terms of average center errors (in pixels). The best three results are shown in red, Blue and Green fonts.

sequences	ASLA	FCT	IVT	L1APG	SCM	STC	Our
Board	84.1666	93.6998	164.9847	216.7564	31.7272	21.3653	11.7404
Boy	2.7294	8.4897	42.5409	6.0128	2.6866	10.7276	2.2062
car4	3.4738	178.9681	3.7852	13.2720	3.9399	14.7730	1.9223
car11	2.0578	26.9202	2.9854	2.4786	1.9946	3.5326	1.4043
caviar2	1.5334	61.1609	4.5668	12.3992	2.2586	6.5082	2.2100
davidin300	18.5372	10.1545	3.6339	24.0785	22.3952	8.5710	3.2389
Face	98.2796	30.6608	29.6935	28.9030	141.8946	28.3821	4.0860
Girl	16.3122	34.5508	30.4518	11.6667	29.6594	14.6307	10.2709
occlusion1	7.5252	38.3269	9.8410	7.7760	4.0931	34.2394	3.0815
Owl	27.7134	26.8186	99.2821	27.2591	29.8939	197.1934	1.7551
Stone	3.7389	27.6678	11.0253	7.6665	2.6097	19.8194	2.2837
average	24.1880	48.8562	36.6173	32.5699	24.8321	32.7039	4.0181

Table 2. Comparison results in terms of average overlap rates (in pixels). the best three results are shown in red, blue and green fonts.

sequences	ASLA	FCT	IVT	L1APG	SCM	STC	Our
Board	0.4007	0.3227	0.1462	0.0729	0.7048	0.3828	0.5601
Boy	0.7914	0.5764	0.3402	0.7215	0.7954	0.5389	0.8238
car4	0.9011	0.2607	0.9136	0.6871	0.8992	0.6873	0.9265
car11	0.8163	0.3762	0.7542	0.7930	0.7961	0.6341	0.8422
caviar2	0.7706	0.3109	0.6054	0.5991	0.6731	0.6681	0.7707
davidin300	0.4758	0.4733	0.6324	0.4488	0.5915	0.5605	0.7440
Face	0.2033	0.5048	0.5109	0.5374	0.3280	0.5235	0.9216
Girl	0.6443	0.5024	0.5888	0.7186	0.5763	0.5468	0.7016
occlusion1	0.8590	0.5172	0.8192	0.8461	0.9026	0.5012	0.9361
Owl	0.4959	0.4897	0.1976	0.4969	0.4693	0.0998	0.9417
Stone	0.5289	0.3338	0.5206	0.6309	0.6118	0.3446	0.6401
average	0.6261	0.4244	0.5481	0.5957	0.6680	0.4989	0.8008

Table 1 and Table 2 show the average center error and overlapping ratio where the red, blue and green fonts represent the top three tracking results. The sequences caviar2, girl, occlusion1 and stone have serious occlusion, even totally occluded. The average center errors of FCT, IVT and STC are larger in table 1, meanwhile, the average overlap rates in table 2 is smaller, which indicates these trackers lost their target, however L1APG, SCM and ASLA contain partial occlusion handling mechanism and the proposed tracker using a template structure rearrangement method and dynamic template mechanism to deal with occlusion. The sequences car4, car11 and davidin300 undergo a large illumination variation, the FCT method is not perform well, and the sparse based methods have more robustness to illumination. The average center errors is large and the average overlap rate is small for most trackers in sequences boy, Owl and face with motion blur, which indicates they almost lose target, however the proposed method introduce a two-step search mechanism to resist drifting. In the sequences board and stone with complex background the proposed method has the smallest average center errors and a larger average overlap rate, which indicates the proposed appearance model is robust to complex background. The last row in table 1 and table 2 represents the comprehensive performance of each algorithm in all sequences. Overall, the proposed algorithm is better than the other six kinds of popular algorithms.

4 Conclusion

In this paper, we propose a robust object tracking algorithm via structure rearrangement and multi-scale block appearance model. The structure rearrangement method in this algorithm rearranges the spatial structure of template without loss of target information and is ability to fragment the partial occlusion appeared anywhere, plus the multi-scale patches method the proposed tracker is robust to target appearance changes caused by the light, posture changes and heavily occlusion and so on. Furthermore, the use of labeled template pool and dynamic template set can not only effectively deal with the long-term occlusion and permanent deformation, but also reduce the negative impact of template update strategy for tracking performance. We also design a two-step search method to trim tracking results. Finally, both qualitative and quantitative evaluations on eleven challenging image sequences demonstrate that the proposed tracking algorithm performs favorably against the other six kinds of popular algorithms.

Acknowledgement. This work was supported by the China National Natural Science Funds (61302156) and the University Natural Science Research Project of Jiangsu province (13KJB510021).

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38**(4) (2006)
2. Cannons, K.: A review of visual tracking, York University, Tech. Rep. (2008)
3. Trigueiros, P., Ribeiro, F., Reis, L.P.: Generic system for human-computer gesture interaction. In: Proceedings of the IEEE Conference on ICARSC (2014)

4. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. The Visual Computer **29**(10), 983–1009 (2013)
5. Collins, R.T.: Mean-shift blob tracking through scale space. In: Proceedings of the IEEE Conference on CVPR (2009)
6. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10), 1631–1643 (2005)
7. Yang, M., Yuan, J.: Spatial selection for attentional visual tracking. In: Proceedings of the IEEE Conference on CVPR (2007)
8. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Proceedings of the IEEE Conference on CVPR (2010)
9. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: Proceedings of the ICCV (2011)
10. Li, H., Shen, C.: Real-time visual tracking using compressive sensing. In: Proceedings of the IEEE Conference on CVPR (2011)
11. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: Proceedings of the IEEE Conference on CVPR (2012)
12. Oron, S., Bar-Hillel, A.: Locally orderless tracking. In: Proceedings of the IEEE Conference on CVPR (2012)
13. Zhang, T.: Robust visual tracking via multi-task sparse learning. In: Proceedings of the IEEE Conference on CVPR (2012)
14. Avidan, S.: Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(2), 261 (2007)
15. Babenko, B.: Visual tracking with online multiple instance learning. In: Proceedings of the IEEE Conference on CVPR (2009)
16. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings of the IEEE Conference on CVPR (2006)
17. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Proceedings of the ECCV (2008)
18. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: Bootstrapping binary classifiers by structural constraints. In: Proceedings of the IEEE Conference on CVPR (2010)
19. Wang, S., Lu, H., Yang, F., Yang, M.-H.: Superpixel tracking. In: Proceedings of the on ICCV (2011)
20. Wen, L., Cai, Z.: Robust online learned spatio-temporal context model for visual tracking. IEEE Transactions on Image Processing **23**(2), 785–796 (2014)
21. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5), 564–575 (2003)
22. Matthews, I.: The template update problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 810–815 (2004)
23. Ross, D., Lim, J., Lin, R., Yang, M.-H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77**(1), 125–141 (2008)
24. Bai, S., Liu, R., Su, Z.: Incremental robust local dictionary learning for visual tracking. In: Proceedings of the ICME (2014)
25. Zhang, J., Cai, W., Tian, Y., Yang, Y.: Visual tracking via sparse representation based linear subspace model. In: Proceedings of the IEEE Conference on Computer and Information Technology (2009)
26. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: Proceedings of the IEEE Conference on CVPR (2011)

27. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 127–141. Springer, Heidelberg (2014)
28. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of the IEEE Conference on CVPR (2012)
29. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
30. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust L1 tracker using accelerated proximal gradient approach. In: Proceedings of the IEEE Conference on CVPR (2012)
31. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 2259–2272 (2011)
32. Mei, X., Ling, H.: Robust visual tracking using L1 minimization. In: Proceedings of the ICCV (2009)
33. Zhong, W., Lu, H., Yang, M.-H.: Robust object tracking via sparse collaborative appearance model. IEEE Transaction on Image Processing **23**(5), 2356–2368 (2014)
34. Bai, T., Li, Y., Tang, Y.: Structured sparse representation appearance model for robust visual tracking. In: Proceedings of the IEEE Conference on Robotics and Automation (2011)