Patch-based Object Tracking via Locality-constrained Linear Coding

Kunqi Gu[1], Mingna Liu[3,4], Tao Zhou[1], Fanghui Liu[1], Xiangjian He[5], Jie Yang[1,2], Yu Qiao[1,2]

1. Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University

2. The Key Laboratory of Ministry of Education for System Control and Information Processing, China

3. Shanghai Institute of Spaceflight Control Technology

4. Infrared Detection Technology Research and Development Center, China Aerospace Science and Technology Corporation

5. School of Computing and Communications, University of Technology, Sydney, Australia

Abstract: In this paper, the Locality-constrained Linear Coding(LLC) algorithm is incorporated into the object tracking framework. Firstly, we extract local patches within a candidate and then utilize the LLC algorithm to encode these patches. Based on these codes, we exploit pyramid max pooling strategy to generate a richer feature histogram. The feature histogram which integrates holistic and part-based features can be more discriminative and representative. Besides, an occlusion handling strategy is utilized to make our tracker more robust. Finally, an efficient graph-based manifold ranking algorithm is exploited to capture the relevance between target templates and candidates. For tracking, target templates are taken as labeled nodes while target candidates are taken as unlabeled nodes, and the goal of tracking is to search for the candidate that is the most relevant to existing labeled nodes by manifold ranking algorithm. Experiments on challenging video sequences have demonstrated the superior accuracy and robustness of the proposed method in comparison to other state-of-the-art baselines.

Key Words: object tracking, LLC, pyramid max-pooling, local appearance model, manifold ranking

1 Introduction

Visual tracking has widespread research interest due to its applications in behavior analysis, activity recognition, video surveillance and human-computer interaction. Although this field has made significant progress in the past decade [18], developing an efficient and robust tracking algorithm stil-l remains a challenging problem. This is mainly attributed to factors such as partial occlusion, illumination variation, pose change, background clutter, etc.

The main tracking algorithm can be classified into two kinds: generative and discriminative methods. The generative methods [1, 4–6, 11] formulate the tracking problem as searching for the regions with the highest likelihood.Generally, a target appearance model need to be updated dynamically to adapt to the target appearance variations caused by pose changes and illumination changes. Discriminative methods [7–9] formulate tracking as a binary classification problem which aims to distinguish a tracked target from its background. It employs the information from both the target and the background.

In recent years, methods based on sparse representation have also been used in object tracking. This type of methods has been used in the ℓ_1-tracker [6] where an object is modeled by a sparse linear combination of target templates and trivial templates. The templates are dynamically updated according to the similarity between the tracking result and the template set. However, occlusion is still one of the most challenging problems in these trackers. Jia et al. [5] develops a simple yet robust tracking method based on a structurally local sparse appearance model. This representation exploits both the partial information and spatial information of a target based on a novel alignment-pooling method, so that it helps not only locate the target more accurately but also handle occlusion. Zhong et al. [10] proposes a robust appearance model that exploits both holistic templates and local representations. Furthermore, the updated scheme considers both the latest observations and the original templates, thereby enabling that the tracker can deal with appearance changes effectively and alleviate a drift problem. The above trackers have obtained promising performance compared with many existing trackers. However, ℓ_1-minimization used in these trackers is very time consuming.

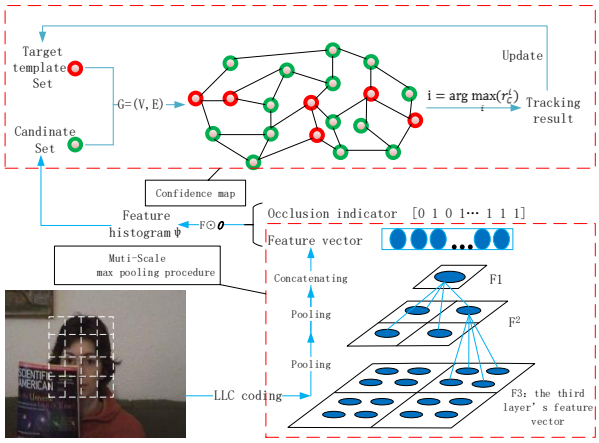

Fig. 1: Basic flow of our tracking method. In the max pooling procedure, we illustrate the strategy to get the feature histogram. In the confidence map module, a graph is established based on the feature histogram of labeled nodes(target templates) and unlabeled nodes(candidates). Then, we compute the ranking scores of all candidates. The candidate with the largest score is deemed as the tracking result of the current frame and we utilize it to update the target template set.

Wang et al. [12] presents a simple but effective coding scheme called Locality-constrained Linear Coding (LLC) and proposes a fast approximated LLC method. The method

Corresponding author: Yu Qiao, qiaoyu@sjtu.edu.cn. This research is partly supported by USCAST2013-07, NSFC (No: 61273258) and 863 Plan China (No. 2015AA042308).

Table 1: Comparison of the selected trackers. The tracker is based on local or holistical template, and based on sparse coding or LLC coding. H: holistical, L: local, SC: sparse coding, LLC: LLC coding. The trackers are including: ASLA [5], SGM [10], SPT [3]

Tracker	ASLA	SGM	SPT	Ours
Template	L	$H\&L$	L	H&L
Representation	SC	SC	LLC	LLC

first performing a K-nearest-neighbor search and then solving a constrained least square fitting problem. Compared with the sparse coding strategy, LLC can not only ensure an analytical solution but also that the similar patches will have similar codes. Liu et al. [3], for the first time, incorporates LLC into a mean-shift based tracking framework, and hence achieves better performance on comprehensive experiments.

Motivated by the challenges of the work mentioned above, we propose a novel tracking method, the part-based LLC tracker. We firstly construct a feature vector based on the coefficients of the LLC solution, and then utilize a pyramid representation with max pooling to get the final feature histogram. Finally, an efficient manifold ranking algorithm is adopted to choose the best target candidate in the current frame. The contributions of this paper are as follows. (1) A novel part-based object tracking framework based on LLC is proposed. (2) A multi-scale representation strategy through pyramid max pooling strategy is utilized. Compared with the holistic model, this method can handle partial occlusion and other challenging factors. (3)Our tracker can be adaptively updated so that it keeps the representative templates of each part throughout the tracking process. The comparison with related trackers is shown in table 1.

2 Tracking Algorithm

In this section, we present the proposed algorithm (Fig. 1) in details. Sec 2.1 shows how the appearance model is created. The strategy to get the final feature histogram is discussed in Sec 2.2. Sec 2.3 shows the way to choose the best candidate. In the end, the update scheme of our appearance model is then presented in Sec2.4.

2.1 Local target appearance model

Given the target location at the first frame, we slide the target window pixel by pixel to get n templates $T = [T_1, T_2, \cdots, T_n]$. For simplicity, we use the gray-value features to represent local information. Each template is divided into M patches and each patch is converted to a ℓ_2 normalized vector b_i, These patches are used to form a dictionary D to encode the local patches inside possible candidate regions, i.e.

$$B = [b_1, b_2, \cdots, b_{n\times M}] \in R^{d\times(n\times M)}, \tag{1}$$

where d denotes the dimension of each vectorized patch, n is the number of target templates and M is the number of local patches sampled within a target region. Each column in B is obtained by normalization on a vectorized local image patch exacted from a template in T. Each local patch represents one fixed part of the target object, so the local patches together can represent a complete structure of the target. For a target candidate, we extract local patches within it and turn them into vectors in the same way. These vectors construct the columns of the matrix represented by $Y = [y_1, y_2, \cdots, y_M] \in R^{d\times M}$. These patches can be represented by only a few basis elements of the dictionary. Here We utilize the Locality-constrained Linear Coding (LLC) method [12] to solve the representation problem. Because compared with the sparse coding strategy, LLC not only ensure an analytical solution but also the locality. And locality is more important than sparsity in our appearance model. The coefficient vector β is computed by

$$\begin{aligned} \underset{\beta}{\operatorname{argmin}} \quad & \sum_{i=1}^{N} \|y_i - B\beta_i\|^2 + \lambda \|d_i \odot \beta_i\|^2 \\ \text{s.t.} \quad & \mathbf{1}^{\mathrm{T}}\beta_i = 1, \forall i \end{aligned} \tag{2}$$

where $\odot$ denotes the element-wise multiplication, and $d_i \in R^{nM}$ is the locality adaptor that gives different freedom for each basis vector proportional to its similarity to the input vector y_i. Specifically,

$$d_i = exp(\frac{dist(y_i, B)}{\sigma}) \tag{3}$$

where $dist(y_i, B) = [dist(y_i, b_1), \cdots, dist(y_i, b_{n\times M})]^{\mathrm{T}}$ with $dist(y_i, b_j)$ being the Euclidean distance between y_i and b_j, and σ is used for adjusting the weight decay speed for the locality adaptor. The constraint $\mathbf{1}^{\mathrm{T}}\beta_i = 1$ follows the shift-invariant requirements of the LLC code.

2.2 Spatial pyramid representation

After obtaining sparse codes of the local patches, we build the feature vector layer by layer to model the appearance of a target. An image at layer l contains 4^{l-1} patches, We group four adjacent patches as a cell. A spatial max pooling algorithm (as Fig. 1 shows) is applied to process the sparse codes within a cell. Then we can get a feature vector the same dimension with the sparse code. We treat the feature vectors we get as the inputs of next layer and process them in the same way. The feature vector is the component-wise maxima of the sparse codes in each cell Ω and it is denoted by

$$F_\Omega = \max_{i\in\Omega}[\beta_{i1}, \beta_{i2}, \cdots, \beta_{i(n\times M)}] \tag{4}$$

where i ranges over all patches in the cell, and β_{ik} represents the k-th channel of the corresponding sparse code β_i. Fig. 2 illustrates the max-pooling strategy. Feature vector F^l at

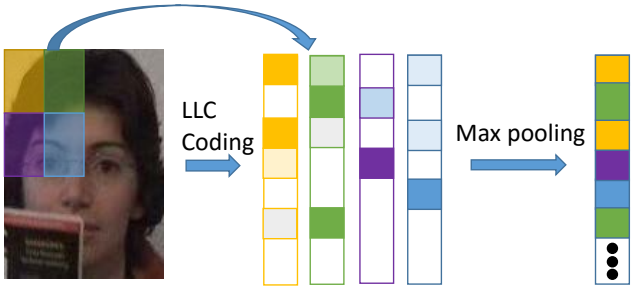

Fig. 2: Illustration of the max-pooling strategy.

layer l is the concatenation of aggregated sparse codes of the individual cells, denoted by

$$F^l = [F^l_{\Omega_1}, \cdots, F^l_{\Omega_s}, \cdots, F^l_{\Omega_{l\times l}}]. \tag{5}$$

Next, we concatenate pooling features over different layers and form the spatial pyramid representation.

$$F = [F^1, \cdots, F^l, \cdots, F^L], \tag{6}$$

where L is the number of layers in our multi-scale representation. Features of lower scales correspond to a local spatial configuration of the target, and are effective in dealing with partial occlusions. Features of higher scales represent global properties of the target, and they are robust to displacement of individual fragments that belong to the target during tracking. We utilize pyramid representation with max pooling, our model generates part-based decomposition of the target and this results in improved tracking performance. Our method integrates holistic and part-based sparse signals, thereby generating a richer feature set that contains more structures. Also, structures from different scales and data channels will lead to more discriminative patch descriptors.

In our method, cells with large reconstruction errors are regarded to be due to occlusion and they are ignored when we compute a histogram representation of features. The histogram ψ is generated by

$$\psi = F \odot O, \tag{7}$$

where $\odot$ denotes the element-wise multiplication, and O is a vector with each element o_i being an indicator of occlusion of the corresponding patch and is obtained by

$$o_i = \begin{cases} 1 & \text{if } \varepsilon_i < \varepsilon_0 \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $\varepsilon_i = \|y_i - D\beta_i\|_2^2$ is the reconstruction error in i-th patch by comprehensively considering all positive samples and the elements corresponding to the higher layers are all set 1.

2.3 Confidence map

After we obtain the feature histograms of all target templates and candidates, we formulate the tracking problem as a ranking task. Here, we utilize an efficient graph-based manifold ranking algorithm to solve the problem as shown in [2, 13]. In this paper, we treat the feature histograms of target templates and candidates as query nodes and unlabel nodes respectively. Then, we compute the ranking scores for all candidates. The candidate with the largest score will be the tracking result in the current frame.

The manifold ranking method is described as follows: given a query node, the remaining unlabeled nodes are ranked based on their relevance to the given query. The goal is to learn a ranking function to define the relevance between unlabeled nodes and this query [2, 13]. In [2], a ranking method that exploits the intrinsic manifold structure of data for graph labelling is proposed. Given a data set $X = \{x_1, x_2, \cdots, x_l + 1, \cdots, x_n\} \in \Re^{m \times n}$, some data points are labelled queries and the rest need to be ranked according to their relevance to the queries. $W \in \Re^{n*m}$ denotes the adjacency matrix with element W_{ij} that indicates the weight of the edge between point i and j. Generally, the weight can be defined by the kernel $w_{ij} = e^{d^2(x_i, y_j)/2\sigma^2}$ if there is an edge linking x_i and y_j, otherwise $w_{ij} = 0$. The function $d(x_i, y_j)$ represents a distance metric between x_i and y_j.

Let $f : X \to \Re^n$ denotes a ranking function which assigns a ranking value r_i to each point x_i , and r can be defined as a vector $r = [r_1, r_2, \cdots, r_n]^T$. Let $y = [y_1, y_2, \cdots, y_n]^T$ denote an indication vector, in which $y_i = 1$ if x_i is a query, and $y_i = 0$ otherwise. Suppose all data points represent a graph $G = (V, E)$, where V represents vertex set, and E represents the edge set with $W = W_(ij), i, j = 1, 2, \cdots, N$. The strength of edge reflects the similarity between two vertices. To solve the optimal ranking of queries, the cost function associated with f is defined as follows:

$$O(r) = \frac{1}{2}\left(\sum_{i,j=1}^{n} \| \frac{1}{\sqrt{D_{ii}}} r_i - \frac{1}{\sqrt{D_{jj}}} r_j \|^2 + \mu \sum_{i=1}^{n} \|r_i - r_j\|^2\right) \tag{9}$$

where $\mu > 0$ is the regularization parameter and D is a diagonal matrix with the element $D_{ii} = \sum_{j=1}^{n} w_{ij}$. To minimize the cost function, we can obtain the closed form solution as:

$$r^* = (I - \alpha S)^{-1} y \tag{10}$$

where I is an identity matrix, $\alpha = \frac{1}{1+\mu}$ and $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Then, we use the iteration scheme to solve this optimal problem:

$$r(t+1) = \alpha S(r(t) + (1-\alpha)y) \tag{11}$$

where α is control parameter, which balances each points information from its neighbors and that of initial information.

2.4 Update scheme

Here, we develop an update scheme to adapt to the target's appearance changes. Firstly, the update index is denoted by ρ_t and it is computed by

$$\rho_t = \frac{r_c}{r_q}, \tag{12}$$

where r_c and r_q are the largest scores in the candidates and templates, respectively.

We set two thresholds ρ_l and ρ_h. If $\rho_t < \rho_l$, it indicates that the dictionary has been deteriorated by the update of tracking failures or occlusion. In order to recover the object from occlusion and capture a new appearance, the template histogram is updated by

$$\psi_t^i = \mu \psi_f^i + (1-\mu)\psi_c, i = 1, 2, \cdots, n, \tag{13}$$

where the new i-th template's histogram ψ_t^i is composed of the histogram ψ_f^i at the first frame and the histogram ψ_c of the current tracking result according to the weights determined by the constant μ, and n is the number of the templates forming the dictionary. If $\rho_l < \rho_t < \rho_h$, we just replace the template of the lowest score with the current tracking result.

3 Experimental Results and Discussion

3.1 Experimental setup

Setup: The proposed tracker was implemented in MATLAB with a PC with Intel(R) Core(TM) i3-4130 CPU(3.4GHz), 4GB memory. The implementation details of the proposed method and the choice of baseline is described as follows. In all experiments, 10 target templates were used for modelling the target appearance. we empirically resized each image

Fig. 3: Screenshots of some sampled tracking results. Subfigures from left to right, top to bottom: (a) Car11, (b)Caviar2, (c)Stone, (d)Occlusion2, (e)Caviar1, (f)Singer1, g)Occlusion1, (h) Car4.

Table 2: Center location error (CLE). Red fonts indicate the best performance while the blue fonts indicate the second best ones.

Sequence	L1	MIL	IVT	Frag	WMIL	LOT	FCT	L1APG	Ours
Car4	4.1	60.1	2.8	197.8	162.5	183.8	183.9	197.3	13.9
Caviar1	119.9	48.5	45.3	5.7	23.8	2.2	49.9	51.9	1.3
Car11	33.3	5.6	2.1	63.9	96.1	47.7	24.8	32.3	2.2
Stone	19.2	32.3	2.2	65.9	49.6	28.1	34.2	6.1	2.2
Occlusion1	6.5	25.8	10.3	5.6	23.5	21.3	26.8	7.5	5.2
Caviar2	3.2	70.3	8.6	5.8	59.8	3.4	87.7	32.3	2.9
Occlusion2	11.2	14.1	10.3	15.5	16.7	18.9	14.1	35.9	6.7
Singer1	48.5	15.2	8.5	22.1	16.6	156.8	14.3	4.5	8.5
Average	30.7	38.7	11.3	47.8	56.1	57.8	54.5	46.0	5.4

Table 3: Success rate (SR). Red fonts indicate the best performance while the blue fonts indicate the second best ones.

Sequence	L1	MIL	IVT	Frag	WMIL	LOT	FCT	L1APG	Ours
Car4	0.57	0.02	0.92	0.22	0.27	0.02	0.34	0.14	0.93
Caviar1	0.3	0.82	0.28	0.68	0.31	0.88	0.77	0.3	0.89
Car11	0.68	0.01	0.81	0.08	0.01	0.16	0.25	0.36	0.91
Stone	0.82	0.01	0.92	0.13	0.48	0.19	0.26	0.86	0.92
Occlusion1	0.87	0.85	0.85	0.9	0.73	0.49	0.86	0.93	1
Caviar2	0.84	0.25	0.43	0.58	0.37	0.86	0.3	0.36	0.95
Occlusion2	0.63	0.61	0.58	0.6	0.66	0.45	0.77	0.01	0.86
Singer1	0.02	0.22	0.66	0.34	0.23	0.23	0.25	0.85	0.94
Average	0.59	0.35	0.68	0.44	0.38	0.41	0.47	0.48	0.92

patch to 32×32 pixels and extracted non-overlapped 8×8 local patches. The number of layers as mentioned in Sec. 2.1 was set to 3. The threshold ε_0 in Eq. (8) was set to 0.4. The update rate μ in Eq. (13) was set to be 0.95. The threshold ρ_l and ρ_h in Sec. 2.4 were set to 0.6 and 0.8, respectively. All the parameters above were fixed for all sequences.

Methods compared: In order to evaluate the proposed method against the state-of-the-art methods, 8 baseline techniques were chosen, including: L1 tracker [6], FCT tracker [15], MIL tracker [7], IVT tracker [4], Frag tracker [1], WMIL tracker [9], LOT tracker [16], L1APG tracker [17]. The codes of all trackers were from original authors.

Table 4: Average FPS comparison of the selected trackers.

Algorithm	L1	MIL	IVT	Frag	WMIL	LOT	FCT	L1APG	Ours
Average FPS	0.32	4.03	16.4	6.7	24.4	0.23	32.9	1.73	5.68

3.2 Quantitative analysis

We perform experiments on eight publicly available standard video sequences. As the ground truth, the center position of a target in a sequence is labeled manually. This ground truth is provided in Wus work [18]. For quantitative analysis, we use average center location errors as evaluation criteria to compare the performance, and the pixel error in every frame is defined as follows.

$$CLE = \sqrt{(x' - x)^2 + (y' - y)^2} \tag{14}$$

where $(x^{'}, y^{'})$ represents the object position obtained by different tracking methods, and (x, y) is the ground truth. The second evaluated metric is the success rate [19], and the score in every frame is defined as follows.

$$score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)} \tag{15}$$

where R_T is the tracking bounding box and R_G is the ground truth bounding box. If the $score$ is larger than 0.5 in one

frame, the tracking result is considered as a success. The third evaluated metric is the speed of the tracking algorithm, which is usually defined as FPS(frames per second).

Fig. 3 shows screenshots of some sampled tracking results from different trackers. Table. 2 reports the center location error (CLE), where a smaller value of CLE indicates a more accurate tracking result. Table. 3 reports the success rates, where larger scores indicate more accurate results. Table. 4 reports the speed of the selected trackers, the larger the value, the faster the tracker.

3.3 Qualitative analysis

Partial Occlusion: The sequences Occlusion1, Occlusion2, Caviar1 and Caviar2 are chosen to demonstrate the effect of partial occlusion, scale changes, deformation and rotation of a target. In Fig. 3(a), it can be seen that the proposed method can track the target accurately in all displayed frames. The APGL1, FCT, IVT and L1 trackers completely fail to track in frames #149, #189 and #261. The MIL and WMIL trackers suffer some drifts in these frames. In our patch-based module, we utilize a multi-scale max pooling and occlusion handling strategy to get a feature histogram. When there is patch occluded, the other patches will be more representative and discriminative. Therefore, our tracker can be robust to partial occlusion.

Background clutter: One common failure mode in visual tracking is when the target appearance matches with that of the background.Fig. 3(b) and Fig. 3(d) demonstrate the tracking results in the Car11 and Stone sequences with background clutter. Comparatively, our method and IVT exhibit better discriminative ability and outperform other methods as shown in frames #5, #21, #137, #277 and #381. The MIL and WMIL trackers completely drift to the background in frames #137, #277 and #381. This verifies that the features selected by the MIL and WMIL trackers are less informative of distinction. Both the Frag tracker and the CT method suffer severe drifts in all displayed frames. While the Fraq tracker does not update its template online, the CT method only uses the compressive features in a Bayesian classifier and hence is sensitive to background clutter. Our tracker utilize a multi-scale max pooling strategy to get the feature, and the features from the higher scales represent holistic properties of the target, which is distinctive from the background.

Illumination Variations: Fig. 3 shows the tracking results in the Singer1 sequence where significant illumination variation, and scale and pose changes can be noticed. The proposed method performs better than all other evaluated algorithms. The L1 tracker suffers completely from drifts to the background. Other methods have managed to track the target with large tracking errors. The results under illumination variations are further validated using the Car11 (Fig. 3(b)) sequence. Overall, the proposed method could obtain much better performance, thereby verifying that multi-features can be robust in dynamic environments.

Complexity analysis: Here we'll discuss the computational complexity of our proposed algorithm comparing to IVT and l_1 tracker.Suppose the dimension of dictionary matrix is $d \times M$ In the IVT method, the computation involves matrix-vector multiplication and the computation complexity is $O(dM)$. The computation complexity of LASSO algorithm to compute the sparse coefficients for a sparse representation is $O(d^2 + dM)$. In our method, we utilize a fast approximated LLC method [12] to solve Eq. 2, and the computational complexity is $O(d + k^2)$. Besides, the manifold ranking complexity is $O(\eta md)$ (m is the positive template number, η is the concatenation coefficient of max pooling. Here m=10, η=21). Considering that $d >> k$, the complexity of total operation is approximate $O((\eta m + 1)d)$.

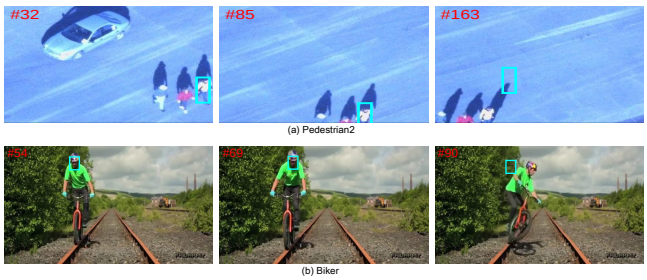

Fig. 4: Two failed tracking cases:(a) out of plane rotation and abrupt motion; (b) object of interest leaves completely out of screen and reappears.

3.4 Discussion

As shown in our experiments, our method can address these factors including abrupt motion, cluttered background, occlusion, and Illumination variation more effectively. This can be attributed to some reasons listed as follows. (1) We learn the discriminative and representative feature histogram through LLC solving and pyramid max-pooling, which help our method to distinguish a target from a cluttered background accurately. (2) The confidence map is solved by graph based manifold ranking algorithm, which makes full use of the manifold structure information of the target. That helps to handle the target appearance change and background clutter effectively.

However, our proposed method may fail when an object of interest leaves completely out of screen and reappears or an out-of-plane rotation in the current sequences(see Fig.4), due to the reason that there is no target redetection strategy in our algorithm. Fig.4(a) shows the tracked object completely out of the screen and reappears after some frames. Our tracker can not track the object in a long time when an object of interest leaves completely out of screen, so there are big errors to update the subspace appearance model. Fig.4(b) shows an out-of-plane rotation and an abrupt motion after #69. Our method drifts away the ground truth because the appearance model can not match well between the object model and the candidates, and it cannot distinguish the object from the changed background when abrupt motion.

Overall, our method performs favorably against the state-of-the-art tracking methods in the challenging sequences.

4 Conclusion

This paper has proposed an novel tracking method based on the LLC algorithm and pyramid max-pooling strategy. A multi-scale representation is adopted to form a robust histogram which considers the spatial information among local patches with an occlusion handling module, and it enables our tracker to be more discriminative. In addition, an efficient graph-based manifold ranking algorithm has been exploited to obtain the best candidate. Quantitative and qualitative experiments have demonstrated the robustness of our tracker.

References

[1] Adam, Amit and Rivlin, Ehud and Shimshoni, Ilan, Robust fragments-based tracking using the integral histogram, *Computer Vision and Pattern Recognition*, 2006: 798–805.

[2] Zhou, Tao and He, Xiangjian and Xie, Kai and Fu, Keren and Zhang, Robust visual tracking via efficient manifold ranking with low-dimensional compressive features *Pattern Recognition*, 2015: 2459–2473.

[3] Liu, Baiyang and Huang, Junzhou and Yang, Lin and Kulikowsk, Casimir, Robust tracking using local sparse appearance model and k-selection, *Computer Vision and Pattern Recognition (CVPR)*, 2011:1313–1320.

[4] Ross, David A and Lim, Jongwoo and Lin, Ruei-Sung and Yang, Ming-Hsuan, in *International Journal of Computer Vision*, 2008: 125–141.

[5] X. Jia, H. Lu, and M.-H. Yang, Visual Tracking via Adaptive Structural Local Sparse Appearance Model, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference*, 2012: 1822–1829.

[6] Mei, Xue and Ling, Haibin, Robust visual tracking using 1 minimization, *Computer Vision, 2009 IEEE 12th International Conference*, 2009:1436–1443.

[7] Babenko, Boris and Yang, Ming-Hsuan and Belongie, Serge, Visual tracking with online multiple instance learning, *Computer Vision and Pattern Recognition*, 2009: 983–990.

[8] Zhang K, Zhang L, Yang M H, Real-time compressive tracking, *Computer VisionCECCV 2012*, Springer Berlin Heidelberg, 2012: 864-877.

[9] Zhang K, Song H, Real-time visual tracking via online weighted multiple instance learning, *Pattern Recognition*, 2013, 46(1): 397-411.

[10] Zhong W, Lu H, Yang M H, Robust object tracking via sparsity-based collaborative model, *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012: 1838-1845.

[11] Zhang T, Ghanem B, Liu S, et al, Robust visual tracking via multi-task sparse learning, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012: 2042-2049.

[12] Wang J, Yang J, Yu K, et al. Locality-constrained linear coding for image classification, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010: 3360-3367.

[13] Xu B, Bu J, Chen C, et al. Efficient manifold ranking for image retrieval, *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011: 525-534.

[14] Kang Z, Wong E K. Learning multi-scale sparse representation for visual tracking, *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014: 4897-4901.

[15] Zhang K, Zhang L, Yang M H. Fast compressive tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014, 36(10): 2002-2015.

[16] Oron S, Bar-Hillel A, Levi D, et al. Locally orderless tracking, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012: 1940-1947.

[17] Bao C, Wu Y, Ling H, et al. Real time robust l1 tracker using accelerated proximal gradient approach, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012: 1830-1837.

[18] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark, *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013: 2411-2418.

[19] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010, 88(2): 303-338.