Online Visual Tracking via Coupled Object-Context Dictionary

Mingquan Ye[1][2]
mingquan.ye@vipl.ict.ac.cn
Hong Chang[1]
hong.chang@vipl.ict.ac.cn
Xilin Chen[1]
xilin.chen@vipl.ict.ac.cn

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China

[2] University of Chinese Academy of Sciences, Beijing, 100049, China

Abstract

Sparse representation and context information have been extensively applied in visual tracking. In this paper, we make the most of context information outside the target bounding box to construct the distinct background dictionary. The pure target dictionary is then constructed by filtering out background patches from the target bounding box. At each frame, all relevant patches are encoded by the coupled dictionaries. Based on the reconstruction errors, we can efficiently compute the confidence value of each bounding box candidate. By investigating the changes of the reconstruction errors on the coupled dictionaries, we can effectively handle occlusion. Both quantitative and qualitative results demonstrate that the proposed tracker performs favorably compared with several state-of-the-art trackers on some challenging video sequences.

1 Introduction

Visual object tracking is one of the most active and important topics in computer vision community with a wide range of applications [26], including surveillance, vehicle navigation, human computer interaction, to name just a few. Although significant progresses have been made in recent years, it is still challenging to build a robust tracker due to various factors such as occlusion, illumination changes, background clutter and variations in pose and scale.

In general, tracking algorithms can be roughly categorized into either generative or discriminative methods. Discriminative trackers formulate tracking as a binary classification problem which divides the current image into target and background. Some representative methods include *MIL tracker* [1], *compressive tracker* (CT) [27], *P-N tracker* [7], *Struck tracker* [4], *CSK tracker* [5], etc.

Generative trackers usually learn an appearance model to represent the target object and search for the best candidate based on the similarities (or dissimilarities) between candidates and target model. Motivated by the successful application of sparse representation to face recognition [24], Mei et al. [13] assume that the tracked object can be represented well by a sparse linear combination of target templates and trivial templates. Since then, many tracking approaches based on sparse coding have been proposed. Readers can refer to [29]. In this paper, we focus on generative methods.

In conventional sparse coding based tracking algorithms, the dictionary is composed of holistic target templates and background templates [20]. The target templates always contain some background parts due to the non-rigidness of the tracked object. The background templates are produced around the labeled target position with big perturbations, but some target contents may still be included. Such circumstances will certainly decrease the discriminative ability of trackers. And worse yet, when the target is occluded for a long time, some target templates will be updated with the tracking result at current frame. Hence, some basis vectors will be inevitably replaced by the false positive ones. After that, when the target reoccurs, both the real and false targets can be well expressed by the dictionary because of the sparsity restriction, which leads to tracking drift. Sparse coding on local patches have also been proposed as tracking methods [6]. In this paper, the target bounding box is divided into overlapped image patches and alignment-pooling is carried out to keep the corresponding position relationship. However, it has two drawbacks. First, the background information is not fully utilized. Second, some target patches contain background parts. Found on the above situation, we tend to construct pure background and target dictionaries.

Recently, Lu et al. [11] propose a method that detects abnormal events by learning the normal patterns. Contextual learning [8, 28] has been successfully applied in visual tracking. For online tracking, the available information about the object is quite limited, while the background information around the target can be fully utilized. Hence, we can construct the pure background dictionary, by which the true target features can be effectively learned and characterised.

Inspired by [8, 11, 28], we propose a robust tracking method based on sparse coding with elaborate target (positive) and background (negative) dictionaries. More specifically, we utilize context information outside the target bounding box to construct the distinct background dictionary. The target dictionary is then constructed by real target patches inside the bounding box, which are identified by their reconstruction errors on the background dictionary. For each video frame, all relevant patches are encoded by the coupled background and target dictionaries respectively. Based on the reconstruction errors, we can compute the confidence map and handle possible occlusions. In summary, the contributions of our method are as follows:

(1) We propose an effective and efficient method to construct pure target and background dictionaries, which are more discriminative than traditional dictionaries.

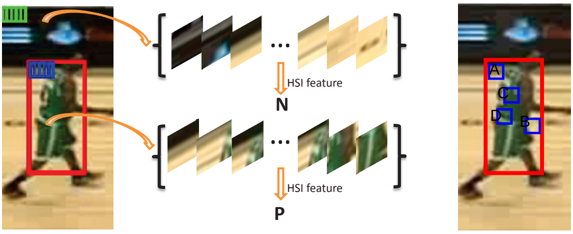
(2) The patches in each bounding box candidate are encoded by the coupled dictionaries. Therefore our method can obtain reliable confidence map and handle occlusions.

(3) Although under the particle filter tracking framework, our method can efficiently compute the scores of bounding box candidates based on the estimated confidence map.

1.1 Other related work

Many algorithms have been proposed for visual tracking over the past decades. For a comprehensive review and comparison, we refer readers to several survey papers [10, 16, 26]. Here we briefly introduce some typical methods.

Transfer learning and ensemble learning have been applied on visual tracking. In [22], an overcomplete dictionary is learned to represent visual prior by a collection of real-world



(a) (b)

Figure 1: Illustration of constructing the coupled dictionaries. The red rectangles in (a) and (b) represent the target bounding boxes. The green squares in (a), which are generated by sliding windows outside the target bounding box, correspond to basis patches involved in the background dictionary **N**. The blue squares in (a) are generated inside the target bounding box in a similar way, which constitute the noisy target dictionary **P**. Although the patches *A*, *B*, *C* and *D* in (b) are all inside the target bounding box, *A* and *B* have small reconstruction errors w.r.t. dictionary **N**, while *C* and *D* have big reconstruction errors.

images. Then the learned prior is transferred to online tracking by sparse coding. With this representation, a linear classifier is trained online to distinguish the target from background. To make full use of the advantages of different trackers, Wang et al. [19] fuse *ASLA* [6], *Struck* [4], *DLT* [18], *CSK* [5] and *LSST* [17] to construct an ensemble. Then a factorial hidden Markov model (FHMM) for ensemble-based tracking is proposed by learning jointly the unknown trajectory of the target and the reliability of each tracker in the ensemble.

Recently, correlation filter and circulant structure have demonstrated their advantages in visual tracking. [2] proposes an adaptive correlation filter by minimizing the output sum of squared errors. Motivated by [2], Danelljan et al. [3] propose to learn separate filters for translation and scale estimation for robust tracking. Besides training two regression models based on correlation filters for translation and scale estimation of objects, the most recent work [12] trains an online random fern classifier to re-detect targets to realize long-term tracking. With circulant structure, [5] increases the tracking speed on the benchmark [25], by computing and dense sampling in Fourier domain.

Deep learning has shown its initial success in tracking. [18] puts more emphasis on the feature learning problem by training a stacked denoising autoencoder to learn generic image features. The offline CNN model in [21] is trained with ImageNet 2014 detection dataset[1]. In online tracking, the CNN is finetuned to adapt to the target appearance. While [9] automatically relearns the useful feature representation during the tracking process without offline training.

2 Our Approach

2.1 Constructing Pure Coupled Dictionaries

The surrounding scene of the target provides useful context information for target localization [28]. Normally, the outside of the target bounding box is pure background, while the inside includes both target and background regions [8]. Hence, we first construct the pure background dictionary with the patches in the outside context region. As shown in Figure 1,

[1] http://image-net.org/challenges/LSVRC/2014/

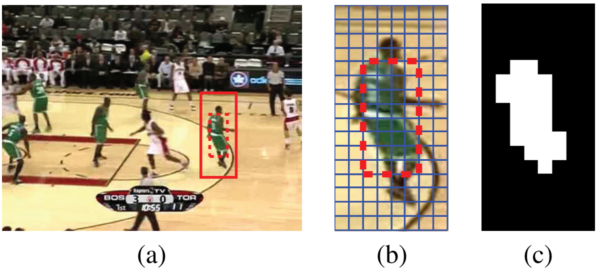


Figure 2: Illustration of computing the confidence map. (a) The t-th frame image. (b) The context window Ω is divided into nonoverlapping patches, and the red dashed line represents the target bounding box at frame $t-1$. (c) The confidence map of Ω at frame t.

the red rectangle represents the target bounding box, and the whole cropped image represents the context region Ω whose size is α ($\alpha > 1$) times the target size. The green squares, which are generated by sliding windows with padding size p in the context region, correspond to basis patches involved in the background dictionary. We denote the background dictionary as $\mathbf{N} \in \mathbb{R}^{m \times n}$, where m is the feature dimensionality of the basis patches and n is the cardinality of the dictionary.

The blue squares, as shown in Figure 1 (a), are generated inside the target bounding box in a similar way. These patches constitute a noisy target dictionary $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_l] \in \mathbb{R}^{m \times l}$, where $\mathbf{p}_i$ denotes the i-th basis patch as well as its feature representation. Then, we pick out real target patches from $\mathbf{P}$ based on their coding quality with respect to dictionary $\mathbf{N}$. Formally, we compute the sparse coding of $\mathbf{P}$ as

$$\widehat{\mathbf{C}} = \arg\min_{\mathbf{C}} \|\mathbf{P} - \mathbf{N}\mathbf{C}\|_2^2 + \lambda \sum_{i=1}^{l} \|\mathbf{c}_i\|_1 \,, \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{n \times l}$ is the coefficient matrix with the i-th column $\mathbf{c}_i$ being the sparse coding of $\mathbf{p}_i$, $\|\cdot\|_1$, $\|\cdot\|_2$ denote ℓ_1 and ℓ_2 norm, respectively. Then the set of reconstruction errors is expressed as $\mathcal{R} = \{\|\mathbf{p}_i - \mathbf{N}\widehat{\mathbf{c}}_i\|_2^2\}_{i=1}^{l}$. Obviously, the larger the reconstruction error, the higher the probability of this basis belonging to the pure target dictionary. As illustrated by Figure 1 (b), the patches A and B belonging to background have small reconstruction errors. In contrast, C and D are parts of the target object, thus have big reconstruction errors. Therefore, we choose l' ($l' = \lfloor \beta \times l \rfloor$ and $\beta \in (0,1)$) basis vectors with top l' highest reconstruction errors to construct the pure target dictionary $\mathbf{P}'$, which is expressed as $\mathbf{P}' = [\mathbf{p}_{i_1}, \ldots, \mathbf{p}_{i_{l'}}]$, where $i_1, \ldots, i_{l'}$ are the indexes of the top l' values in $\mathcal{R}$.

2.2 Computing Confidence Map

The target patches should have big and small reconstruction errors when encoded by $\mathbf{N}$ and $\mathbf{P}'$, respectively, while the background patches have the reversed situations. Based on this fact, we can effectively discriminate the target object from the backgrounds. At online tracking in t-th frame, we construct the confidence map of the context region Ω which is centered at the target position of frame $t-1$. Specifically, we normalize and divide Ω into q nonoverlapping patches of same size as those in the dictionaries (as shown in Figure 2 (b)), denoted as $\mathbf{O} \in \mathbb{R}^{m \times q}$. Then the q patches are encoded by $\mathbf{N}$ and $\mathbf{P}'$ as follows:

$$\widehat{\mathbf{C}}_1 = \arg\min_{\mathbf{C}_1} \|\mathbf{O} - \mathbf{N}\mathbf{C}_1\|_2^2 + \lambda \sum_{i=1}^{q} \|\mathbf{c}_{1i}\|_1 , \tag{2}$$

$$\widehat{\mathbf{C}}_2 = \arg\min_{\mathbf{C}_2} \left\|\mathbf{O} - \mathbf{P}'\mathbf{C}_2\right\|_2^2 + \lambda \sum_{i=1}^{q} \|\mathbf{c}_{2i}\|_1 , \tag{3}$$

where $\mathbf{C}_1 \in \mathbb{R}^{n \times q}$, $\mathbf{C}_2 \in \mathbb{R}^{l' \times q}$ are the coefficient matrices, $\mathbf{c}_{1i}$ and $\mathbf{c}_{2i}$ are the i-th column vectors of $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively.

We define the score (confidence value) of the i-th patch as

$$s_i = \|\mathbf{o}_i - \mathbf{N}\widehat{\mathbf{c}}_{1i}\|_2^2 - \left\|\mathbf{o}_i - \mathbf{P}'\widehat{\mathbf{c}}_{2i}\right\|_2^2, i = 1, \ldots, q. \tag{4}$$

As analyzed above, the score of target patch tends to be positive while the score of background patch is on the contrary. In order to alleviate the negative effects caused by outliers, we adjust the patch score as below,

$$s'_i = \begin{cases} 0, \; s_i \le 0 \text{ or } \sum\limits_{j \in \mathcal{N}(i)} \mathbf{1}(s_j \le 0) \ge \tau \\ 1, \; \text{otherwise} \end{cases}, i = 1, \ldots, q, \tag{5}$$

where $\mathbf{1}(\cdot)$ is the indicator function, $\mathcal{N}(i)$ denotes the set of 8 neighbors of the i-th patch. That is, we only care about whether a patch is target part or not, but not how large its score is. Besides, in the light of target connectivity, a patch should be regarded as background patch if a majority of its neighbors are background regions.

We assign the score of each patch to all the pixels inside it, then the confidence map of the context region Ω can be obtained, as shown in Figure 2 (c).

2.3 Bayesian Tracking Framework

Our algorithm is under the Bayesian sequential estimation framework, which performs tracking by solving the maximum a posterior (MAP) problem,

$$\hat{x}_t = \arg\max_{x_t} p(x_t | y_{1:t}), \tag{6}$$

where x_t is the state at time t, $y_{1:t} = \{y_1, \ldots, y_t\}$ represents all the observations up to the t-th frame.

In this work, the target state is defined as $x_t = (x, y, w, h)$, where x, y represent the center location of the target and w, h denote its width and height, respectively. The motion model is assumed to be Gaussian distributed:

$$p(x_t | x_{t-1}) = N(x_t; x_{t-1}, \Psi), \tag{7}$$

where Ψ is a diagonal covariance matrix whose elements are the standard deviations of the four parameters. The observation model $p(y_t | x_t)$, which is of fundamental importance to the success of the tracker, is modeled by

$$p(y_t | x_t^i) \propto \sum_{(j,k) \in B_i} s_{j,k}, \tag{8}$$

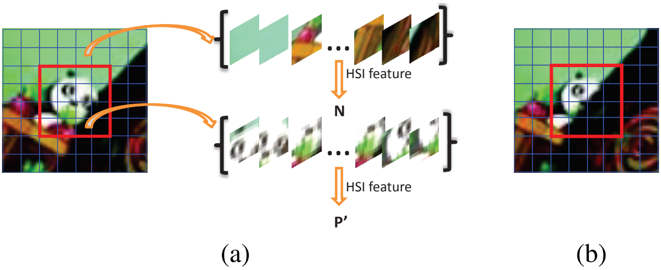


Figure 3: Illustration of occlusion handling. The background dictionary **N** keeps track of the backgrounds around the panda. When occlusion happens, ε_N and $\varepsilon_{P'}$ will decrease and increase, respectively, because some parts of panda are replaced by background pixels.

where x_t^i represents the i-th sample (particle) of state x_t, B_i is its corresponding bounding box, and $s_{j,k}$ is the pixel score at location (j,k).

Analysis: With the confidence map of the context window Ω, we can quickly compute the likelihood of each bounding box candidate (particle) by summating the scores of all pixels inside it. Whereas in conventional particle filter based tracking algorithms, the reconstruction error or classifier response for each bounding box candidate needs to be computed, which is more time consuming.

2.4 Occlusion Handling and Dictionaries Update

Handling Occlusion: Patch based tracking methods have advantages in handling occlusions. With elaborate coupled dictionaries, our method is more robust against occlusions. Figure 3 shows that the background dictionary **N** keeps track of the backgrounds around the target. When the target is occluded at right in Figure 3, the sum of reconstruction errors in target bounding box on **N** (denoted as ε_N) will decrease because some target patches are replaced by background patches, while that on $\mathbf{P}'$ (denoted as $\varepsilon_{P'}$) will increase.

By investigating the above defined reconstruction errors at the current t-th frame and the t'-th $(1 \leq t' < t)$ frame which is the latest one without occlusion, we can decide if there is occlusion in the current frame. Formally, if

$$\varepsilon_N(t) < \gamma \varepsilon_N(t') \text{ and } \varepsilon_{P'}(t) > \eta \varepsilon_{P'}(t') \tag{9}$$

are satisfied simultaneously, we conclude that there is occlusion at the current frame, where $\gamma \in (0,1)$ and $\eta \in (1,+\infty)$ are the given parameters. The changes of ε_N and $\varepsilon_{P'}$ for the case illustrated above are listed in Table 1.

Table 1: The changes of ε_N and $\varepsilon_{P'}$ from (a) to (b) illustrated in Figure 3.

	(a)	(b)
ε_N	293.1	245.3
$\varepsilon_{P'}$	354.2	468.1

Updating Dictionaries: To effectively adapt to the variations of target and backgrounds, we design principled update schemes for **N** and $\mathbf{P}'$ as follows:

Table 2: Experiment results evaluated with CLE and SR, the best 3 values are shown in boldfaced red, purple and blue.

	L1T	MTT	CT	STC	SPT	SRMT	DFT	IVT	Ours
Ball	95	117	12.7	113	**3.77**	102	**5.47**	85.7	**12.6**
Basketball	97	248	122	74.3	**5.57**	40.7	**18**	27.8	**6.76**
Boy	42	**25.7**	**21.9**	25.9	44.7	35.1	106	73.7	**7.15**
CarScale	63.1	92.5	25.9	57.1	**11.5**	25.9	75.8	**11.5**	**11.2**
Fish	40.6	52.5	25.6	**3.98**	33.9	9.49	8.84	**3.94**	**7.42**
Football1	48.4	9.58	11	48.4	**7.71**	29.8	**1.97**	8.83	**5.37**
Kitesurf	39	**11.8**	**10.3**	66.2	67.7	24.4	29.5	74.8	**5.89**
Mhyang	25	**4.03**	24.1	4.53	12.7	**2.68**	9.06	**2.06**	8.12
Panda	92.2	93.4	117	**76**	80.1	**67.5**	183	175	**5.37**
Polarbear	29.9	**12.4**	20.4	21.5	16.9	20.9	**12.5**	25.2	**7.69**
Skiing	**158**	276	257	227	**145**	262	276	255	**6.38**
Subway	**4.86**	193	11.5	143	113	137	**3.31**	135	**10.7**
Average	61.3	94.7	**55**	71.7	**45.2**	63.1	60.8	73.2	**7.89**

	L1T	MTT	CT	STC	SPT	SRMT	DFT	IVT	Ours
Ball	15.4	12	52	11.5	**97.8**	13.6	**83.4**	11.1	**59**
Basketball	23.3	2.76	24.3	23.6	**98.5**	44	**71.6**	37.5	**96.4**
Boy	48.2	45	61	**66.3**	**65**	16.6	48.3	62.1	**86.4**
CarScale	36.9	55.6	44.8	46.8	**93.3**	68.3	44.8	**96.8**	**79**
Fish	12.6	3.99	22.5	37.2	39.1	83.2	**86.1**	**100**	**99.4**
Football1	12.2	73	32.4	35.1	**78.4**	35.1	**100**	68.9	**94.6**
Kitesurf	29.8	**57.1**	41.7	28.6	35.7	29.8	**56**	16.7	**76.2**
Mhyang	68.8	**100**	31.5	86	80.5	96.4	77.5	**97.8**	**98**
Panda	1.66	3.32	4.98	**47.3**	**41.9**	14.9	22	16.6	**78**
Polarbear	43.4	**65**	36.9	31	32.3	29.6	**59.8**	35.6	**97.3**
Skiing	1.23	**11.1**	7.41	**11.1**	**12.3**	**11.1**	6.17	7.41	**40.7**
Subway	**82.3**	7.43	**76.6**	22.3	28.6	22.3	**99.4**	20.6	75.4
Average	31.3	36.4	36.3	37.2	**58.6**	38.7	**62.9**	47.6	**81.7**

- Dictionary $\mathbf{N}$ stores the background patches of the latest κ frames. When the target has been identified at t-th frame, we obtain the background patches as presented in Section 2.1, and use them to partially update $\mathbf{N}$.

- Dictionary $\mathbf{P}'$ contains the target patches of the first frame and the latest $\kappa - 1$ frames. When the target is occluded by judging condition (9), we do not update $\mathbf{P}'$. Otherwise, we partially update $\mathbf{P}'$ with the target patches at the current frame.

3 Experiments

In this section, we evaluate the performance of the proposed algorithm with several state-of-the-art trackers, including L1T [13], MTT [30], CT [27], STC [28], SPT [23], SRMT [31], DFT [15], IVT [14]. These trackers are run with publicly available source codes provided by the authors. For fair comparison, all the trackers are executed with well adjusted parameters to get the performances. And because the trackers involve randomness, we repeat the experiments several times on each sequence and choose the best results. We choose 12 challenging video sequences from the benchmark [25] and VOT Challenge 2014[2].

3.1 Implementation Details

In this paper, we solve sparse coding problems using the public sparse learning package SPAMS [3]. We utilize a normalized histogram in the HSI color space [23] as the feature for

[2] http://www.votchallenge.net/vot2014/

[3] http://spams-devel.gforge.inria.fr/

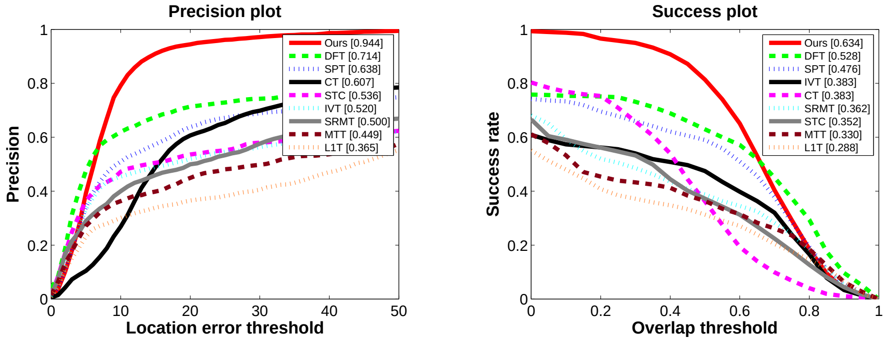


Figure 4: Precision and success plots over all the 12 tested sequences. The precision score of each tracker at 20 pixels is listed in the legend of the left plot. The right plot presents the area-under-the-curve (AUC) score for each method.

each patch, which is robust to lighting changes. All the experiments are implemented in MATLAB R2014a on a 3.10GHz CPU with 6 GB memory. The current tracking speed of our tracker runs at 1.2 frames per second (fps) without optimization. The patch size r is set to $6 \sim 16$ based on the input target size and the padding size p is half of the patch size. Suppose the target's width and height are w and h in the first frame, the normalized width and height of Ω are $\lceil \frac{w}{\alpha \times r} \rceil \times \alpha^2 r$ and $\lceil \frac{h}{\alpha \times r} \rceil \times \alpha^2 r$. The ratio α which decides the size of window Ω is chosen as 2. The proportion β in Section 2.1 is 0.55. We use $\gamma = 0.9$ and $\eta = 1.1$ in inequations (9). The parameter κ in Section 2.4 is empirically defined as 3. The threshold τ is fixed to 7 in Eq. (5). We sample 400 target candidates in each frame.

3.2 Empirical Results

Two conventional performance metrics are adopted for quantitative comparison: average center location error (CLE) and success rate (SR), as shown in Table 2. For each frame, the object is considered being successfully tracked if the overlap percentage is above 0.5. Besides, we provide the precision and success plots over all the 12 tested sequences, as shown in Figure 4.

3.3 Analysis

Overall, our method significantly outperforms the 8 state-of-the-art trackers in both precision and success plots. In Table 2, our approach is almost always among the best two and the best items occupy half of the tested data. The qualitative result is presented in Figure 5. Then we conduct a qualitative analysis of our tracker on several challenging sequences, combined with Figure 5.

In the *Basketball* sequence, most trackers do not drift when the target is occluded at frame 18. Our track performs consistently well even there are some distractors wearing the same jersey, e.g., at frame 644. At the end of this sequence when the illumination changes dramatically, our tracker can still track the object accurately due to the robust HSI features. In *Boy* sequence where the object undergoes fast motion and motion blur, only our tracker does not drift at the end. There exists occlusion and scale variations in the *CarScale* sequence, but our tracking performance is pretty good. In sequences *Kitesurf, Panda* and

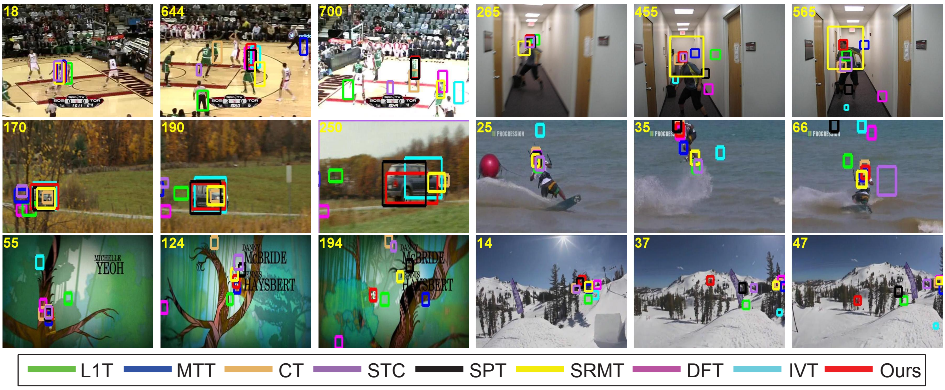


Figure 5: The experimental comparisons of our method and 8 state-of-the-art trackers on 6 sequences.

Skiing, the objects suffer severe rotation, shape deformation and occlusion, our tracker acquires robust performances based on the accurate representations of targets and backgrounds and the robust dictionaries update strategies.

4 Conclusion and Discussion

In this paper, we propose to construct elaborate dictionaries for target and backgrounds respectively. Each patch candidate is sparsely represented by the coupled dictionaries, which obviously enlarges the gaps between background parts and target parts, thus improves the discriminative ability. Experiments on challenging video sequences demonstrate that the proposed tracker performs favorably compared with several state-of-the-art trackers.

In our future work, we will pursue more effective schemes for patch scoring and abnormal patch detection, and the possibility of incorporating structure and location information.

5 Acknowledgement

This work is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390515, 61272319, and 61272321.

References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, 2011.

[2] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550. IEEE, 2010.

[3] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*. BMVA Press, 2014.

[4] Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270. IEEE, 2011.

[5] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715. Springer, 2012.

[6] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829. IEEE, 2012.

[7] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56. IEEE, 2010.

[8] Junseok Kwon, Junha Roh, Kyoung Mu Lee, and Luc Van Gool. Robust visual tracking with double bounding box model. In *ECCV*, pages 377–392. Springer, 2014.

[9] Hanxi Li, Yi Li, and Fatih Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In *BMVC*. BMVA Press, 2014.

[10] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):58, 2013.

[11] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, pages 2720–2727. IEEE, 2013.

[12] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *CVPR*, pages 5388–5396, 2015.

[13] Xue Mei and Haibin Ling. Robust visual tracking using ℓ_1 minimization. In *ICCV*, pages 1436–1443. IEEE, 2009.

[14] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3): 125–141, 2008.

[15] Laura Sevilla-Lara and Erik Learned-Miller. Distribution fields for tracking. In *CVPR*, pages 1910–1917. IEEE, 2012.

[16] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.

[17] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Least soft-threshold squares tracking. In *CVPR*, pages 2371–2378. IEEE, 2013.

[18] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *NIPS*, pages 809–817, 2013.

[19] Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *ICML*, pages 1107–1115, 2014.

[20] Naiyan Wang, Jingdong Wang, and Dit-Yan Yeung. Online robust non-negative dictionary learning for visual tracking. In *ICCV*, pages 657–664. IEEE, 2013.

[21] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

[22] Qing Wang, Feng Chen, Jimei Yang, Wenli Xu, and Ming-Hsuan Yang. Transferring visual prior for online object tracking. *Image Processing, IEEE Transactions on*, 21 (7):3296–3305, 2012.

[23] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *ICCV*, pages 1323–1330. IEEE, 2011.

[24] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[25] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418. IEEE, 2013.

[26] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[27] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In *ECCV*, pages 864–877. Springer, 2012.

[28] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, pages 127–141. Springer, 2014.

[29] Shengping Zhang, Hongxun Yao, Xin Sun, and Xiusheng Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46(7): 1772–1788, 2013.

[30] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049. IEEE, 2012.

[31] Zhe Zhang and Kin Hong Wong. Pyramid-based visual tracking using sparsity represented mean transform. In *CVPR*, pages 1226–1233. IEEE, 2014.