Hindawi
Advances in Multimedia
Volume 2018, Article ID 7357284, 10 pages
https://doi.org/10.1155/2018/7357284

Research Article

Robust Visual Tracking with Discrimination Dictionary Learning

Yuanyun Wang,[1,2] Chengzhi Deng,[1] Jun Wang,[1,2] Wei Tian,[1] and Shengqian Wang[1]

[1]*Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, China*
[2]*School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China*

Correspondence should be addressed to Chengzhi Deng; dengchengzhi@126.com

Received 3 June 2018; Revised 3 August 2018; Accepted 10 August 2018; Published 2 September 2018

Academic Editor: Shih-Chia Huang

Copyright © 2018 Yuanyun Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is a challenging issue to deal with kinds of appearance variations in visual tracking. Existing tracking algorithms build appearance models upon target templates. Those models are not robust to significant appearance variations due to factors such as illumination variations, partial occlusions, and scale variation. In this paper, we propose a robust tracking algorithm with a learnt dictionary to represent target candidates. With the learnt dictionary, a target candidate is represented with a linear combination of dictionary atoms. The discriminative information in learning samples is exploited. In the meantime, the learning processing of dictionaries can learn appearance variations. Based on the learnt dictionary, we can get a more stable representation for target candidates. Additionally, the observation likelihood is evaluated based on both the reconstruct error and dictionary coefficients with ℓ_1 constraint. Comprehensive experiments demonstrate the superiority of the proposed tracking algorithm to some state-of-the-art tracking algorithms.

1. Introduction

Visual tracking is a fundamental task in computer vision, which is applied in a wide range of applications, such as intelligent transportation, video surveillance, human-computer interaction, and video editing. The goal of visual tracking is to estimate target states of a tracked target in each frame. Although many tracking algorithms are proposed in recent decades [1], designing a robust tracking algorithm remains a challenging issue due to factors such as fast motion, out-of-rotation, nonrigid deformation, and background clutters.

Based on the types of target observations, visual tracking algorithms can be classified as either generative [2–6] or discriminative [7–14]. Generative tracking algorithms search for an image patch that has the most similarity to the tracked target model as the tracking result in the current frame. For a generative algorithm, the prime problem is to build an effective appearance model that is robust to complicated appearance variations.

The discriminative tracking algorithms consider visual tracking as a binary classification problem. The tracked target is distinguished from the surround background by learnt classifiers. The classifiers compute the confidence value for target candidates and distinguish each as a foreground target or a background block. In this work, we will propose a generative algorithm. Next, we briefly review some related works to our tracking algorithm and some recent tracking algorithms.

Kwon et al. [3] decompose the observation model and motion models into multiple basic observation models and multiple basic motion models, respectively. Each basic observation model covers a target appearance variation. Each basic motion model covers a special motion model. A basic observation model and a basic motion model are combined into a basic tracker. The tracking algorithm is robust to drastic appearance changes. He et al. [4] propose an appearance model based on locality sensitive histograms at each pixel location. The proposed observation model is robust to drastic illumination variations. In [2], a target candidate is represented by a set of intensity histograms of multiple image patches, which has a vote value on the corresponding position. A target candidate is represented

by fixed target templates, which is not robust to drastic appearance variations. Wang et al. [6] represent a target candidate based on target templates with affine constraint. The observation likelihood is computed based on a learnt distance metric.

The representation technique with sparse constraint is applied into visual tracking [15–19]. The target representations with sparsity are robust to outliers and occlusions. In [15], Mei et al. use a set of target templates to represent target candidates and represent partial occlusions with trivial templates. The algorithm in [15] is robust to partial occlusion. While severe occlusions occur, the algorithm is not effective. Zhong et al. [18] propose a collaborative model with sparsity constraint. In order to improve the tracking performance, the tracking algorithm combines the generative model and discriminative model. Zhang et al. [16] exploit the spatial layout structure of a target candidate and represent target appearance based on local information and spatial structure. In [19], a target candidate is represented by underlying low-rank with sparse constraints, in which the temporal consistency is used.

Recently, correlation filter [21–24] and deep network [25–28] techniques are applied into visual tracking. In [23], the tracking algorithm takes different features to learn correlation filter. The proposed appearance model is robust to large-scale variations and maintain multiple modes in a particle filter tracking framework. Liu et al. [22] exploit the part-based structure information for correlation filter learning. The learnt filters can accurately distinguish foreground parts from the background. In [28], Ma et al. exploit object features from deep convolutional neural networks. The output of the convolutional layers includes semantic information and hierarchies, which is robust to appearance variations. Huang et al. [27] propose deep feature cascades based tracking algorithm, which considers the visual tracking as a decision-making process.

Motivated by the above-mentioned work, we propose a learnt dictionary based appearance model. A target candidate is represented by a linear combination of the learnt dictionary atoms. The dictionary learning process can learn appearance variations. The dictionary atoms cover recent appearance variations and a stable target representation is obtained. The observation likelihood is evaluated based on the reconstruction error with sparse constrain on dictionary coefficients. Extensive experimental results on some challenging video sequences show the robustness and effectiveness of the proposed appearance model and the tracking algorithm.

The remainder of this paper is organized as follows. Section 2 proposes the novel tracking algorithm, which includes the appearance model, the dictionary learning, the observation likelihood evaluation, and the dictionary update. Section 3 compares the tracking performance of the proposed tracking algorithm with some state-of-the-art algorithms. Section 4 concludes this work.

2. Proposed Tracking Algorithm

In this section, we detail the proposed tracking algorithm including an appearance model based on a learnt dictionary, discrimination dictionary learning for target representation, and a novel likelihood function. In this work, we propose the tracking algorithm in a particle filter tracking framework [29]. The particle filter framework is widely used in visual tracking due to its effectiveness and simplification.

In our tracking algorithm, the target state in the first frame is given as $\mathbf{s}_1$. $\mathbf{y}_1$ denotes the corresponding target observation of $\mathbf{s}_1$. In the first frame, a set of particles (i.e., target candidates) are extracted and denoted as $\mathbf{X}_1 = \{\mathbf{x}_1^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_1^m\}$. These particles are collected by cropping out an image regions surrounding the location of $\mathbf{s}_1$. These particles have same sizes as $\mathbf{s}_1$ and they have same important weights as $w_1^i = 1/m$, $i = 1, 2, \ldots, m$. The particles in frame t are denoted as $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^m\}$. The states and the corresponding observation of particle $\mathbf{x}_t^i$ are denoted as $\mathbf{s}_t^i$ and $\mathbf{y}_t^i$ with important weights w_t^i, respectively.

The particles $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^m\}$ in frame t are propagated from frame $t-1$ according the state transition model $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$. $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$ is assumed to be a Gaussian distribution:

$$p\left(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i\right) \sim \mathbf{G}\left(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i, \Sigma\right), \tag{1}$$

where the covariance Σ is a diagonal matrix, in which the diagonal entries denote the variances of the 2D position and the scale of a target candidate.

The target state and the corresponding observation in the t-th frame are denoted as $\mathbf{s}_t$ and $\mathbf{y}_t$, respectively. In the particle filter framework, the $\mathbf{s}_t$ in the frame t is approximately estimated by $\mathbf{x}_t^i$ as

$$\mathbf{s}_t = \sum_{i=1}^{m} w_t^i \mathbf{x}_t^i, \tag{2}$$

where w_t^i is the weight of the particle $\mathbf{x}_t^i$. In the tracking, the particle weights are dynamically updated according to the likelihood of the particle $\mathbf{x}_t^i$ as

$$w_t^i = w_{t-1}^i p\left(\mathbf{y}_t^i \mid \mathbf{x}_t^i\right), \tag{3}$$

where $p(\mathbf{y}_t^i \mid \mathbf{x}_t^i)$ is the likelihood function of particle $\mathbf{x}_t^i$, which is introduced in (15).

2.1. Target Representations. In existing algorithm, a target candidate is represented by a linear combination of a set of target templates. These templates are usually generated from tracking results in previous frames. There are some noises and uncertain information in these templates due to complicated appearance variations. These tracking algorithms are not robust to drastic variations. Thus, in our tracking algorithm, a target candidate is approximately represented by the atoms of a learnt dictionary.

Based on a learnt dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$, a target candidate $\mathbf{y}$ is approximately represented as

$$\mathbf{y} = \mathbf{d}_1\alpha_1 + \mathbf{d}_2\alpha_2 + \cdots + \mathbf{d}_n\alpha_n, \tag{4}$$

where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$ is a learnt dictionary. $\mathbf{d}_i$ is an atom of the learnt dictionary $\mathbf{D}$. n is the number of the atoms. $\boldsymbol{\alpha} =$

$[\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the coefficient of the dictionary $\mathbf{D}$. The dictionary coefficient $\boldsymbol{\alpha}$ is evaluated by solving

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left\| \mathbf{y} - \mathbf{D}\boldsymbol{\alpha} \right\|_2^2, \tag{5}$$

where $\boldsymbol{\alpha}$ are the coefficient vector for the target candidate $\mathbf{y}$ associated with a learnt dictionary $\mathbf{D}$.

2.2. Dictionary Learning. In existing tracking algorithms, target candidates are represented by target templates, which are some tracking results from previous frames. To improve the tracking performance, we use a learnt dictionary to approximately represent target candidates.

Denote by $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]$ the set of training samples. Denote by $\mathbf{V}$ the coding vector matrix of training samples $\mathbf{T}$ over $\mathbf{D}$, i.e., $\mathbf{T} = \mathbf{DV}$. The learnt dictionary should have discriminative capability and can adapt to learn appearance variations like partial occlusions, nonrigid deformation, illumination variations, and so on. Based on the learnt dictionary, a stable target representation model is obtained. Motivated by a dictionary learning method [30], we use a discriminative dictionary learning model as

$$J_{D,V} = \arg\min_{\mathbf{D},\mathbf{V}} \left\{ \Phi(\mathbf{T}, \mathbf{D}, \mathbf{V}) + \gamma_1 \|\mathbf{V}\|_1 + \gamma_2 f(\mathbf{V}) \right\}, \tag{6}$$

where $\Phi(\mathbf{T}, \mathbf{D}, \mathbf{V})$ is the discriminative fidelity term; $\|\mathbf{V}\|$ is the sparsity constraint on the coefficient matrix $\mathbf{V}$; $f(\mathbf{V})$ is a discriminate constraint on the coding coefficient matrix $\mathbf{V}$; γ_1 and γ_2 are parameters for balancing this constraint terms. In [30], the dictionary $\mathbf{D}$ includes a set of subdictionaries for all classes. Different from the dictionary learning in [30], in our tracking algorithm, $\mathbf{D}$ is a one-class dictionary and it is learnt from only a set of positive training samples.

In the dictionary learning process, based on the reconstruction error between the training samples $\mathbf{T}$ and the dictionary $\mathbf{D}$, the discriminative fidelity term is defined as

$$\Phi(\mathbf{T}, \mathbf{D}, \mathbf{V}) = \|\mathbf{T} - \mathbf{DV}\|_F^2. \tag{7}$$

To improve the discriminative performance of the learnt dictionary, we add the Fisher discriminative criterion to minimize the within-class scatter of the coefficient matrix $\mathbf{V}$. Denote by $S(V)$ the with-class scatter, which is defined as

$$S(V) = \Sigma_{v_i \in \mathbf{V}} (v_i - m)(v_i - m)^T, \tag{8}$$

where v_i is a vector of the coefficient matrix $\mathbf{V}$, m is the mean vector of the coefficient vector $\mathbf{V}$. In the learning process, we use the trace of $S(V)$ as constraint term $f(\mathbf{V})$ in (6). To prevent some coefficients that are too large in constraint term, a regularized term $\|\mathbf{V}\|_F^2$ is added to $f(\mathbf{V})$

$$f(\mathbf{V}) = \mathrm{tr}(S(\mathbf{V})) + \mu \|\mathbf{V}\|_F^2, \tag{9}$$

where μ is a balancing parameter.

Based on (7), (8), and (9), the dictionary learning model can be rewritten as

$$J_{(D,V)} = \arg\min_{(\mathbf{D},\mathbf{V})} \left\{ \|\mathbf{T} - \mathbf{DV}\|_F^2 + \gamma_1 \|\mathbf{V}\|_1 + \gamma_2 \mathrm{tr}(S(\mathbf{V})) + \gamma_3 \|\mathbf{V}\|_F^2 \right\}, \tag{10}$$

where γ_1, γ_2, and γ_3 are positive scalar parameters.

In (10), we update the dictionary $\mathbf{D}$ and the corresponding coefficient $\mathbf{V}$, iteratively. In the updating processing, one is updated when the other is fixed. When the dictionary $\mathbf{D}$ is fixed, the optimization function is reduced to the following:

$$J_{(V)} = \arg\min_{(\mathbf{D},\mathbf{V})} \left\{ \|\mathbf{T} - \mathbf{DV}\|_F^2 + \gamma_1 \|\mathbf{V}\|_1 + \gamma_4 \|\mathbf{V}\|_F^2 \right\}, \tag{11}$$

and

$$f(\mathbf{V}) = (v_i - m)^T (v_i - m) + \gamma_3 \|\mathbf{V}\|_F^2, \tag{12}$$

where v_i is a vector of the coefficient matrix $\mathbf{V}$ and m is the mean vector of the coefficient vector $\mathbf{V}$.

When $\mathbf{V}$ is fixed, the dictionary $\mathbf{D}$ in (10) is updated as

$$J_{(D)} = \arg\min_{(\mathbf{D})} \|\mathbf{T} - \mathbf{DV}\|_F^2. \tag{13}$$

In the learning process, the training samples are primely important, which should reflect the recent variations of the tracked target and keep diversity to adapt to target appearance variations. In the first frame, a set of training samples are collected. Firstly, the initialized target is selected as training samples. In the meantime, the other training samples are selected by perturbing a few pixels surrounding the center location of the tracked target.

In order to keep the diversity of the learnt dictionary to appearance variations, we set the size of the training samples to 25. In the subsequent frames, we should update the training samples and relearn a dictionary to adapt to target appearance variations. At the current frame, when the tracked target state is computed and located, we crop the corresponding image and extracted the feature vector as a new training sample. Then, the new training sample is added to the set of current training samples and the oldest training sample is swapped.

2.3. Likelihood Evaluation. The similarity metric of a target candidate and the corresponding candidate is an important issue in visual tracking. In this work, the similarity is measured as

$$d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}}) = (\mathbf{y} - \mathbf{D}\widehat{\boldsymbol{\alpha}})^T (\mathbf{y} - \mathbf{D}\widehat{\boldsymbol{\alpha}}), \tag{14}$$

where $\mathbf{D}$ is the learnt dictionary and $\widehat{\boldsymbol{\alpha}}$ is the coefficient vector of the learnt dictionary computed in (6).

Based on the distance between a target candidate and the corresponding template dictionary, the target observation likelihood is computed as

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left\{ -\psi \left(d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}}) \right) - \zeta \|\widehat{\boldsymbol{\alpha}}\|_1 \right\}, \tag{15}$$

where $d(\mathbf{y}, \mathbf{D}\widehat{\boldsymbol{\alpha}})$ is the distance between a target candidate $\mathbf{y}$ and the corresponding dictionary $\mathbf{D}$, ψ is the standard deviation of the Gaussian, and ζ is a positive parameter.

2.4. Visual Tracking with Dictionary Based Representation. By integrating the proposed target representation and the online dictionary learning and updating and the observation evaluation, the proposed visual tracking algorithm is outlined in Algorithm 1. The particle filter framework is used for all

(1) In the first frame, manually select the tracked target state $\mathbf{s}_1$; collect n training samples $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]$; learn a dictionary $\mathbf{D}_1 = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ according to the learning scheme in Section 2.2; sample m particles $\{\mathbf{x}_1^i\}_{i=1}^m$ with equal weights as $1/m$.
Input: t-th video frame.
(2) Resample m particles $\{\mathbf{x}_t^i\}_{i=1}^m$ according to motion model $p(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i)$.
(3) Extract feature vectors $\{\mathbf{y}_t^i\}_{i=1}^m$ according to $\{\mathbf{x}_t^i\}_{i=1}^m$.
(4) **for** $i = 1$ to m **do**
(5) Compute observation likelihoods $p(\mathbf{y}_t^i \mid \mathbf{x}_t^i)$ via Eqn. (15).
(6) Update particle weight w_t^i via Eqn. (3).
(7) **end**
(8) Compute target state $\widehat{\mathbf{s}}_t$ with Eqn. (2).
(9) Extract feature vector $\mathbf{y}_t$ according to $\widehat{\mathbf{s}}_t$.
(10) Update the training samples with $\mathbf{y}_t$.
(11) Learn dictionary $\mathbf{D}_t$ according to Eqns. (11) and (13) in Section 2.2.
(12) Obtain dictionary $\mathbf{D}_t$.
(13) Return $\widehat{\mathbf{s}}_t$.

Algorithm 1: Proposed tracking algorithm.

Table 1: Average frames per second (FPS).

Sequence	Coupon	Fish	Football	Football1	Man	Singer2	Sylv	Walking
Frames	327	476	362	74	134	366	1345	412
Total times	269	430	196	39	67	437	724	253
FPS	1.21	1.11	1.84	1.90	1.99	0.84	1.86	1.63

video sequences. For a video sequence, the tracked target is manually selected by a bounding box in the first frame. A set of particles (i.e., target candidates) are selected with same weights in the particle framework. The training samples are collected and a dictionary is learnt in the first frame. In the subsequent tracking processing, when the current target states are evaluated, the current tracking result is added to the training samples. The dictionary is relearnt according to the updated training samples.

3. Experiments

We conduct comprehensive experiments on some challenging video sequences and compare the proposed tracking algorithm against some state-of-the-art tracking algorithms. These tracking algorithms include Struck [12], SCM [18], VTD [3], Frag [2], L1 [15], LSHT [4], LRT [19], and TGPR [20]. For fairness, we use the source codes or binary codes provided by the authors and initialize all the evaluated algorithms with default parameters in all experiments. 8 challenging video sequences from a recent benchmark [1] are used to evaluate the tracking performance. Table 2 shows the main challenging attributes in these test sequences.

The proposed tracking algorithm is implemented in MATLAB. All experimental results are conducted on a PC with Intel(R) Core(TM) i5-2400 3.10GHZ and 4 GB memory. The number of particles is set to 300. The target features are described by the histograms of sparse coding (HSC) [31]. The value of ψ in (15) is set to 20. In the proposed tracking algorithm, the number of atoms of a learnt dictionary is set to 25.

The proposed tracking average processing time of the proposed tracking algorithm is 1.55 frames per second (FPS). We show the average tracking speed for each sequence in Table 1. Compared with some state-of-the-art tracking algorithms [1], the proposed tracking algorithm is superior to SCM. However, it is slow to some tracking algorithms, e.g., Struck, VTD, and LSHT. This is due to online dictionary learning spending some time in optimizing the target representation. We can learn the dictionary every five frames. But this may influence the dictionary adaption to complicated tracking surrounding. In our tracking algorithm, the dictionary is learnt in each frame.

3.1. Quantitative Evaluation. We use four evaluation measures in the experiments including average center location error, success rate, overlap rate, and precision. These measures are adopted in recent tracking benchmark [1].

We show the precision plots for these tracking algorithms in Figure 1 for 9 tracking algorithms. The average center location errors (CLE) are shown in Table 3. From Figure 1 and Table 3, we can see that the proposed tracking algorithm obtains the best two results in six of eight sequences. The proposed tracking algorithm achieves the smallest CLE over all the 8 sequences. TGPR achieves robust tracking results in the *Football*1, *Sylv*, and *Singer*2 video sequences. LRT obtains the best tracking results in the *Coupon*, *Walking*, and *Football* video sequences. Struck tracks the *Fish* and *Man* video sequences well and achieves the best tracking results in CLE.

Table 4 presents success rates for 9 tracking algorithms on the 8 sequences. Figure 2 also shows the success rate

Table 2: The main attributes of the 8 video sequences. Target size: the initial target size in the first frame; OPR: out-of-plane rotation; IPR: in-plane rotation; BC: background clutter; IV: illumination variation; Occ: occlusion; Def: deformation; SV: scale variation.

Sequence	Frames	Image size	Target size	OPR	IPR	BC	IV	Occ	Def	SV
Coupon	327	320×240	62×98			√		√		
Fish	476	320×240	60×88				√			
Football	362	624×352	39×50	√	√	√		√		
Football1	74	352×288	26×43	√	√	√				
Man	134	241×193	26×40				√			
Singer2	366	624×352	67×122	√	√	√	√		√	
Sylv	1345	320×240	51×61	√	√		√			
Walking	412	768×576	24×79					√	√	√

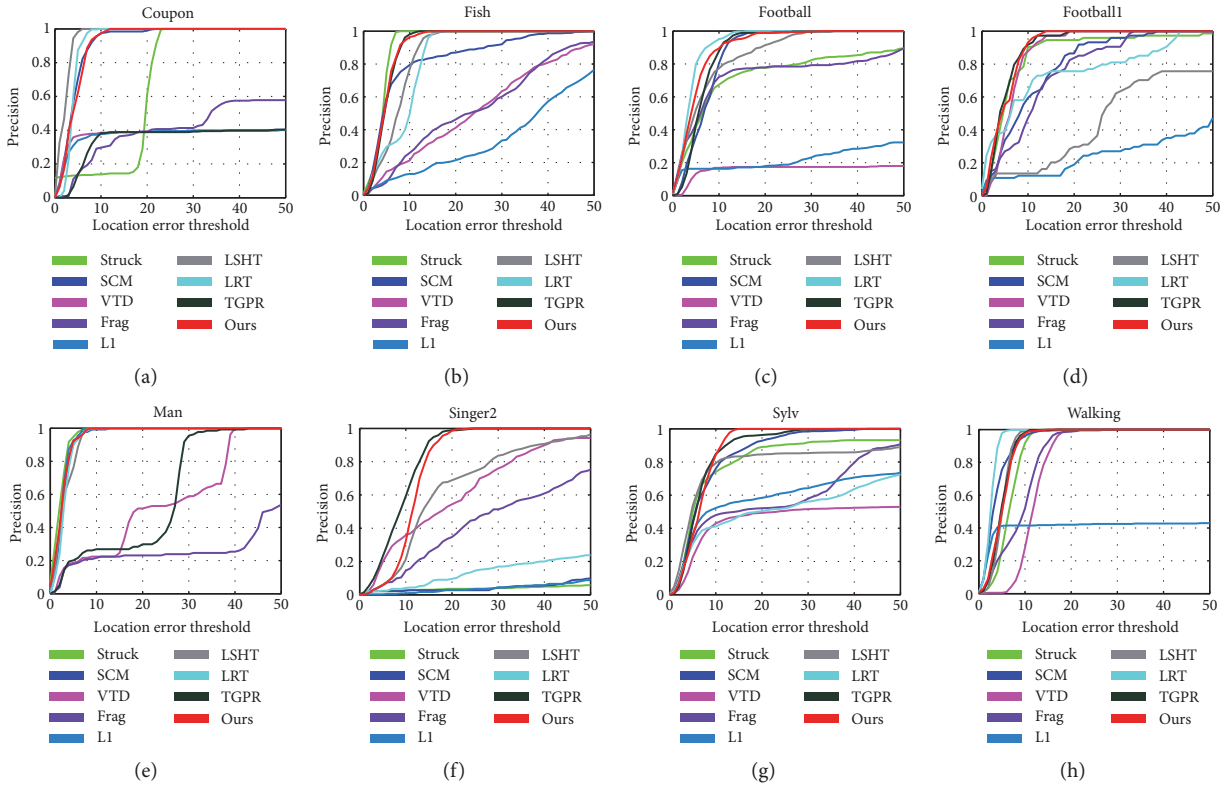


Figure 1: Precision plots in terms of location error threshold (in pixels).

plots for the evaluated tracking algorithms. From Table 4 and Figure 2, it can be seen that the proposed tracking tracks well in most of these video sequences. The proposed tracking algorithm obtains the best tracking results in most of these video sequences. Additionally, TGPR achieves accurate tracking results in the *Fish* and *Singer*2 video sequences. LRT achieves the best tracking results in the *Coupon*, *Walking*, and *Football* video sequences. LSHT obtains the best tracking results in the *Fish*, *Man*, and *Coupon* video sequences.

Table 5 presents the average overlap rates for the 9 tracking algorithms. It can be seen that the proposed tracking algorithm achieves the best or the second best tracking results in most video sequences. Struck, LSHT, LRT, and TGPR also achieve robust tracking results in some video sequences.

3.2. Qualitative Evaluation. Next, we will analyze the tracking performance of these tracking algorithms on the 8 video sequences.

In the *Coupon* sequence shown in Figure 3(a), the tracked target is a coupon book with cluttered background. When the target is occluded by himself, VTD and L1 drift away from the target. Frag, TGPR, VTD, and L1 lose the target and track the other similar object until the end of whole sequence. Struck, SCM, LSHT, LRT, TGPR, and the proposed tracking algorithm can accurately track the target throughout the video sequence.

Figure 3(b) presents some tracking results for the 9 tracking algorithms on the *Fish* sequence. Struck, LSHT, and TGPR achieve accurate tracking results. The proposed

Table 3: Average center location errors (in pixels). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	15.0	6.0	65.2	56.2	66.3	**4.3**	*3.4*	65.7	**4.3**
Fish	*3.9*	8.3	24.7	24.7	36.4	7.3	8.5	4.8	**4.7**
Football	15.3	6.9	218.3	**4.6**	68.4	7.2	*3.8*	6.2	4.9
Football1	7.0	10.4	6.4	11.9	59.3	30.8	12.1	**5.0**	*4.9*
Man	*2.3*	2.9	22.8	44.6	**2.6**	3.1	3.0	20.8	**2.6**
Singer2	174.7	172.2	20.2	35.9	145.8	18.4	126.6	*8.8*	**11.4**
Sylv	11.7	7.9	58.4	22.7	31.0	13.6	28.9	**6.8**	*6.5*
Walking	6.5	**3.9**	11.8	8.9	125.3	5.0	*2.6*	5.0	5.0
Average	29.5	27.3	53.5	26.2	66.9	*11.2*	23.6	15.4	*5.5*

Table 4: Success rates (%). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	*100*	*100*	39.4	40.9	39.4	*100*	*100*	39.4	*100*
Fish	*100*	86.6	39.7	47.3	20.2	*100*	*100*	*100*	*100*
Football	69.3	88.7	16.9	72.9	16.3	79.6	*96.7*	**94.8**	92.5
Football1	**89.2**	39.2	81.1	43.2	12.2	14.9	58.1	85.1	*90.5*
Man	99.3	98.5	22.4	21.0	98.5	*100*	99.3	26.1	*100*
Singer2	3.6	3.0	33.3	45.9	4.1	67.5	9.8	*100*	*100*
Sylv	80.3	86.6	48.4	50.3	55.5	83.3	48.1	**94.3**	*100*
Walking	54.9	**79.1**	16.3	50.2	41.5	54.6	*97.3*	55.1	*77.7*
Average	74.6	72.7	37.2	46.5	36.0	75.0	**76.2**	74.3	*95.1*

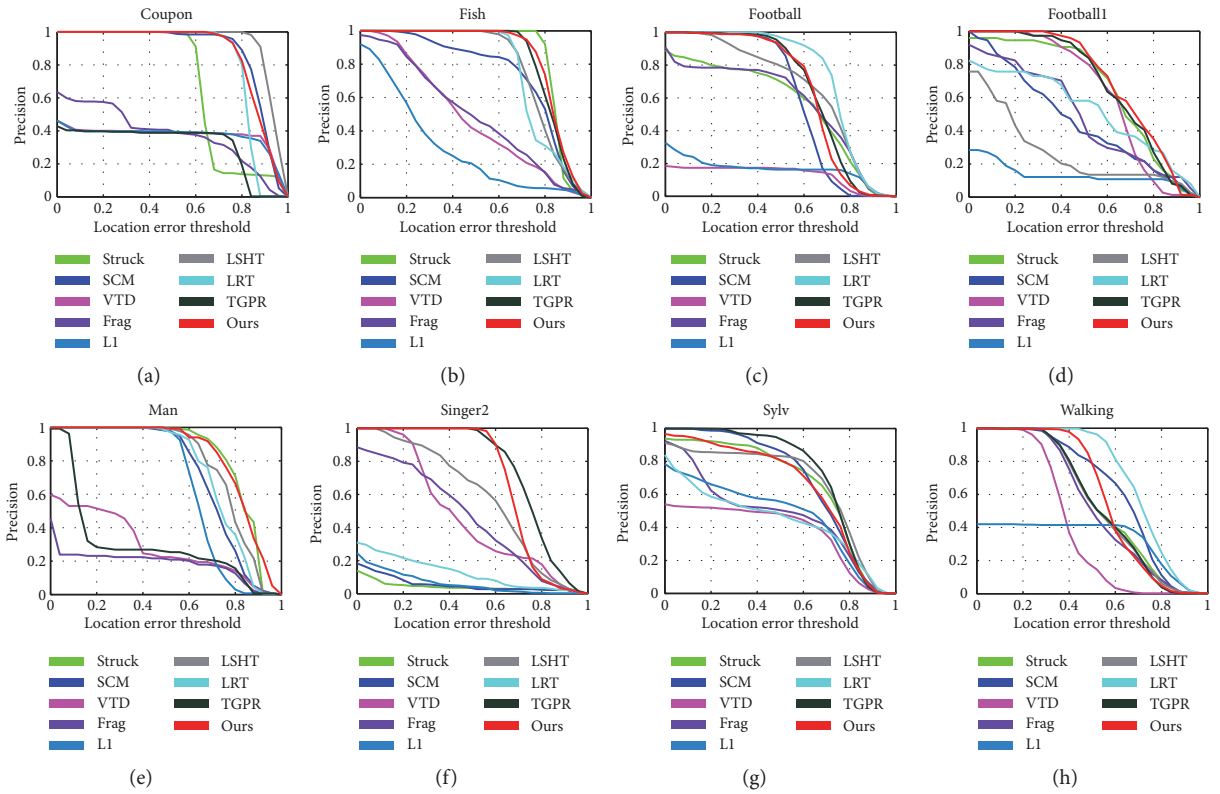
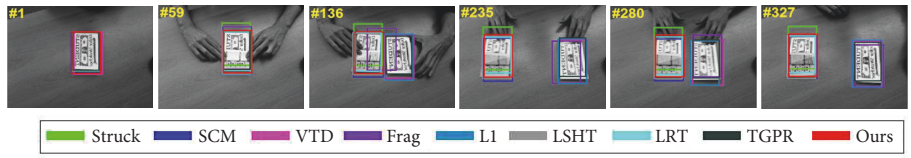
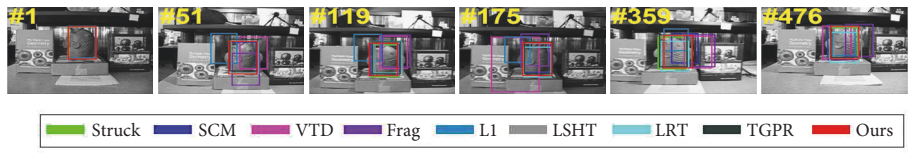


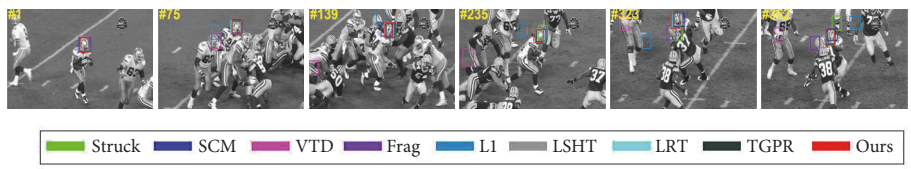
Figure 2: Success plots in terms of overlap threshold.



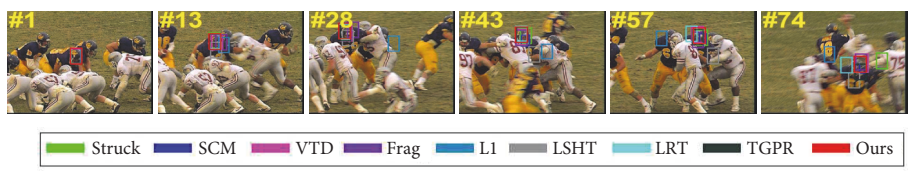
(a) *Coupon*



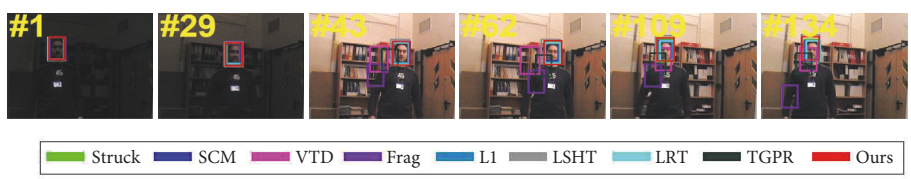
(b) *Fish*



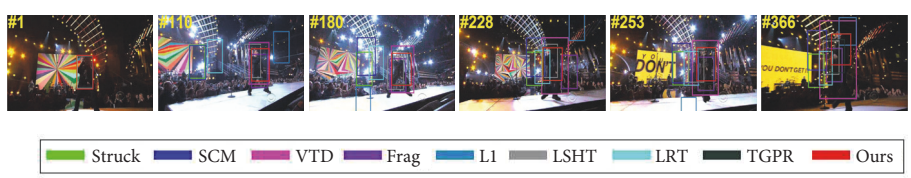
(c) *Football*



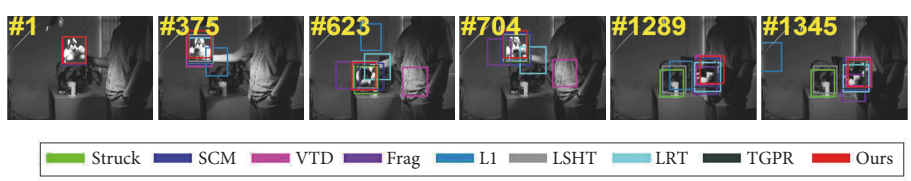
(d) *Football1*



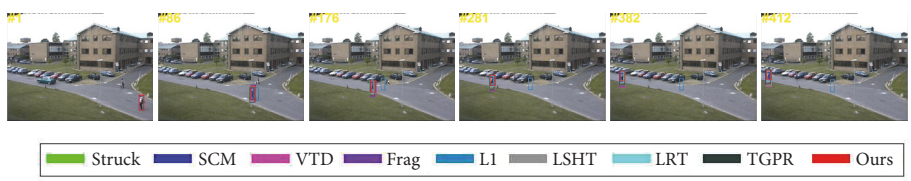
(e) *Man*



(f) *Singer2*



(g) *Sylv*



(h) *Walking*

Figure 3: The tracking results on the 8 sequences.

Table 5: Average overlap rates (%). The best two results are shown in italic and bold colors, respectively.

Sequence	Struck [12]	SCM [18]	VTD [3]	Frag [2]	L1 [15]	LSHT [4]	LRT [19]	TGPR [20]	Ours
Coupon	70.2	82.3	36.2	37.1	35.2	*89.2*	80.7	32.8	**84.1**
Fish	*84.3*	74.0	47.3	48.9	28.6	78.1	76.5	82.4	**83.7**
Football	55.7	60.3	13.0	56.2	16.2	66.0	*75.0*	**66.9**	65.5
Football1	66.0	45.4	63.1	48.4	13.1	26.0	52.2	**67.9**	*69.6*
Man	**81.9**	71.9	28.8	17.5	65.3	77.8	74.1	29.4	*82.9*
Singer2	4.2	5.3	46.9	44.9	6.0	58.1	12.3	*75.4*	**69.5**
Sylv	66.0	67.8	37.4	46.4	46.4	65.8	43.4	*72.5*	**71.0**
Walking	55.9	**63.4**	39.2	53.4	32.8	55.0	*71.6*	55.1	58.9
Average	60.5	58.8	39.0	44.1	30.4	**64.5**	60.7	60.3	*73.2*

tracking algorithm can learn the appearance variations in the dictionary learning processing. It can accurately track the target throughout the video sequence. Struck, LSHT, LRT, and TGPR also track the target until the end of the video sequence.

In the *Football* sequence shown in Figure 3(c), a football match is going on. The tracked target is a player, which is similar to others in color and shape. The target undergoes partial occlusion, background clutter, and in-plane and out-of-plane rotations. Due to the influence of background clutter, VTD and L1 lose the target. When some similar objects appear surrounding the target, Struck, SCM, Frag, and LSHT track the other similar distracters and lose the target. Compared with these algorithms, LRT, TGPR, and the proposed tracking algorithm can track the target successfully.

As shown in Figure 3(d), the tracked target is influenced by out-of-plane and in-plane rotations. And there are some other objects that are similar to the tracked target. Frag loses the target when other distracters appear surrounding the target, which is very similar to the tracked target. L1 and Frag achieve inaccurate tracking results when the target rotates in-plane and out-of-plane. Struck, TGPR and the proposed tracking algorithm can track the target throughout the sequence. In the three algorithms, the proposed algorithm achieves the most accurate tracking results in average center location error, success, and overlap rates.

Figure 3(e) shows some tracking results in the *Man* sequence. The tracked target is a moving face in an indoor room. Influenced by drastic illumination variations, VTD, Frag, and TGPR lose the target after the 40th frame until the end of the sequence. Struck, SCM, LSHT, LRT, and the proposed tracking algorithm can successfully track the whole sequence. The proposed tracking algorithm obtains the most robust tracking result.

The *Singer*2 video sequence shown in Figure 3(f) is captured on an indoor stage with drastic illumination variations. The target is also affected by nonrigid deformation, background clutters, and in-plane and out-of-plane rotations. Struck, SCM, L1, and LRT only track the target before the first 47th frame due to the appearance variations. VTD obtains inaccurate scale evaluation when the target undergoes nonrigid deformation. The proposed tracking algorithm can successfully track the target in the whole sequence. The learnt dictionary covers the target variations, so the linear combination of the atoms in the dictionary can represent these variational appearance.

In the *Sylv* video sequence shown in Figure 3(g), the target is a moved toy, which undergoes illumination variations and in-plane and out-of-plane rotations. L1 loses the target due to the influence of illumination variations and rotation from up to down. It locates the target again after the 495th frame when the tracked target rotates from down to up. When the target is affected by illumination variation, LRT drift away from the tracked target until the end of the video sequence. Frag uses fixed target templates, so it cannot adapt to the appearance variations. It achieves inaccurate tracking results. Compared with these algorithms, the proposed tracking algorithm obtains more tracking results. This is attributed to the fact that the proposed target representation can learn the appearance variations.

As shown in Figure 3(h), the tracked target is a walking man in an outdoor scene. Due to the influence of partial occlusion, nonrigid deformation, and scale variation, L1 loses the target until the end of this sequence. VTD, TGPR, and LSHT can not accurately evaluate the scale variation. SCM, Struck, LRT, and the proposed tracking algorithm obtain more accurate tracking results.

From the above analysis, we can see that the proposed appearance model is effective and efficient. The proposed tracking algorithm is robust to significant appearance variations, e.g., drastic illumination variations, partial occlusion, and out-of-plane rotation.

4. Conclusion

We have presented an effective tracking algorithm based on learnt discrimination dictionary. Different from exist tracking algorithms, a target candidate is represented with a linear combination of dictionary atoms. The dictionaries are leant and updated in the tracking processing, which can learn the target appearance variation and exploit the discriminative information in the learning samples. The learning samples are collected from previous tracking results. The proposed tracking algorithm is robust to drastic illumination variations, nonrigid deformation, and rotation. Conducted experiments on some challenging video sequences demonstrate the robustness in comparison with state-of-the-art tracking algorithms.

Data Availability

(1) The video sequences data used to support the findings of this study have been deposited in http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html or http://www.visual-tracking.net. (2) The measure metrics and the challenging attributes in videos used to support the finding of this study are included within the articles: Y. Wu, J. Lim, and M. Yang, Online Object Tracking: A Benchmark, IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2411-2418.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Jiangxi Science and Technology Research Project of Education Department of China (Nos: GJJ151135 and GJJ170992), the National Natural Science Foundation of China (No: 61661033), the Jiangxi Natural Science Foundation of China (Nos: 20161BAB202040 and 20161BAB202041), and the Open Research Fund of Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing (No: 2016WICSIP020).

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[2] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[3] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, June 2010.

[4] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2427–2434, June 2013.

[5] J. Wang, H. Z. Wang, and Y. Yan, "Robust visual tracking by metric learning with weighted histogram representations," *Neurocomputing*, vol. 153, pp. 77–88, 2015.

[6] J. Wang, Y. Wang, and H. Wang, "Adaptive Appearance Modeling with Point-to-Set Metric Learning for Visual Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1987–2000, 2017.

[7] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, pp. 47–56, September 2006.

[8] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[9] K. H. Zhang, L. Zhang, and M. H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[11] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization," in *Computer Vision – ECCV 2014*, vol. 8694 of *Lecture Notes in Computer Science*, pp. 188–203, Springer International Publishing, Cham, 2014.

[12] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, IEEE, Barcelona, Spain, November 2011.

[13] L. Ma, X. Zhang, W. Hu, J. Xing, J. Lu, and J. Zhou, "Local Subspace Collaborative Tracking," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4301–4309, Santiago, Chile, December 2015.

[14] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3002–3010, Chile, December 2015.

[15] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[16] T. Zhang, S. Liu, C. Xu, S. Yan, and B. Ghanem, "Structure sparse tracking, IEEE Conference on Computer Vision and Pattern Recognition," in *Proceedings of the Structure sparse tracking, IEEE Conference on Computer Vision and Pattern Recognition*, pp. 150–158, 2015.

[17] L. Zhang, H. Lu, D. Du, and L. Liu, "Sparse hashing tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 840–849, 2016.

[18] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2356–2368, 2014.

[19] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, pp. 1–20, 2014.

[20] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8691 of *Lecture Notes in Computer Science*, pp. 188–203, 2014.

[21] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 4310–4318, Santiago, Chile, December 2015.

[22] T. Zhang, S. Liu, C. Xu, B. Liu, and M.-H. Yang, "Correlation particle filter for visual tracking," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2676–2687, 2018.

[23] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 4819–4827, USA, July 2017.

[24] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1387–1395, USA, July 2017.

[25] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, "Robust Object Tracking Based on Temporal and Spatial Deep Networks," in *Proceedings of the 16th IEEE International Conference*

on Computer Vision, ICCV 2017, pp. 1153–1162, Italy, October 2017.

[26] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4293–4302, July 2016.

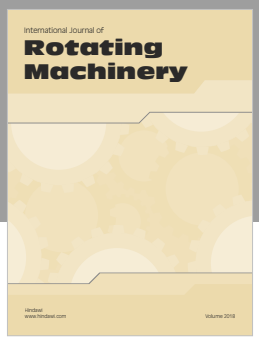
[27] C. Huang, S. Lucey, and D. Ramanan, "Learning Policies for Adaptive Tracking with Deep Feature Cascades," in *Proceedings of the 16th IEEE International Conference on Computer Vision, ICCV 2017*, pp. 105–114, Italy, October 2017.

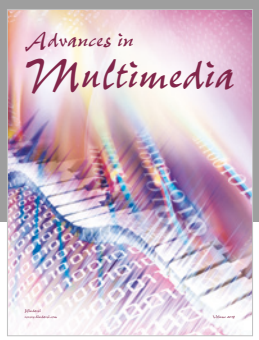
[28] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3074–3082, Chile, December 2015.

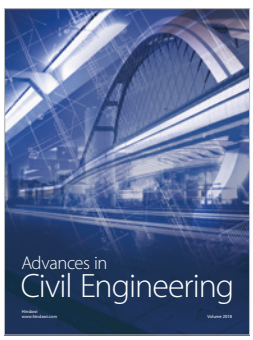
[29] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

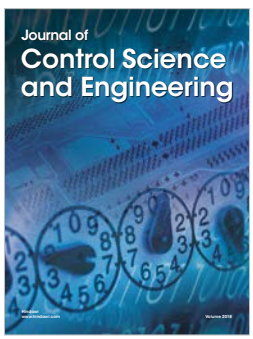
[30] M. Yang, L. Zhang, X. C. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 543–550, Barcelona, Spain, November 2011.

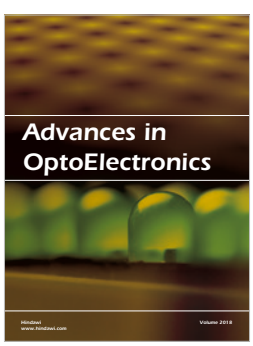
[31] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 3246–3253, USA, June 2013.

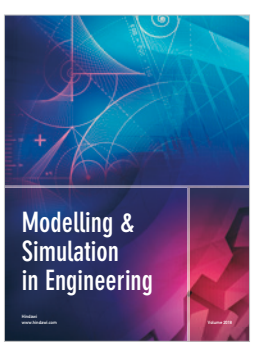


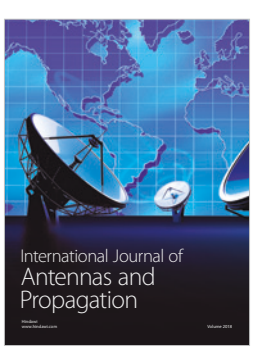


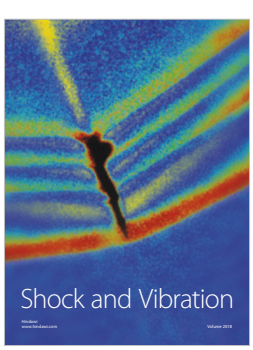


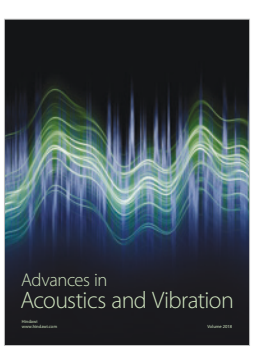












Hindawi

Submit your manuscripts at
www.hindawi.com