A Biologically Inspired Appearance Model for Robust Visual Tracking

Shengping Zhang, *Member, IEEE*, Xiangyuan Lan, *Student Member, IEEE*, Hongxun Yao, *Member, IEEE*, Huiyu Zhou, Dacheng Tao, *Fellow, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*— In this paper, we propose a biologically inspired appearance model for robust visual tracking. Motivated in part by the success of the hierarchical organization of the primary visual cortex (area V1), we establish an architecture consisting of five layers: whitening, rectification, normalization, coding, and pooling. The first three layers stem from the models developed for object recognition. In this paper, our attention focuses on the coding and pooling layers. In particular, we use a discriminative sparse coding method in the coding layer along with spatial pyramid representation in the pooling layer, which makes it easier to distinguish the target to be tracked from its background in the presence of appearance variations. An extensive experimental study shows that the proposed method has higher tracking accuracy than several state-of-the-art trackers.**

***Index Terms*— Appearance modeling, biologically inspiration, sparse coding, visual tracking.**

I. Introduction

VISUAL tracking is a task that continuously infers the state of a specific target from an image sequence. It is a specific task of computer vision, which has attracted increasing interest in recent years [1]–[15]. Given a target template, a visual tracking process usually consists of four stages: 1) candidate sampling, whereas we sample a set of target candidates from the current image frame; 2) appearance modeling, where we describe the target template and a number of candidates for correspondence using an appearance representation method; 3) target searching, which allows us to find which candidate has the best similarity to the target template; and 4) template updating, where we update the appearance of the target template in order to adapt to the variations of the target appearance over time.

In these four stages, appearance modeling attracts large attention in the last few decades [3], [16]–[29]. A good appearance model should not only be used to distinguish the target from its background but should also be robust against appearance variations due to pose changes, illumination, and occlusions. Although many appearance models have been reported in the literature, they still do not have sufficient capabilities to handle these challenges. One of the main reasons is that most of these models are based on handcrafted features [30]–[32]. The computation procedures of these features are fixed regardless of input data and, therefore, cannot be adapted to different tracking scenarios. Several recent models resort to machine learning methods and devise suitable training criteria and optimization methods to learn features from data [33]–[35]. In spite of their high adaptive capability, these methods still somehow depend on prior knowledge, which does not easily transfer to other applications.

While robust visual tracking remains very challenging to computers, it is effortless for humans to accomplish this task. The progress made in understanding the brain mechanisms of visual information processing, especially the primary visual cortex (area V1), enables us to develop effective appearance models to handle the tracking problems. In this paper, we focus on developing a biologically inspired appearance model for robust visual tracking. The motivation is twofold.

1) *Motivation (i):* Psychophysical evidence has shown that V1 can process visual information using a hierarchical structure [36], [37], which alternates between simple and complex cells. Riesenhuber and Poggio [38] proposed a quantitative model to simulate such a hierarchy, where the responses of simple cells were produced using local filters, while the invariance properties of complex cells were created using a max pooling process over the neighboring positions.
2) *Motivation (ii):* Olshausen and Field [39] proposed an unsupervised learning method, called sparse coding, to represent a natural image as a linear combination of a set of basis functions. Assume the coefficients are sparse and statistically independent. People have observed that these basis functions captured the underling image

Manuscript received July 15, 2015; revised February 7, 2016 and March 18, 2016; accepted June 22, 2016. This work was supported in part by the National Natural Science Foundation of China (Nos. 61300111 and 61472103) and Key Program (No. 61133003); in part by the Australian Research Council Projects DP-140102164, FT-130101457, and LE140100061. The work of H. Zhou was supported by the UK EPSRC under Grants EP/G034303/1, EP/N508664/1 and EP/N011074/1. *(Corresponding author: Shengping Zhang.)*

S. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: s.zhang@hit.edu.cn).

X. Lan is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: xylan@comp.hkbu.edu.hk).

H. Yao is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: s.zhang@hit.edu.cn).

H. Zhou is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT3 9DT, U.K. (e-mail: h.zhou@qub.ac.uk).

D. Tao is with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, 81 Broadway Street, Ultimo, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au).

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2586194

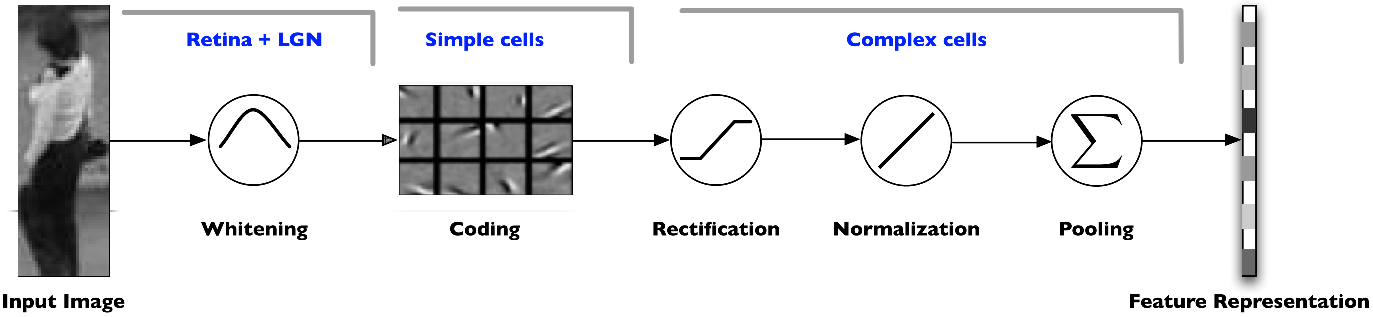


Fig. 1. Architecture of the proposed biologically inspired appearance modeling framework. LGN = lateral geniculate nucleus.

structures (lines and edges). More importantly, the basis functions have similar response properties to those of the simple cells in area V1.

Motivation (i) suggests that we can perform appearance modeling for visual tracking by simulating the hierarchical structure of area V1. In the literature, several appearance models with this motivation have been reported for object recognition [40]–[42], which used one or two stages of V1-like feature representation. The responses of simple cells at the bottom of the hierarchy were generated using Gabor filters. At the top of the hierarchy, complex cells were simulated by performing max pooling over the local neighboring responses. Their experimental results show that these models outperform the other state-of-the-art object recognition systems on several standard data sets [40], [41].

Although these models are biologically inspired, simple cells were only simulated by handcrafted Gabor filters, resulting in a relatively poor evolutionary capability in their systems. In fact, neurons in the visual cortex should be adapted to different statistical properties of the visual signals to which they are exposed [43], [44]. Therefore, a more reasonable simulation of simple cells in visual cortex is to use learning-based methods to capture the statistical properties of the visual signals. Sparse coding shown in motivation (ii) is a model that meets this requirement. Therefore, we here propose to use sparse coding to model the responses of simple cells in V1. Furthermore, considering the importance of the discriminative ability of an appearance representation for visual tracking, we, therefore, use discriminative sparse coding and spatial pyramid pooling to make the representation more capable of distinguishing the target from its background.

The architecture of the proposed biologically inspired appearance model is shown in Fig. 1, which consists of five layers. The whitening, rectification, and normalization layers stem from object recognition models and perform very well in various applications [41], [42]. However, in visual tracking applications, we need to maintain the discriminative capability of the appearance model employed in the tracker all the time, which is more challenging than the case of object recognition, and therefore, our attention in this paper focuses on the coding and pooling layers. In particular, in the coding layer, we will adopt a discriminative sparse coding technique to represent local patches densely sampled from the input image. This makes it easy to distinguish between the target and the background patches. In the pooling layer, average pooling with spatial pyramid representation will be used to aggregate the local codes into a global representation of the input image. The multiple scale pyramid structure can be used to preserve the spatial relationship of the local codes, and hence increase the global discriminative capability of the final feature representation with strong robustness against local appearance variations.

The rest of this paper is organized as follows. In Section II, we review the work closely related to our proposed approach. Section III gives a detailed description of the proposed biologically inspired appearance model. The tracking algorithm based on the proposed appearance model is introduced in Section IV. Experimental results are reported and analyzed in Section V. We conclude this paper in Section VI.

II. Related Work

In this section, we review the biologically inspired appearance models used for the object recognition, state-of-the-art tracking approaches, and corresponding benchmarks.

A. Biologically Inspired Models for Object Recognition

Based on the findings in human psychophysics and electrophysiology studies [36], [37], [45], Riesenhuber and Poggio [38] proposed a quantitative model to simulate the hierarchical structure in human vision for object recognition, where the response of the simple cells was obtained using the second derivative of Gaussian filters, and the invariance properties of the complex cells were obtained using a max pooling operator over the neighboring positions. Serre *et al.* [40] proposed the HMAX model, which uses Gabor filters to replace the second derivative of Gaussian filters used in [38]. In addition, they also proposed to use a two-stage V1-like feature representation. In the second stage of the system, the filters were built by randomly sampling the outputs of the first stage. The max pooling of the filters' response over the entire image forms the final feature representation. Pinto *et al.* [41] simplified the HMAX model and only used the first stage of the HMAX model with additional preprocessing and normalization operators. In addition, they used a sum operator instead of a max operator in the pooling stage. The experiments against several object recognition data sets demonstrate that their system achieved better performance than the other state-of-the-art systems. Yang *et al.* [46] improved the bag-of-features model [47] by using sparse coding to replace the K-means in the vector quantization step. Although they do not claim that their method is biologically inspired, their feature extraction method is very similar to that of [41]. Other researchers, such as [42], [48], and [49], conducted extensive experiments in object recognition in order to compare the performance of different hierarchical architectures. The evaluation results indicate that a single-layer model with coding, pooling, and suitable preprocessing and normalization operators can achieve the expected recognition performance.

B. State-of-the-Art Tracking Approaches and Corresponding Benchmarks

Visual tracking has attracted increasing interests in recent years. A large number of novel tracking approaches, large benchmark data sets, and experimental comparison were reported [1], [6], [50]–[54]. In [50], a kernelized structured output support vector machine (SVM) was learned online to achieve adaptive tracking. Real-time tracking was achieved by introducing a budgeting mechanism, which prevents the unbounded growth of the number of support vectors that would otherwise occur during tracking. Very recently, deep learning has been successfully applied to different vision tasks. Inspired by recent advances in deep learning architectures, Wang and Yeung [51] proposed a deep compact image representation for visual tracking, which first trains a stacked denoising autoencoder offline to learn generic image features that are more robust against variations and then transfers the offline training to an online tracking process. Chen *et al.* [55] used cells to extract local appearance, and constructed complex cells to integrate the information from cells to provide diverse and important object cues for visual tracking. With different spatial arrangements of cells, complex cells are of various contextual information at multiple scales, which is important to improve the tracking accuracy. Although two biological terms, such as cells and complex cells, were used in their paper, there is little biologically inspiration. Henriques *et al.* [6] proposed a new kernelized correlation filter (KCF) for visual tracking, which can achieve fast and accurate tracking. To evaluate the progress in visual tracking, there are also several benchmark databases and experimental surveys on recent tracking methods. In [1], a benchmark with 50 test sequences was built for evaluating the state-of-the-art tracking methods, which is then further extended in [54] to have 100 test sequences. In [53], a set of 19 trackers were evaluated on 315 video fragments, which provides objective insight into the strengths and weaknesses of these trackers.

In the literature, several appearance models based on sparse coding have been proposed for visual tracking, which usually consist of two steps: local coding and global pooling. After the input image is divided into a set of image patches, each patch is sparsely coded by a linear combination of a set of basis functions. The combination coefficients are used as codes to describe the image patch. The final appearance representation of the input image is obtained by pooling the codes of all the patches to form a histogram-like feature vector. Zhang *et al.* [56] proposed to code image patches using independent component analysis basis functions, which were learned from a set of randomly sampled image patches. Average pooling was used to obtain the final appearance representation. In [19], each patch was coded by the basis functions of sparse coding, which were learned using the sampled patches out of the target image. Similar to [56], average pooling was used to construct the final feature representation. Wang *et al.* [57] combined the sampled patches from the target image and the identity vectors as basis functions to code each patch. All the codes were concatenated together to form the final feature representation. Zhong *et al.* [58] learned the basis functions of sparse coding from the patches sampled from the target image via K-means. The pooling step is similar to the one used in [57]. Jia *et al.* [20] maintained multiple target templates. Local patches sampled from each template were collected and formed as basis functions to code patches sampled from the input image with the same spatial layout. They used an alignment pooling operator to preserve the structural relationship between local codes.

Although these appearance models, based on sparse coding, have shown promising results in some applications, they also accompany some inevitable shortcomings.

1) The basis functions of sparse coding are obtained by either learning from images [56] or sampling the patches from the tracking data [20]. The basis functions are so generative that the resulting sparse codes from the target are not discriminative enough to distinguish the target from its background.
2) The sparse codes are pooled by either summing all the coefficient vectors over the entire image that losses the spatial relationship [19], [56], or concatenating all the coefficient vectors that preserve the spatial relationship but also violating the local invariance of the appearance representation [57].
3) Supported by the biological studies [59], [60] and empirical evaluation [41], [42], nonlinear operators, such as whitening, rectification, and normalization operators, can also enhance the representation capability. However, the existing appearance models based on sparse coding only contain the basic coding and pooling operators and, therefore, ignore the role of the nonlinear operators.

III. Proposed Biologically Inspired Appearance Model

The visual information received by simple cells in V1 is just part of the information contained in the whole retina image, which is determined by the size of the receptive field of simple cells. To simulate this mechanism, first of all, a set of n patches are densely sampled from an input image $\mathbf{I}$. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ be the sampled patch set, where $\mathbf{x}_i \in \mathbb{R}^d$ refers to the stacked pixel intensities from the ith patch. To reduce the impact of noise, each patch is first normalized by subtracting its mean and then divided by its standard deviation.

The proposed appearance modeling method is used to find a feature mapping function $\mathbf{y} = \mathcal{F}(\mathbf{X})$, which takes in the patch set $\mathbf{X}$ and then outputs a feature vector $\mathbf{y}$ that is capable of

Algorithm 1 Proposed Appearance Modeling Method

```
Input  : Input image $\mathbf{I}$, parameters $n$ and $\lambda$
Output: Appearance representation $\mathbf{y}$
Sample $n$ patches $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ from the input image $\mathbf{I}$;
Whiten each patch $\mathbf{x}_i$ in $\mathbf{X}$ to have the whitened patch $\mathbf{w}_i$;
Code each whitened patch $\mathbf{w}_i$ using the learned discriminative dictionary to get the responses $\mathbf{c}_i$;
Rectify the responses of a whitened patch to obtain the rectified responses $\mathbf{r}_i$;
Normalize the rectified responses of a whitened patch to gain the normalised responses $\mathbf{n}_i$;
Integrate the normalised responses in a pyramid division to obtain the feature representation
$\mathbf{y} = \{\mathbf{y}_{s,v,h} | s = 0, 1, 2, v = 1, 2, \dots, 2^s, h = 1, 2, \dots, 2^s\}$
```

representing the appearance of the image $\mathbf{I}$. Under the hierarchical framework, as shown in Fig. 1, let $\mathcal{W}$, $\mathcal{C}$, $\mathcal{R}$, $\mathcal{N}$, and $\mathcal{P}$ be the feature mappings of the whitening, coding, rectification, normalization, and pooling layers, respectively. The feature mapping function $\mathcal{F}$ can be obtained by sequencing all the layers as $\mathcal{F} = \mathcal{P} \circ \mathcal{N} \circ \mathcal{R} \circ \mathcal{C} \circ \mathcal{W}$. Algorithm 1 summarized the steps of our appearance modeling method. In Sections III-A–III-E, we will introduce how to implement these feature mappings in detail.

A. Whitening Layer

A natural scene observed by the human eye is transmitted to the retina. Since the adjacent pixels of the image tend to be highly correlated in intensity, it is wasteful for the retina to transmit every pixel separately to the brain for further processing. Instead of doing this, the retina performs a decorrelation operation via retinal neurons, which can be implemented using the Zeros Component Analysis whitening operator [61]. To whiten each patch in $\mathbf{X}$, the covariance matrix is first computed as $\mathbf{X}\mathbf{X}^\top$, where $\top$ denotes the matrix transpose. The eigenvalue decomposition of the covariance matrix is $\mathbf{E}\mathbf{D}\mathbf{E}^\top$, where the orthogonal matrix $\mathbf{E}$ consists of eigenvectors and the diagonal matrix $\mathbf{D}$ consists of eigenvalues. Then, the whitening operator can be achieved as follows:

$$\mathbf{w}_i = \mathcal{W}(\mathbf{x}_i) = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^\top \mathbf{x}_i. \tag{1}$$

After the whitening, all the dimensions of the image patches are uncorrelated and their variances are equal. For computation, the whitening operator is also useful to speed up the convergence of the coefficient inference in the subsequent coding layer.

B. Coding Layer

The goal of the coding layer is to extract local features from each image patch. In contrast to other handcrafted features, in this paper, we consider using sparse coding to learn sparse features for each image patch. The motivation of using sparse coding to extract features is twofold: 1) the sparse features are capable of capturing the underlying structural properties of the images due to small reconstruction errors and the sparsity constraints in the dictionary learning and 2) studies in neuroscience clearly reveal that the human brain codes visual information in a sparse manner.

Given a dictionary with K basis functions $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K] \in \mathbb{R}^{d \times K}$, the coding responses of a whitened patch $\mathbf{w}_i$ can be computed as

$$\mathbf{c}_i = \mathcal{C}(\mathbf{w}_i) = \arg\min_{\mathbf{c}} \|\mathbf{w}_i - \mathbf{V}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1 \tag{2}$$

where λ is a regularization parameter that controls the importance of the sparsity constraint of the responses to the reconstruction error. Similar to the design of the handcrafted filters [38], [40], a critical step in the coding layer is to learn the dictionary $\mathbf{V}$ from a set of image patches $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots \tilde{\mathbf{x}}_{\tilde{n}}] \in \mathbb{R}^{d \times \tilde{n}}$, which are sampled from a set of training images. After the whitening, these image patches supply training samples $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \dots \tilde{\mathbf{w}}_{\tilde{n}}] \in \mathbb{R}^{d \times \tilde{n}}$ for the dictionary learning, where $\tilde{\mathbf{w}}_i = \mathcal{W}(\tilde{\mathbf{x}}_i)$. The traditional sparse coding scheme learns the dictionary by solving the following equations:

$$\arg\min_{\mathbf{V}, \mathbf{c}_i} \sum_{i=1}^{\tilde{n}} \|\tilde{\mathbf{w}}_i - \mathbf{V}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_1. \tag{3}$$

The objective function above only contains the reconstruction error and sparse coding responses, which is essentially required for learning a sparse but approximate representation. However, such a representation is not good enough for those applications, which require high discriminative ability, e.g., the visual tracking task studied in this paper. To make the coding responses easier to discriminate the target from its background, in this paper, we utilize a discriminative dictionary learning method [62], which learns two dictionaries for the target and the background, respectively. We call them as the target dictionary and the background dictionary accordingly in the following sections. The target dictionary represents the patches sampled from the target image with the minimal reconstruction error, while representing the patches sampled from the background image with the maximal reconstruction error. Similarly, the background dictionary represents the background patches with the minimal reconstruction error, while representing the target patches with the maximal reconstruction error. The objective of using this dictionary learning method is to produce discriminative codes in the coding layer when we use the dictionary consisting of both the target and background dictionaries.

Let $\tilde{\mathbf{W}}_f = [\tilde{\mathbf{w}}_{f,1}, \tilde{\mathbf{w}}_{f,2}, \dots, \tilde{\mathbf{w}}_{f,n_f}] \in \mathbb{R}^{d \times n_f}$ and $\tilde{\mathbf{W}}_b = [\tilde{\mathbf{w}}_{b,1}, \tilde{\mathbf{w}}_{b,2}, \dots, \tilde{\mathbf{w}}_{b,n_b}] \in \mathbb{R}^{d \times n_b}$ be the training patches sampled from the target and background images, respectively. Let $\mathbf{V}_f \in \mathbb{R}^{d \times K_f}$ and $\mathbf{V}_b \in \mathbb{R}^{d \times K_b}$ be the dictionary pair learned from the target and background patches, respectively. The reconstruction error of coding a sample $\tilde{\mathbf{w}}$ using the dictionary $\mathbf{V}$ is computed as $\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}) = \|\tilde{\mathbf{w}} - \mathbf{V}\mathcal{C}(\tilde{\mathbf{w}})\|_2^2$. To learn and create the dictionary pair $\mathbf{V}_f$ and $\mathbf{V}_b$, we define the following discriminative cost function:

$$\mathcal{L}(\tilde{\mathbf{w}}, \mathbf{V}_f, \mathbf{V}_b) = \log(1 + e^{\lambda(\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_f) - \mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_b))}) \tag{4}$$

where $\lambda > 0$ is a parameter to be determined. $\mathcal{L}(\tilde{\mathbf{w}}, \mathbf{V}_f, \mathbf{V}_b)$ is close to zero when $\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_f) < \mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_b))$. The discriminative dictionary pair $\mathbf{V}_f$ and $\mathbf{V}_b$ can be established by optimizing this cost function

$$\min_{\mathbf{V}_f,\mathbf{V}_b} = \frac{1}{n_f}\sum_{i=1}^{n_f}\mathcal{L}(\tilde{\mathbf{w}}_{f,i},\mathbf{V}_f,\mathbf{V}_b) + \gamma\,\mathcal{E}(\tilde{\mathbf{w}}_{f,i},\mathbf{V}_f) + \frac{1}{n_b}\sum_{i=1}^{n_b}\mathcal{L}(\tilde{\mathbf{w}}_{b,i},\mathbf{V}_b,\mathbf{V}_f) + \gamma\,\mathcal{E}(\tilde{\mathbf{w}}_{b,i},\mathbf{V}_b) \quad (5)$$

where $\gamma > 0$ is a parameter to be determined, and n_b and n_f are the two numbers of the investigated patches. Given the randomly initialized $\mathbf{V}_f$ and $\mathbf{V}_b$, this optimization problem can be solved by a standard K-SVD technique [63] with typically five iterations, consisting of two steps.

1) *Coding:* Compute codes for each patch in $\tilde{\mathbf{W}}_f$ (or $\tilde{\mathbf{W}}_b$) using dictionary $\mathbf{V}_f$ (or $\mathbf{V}_b$).
2) *Updating:* Update each column of $\mathbf{V}_f$ or $\mathbf{V}_b$ to reduce the residual of the cost function. After learning and updating the dictionaries $\mathbf{V}_f$ and $\mathbf{V}_b$, we then create a combination form as $\mathbf{V} = [\mathbf{V}_f, \mathbf{V}_b] \in \mathbb{R}^{d\times K}$, where $K = K_f + K_b$.

After obtaining the dictionary, we have the coding outputs of the whitened patches $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$ as $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$, where $\mathbf{c}_i = \mathcal{C}(\mathbf{w}_i)$ is computed using (2) with the learned dictionary.

C. Rectification Layer

Neuroscience studies [59], [64] indicate that simple cells in visual cortex are rarely in their maximum saturation regime and suggest that their response can be approximated by a rectification operator

$$\mathbf{r}_i = \mathcal{R}(\mathbf{c}_i) = \max(\mathbf{0}, \mathbf{c}_i) \quad (6)$$

where $\max(\cdot,\cdot)$ operates on two vectors and returns the maximal value of each dimension. The rectification layer makes the response to the opposite of an excitatory pattern to be 0 (i.e., no response). On the other hand, the outputs of the coding layer and the coefficients still possess some small but nonzero values. The rectification operator enables the codes much sparser than an usual case, which has been proved to achieve better object recognition performance [42].

D. Normalization Layer

Empirical evidence shows that the rectified coefficients, corresponding to the basis functions at the neighboring spatial positions, orientations, and scales, are highly correlated. An operator of removing these dependences is divisive normalization [60], [65], which represents each coefficient by a weighted combination of its rectified neighbors. Let $\mathbf{r}_i$ and $\mathbf{r}_i^1, \mathbf{r}_i^2, \ldots, \mathbf{r}_i^P$ be the rectification layer's outputs of the image patch $\mathbf{x}_i$ and its P neighboring patches, respectively, the divisive normalization for the kth dimension can be implemented by

$$\mathbf{n}_i(k) = \mathcal{N}(\mathbf{r}_i(k)) = \frac{\mathbf{r}_i(k)}{\sum_{p=1}^{P} w_p \mathbf{r}_i^p(k) + \delta} \quad (7)$$

where $\mathbf{r}(k)$ denotes the kth element of the vector $\mathbf{r}$, $\mathbf{w} = [w_1, w_2, \ldots, w_P]$ are the weights of the neighboring patches, and δ is a constant. The parameters $\mathbf{w}$ and δ can be learned by collecting a set of patches and then minimizing the prediction error. To simplify the model, we here use equal weights $w_p = 1/P$ and an infinitesimal value δ.

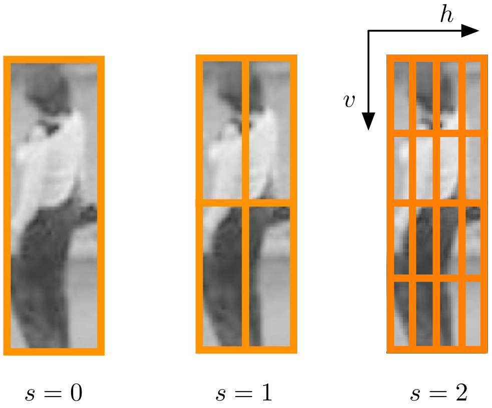


Fig. 2. Spatial pyramid structure with three scales.

E. Pooling Layer

The purpose of appearance modeling is to find a global feature representation to describe the target to be tracked. After obtaining the rectified and normalized coding responses of the local patches sampled from the input image, we combine these local codes to form a global representation. This can be implemented by a pooling operator, which employs the statistics of the local sparse codes. Biological studies have shown that the complex cells in area V1 also use a pooling operator to make the final representation invariant to small distortions. Some biological models recruited sum pooling [37], which computed the sum of responses to a specific stimuli over a neighboring space. Recently, people proposed to use max pooling instead of sum pooling [40], [66], which used a max operator to replace the sum operator in the sum pooling. Although max pooling can produce a more invariant representation, our experimental results indicate that it is not a good pooling operator for representing the appearance model in visual tracking. The reason is that max pooling increases the invariance while losing much discriminative elements. Comparison experiments in object recognition also indicate that when an appropriate rectification layer is adopted, sum pooling slightly outperforms max pooling [42]. Therefore, in this paper, we choose to use sum pooling rather than max pooling. However, sum pooling over the entire image has a significant drawback—the spatial relationships of the local codes will be lost. Since the spatial relationships are important for visual tracking, we propose to use spatial pyramid pooling proposed in [67], which effectively preserves the spatial relationships and effectively increases the global discriminative capability with sufficient local robustness against appearance variations.

In particular, we evenly partition the input image into $2^s \times 2^s$ subimages in three different scales $s = \{0, 1, 2\}$ (see Fig. 2). Let v and h be the indices of the subimages along the vertical and horizontal directions, respectively. The (v, h)th subimage in the scale s is denoted by $\mathbf{I}_{s,v,h}$, where $v = 1, 2, \ldots, 2^s$ and $h = 1, 2, \ldots, 2^s$. Let $\mathbf{N}_{s,v,h} = [\mathbf{n}_{s,v,h,1}, \mathbf{n}_{s,v,h,2}, \ldots, \mathbf{n}_{s,v,h,L}] \in \mathbb{R}^{K\times L}$ be the normalization layer outputs of L image patches sampled in the subimage $\mathbf{I}_{s,v,h}$. Because each patch in the subimage is encoded by all the basis functions, the distribution of responses of all the basis functions can be used as features to describe the subimage. Taking this into account, the feature representation $\mathbf{y}_{s,v,h}$ of the subimage $\mathbf{I}_{s,v,h}$ can be obtained by sum pooling along each dimension over all the image patches in the subimage, which calculates the activation frequencies of basis functions when encoding the image patches in the subimage. For example, the kth element of $\mathbf{y}_{s,v,h}$ is obtained by

$$\mathbf{y}_{s,v,h}(k) = \mathcal{P}(\mathbf{N}_{s,v,h}(k)) = \sum_{l=1}^{L} \mathbf{n}_{s,v,h,l}(k) \tag{8}$$

where $\mathbf{N}(k)$ denotes the kth row of the matrix $\mathbf{N}$, and $\mathbf{n}(k)$ denotes the kth element of the vector $\mathbf{n}$. After pooling along all the dimensions, we then normalize $\mathbf{y}_{s,v,h}$ to make the sum of its all elements to be one. The feature set $\{\mathbf{y}_{s,v,h} | s = 0, 1, 2, v = 1, 2, \ldots, 2^s, h = 1, 2, \ldots, 2^s\}$, computed from all the subimages, is used as the final appearance representation of the input image.

Please note that the multiscale pooling is implemented for each input image to compute its appearance representation. Here, the input image can be either a target template image (as shown in Fig. 2) or a target candidate image (e.g., an image with the same size as the target image but which may contain some background regions). The number of all the subimages in all the three scales is 21.

IV. Proposed Tracking Algorithm

Based on the proposed appearance model presented earlier, in this section, we introduce the proposed tracking algorithm using a multiscale pyramid matching scheme within a standard particle filter framework. The detailed algorithm for tracking is outlined in Algorithm 2.

A. Particle Filter Framework

Particle filtering [68] is a popular computation method to recursively approximate the posterior distribution of the state variables characterizing a dynamic system. It consists of two stages: prediction and update. Let $\mathbf{z}_t$ and $\mathbf{I}_t$ be the state variable and the observation at time t, respectively. The posterior distribution of $\mathbf{z}_t$ given all the available observations $\mathbf{I}_{1:t-1} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{t-1}\}$ up to time $t-1$ can be predicated using the state transition model $p(\mathbf{z}_t | \mathbf{z}_{t-1})$ as

$$p(\mathbf{z}_t | \mathbf{I}_{1:t-1}) = \int p(\mathbf{z}_t | \mathbf{z}_{t-1}) p(\mathbf{z}_{t-1} | \mathbf{I}_{1:t-1}) d\mathbf{z}_{t-1}. \tag{9}$$

At time t, the observation $\mathbf{I}_t$ is available, and the posterior distribution of $\mathbf{z}_t$ is updated using Bayes' rule as

$$p(\mathbf{z}_t | \mathbf{I}_{1:t}) = \frac{p(\mathbf{I}_t | \mathbf{z}_t) p(\mathbf{z}_t | \mathbf{I}_{1:t-1})}{p(\mathbf{I}_t | \mathbf{I}_{1:t-1})}. \tag{10}$$

Using a sequential importance sampling technique, we can approximate the posterior distribution $p(\mathbf{z}_t | \mathbf{I}_{1:t})$ by a set of N weighted samples (also called particles) $\{\mathbf{z}_t^i, w_t^i\}_{i=1,\ldots,N}$, where w_t^i are the importance weights of particles $\mathbf{z}_t^i$. Let $q(\mathbf{z}_t | \mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})$ be the importance distribution from which the particles are drawn, the importance weights w_t^i are updated as

Algorithm 2 Proposed Tracking Algorithm Using a Novel Appearance Modeling Technique

```
Input  : Image sequence $\{\mathbf{I}_t\}_{t=1:T}$, initial target state $\mathbf{z}_0$,
         target templates $\{\hat{\mathbf{I}}_0^j\}_{j=1:J}$, parameters $N$ and $\Sigma$
Output: The tracking results $\{\mathbf{z}_t^{i^*}\}_{t=1:T}$
Compute appearance representations $\{\hat{\mathbf{y}}_0^j\}_{j=1:J}$ for
template images $\{\hat{\mathbf{I}}_0^j\}_{j=1:J}$ using Algorithm 1;
for $t = 1, 2, \ldots$ do
    Sample a set of $N$ particle $\{\mathbf{z}_1^i\}_{i=1:N}$ from
    $\mathcal{N}(\mathbf{z}_{t-1}, \Sigma)$;
    Determine images $\{\mathbf{I}_t^i\}_{i=1:N}$ from target candidates
    paramerized using $\{\mathbf{z}_t^i\}_{i=1:N}$;
    Compute appearance representations $\{\mathbf{y}_t^i\}_{i=1:N}$ for
    $\{\mathbf{I}_t^i\}_{i=1:N}$ using Algorithm 1;
    Compute particle weights $\{w_t^i\}_{i=1:N}$ using Eq. (14);
    Obtain the tracking result at time $t$, $\mathbf{z}_t^{i^*}$ using Eq. (15);
    Update target templates using the procedure described
    in Section IV-C;
    Resample $N$ particles $\{\mathbf{z}_t^i\}_{i=1:N}$ with replacement from
    the current particle set according to probabilities
    $\{w_t^i\}_{i=1:N}$;
end
```

$$w_t^i = w_{t-1}^i \frac{p\left(\mathbf{I}_t | \mathbf{z}_t^i\right) p\left(\mathbf{z}_t^i | \mathbf{z}_{t-1}^i\right)}{q(\mathbf{z}_t | \mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})}. \tag{11}$$

To avoid the degeneracy case where the weights of some particles possibly keep increasing, particles are resampled according to their importance weights so as to generate a set of equally weighted particles. In the case of bootstrap filtering [68], the state transition distribution is chosen as the importance distribution $q(\mathbf{z}_t | \mathbf{I}_{1:t}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t | \mathbf{z}_{t-1}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where Σ is a diagonal matrix, and the weights are updated as the observation likelihood $w_t^i = p(\mathbf{I}_t | \mathbf{z}_t^i)$.

Particle filtering has found its applications in contour tracking [69]. Pérez *et al.* [70] used particle filtering for tracking a target parameterized by a rectangle region. The key step of particle filtering for visual tracking is to compute the weight for each particle using the observation likelihood. In practice, the observation likelihood $p(\mathbf{I}_t | \mathbf{z}_t^i)$ is computed as the similarity between the target template image and the target candidate image parameterized by the particle $\mathbf{z}_t^i$. In Section IV-B, we present how to combine our proposed appearance model with a multiscale pyramid matching to assign an appropriate weight to each particle.

B. Particle Weighting Based on the Proposed Appearance Model

To compute the weight w_i for the ith particle $\mathbf{z}_i$, we first crop a candidate image $\mathbf{I}_i$ parameterized by $\mathbf{z}_i$ from the current image.[1] Using the proposed appearance model, we can compute a set of feature descriptors

[1] The time index is omitted for readability.

$\{\mathbf{y}^i_{s,\upsilon,h} | s = 0, 1, 2, \upsilon = 1, 2, \ldots, 2^2, h = 1, 2, \ldots, 2^s\}$ from the candidate image $\mathbf{I}_i$. In addition, we maintain a set of J target templates $\{\hat{\mathbf{I}}_1, \hat{\mathbf{I}}_2, \ldots, \hat{\mathbf{I}}_J\}$. The feature descriptors of the jth target template are $\{\hat{\mathbf{y}}^j_{s,\upsilon,h} | s = 0, 1, 2, \upsilon = 1, 2, \ldots, 2^2, h = 1, 2, \ldots, 2^s\}$. The distance between the jth template $\hat{\mathbf{I}}_j$ and the candidate $\mathbf{I}_i$ is computed by summing the pairwise distance between their corresponding descriptors, shown as follows:

$$\rho(\mathbf{I}_i, \hat{\mathbf{I}}_j) = \sum_{s=0}^{2} \sum_{\upsilon=1}^{2^s} \sum_{h=1}^{2^s} \sum_{k=1}^{K} \frac{\left(\mathbf{y}^i_{s,\upsilon,h}(k) - \hat{\mathbf{y}}^j_{s,\upsilon,h}(k)\right)^2}{\mathbf{y}^i_{s,\upsilon,h}(k) + \hat{\mathbf{y}}^j_{s,\upsilon,h}(k)}. \tag{12}$$

Based on the above distance, we further define the similarity between the jth template $\hat{\mathbf{I}}_j$ and the candidate $\mathbf{I}_i$ as

$$\eta(\mathbf{I}_i, \hat{\mathbf{I}}_j) = \mathcal{D}(\rho(\mathbf{I}_i, \hat{\mathbf{I}}_j); 0, \sigma^2) \tag{13}$$

where $\mathcal{D}(x; \mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2. We fix $\mu = 0$ and σ^2 to 15 in all the experiments reported in this paper.

We use the weighted sum of the similarities between the candidate $\mathbf{I}_i$ and all the target templates $\{\hat{\mathbf{I}}_1, \hat{\mathbf{I}}_2, \ldots, \hat{\mathbf{I}}_J\}$ as the weight of the ith particle

$$w_i = \sum_{j=1}^{J} \pi_j \eta(\mathbf{I}_i, \hat{\mathbf{I}}_j) \tag{14}$$

where π_j is the weight associated with the jth target template image. All the target templates are initialized to have the same weights, and then, the weights are updated using the procedure that will be introduced in Section IV-C.

The tracking result in the current frame is the particle with the largest weight with an index computed as

$$i^* = \arg \max_{i=1,\ldots,N} w_i. \tag{15}$$

Please note that discriminative approaches, such as SVM or structured SVM, can be used for visual tracking with the proposed appearance model. However, in this paper, we still use distance matching [see (12)] due to its simplicity and efficiency.

C. Online Multiple Template Update

In order to capture the target's appearance variations during tracking, we maintain a set of target templates and update them online. For the purpose of dynamical updating, we assign a weight to each template, which reflects how much the template is similar to the tracked target. After the target has been tracked, the tracking result will be added to the template set, so that we capture the latest target appearance. However, we do not know whether the tracking result is correct. Therefore, to avoid the accumulation of errors, we only include the recent tracking results that are similar enough to the corresponding templates for template updating. The index used to indicate this similarity is represented by $j^* = \arg\max_j \eta(\mathbf{I}_{i^*}, \hat{\mathbf{I}}_j)$. We set a threshold τ, so that if the cosine angle between $\mathbf{I}_{i^*}$ and $\hat{\mathbf{I}}_{j^*}$ is less than the threshold, we then replace the Jth template $\hat{\mathbf{I}}_J$ using the available tracking result $\mathbf{I}_{i^*}$. The weight π_J of the Jth template is assigned to be the median of the previous weights of all the templates before the template updating starts. The weights of the remaining templates are updated as $\pi_j = \eta(\mathbf{I}_{i^*}, \hat{\mathbf{I}}_j)$, where $j = 1, 2, \ldots, J - 1$. After the updating, the weights of all the templates are then normalized to have the sum as one, and the templates are finally sorted in a descending order according to their normalized weights.

V. Experimental Results

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed method. We introduce the implementation details and the evaluation protocol in Sections V-A and V-B, respectively, including the parameter setup, baseline trackers, test sequences, and evaluation criteria. In Section V-C, we evaluate the proposed method when different layer architectures are used. The comparison results against the other baseline trackers are presented in Section V-D.

A. Implementation Details

For each sequence, the four corner points of the target are manually labeled in the first frame. Similar to [71], the state variable $\mathbf{z}_t$ consists of six affine transformation parameters. We set the variance matrix of the importance distribution $\Sigma = [0.005, 0.0005, 0.0005, 0.005, 3, 3]^T$ for all the sequences. By applying affine transformation using $\mathbf{z}_0$ as the initial parameters, we define the target template image as the first image frame and then duplicate it ten times to form the target template set. To learn the dictionary, we obtain a foreground patch set by densely extracting 8 × 8 local patches from the target template images with four pixels as the step length. The background patch set is obtained by sampling 8 × 8 patches around the target region. For all the experiments, we manually set the number of particles to be $N = 600$, the number of basis functions of the dictionaries $K_f = K_b = 64$, and the regularization parameter $\lambda = 0.01$ to achieve a tradeoff between the tracking accuracy and speed. The proposed algorithm is implemented in the MATLAB and runs around 4 frames/s on a PC with a 2.4-GHz Intel Core i5 processor and a 4-GB memory.

B. Experiment Protocols

Recently, a large scale benchmark library[2] for visual tracking was built by Wu *et al.* [1] and further extended [54], which contains the source codes of most state-of-the-art trackers and a large number of annotated test sequences. Most importantly, they propose robust evaluation metrics, which are used to reliably measure the performance of a tracker on different challenging scenarios. In this paper, we follow the evaluation protocols reported in the literature. In particular, we choose all the 100 test sequences from their data set including different tracking challenges, such as illumination variation, low-resolution, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, and background clutter.

To make the comparisons fair, we deliberately chose two state-of-the-art sparse coding-based trackers, including the L1 tracker using accelerated proximal gradient [72]

[2] http://visual-tracking.net

TABLE I

TRACKING SUCCESS RATES ON EIGHT TEST SEQUENCES BY THE PROPOSED TRACKER USING DIFFERENT PYRAMID SCALES AND DISCRIMINATIVE SPARSE CODING OR GENERIC SPARSE CODING. FOR EACH SEQUENCE, THE BEST RESULT IS HIGHLIGHTED IN RED

	$s = \{0\}$		$s = \{1\}$		$s = \{2\}$		$s = \{0, 1\}$		$s = \{0, 2\}$		$s = \{1, 2\}$		$s = \{0, 1, 2\}$	
car4	0.37	0.19	0.38	0.21	0.57	0.39	0.36	0.21	0.80	0.61	**0.89**	0.68	0.57	0.37
david_outdoor	0.28	0.21	0.32	0.25	**0.95**	0.85	0.75	0.59	0.69	0.59	0.76	0.57	0.44	0.34
david_indoor	0.36	0.19	0.33	0.24	**1.00**	0.79	0.36	0.21	0.77	0.61	0.85	0.64	0.69	0.44
sylv	0.86	0.72	0.56	0.47	0.62	0.52	**1.00**	0.77	0.79	0.65	0.98	0.72	0.96	0.79
lemming	0.18	0.11	0.18	0.12	**0.63**	0.40	0.19	0.13	0.35	0.21	0.45	0.35	0.57	0.41
box	0.71	0.62	0.86	0.77	**0.95**	0.85	0.92	0.79	0.63	0.50	0.69	0.58	0.92	0.72
basketball	0.21	0.14	0.38	0.21	**0.76**	0.58	0.42	0.31	0.55	0.46	0.52	0.37	0.47	0.35
faceocc2	0.82	0.65	0.75	0.59	0.70	0.53	0.83	0.69	**1.00**	0.79	0.84	0.75	0.83	0.62

TABLE II

TRACKING SUCCESS RATES ON THE EIGHT TEST SEQUENCES BY THE PROPOSED TRACKER USING DIFFERENT LAYER ARCHITECTURES. FOR EACH SEQUENCE, THE BEST RESULT IS HIGHLIGHTED IN RED

	basic	*basic*-$\mathcal{W}_{no}$	*basic*-$\mathcal{R}_{abs}$	*basic*-$\mathcal{R}_{sq}$	*basic*-$\mathcal{N}_{no}$	*basic*-$\mathcal{P}_{max}$
car4	**0.89**	0.60	0.73	0.71	0.80	0.81
david_outdoor	**0.95**	0.69	0.75	0.80	0.85	0.93
david_indoor	**1.00**	0.77	0.84	0.81	0.90	0.94
sylv	**1.00**	0.79	0.86	0.81	0.90	0.92
lemming	**0.63**	0.46	0.52	0.51	0.55	0.51
box	**0.95**	0.77	0.86	0.81	0.89	0.87
basketball	**0.76**	0.60	0.66	0.70	0.72	0.68
faceocc2	**1.00**	0.72	0.90	0.86	0.92	0.94

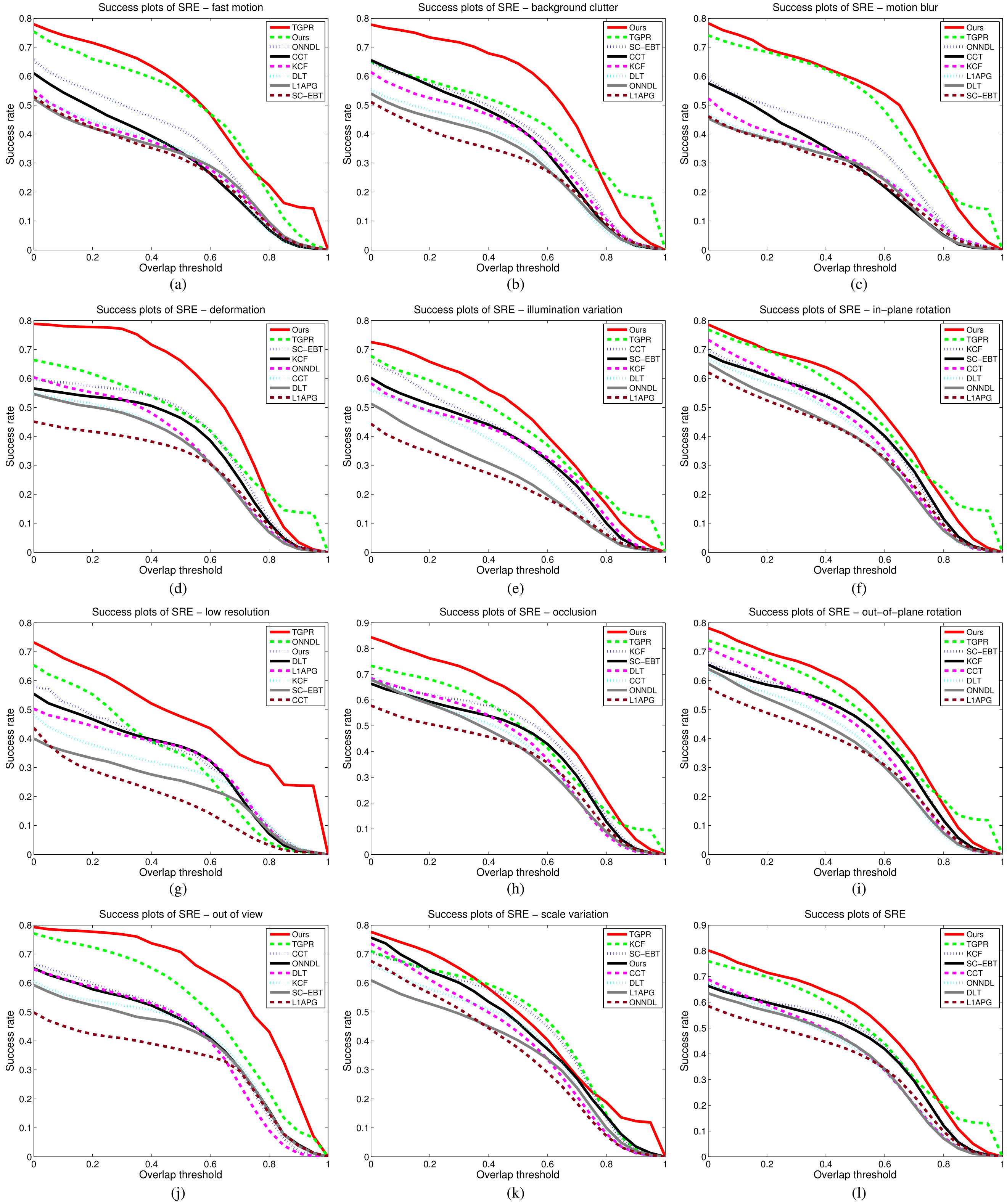
and the online robust nonnegative dictionary learning (ONNDL) [73]. We also selected five latest trackers, including the deep learning-based tracker [51], the complex cell-based tracker (CCT) [52], the self-correction ensemble-based tracking (SC-EBT) [74], the transfer learning-based visual tracking with Gaussian process regression (TGPR) [75], and the KCF-based tracker [6] for comparisons. For these trackers, we run the publicly available source codes on the benchmark in order to obtain their results. Please note that the results reported here may be of slight difference from the results reported in their original papers due to the utilization of parameters.

Two frame-based metrics widely used in tracking performance evaluation are: 1) center location error, which is defined as the Euclidean distance between the central location of the tracked target and the manually labeled ground-truthed position and 2) bounding box overlap, which is the ratio of the areas of the intersection and the union of the bounding box indicating the tracked subject and the ground-truthed bounding box. To measure the overall performance of a tracker on a test sequence, success rate and precision score are adopted. The former is computed as the percentage of image frames that have a bounding box overlap larger than a given threshold. The latter is the percentage of image frames that have a central position error less than a given threshold. In each case, when multiple thresholds are used, a curve is drawn to show how success rates or precision scores are affected by different thresholds. These curves are, namely, success plot and precious plot, respectively. In practical evaluations, we average the curves of a tracker over all the sequences that have the same challenge and show a curve for each challenge item rather than a test sequence. In addition, we use the area under curve (AUC) of the success plot to quantitatively measure the overall performance of a tracker on a challenge item.

The conventional way to evaluate trackers is to run a tracker throughout a test sequence with an initialization from the ground-truthed position in the first frame. However, we found that the initialization usually affects the performance of a tracker significantly. Therefore, it is necessary to test how robust a tracker is against different initialization states. Wu *et al*. [1] proposed two ways to analyze a tracker's robustness against initialization: temporal robustness evaluation that perturbs the initialization by starting a tracker at different frames and spatial robustness evaluation (SRE) that perturbs the initialization spatially by starting a tracker at different bounding boxes. In this paper, we adopt the SRE for all the comparison shown in this paper.

C. Comparison With Different Layer Architectures

We compare different layer architectures on eight test sequences. First of all, we study whether or not the discriminative sparse coding in the coding layer and the multiple scale pyramid representation in the pooling layer help to improve the tracking accuracy. To find out the benefits of using different pyramid scales, we design seven experiments with multiple pyramid scale $s = \{0\}$, $s = \{1\}$, $s = \{2\}$, $s = \{0, 1\}$, $s = \{0, 2\}$, and $s = \{0, 1, 2\}$. For each scale, to evaluate the benefits of using discriminative sparse coding, we compare the proposed tracker using the discriminative basis functions $\mathbf{V}_d$ and the one using the generic basis functions that are learned using local patches extracted from the target image via 3. The AUC of these experiments is shown in Table I, where for each scale, the left column indicates the results when the discriminative basis functions are used and the right column refers to the results when the generic basis functions are used. It clearly shows that for most of the test sequences, pyramid scale $s = \{2\}$ leads to the highest tracking success rates. In the meantime, $s = \{0\}$ (corresponding to no pyramid structure) has the worst results. These results justify that pyramid representation especially with a large scale helps to achieve good visual tracking accuracy. From Table I, we also observe that using the discriminative basis functions in the coding



(a) (b) (c)

(d) (e) (f)

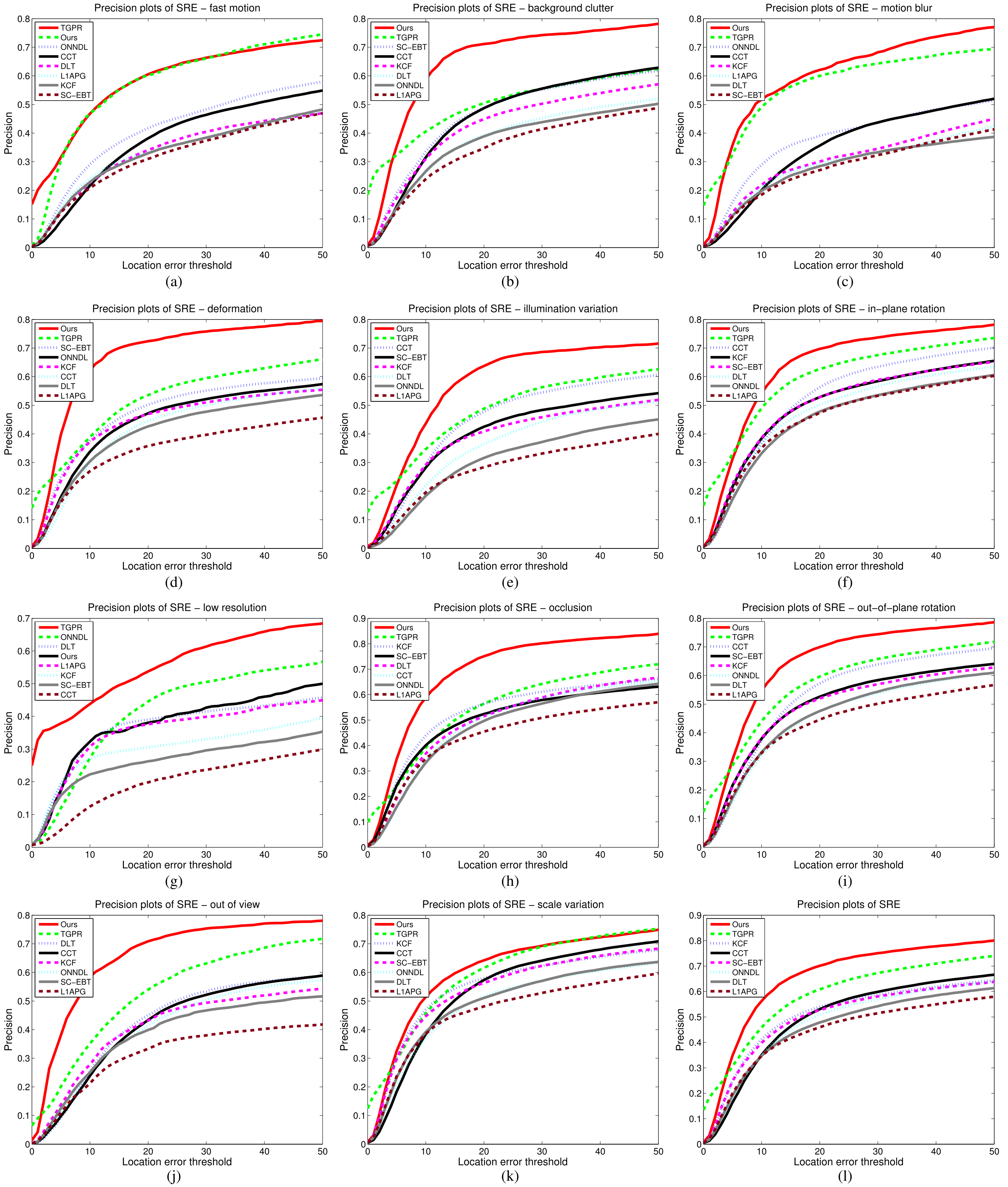
(g) (h) (i)

(j) (k) (l)

Fig. 3. Success plots for the challenges considered in this paper. (a) Fast motion. (b) Background clutter. (c) Motion blur. (d) Deformation. (e) Illumination variations. (f) In-plane rotation. (g) Low resolution. (h) Occlusion. (i) Out-of-plane rotation. (j) Out of view. (k) Scale variation. (l) Overall.

layer brings us better results than only using general basis functions.

We further investigate whether or not all the five layers are necessary for the proposed appearance model. We choose the architecture with all the five layers as the basic architecture and then remove or replace some of the operators used in the basic architecture to form different architectures. For example, if the whitening layer or the normalization layer is removed



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 4. Precision plots for the challenges considered in this paper. (a) Fast motion. (b) Background clutter. (c) Motion blur. (d) Deformation. (e) Illumination variations. (f) In-plane rotation. (g) Low resolution. (h) Occlusion. (i) Out-of-plane rotation. (j) Out of view. (k) Scale variation. (l) Overall.

from the basic architecture, we call it basic-$\mathcal{W}_{\text{no}}$ or basic-$\mathcal{N}_{\text{no}}$, respectively. If we replace the max operator in the rectification layer with an absolute or square operator, we call them basic-$\mathcal{R}_{\text{abs}}$ or basic-$\mathcal{R}_{\text{sq}}$. If we replace the sum operator in the pooling layer with a max operator, we call it basic-$\mathcal{P}_{\max}$. Table II shows the tracking success rates of such experiments based on the first eight test sequences, from which we can see that the basic architecture achieves the best accuracy on all

TABLE III

AUC Comparison of the Studied Trackers per Each Challenging Scenario. Note That the Last Row Is Not Simply the Average of the Overall Rows. For Each Sequence, the Best Result Is Highlighted in Red

	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Occlusion	0.514	0.339	0.363	0.386	0.372	0.400	0.424	0.443
Illumination variation	0.428	0.215	0.242	0.292	0.341	0.331	0.328	0.400
Scale variation	0.412	0.342	0.341	0.376	0.377	0.428	0.438	0.454
Background clutter	0.512	0.274	0.295	0.304	0.356	0.373	0.345	0.425
Deformation	0.522	0.280	0.341	0.316	0.320	0.385	0.358	0.422
Fast motion	0.454	0.283	0.350	0.288	0.303	0.275	0.288	0.487
Motion blur	0.492	0.255	0.328	0.244	0.275	0.242	0.267	0.477
In-plane rotation	0.472	0.339	0.342	0.363	0.385	0.397	0.398	0.467
Out-of-plane rotation	0.467	0.316	0.334	0.345	0.382	0.390	0.387	0.448
Out of view	0.589	0.309	0.394	0.382	0.398	0.365	0.380	0.492
Low resolution	0.312	0.297	0.314	0.0.304	0.0.179	0.222	0.260	0.458
Overall	0.485	0.335	0.360	0.355	0.366	0.398	0.3405	0.466

TABLE IV

Precious Scores for the Studied Trackers per Each Challenging Scenario. Note That the Precious Score Is Computed for the Threshold Value of 20. For Each Sequence, the Best Result Is Highlighted in Red

	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Occlusion	0.750	0.454	0.496	0.515	0.513	0.524	0.563	0.564
Illumination variation	0.637	0.284	0.316	0.364	0.478	0.425	0.409	0.488
Scale variation	0.642	0.481	0.515	0.511	0.573	0.563	0.580	0.616
Background clutter	0.711	0.348	0.389	0.393	0.487	0.488	0.449	0.505
Deformation	0.724	0.358	0.472	0.426	0.446	0.502	0.469	0.535
Fast motion	0.602	0.334	0.418	0.0.341	0.382	0.310	0.329	0.605
Motion blur	0.621	0.293	0.391	0.284	0.356	0.271	0.301	0.600
In-plane rotation	0.697	0.473	0.478	0.502	0.563	0.529	0.529	0.627
Out-of-plane rotation	0.700	0.446	0.476	0.476	0.573	0.526	0.520	0.596
Out of view	0.709	0.333	0.416	0.450	0.434	0.399	0.431	0.539
Low resolution	0.381	0.376	0.445	0.391	0.198	0.263	0.306	0.538
Overall	0.701	0.459	0.508	0.478	0.531	0.527	0.539	0.608

the test sequences. When the whitening layer is removed, the tracking success rates on all the sequences witness significant drops. The reason is that the whitening operator helps speed up the convergence of the dictionary learning in the coding layer and if it is removed, then the system accuracy degrades. From the comparison of using different rectification operators, we observe that the max operator outperforms both the absolute and square operators. As explained in Section III (C) , the max operator increases the sparsity of the codes obtained in the coding layer and hence reduces the influence of the background. Finally, we also see that the sum pooling-based method outperforms the max pooling-based method.

D. Comparison Against Different Methods

We compare the proposed method with seven trackers on all the 100 sequences used in [54]. Figs. 3 and 4 show the success plots and precious plots for all the involved trackers over all the test sequences containing a specific challenge (e.g., fast motion and occlusion) as well as the overall success plots and precious plots over all the test sequences. Fig. 3(a)–(f) and (h)–(j) shows the success plots of the studied trackers for the challenges of fast motion, background clutter, motion blur, deformations, illumination variations, in-plane rotation, occlusions, out-plane rotation, and out of view, respectively. Our tracker significantly outperforms the other trackers with different thresholds.

Low resolution, fast motion, and scale variation are particularly challenging for visual tracking. As shown in Fig. 3(a), (g), and (k) [and also Fig. 4(a) and (g)], our tracker is slightly worse than TGPR, which may be caused by our dictionary learning methods on sampled patches with size 8 × 8. For low resolution or blurred videos caused by fast motion, the sampled patches will contain a few information that may cause the learned dictionaries less discriminative.

In Table III, we provide the AUC values for all the involved trackers per each challenging scenario and the overall performance on the whole data set. As we can see, our tracker achieved the best AUC in 9 out of 11 challenges. In terms of the overall AUC, our tracker outperforms all the other trackers. Precious scores of the studied trackers are shown in Table IV, from which we can see that our tracker achieved the best AUC in 10 out of 11 challenges and the best overall AUC.

1) Comparison on Tracking Speed: To compare the computational efficiency, we compute the time of running the compared trackers on the *car4* test sequence and reported the tracking speeds (frames/s) in Table V. From Table V, we can see that our tracker can achieve about 4 frames/s. Although it is not real time, it is still faster than the ONNDL tracker and the SC-EBT tracker. With code optimization in C++ or running on graphics processing unit, our tracker has potential to achieve real-time tracking. To see the computational cost of each step of our tracker, we also reported the running time of each step of our tracker on the *car4* sequence. From Table VI, we can see that the most computational expensive step is the coding step.

2) Discussion: A number of biologically inspired appearance models that simulate the hierarchical pathway in visual

TABLE V

AVERAGE TRACKING SPEED (IN frames/s) TESTED ON THE *Car4* SEQUENCE

	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Tracking speed	4	25	0.5	5	9	0.8	120	0.59

TABLE VI

RUNNING TIME (IN s) OF EACH STEP TESTED ON THE *Car4* SEQUENCE

	Whitening	Coding	Rectification	Normalization	Pooling	distance computation
Running time (in second)	20.83	87.56	12.16	18.89	15.92	23.26

cortex have been extensively studied in the last two decades, and these existing appearance models were mainly developed for object recognition. The challenge in object recognition is to recognize the same class of objects in the presence of different object transformations, such as translation, scale, and pose/view/illumination changes. To achieve the tasks, the appearance models for object recognition are required to be invariant to translation, scale, view, and illumination changes. Visual tracking indeed is a recognition problem but it has its own characteristics. For example, in a standard particle filter-based tracker, the appearance model has not considered scale changes, and people can take advantage of the temporal consistence constraint to identify the size changes of the target. This makes appearance models used for visual tracking different from those used for object recognition. In particular, visual tracking is a binary classification problem. In this paper, motivated by the success of using the biologically inspired appearance models for object recognition, we developed a biologically inspired appearance model for visual tracking with an emphasis on both discriminativity and appearance invariance. We take advantage of the established hierarchical architecture in visual cortex while introducing the discriminative sparse coding and spatial pyramid representation to the tracking system. Although the proposed appearance model is evaluated in the context of visual tracking, the proposed framework is easy to be extended to other multiple-class classification tasks with well-aligned samples.

VI. CONCLUSION AND FUTURE WORK

In this paper, motivated by the success of the hierarchical organization of the primary visual cortex in object recognition, we have proposed a biologically inspired appearance model for robust visual tracking. The proposed appearance model consists of five layers, simulating the visual information processing pathway from retina to complex cells of the primary visual cortex. Different from the existing biologically inspired appearance models reported in the literature, which overemphasize the invariance to translation, scale and view changes while ignoring the discriminative ability, the proposed model made a good compromise between the discriminative ability and its robustness against appearance variations. Experimental results show that the proposed model is reasonable and achieved promising performance. Comparison results against several state-of-the-art tracking methods also validated the effectiveness of the proposed method.

The proposed appearance modeling method was inspired by biological findings in object recognition of still images and achieved desired tracking accuracy for video applications. This paper disregarded the dynamic nature of cortical computation and the representation of cortical information over time. In the future work, we will develop learning tools to model the temporal relationship between cortical responses, which may further improve the tracking accuracy.

REFERENCES

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[2] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 46, no. 7, pp. 1772–1788, Jul. 2013.

[3] H. Zhou, M. Fei, A. Sadka, Y. Zhang, and X. Li, "Adaptive fusion of particle filtering and spatio-temporal motion energy for human tracking," *Pattern Recognit.*, vol. 47, no. 11, pp. 3552–3567, 2014.

[4] S. Zhang, H. Zhou, H. Yao, Y. Zhang, K. Wang, and J. Zhang, "Adaptive NormalHedge for robust visual tracking," *Signal Process.*, vol. 110, pp. 132–142, May 2015.

[5] X. Yang, M. Wang, and D. Tao, "Robust visual tracking via multi-graph ranking," *Neurocomputing*, vol. 159, pp. 35–43, Jul. 2015.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[7] L. Zhao, X. Gao, D. Tao, and X. Li, "Tracking human pose using max-margin Markov models," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5274–5287, Dec. 2015.

[8] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1749–1760, Nov. 2015.

[9] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[10] L. Zhao, X. Gao, D. Tao, and X. Li, "Learning a tracking and estimation integrated graphical model for human pose tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3176–3186, Dec. 2015.

[11] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, "Robust multitask multiview tracking in videos," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2874–2890, Nov. 2015.

[12] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 749–758.

[13] D. Tao, L. Jin, Y. Wang, and X. Li, "Person reidentification by minimum classification error-based KISS metric learning," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 242–252, Feb. 2015.

[14] J. Pan, X. Li, X. Li, and Y. Pang, "Incrementally detecting moving objects in video with sparsity and connectivity," *Cognit. Comput.*, vol. 8, no. 3, pp. 420–428, 2016.

[15] M. A. A. Aziz, J. Niu, X. Zhao, and X. Li, "Efficient and robust learning for sustainable and reacquisition-enabled hand tracking," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 945–958, Apr. 2016.

[16] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

[17] J. Wen, X. Gao, Y. Yuan, D. Tao, and J. Li, "Incremental tensor biased discriminant analysis: A new color-based visual tracking method," *Neurocomputing*, vol. 73, nos. 4–6, pp. 827–839, 2010.

[18] B. Babenko, M.-S. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 983–990.

[19] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1313–1320.

[20] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1822–1829.

[21] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[22] X. Lu, Y. Yuan, and P. Yan, "Robust visual tracking with discriminative sparse learning," *Pattern Recognit.*, vol. 46, no. 7, pp. 1762–1771, Jul. 2013.

[23] J. Fang, Q. Wang, and Y. Yuan, "Part-based online tracking with geometry constraint and attention selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 854–864, May 2014.

[24] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, vol. 131, pp. 227–236, 2014.

[25] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic, "Efficient online subspace learning with an indefinite kernel for visual tracking and recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1624–1636, Oct. 2012.

[26] Z. Xiao, H. Lu, and D. Wang, "L2-RLS-based object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1301–1309, Aug. 2014.

[27] F. Yang, H. Lu, and M.-H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 242–254, Feb. 2014.

[28] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

[29] X. Liu, D. Tao, M. Song, L. Zhang, J. Bu, and C. Chen, "Learning to track multiple targets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1060–1073, May 2015, doi: 10.1109/TNNLS.2014.2333751.

[30] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[33] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[34] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[35] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1–8.

[36] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, 1962.

[37] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[38] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[39] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[40] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 994–1000.

[41] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?" *PLoS Comput. Biol.*, vol. 4, no. 1, pp. 25–45, 2008.

[42] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. 12th Int. Conf. Comput. Vis.*, 2009, pp. 2146–2153.

[43] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183–193, 1954.

[44] H. B. Barlow, *Possible Principles Underlying the Transformations of Sensory Messages*. Cambridge, MA, USA: MIT Press, 1961.

[45] D. I. Perrett and M. W. Oram, "Neurophysiology of shape processing," *Image Vis. Comput.*, vol. 11, no. 6, pp. 317–333, 1993.

[46] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2009, pp. 1794–1801.

[47] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 524–531.

[48] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[49] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 921–928.

[50] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 263–270.

[51] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[52] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 1113–1120.

[53] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[54] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[55] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[56] S. Zhang, H. Yao, and S. Liu, "Robust visual tracking using feature-based visual attention," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 1150–1153.

[57] Q. Wang, F. Chen, W. Xu, and M.-S. Yang, "Online discriminative object tracking with local sparse representation," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2012, pp. 425–432.

[58] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.

[59] R. J. Douglas, C. Koch, M. Mahowald, K. A. Martin, and H. H. Suarez, "Recurrent excitation in neocortical circuits," *Science*, vol. 269, no. 5226, pp. 981–985, 1995.

[60] M. Carandini and D. J. Heeger, "Summation and division by neurons in primate visual cortex," *Science*, vol. 264, pp. 1333–1336, May 1994.

[61] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[62] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[63] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[64] P. Bush and T. Sejnowski, *The Cortical Neuron*. London, U.K.: Oxford Univ. Press, 1995.

[65] S. Lyu, "Dependency reduction with divisive normalization: Justification and effectiveness," *Neural Comput.*, vol. 23, no. 11, pp. 2942–2973, 2011.

[66] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 45–57, 2008.

[67] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. 2006, pp. 2169–2178.

[68] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[69] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[70] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 661–675.

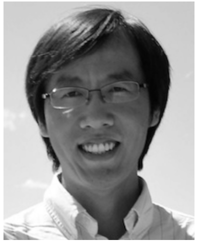
[71] X. Mei, H. Ling, and D. W. Jacobs, "Sparse representation of cast shadows via ℓ_1-regularized least squares," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 583–590.

[72] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1830–1837.

[73] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 657–664.

[74] N. Wang and D.-Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1107–1115.

[75] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.



Shengping Zhang (M'13) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013.

He had been a Post-Doctoral Research Associate with Brown University, Providence, RI, USA, and a Visiting Student Researcher with the University of California at Berkeley, Berkeley, CA, USA. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai, China. He has authored or co-authored over 30 research publications in refereed journals and conferences. His current research interests include sparse coding and its applications in computer vision.

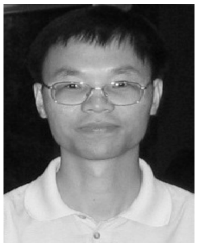
Dr. Zhang is an Associate Editor of *Signal Image and Video Processing*.

Xiangyuan Lan (S'14) received the B.Eng. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong.

He was a Visiting Scholar with the Computer Vision Laboratory, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA, in 2015. His current research interests include computer vision and pattern recognition, particularly, feature fusion and sparse representation for visual tracking.

Hongxun Yao received the B.S. and M.S. degrees from Harbin Engineering University, Harbin, China, in 1987 and 1990, respectively, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, in 2003, all in computer science.

She is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. She has authored five books and over 200 scientific papers. Her current research interests include computer vision, multimedia computing, and human–computer interaction.



Huiyu Zhou was with Guangxi Medical University, Nanning, China, Elscint Ltd., Haifa, Israel, the University of Essex, Colchester, U.K., the University of London, London, U.K., and Brunel University, Uxbridge, U.K. He has taken part in the consortiums of a number of research projects in medical image processing, computer vision, intelligent systems, and data mining. He is currently a Lecturer with the Queen's University of Belfast, Belfast, U.K. He has published over 110 peer-reviewed papers in the field.

Mr. Zhou was a recipient of the CVIU 2012 Most Cited Paper Award and was shortlisted for MBEC 2006 Nightingale Prize. He also won one of the best paper awards in the 1993 Annual Conference of China Association for Medical Devices Industry.

Dacheng Tao (F'15) is currently a Professor of Computer Science with the Centre for Quantum Computation & Intelligent Systems, and the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He mainly applies statistics and mathematics to data analytics. His research results have expounded in one monograph and over 100 publications at prestigious journals and prominent conferences, such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Image Processing, the *Journal of Machine Learning Research*, the *International Journal of Computer Vision*, the Conference on Neural Information Processing Systems, the International Conference on Machine Learning, the Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, the International Conference on Artificial Intelligence and Statistics, the International Conference on Data Mining (ICDM), and the ACM Special Interest Group on Knowledge Discovery and Data Mining. His current research interests include computer vision, data science, image processing, machine learning, neural networks, and video surveillance.

Dr. Tao received several best paper awards, such as the Best Theory/Algorithm Paper Runner Up Award in the IEEE ICDM'07, the Best Student Paper Award in the IEEE ICDM'13, and the 2014 ICDM 10 Year Highest-Impact Paper Award.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, China.