Robust Visual Tracking via Exclusive Context Modeling

Tianzhu Zhang, *Member, IEEE,* Bernard Ghanem, *Member, IEEE,* Si Liu, *Member, IEEE,* Changsheng Xu, *Fellow, IEEE,* and Narendra Ahuja, *Fellow, IEEE*

***Abstract*—In this paper, we formulate particle filter-based object tracking as an exclusive sparse learning problem that exploits contextual information. To achieve this goal, we propose the context-aware exclusive sparse tracker (CEST) to model particle appearances as linear combinations of dictionary templates that are updated dynamically. Learning the representation of each particle is formulated as an exclusive sparse representation problem, where the overall dictionary is composed of multiple group dictionaries that can contain contextual information. With context, CEST is less prone to tracker drift. Interestingly, we show that the popular L_1 tracker [1] is a special case of our CEST formulation. The proposed learning problem is efficiently solved using an accelerated proximal gradient method that yields a sequence of closed form updates. To make the tracker much faster, we reduce the number of learning problems to be solved by using the dual problem to quickly and systematically rank and prune particles in each frame. We test our CEST tracker on challenging benchmark sequences that involve heavy occlusion, drastic illumination changes, and large pose variations. Experimental results show that CEST consistently outperforms state-of-the-art trackers.**

***Index Terms*—Contextual information, exclusive sparse learning, particle filter, tracking.**

I. Introduction

VISUAL tracking is important for automatic surveillance, robotics, human computer interaction, etc. In real-world scenarios, it is very challenging due to the existence of several sources of appearance variations such as occlusion, pose variation, abrupt motion, varying viewpoints, varying lighting conditions, and cluttered background as shown in Fig. 1(a). Over the years, many trackers have been proposed to overcome these challenges, and more details can be found in [2]–[4].

Recently, sparse representation has been successfully applied to visual tracking [1], [5]–[10]. In this case, the tracker represents each target candidate as a sparse linear combination of dictionary templates that can be dynamically updated to maintain an up-to-date target appearance model. This representation has been shown to be robust against partial occlusions, thus, leading to improved tracking performance. However, sparse coding-based trackers perform computationally expensive ℓ_1 minimization at each frame. In a particle filter framework [11], computational cost grows linearly with the number of sampled particles. It is this computational bottleneck that precludes the use of these trackers in real-time scenarios. Consequently, efforts have been made recently to speed up this tracking paradigm [5], [7], [9], [12]–[14]. In addition, these methods focus on building a sparse model to encode the variations of object appearance without considering contextual information (background or other objects) as shown in Fig. 1(b). This renders the tracker more prone to drifting from the target, especially in cases of significant target appearance change and cluttered background. In fact, the issue of tracker drift is a common problem faced in visual tracking, where the representation model of a tracker is unable to persistently describe the changing appearance of a target over time. This inability to represent the target precisely might force the tracker to incorporate more background information in the target's representation, which in turn leads the tracker to drift gradually from the target into the background over time.

Inspired by the above work, we develop a computationally efficient, sparse learning tracker that exploits context information. We generate particles using Gaussian noise models around particles sampled in the previous frame. The next target state is selected to be the particle sample that is represented the best by a dictionary of target templates and poorly by templates from the target's context. As such, we devise an accurate, robust, and discriminative particle representation by making the following considerations.

1) The best particle should have a particle representation that is more similar with the target templates than the context at each frame. In order to handle target appearance changes, these target templates (as well as the context) should be updated dynamically.

Manuscript received August 25, 2014; revised December 8, 2014 and January 1, 2015; accepted January 5, 2015. Date of publication February 9, 2015; date of current version December 14, 2015. This work was supported in part by the Advanced Digital Sciences Center, Singapore's Agency for Science, Technology and Research, under a Research Grant for the Human Sixth Sense Programme. Changsheng Xu was supported in part by the National Program on Key Basic Research Project (973 Program) under Project 2012CB316304 and the National Natural Science Foundation of China under Grant 61225009. This paper was recommended by Associate Editor H. Lu.

T. Zhang is with Advanced Digital Sciences Center, Singapore 138632, and also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: tzzhang10@gmail.com).

B. Ghanem is with the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, 23955-6900, and also with Advanced Digital Sciences Center, Singapore 138632.

S. Liu is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100190, China.

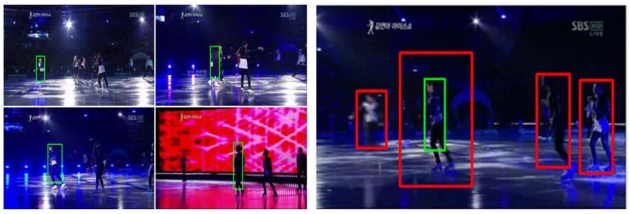
C. Xu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

N. Ahuja is with the Coordinated Science Laboratory, Department of Electrical and Computer Engineering, Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

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/TCYB.2015.2393307

2168-2267 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) (b)

Fig. 1. (a) Frames from a *Skating* sequence. The ground truth is designated in green. (b) Context information of the object (green box) with respect to background (between red and green boxes) and other objects (other red boxes).

2) Since occlusion and noise significantly impact tracking performance, representation error should be incorporated explicitly in the tracking process.
3) Particle representations should encode the target's appearance while also distinguishing it from its context (background or other objects).

Discriminating the target from its context adds another layer of robustness against tracker drift. Generally, a "good" target candidate is effectively represented by the target and not the context templates, thus, leading to a sparse representation. The contrary is true for a "bad" target candidate. In this sense, each particle's sparse representation should be exclusive to either the target or context templates. As compared to other sparse trackers [1], [5]–[7], we exploit context information as well as target appearance to guide tracking. For simplicity, we denote our tracker as the context-aware exclusive sparse tracker (CEST).

A. Contributions

The contributions of this paper are threefold.

1) We propose an exclusive sparse learning method for object tracking, which makes use of context information for more robust performance. To the best of our knowledge, this is the first work to exploit context information through exclusive sparsity in object tracking.
2) Compared with the popular L_1 tracker [1], our CEST exploits context information and can yield the L_1 tracker as a special case.
3) We adopt a coarse-to-fine scheme to make CEST computationally attractive, especially compared to the L_1 tracker.

First, we use the structural properties inherent to the dual of the exclusive sparse learning problem to quickly rank and prune particles. Then, an efficient accelerated proximal gradient (APG) method is used to compute the representations of the particles that are not pruned. This pruning scheme can also be used for exclusive sparse problems outside the domain of object tracking.

II. Related Work

Visual tracking is an important topic in computer vision and it has been studied for several decades. There is extensive literature, and we only briefly review techniques that are most related to ours, including a brief overview of prior work in generic object tracking, as well as, sparse representation, and context information. For a more thorough survey of tracking methods, we refer the readers to [2].

A. Object Tracking

In general, visual tracking methods can be categorized into two groups: 1) generative and 2) discriminative.

1) Generative Visual Tracking: Generative tracking methods adopt an appearance model to describe the target observations, and the aim is to search for the target location that has the most similar appearance to this model. Examples of generative methods include eigentracker [15], context-aware tracker [16], incremental tracker (IVT) [17], fragment-based tracker (Frag) [18], and visual tracking decomposition (VTD) tracker [19]. In [15], a view-based representation is used for tracking rigid and articulated objects. The approach builds on and extends work on eigenspace representations, robust estimation techniques, and parameterized optical flow estimation. The context-aware tracker [16] considers context information in the scene for more robust tracking. Specifically, this method integrates into the tracking process a set of auxiliary objects that are automatically discovered in the video on the fly by data mining. The IVT tracker [17] seeks an adaptive appearance model that accounts for appearance variation of rigid or limited deformable motion. Although it has been shown to perform well when the target object undergoes lighting and pose variation, this method is less effective in handling heavy occlusion or nonrigid distortion as a result of the adopted holistic appearance model.

2) Discriminative Visual Tracking: Discriminative tracking methods formulate object tracking as a binary classification problem, which aims to find the target location that can best distinguish the target from the background. Examples of discriminative methods are on-line boosting (OAB) [20] ensemble tracking [21], and online multiple instance learning tracking [22]. In the OAB tracker [20], online AdaBoost is adopted to select discriminative features for object tracking. Its performance is affected by background clutter and can easily drift. The ensemble tracker [21] formulates the task as a pixel-based binary classification problem. Although this method is able to differentiate between target and background, the pixel-based representation is rather limited and thereby limits its ability to handle occlusion and clutter. Moreover, the multiple instance learning (MIL) tracker [22] extends multiple instance learning to an online setting for object tracking. Although it is able to address the problem of tracker drift, this method does not handle large nonrigid shape deformation well. In [23], a target confidence map is built by finding the most discriminative RGB color combination in each frame. Also, a hybrid approach that combines a generative model and a discriminative classifier is proposed in [24] to capture appearance changes and allow reacquisition of an object after total occlusion. Also, global mode seeking can be used to detect and reinitialize the tracked object after total occlusion [25].

B. Sparse Representation for Object Tracking

Recently, sparse linear representation based on the particle filter framework has been introduced to object

tracking and has been shown to achieve significant tracking performance [1], [5], [7]–[9], [13], [26]–[29]. In the L_1 tracker [1], a tracking candidate is represented as a sparse linear combination of object templates and trivial templates. Sparse representation is computed by solving a constrained ℓ_1 minimization problem with nonnegativity constraints to solve the inverse intensity pattern problem during tracking. The results show good performance at a high computational expense due to the ℓ_1 minimization. In fact, the computational cost grows proportionally with the number of particle samples [30], [31]. In [5], an efficient L_1 tracker with minimum error bound and occlusion detection is proposed. The minimum error bound is quickly calculated from a linear least squares equation, and serves as a guide for particle resampling in a particle filter framework. Without loss of precision during resampling, the most insignificant samples are removed before solving the computationally expensive ℓ_1 minimization problem. In [32], the original L_1 tracker is improved by using principal component analysis subspace as target templates. In [27], dynamic group sparsity is integrated into the tracking problem and high dimensional image features are used to improve tracking robustness. In [9], dimensionality reduction and a customized orthogonal matching pursuit algorithm are adopted to accelerate the L_1 tracker [1]. However, this method may reduce the tracking performance sometimes [9]. In [8], a very fast numerical solver based on the APG approach is developed to solve the ℓ_1 norm minimization problem with guaranteed quadratic convergence. The APG method is also used in [33] to solve the sparse representation for visual tracking. In [13], compressive sensing theory is adopted for real-time tracking. In [34], a fast tracking algorithm is proposed to handle partial occlusion in the tracking problem. In [29], a sparsity-based discriminative classifier and a sparsity-based generative model are designed for tracking. Different from [29], which adopts the background information to select the discriminative features for tracking, our proposed tracker uses the background information via the exclusive sparse learning algorithm. Zhang *et al*. [7], [35] proposed a multitask learning approach to jointly learn the particle representations for robust object tracking. Our proposed method is inspired by the success of these ℓ_1 minimization-based trackers, and we will also adopt the sparsity property for robust tracking.

Different from previous sparse trackers [1], [7], in this paper, we use exclusive sparse model to exploit contextual information to improve visual tracking. The exclusive sparse model has been used for feature selection [36] and multilabel image classification [37]. In [37], the exclusive sparse model is defined as solving a $\ell_{1,2}$-regularized least squares problem, and it encourages that variables in the same group are exclusively selected in the output. In this paper, motivated by the exclusive property between target templates and context templates, we will use the exclusive sparse model to exploit the contextual information to improve visual tracking.

C. Context Information for Object Tracking

Context information has been applied actively in object detection [38], object classification [39], and object recognition [40]. It has been employed recently in several successful tracking methods [16], [41]–[43]. The improved performance of these trackers is attributed to the use of context information in determining the target location. Our proposed method using context information in image domain is inspired by the work above. In this paper, a particle is represented as a sparse linear combination of dictionary templates that are exclusive to either the target or its context. Our tracker is generic, as it can incorporate the forms of context information previously used [16], [41], [42]. For the proposed method, the representation of a target candidate is obtained by efficiently solving an exclusive sparse learning problem.

III. Context-Aware Exclusive Sparse Tracker

In this section, we give a detailed description of our particle filter-based tracking method, which makes use of context information in an exclusive sparse learning framework to represent particle samples. Similar to [1], we assume an affine motion model between consecutive frames. Therefore, the state variable $\mathbf{s}_t$ consists of the six parameters of an affine transformation, consisting of a 2-D linear transformation and a 2-D translation. By applying an affine transformation using $\mathbf{s}_t$ as parameters, we crop the region of interest $\mathbf{x}_t$ from the image and normalize it to the size of the target templates in our dictionary. The state transition distribution $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is modeled to be Gaussian with the components of $\mathbf{s}_t$ assumed independent. The observation model $p(\mathbf{x}_t|\mathbf{s}_t)$ reflects the similarity between a target candidate (particle) and dictionary templates. In this paper, $p(\mathbf{x}_t|\mathbf{s}_t)$ is computed as a function of the reconstruction error obtained by linearly representing $\mathbf{x}_t$ using the target template dictionary. The particle that maximizes this function is selected to be the tracked target at each time instance. Fig. 2 shows the basic idea of our proposed method and how the context information is enforced in the proposed tracking algorithm. Next, we will show how to represent particles using the exclusive sparse learning framework.

A. Representation of Particle Sample

In our particle filter-based tracking method, particles are randomly sampled around the previous states according to zero-mean Gaussian distributions. In the tth frame, we sample n particles, where the observation (pixel color values) of the ith particle is denoted in vector form as: $\mathbf{x}_i \in \mathbb{R}^d$, where d is the dimension of $\mathbf{x}_i$. Each $\mathbf{x}_i$ is represented as a sparse linear combination $\mathbf{z}_i$ of m dictionary templates $\mathbf{D} \in \mathbb{R}^{d\times m}$, as shown in (1). $\mathbf{D}$ is updated dynamically to handle frame-to-frame changes in target appearance. The exact update process is described later. We define three types of templates $\mathbf{D}_F$, $\mathbf{D}_O$, and $\mathbf{D}_C$, which incorporate information about the target, noise/occlusion, and context, respectively. We model $\mathbf{D}$ as a concatenation of $|\mathcal{G}| = G$ predefined groups of dictionary templates, indexed by the set $\mathcal{G}$. Each group in $\mathcal{G}$ contains templates from each of the three types of templates. For example, the templates in the gth group is defined as $\mathbf{D}_g = \left[\mathbf{D}_F^g\ \mathbf{D}_O^g\ \mathbf{D}_C^g\right]$ and $g = 1, \ldots, G$. Here, $\mathbf{D}_F^g$, $\mathbf{D}_O^g$, and $\mathbf{D}_C^g$ are a subset of $\mathbf{D}_F$, $\mathbf{D}_O$, and $\mathbf{D}_C$, respectively. Defining the set

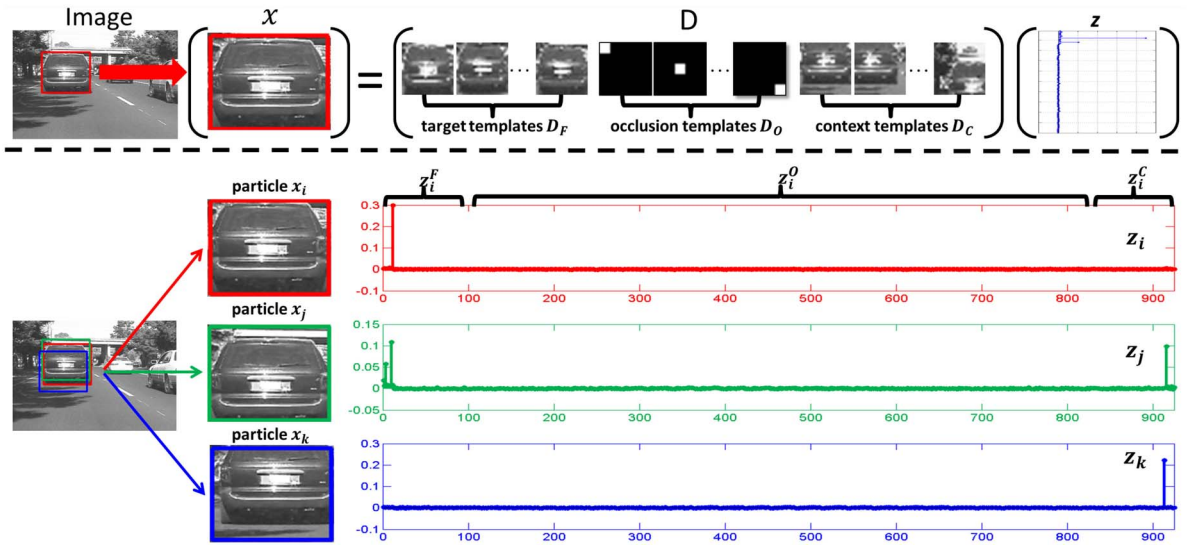


Fig. 2. Schematic example of CEST. Representation $\mathbf{z}$ of particle $\mathbf{x}$ with respect to dictionary $\mathbf{D}$ is computed by solving (2). Note that $\mathbf{z}$ is exclusively sparse in general. Particle $\mathbf{x}_i$ is selected among all particles as the tracking result, since $\mathbf{x}_i$ is represented the best by $\mathbf{D}_F$.

$\mathcal{G}$ and its effect on tracking performance will be addressed in Sections V-B and V-C3, respectively

$$\mathbf{x} = \sum_{g \in \mathcal{G}} \mathbf{D}_g \mathbf{z}^g = \mathbf{D}\mathbf{z}; \quad \text{with} \quad \begin{cases} \mathbf{D} = [\mathbf{D}_F \ \mathbf{D}_O \ \mathbf{D}_C] \\ \mathbf{z} = \left[\mathbf{z}^F; \mathbf{z}^O; \mathbf{z}^C\right]. \end{cases} \tag{1}$$

The three dictionaries ($\mathbf{D}_F$, $\mathbf{D}_O$, and $\mathbf{D}_C$) focus on complimentary aspects of particle representation: 1) accurate target reconstruction; 2) occlusion/noise handling; and 3) discrimination from context. We discuss these issues next.

1) Target Reconstruction ($\mathbf{D}_F$)*:* To reliably represent the appearance of the target, we construct dictionary $\mathbf{D}_F$ from target templates, which are visual observations of the tracked object possibly under a variety of appearance changes. Since our representation is constructed at the pixel level, misalignment between dictionary templates and particles might lead to degraded performance. To alleviate this problem, one of two strategies can be employed.

1) $\mathbf{D}_F$ can be constructed from a dense spatial sampling of the target object, as well as, transformed versions of these samples.
2) Each $\mathbf{x}$ can be aligned to columns of $\mathbf{D}_F$ as in [44].

In this paper, we employ the first strategy, which leads to a larger m but a lower overall computational cost. In the noiseless case, the target at a given frame is the particle $\mathbf{x}$ that is represented the best by only a few templates in $\mathbf{D}_F$. This leads to a sparse and robust target representation.

2) Occlusion/Noise Handling ($\mathbf{D_O}$)*:* In many tracking scenarios, target objects are often corrupted by noise or partially occluded. As in [1], this noise is modeled as sparse additive noise that can take on large values anywhere in its sparse support. Therefore, in the presence of occlusion/noise, we can still represent $\mathbf{x}$ as a linear combination of dictionary templates, so long as $\mathbf{D}_F$ is augmented with occlusion templates $\mathbf{D}_O$, e.g., identity of $\mathbb{R}^{d \times d}$. The nonzero entries of $\mathbf{z}^O$ indicate the pixels in $\mathbf{x}$ that are corrupted or occluded. In general, the nonzero support of $\mathbf{z}^O$ is different for different particles and is assumed to be sparse.

3) Discrimination From Context ($\mathbf{D}_C$)*:* To alleviate the problem of tracker drift, we acknowledge the importance of representing what a target is and what it is not. In that spirit, we go beyond representing the target appearance and exploit context information for more robust object tracking. Here, we define context to include the target's immediate background and any other objects, whose appearance may distract the tracker from the target. Although its exact definition and usage differ in previous tracking methods that use contextual information [16], [41], [42], [45], a target's context has been shown to significantly improve tracking performance. In contrast to these methods that try to exploit the context information, we formulate the tracking problem as an exclusive sparse learning problem that seamlessly incorporates the context information.

To formalize the notion of context, we define $\mathbf{D}_C$ as a dictionary of context templates, which are observations of the target's immediate background and any other objects that might distract tracking. The latter templates are especially important when tracking multiple objects. Clearly, good particles, from which the next target is selected, are particles that are represented well by $\mathbf{D}_F$ and poorly by $\mathbf{D}_C$. Conversely, bad particles tend to be represented better by $\mathbf{D}_C$. As such, augmenting $\mathbf{D}$ with $\mathbf{D}_C$ discriminates the target from its context and reduces tracker drift.

B. Imposing Exclusive Sparsity via the $\ell_{1,2}$-Norm

To allow for robust tracking and because particles are densely sampled around the current target state, particle representations with respect to $\mathbf{D}$ are sparse in general. In previous sparse coding trackers (e.g., L_1 tracker [1]), the sparsity of $\mathbf{z}$ had unstructured support. In other words, each particle could be represented by any set of dictionary templates, as long as that set is small and the reconstruction is accurate. However,

this is not the case when context information is involved. Since $\mathbf{D}_F$ and $\mathbf{D}_C$ describe complimentary aspects of particle representation, we believe that the representation of a particle should ideally be due to either $\mathbf{D}_F$ or $\mathbf{D}_C$ and not both. The sparsity's exclusivity lends some structure to its support. Therefore, when imposing sparsity on $\mathbf{z}$, we need to distinguish between intratype and intertype sparsity. While intratype sparsity describes the sparsity of the representation that uses a particular group of templates (e.g., sparsity of $\mathbf{z}^F$, $\mathbf{z}^O$, and $\mathbf{z}^C$ individually), intertype sparsity describes the sparse selection of target or context templates to represent a particle. Previous sparse coding trackers only consider intragroup sparsity.

For robust context-aware tracking, we take the following into consideration to learn particle representation $\mathbf{z} = \left[\mathbf{z}^F; \mathbf{z}^O; \mathbf{z}^C\right]$.

1) Intratype sparsity should hold, i.e., $\mathbf{z}^F$, $\mathbf{z}^O$, and $\mathbf{z}^C$ should be individually sparse. This allows for a robust representation based on target or context templates and for partial occlusion handling.
2) Intertype sparsity should also hold. This encourages exclusivity in representation by $\mathbf{D}_F$ or $\mathbf{D}_C$.

Based on 1) and 2), we formulate particle representation as an exclusive sparse learning problem (also known as eLasso [36], [37]), as shown in (2). Here, based on the three types of templates ($\mathbf{D}_F$, $\mathbf{D}_C$, and $\mathbf{D}_O$), we define $\mathcal{G}$ to make the learned $\mathbf{z}$ have the above sparse property. Equation (2) is convex and nonsmooth, where λ is a tradeoff parameter between reliable reconstruction and exclusive sparsity. This is the core of our proposed CEST method

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \sum_{g \in \mathcal{G}} \left\|\mathbf{z}^g\right\|_1^2. \tag{2}$$

The solution to (2) is described in Section IV. In Fig. 2, we present an example of how CEST works. The representation $\mathbf{z}$ of particle $\mathbf{x}$ (sampled around the tracked car) in the tth frame is computed by solving (2). Here, we consider three particles ($\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$), where $\mathbf{x}_i$ is chosen as the current tracking result because its reconstruction error with respect to $\mathbf{D}_F$ is smallest. Since $\mathbf{x}_j$ and $\mathbf{x}_k$ are misaligned versions of the target car, they are not represented well by $\mathbf{D_F}$, i.e., $\mathbf{z}_j^F$ and $\mathbf{z}_k^F$ have much smaller values than $\mathbf{z}_i^F$. This precludes the tracker from drifting into the background. Notice that intragroup and intergroup sparsity hold in these representations. Although templates from $\mathbf{D}_F$ and $\mathbf{D}_C$ are both used in representing $\mathbf{x}_j$, more nonzero values exist in $\mathbf{z}_j^F$ than $\mathbf{z}_j^C$, which is a consequence of exclusive sparsity.

C. Dictionary Template Update

The target templates $\mathbf{D}_F$ (m_F elements) are dynamically updated to incorporate variations in target appearance due to changes in illumination, viewpoint, etc. Target appearance remains the same only for a certain period of time, but eventually the target templates no longer provide an accurate representation. This is why a fixed appearance model is not sufficient to handle changes due to occlusion or illumination. Also, if the templates are updated too often, small errors are introduced each time a template is updated, errors accumulate, and the tracker may drift from the target. Our dictionary update scheme is based on [1]. To initialize the object and background dictionaries, we sample equal-sized patches at and around the initial position of the object. In our experiments, we shift the initial bounding box by 1–3 pixels in each direction, thus, resulting in m_F object templates. Also, we initialize $\mathbf{D}_C$ to image patches randomly sampled at a sufficient distance from the initial tracking result, thus, resulting in m_C context templates for a total of $m = m_F + m_C$ templates. Note that m is a user-defined parameter and all templates are normalized.

Each target template in $\mathbf{D_F}$ is assigned a weight ω_k that is indicative of how representative the template is. The more a template is used in representing tracking results, the higher its weight is. When $\mathbf{D_F}$ cannot represent particles well (up to a predefined threshold th), the target template with the smallest weight is replaced by the current tracking result, which is the particle $\mathbf{z}_i$ that is best represented by $\mathbf{D}$ such that $i = \arg\min_{k=1,\ldots,n} \left(\|\mathbf{x}_k - \mathbf{D}_F \mathbf{z}_k^F\|_2\right)$. The weight of this new template is set to the median of the current normalized weight vector ω. Templates in $\mathbf{D}_O$ are fixed, while those in $\mathbf{D}_C$ are updated at every frame by resampling patches at a sufficient distance from the current tracking result. This strategy is very similar to several methods in the literature, e.g., OAB [20] and MIL [22] (for negative samples), and sparse discriminative and generative (SDG) [29] (for background templates).

D. Discussion

As shown in (2), we propose an exclusive sparse learning formulation for robust object tracking by considering context information. It is worth emphasizing the difference between the proposed CEST algorithm and several related tracking methods [1], [7], [35].

1) *Exclusive Sparsity:* As shown in (2),

$$\sum_{g=1}^{G} \left\|\mathbf{z}^g\right\|_1^2 = \sum_{g=1}^{G} \left(\sum_{i=1}^{n_g} \left|\mathbf{z}_i^g\right|\right)^2.$$

Here, n_g is the number of template elements in $\mathbf{D}_g = \left[\mathbf{D}_F^g\ \mathbf{D}_O^g\ \mathbf{D}_C^g\right]$. Due to the $\ell_{1,2}$-norm, the elements in $\mathbf{z}^g$ corresponding to $\mathbf{D}_F^g$, $\mathbf{D}_O^g$, and $\mathbf{D}_C^g$ are exclusively sparse.

2) *Comparison With L_1 Tracker [1]:* The L_1 tracker [1] is a special case of our CEST. When we only use the $\mathbf{D_F}$ and $\mathbf{D_O}$ as one group and discard $\mathbf{D_C}$, our CEST becomes the L_1 tracker [1]. Compared with the L_1 tracker, our experiments show that CEST shows much better performance, thus, demonstrating the effectiveness of context information $\mathbf{D_C}$ for visual tracking as in [16], [41], and [42].

3) *Comparison With MTT Tracker [7], [35]:* The MTT tracker [7], [35] and CEST are different in the following ways.

   a) The two trackers adopt two different norms. The MTT tracker uses the $\ell_{2,1}$-norm $\|\mathbf{Z}\|_{2,1} = \sum_j \left(\sum_i |[\mathbf{Z}]_{ij}|^2\right)^{1/2}$ and our CEST adopts the $\ell_{1,2}$-norm $\|\mathbf{z}\|_{1,2} = \sum_{g \in \mathcal{G}} \|\mathbf{z}^g\|_1^2$. Here, $\mathbf{Z}$ is a matrix and each column is the representation of each

particle, $[\mathbf{Z}]_{ij}$ denotes the entry at the ith row and jth column of $\mathbf{Z}$, and $\mathbf{z}$ is the representation of one particle.

b) The two trackers use different information. The MTT tracker uses the $\ell_{2,1}$-norm $\|\mathbf{Z}\|_{2,1}$ to consider correlations among different particles to learn their representation $\mathbf{Z}$ jointly. However, CEST adopts the $\ell_{1,2}$-norm $\|\mathbf{z}\|_{1,2}$ to consider context information to learn the representation $\mathbf{z}$ of each particle.

IV. Optimization

Solving the exclusive sparsity problem in (2) produces the representation $\mathbf{z}$ of a single particle. Ideally, this optimization problem has to be solved for all n particles in each frame, which leads to a significant computational overhead. In this section, we will describe a two stage process, which allows for an efficient coarse-to-fine handling of these n optimization problems. In the first stage, we propose an approximate sampling method that scores and ranks all particles in the same frame according to their optimal cost functions. We use this ranking to prune out the particles that will obviously not be selected as the tracking result, thus, leaving a much smaller set of potential target candidates, whose representations will be solved for in the next stage. In the second stage, an APG-based method is used to solve (with quadratic convergence) a smooth approximation of (2) for each particle that has not been discarded. This two stage process leads to a significant speedup with minimal loss in accuracy.

A. Stage(1): Particle Pruning Using Dual Formulation

By adding redundant variables and assuming that the reconstruction error term is small (i.e., $\|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2 \leq \alpha \leq 1$), we approximate (2) with its upper bound as in

$$\begin{aligned} &\min_{\mathbf{r},\mathbf{z},c_g \forall g} f(\mathbf{r},\mathbf{z},c_g) = \|\mathbf{r}\|_2 + \lambda \sum_{g\in\mathcal{G}} c_g^2 \\ &\text{such that} \quad \mathbf{r} = \mathbf{x} - \mathbf{D}\mathbf{z};\ \ \|\mathbf{z}^g\|_1 \leq c_g \ \ \forall g. \end{aligned} \tag{3}$$

We construct the Lagrange function $\mathcal{L}(\mathbf{r},\mathbf{z},\mathbf{u}_0,u_g,c_g\ \forall g)$, where the dual variables $\mathbf{u}_0$ and $\{u_g{:}g \in \mathcal{G}\}$ represent the Lagrange multipliers of the constraints in (3). Then, we minimize $\mathcal{L}$ with respect to the primal variables by using identities for the conjugate function of a vector norm [46]. As a result, we obtain the dual problem of (3) in (4). Note that $\mathbf{D}\mathbf{z} = \sum_{g\in\mathcal{G}} \mathbf{D}_g\mathbf{z}^g$, where $\mathbf{D}_g$ and $\mathbf{z}^g$ are the dictionary corresponding to the gth group and its group representation respectively. It is easy to see that (4) is equivalent to (5). We refer the reader to the Appendix for a detailed derivation of the dual problem

$$\begin{aligned} &\max_{\mathbf{u}_0,u_g\geq 0\ \forall g} \ \mathbf{u}_0^T\mathbf{x} - \frac{1}{4\lambda}\sum_{g\in\mathcal{G}} u_g^2 \\ &\text{such that} \quad \|\mathbf{u}_0\|_2 \leq 1;\ \ \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty \leq u_g\ \forall g \end{aligned} \tag{4}$$

$$\begin{aligned} &\max_{\mathbf{u}_0} \quad g(\mathbf{u}_0) = \mathbf{u}_0^T\mathbf{x} - \frac{1}{4\lambda}\sum_{g\in\mathcal{G}} \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty^2 \\ &\text{such that} \quad \|\mathbf{u}_0\|_2 \leq 1 \end{aligned} \tag{5}$$

Since Slater's condition is satisfied and $f(.)$ is convex, strong duality holds between the primal and dual problems [46]. Therefore, sorting the particles according to the regularized reconstruction term in (3) is equivalent to sorting them according to the objective in (5). However, solving the optimization problem in (5) exactly is as hard as the primal problem. But, we observe that (5) possesses two interesting properties, which we will exploit to approximate its optimal objective. In (5), the ball-constraint on the dual variable $\mathbf{u}_0$ is independent of $\mathbf{D}_g$ and $\mathbf{x}$ and is easy to sample from. Consequently, we perform an offline dense sampling of s dual variables $\mathbf{u}_0$ from the unit-ball constraint, thus, generating the set $\mathcal{U} = \{\mathbf{u}_0^j{:}\|\mathbf{u}_0^j\|_2 \leq 1;\ j = 1,\ldots,s\}$. We score each particle $\mathbf{x}$ using the largest objective value obtained when $g(\mathbf{u}_0)$ is evaluated at all the samples in $\mathcal{U}$. This is equivalent to approximating the optimal dual objective $\max_{\|\mathbf{u}_0\|_2\leq 1} g(\mathbf{u}_0)$ with $\max_{\mathbf{u}_0\in\mathcal{U}} g(\mathbf{u}_0)$. Finally, all particles are sorted according to their scores. Particles with scores outside the largest K are discarded immediately and their representations do not need to be computed. The representations of the surviving particles are computed as outlined in the next section.

B. Stage(2): Solving the Primal Problem of (2)

After pruning, each surviving particle is represented by minimizing the objective in (2). Since the cost function is convex but nonsmooth (due to the $\ell_{1,2}$ regularizer), it is well known that any first order method (e.g., gradient descent) will have sublinear convergence. Recently, several methods have been proposed to solve this problem more efficiently [36], [37]. To reduce computational cost, we approximate the nonsmooth part of the original objective with a differentiable one. Then, minimization is done using the APG method [47], which has quadratic convergence (i.e., an ϵ-accurate solution is reached in $O(\epsilon^{-0.5})$ iterations).

Since the dual norm of the ℓ_∞ vector norm is the ℓ_1 norm, we define the nonsmooth regularizer as: $\|\mathbf{z}^g\|_1 = \max_{\|\mathbf{v}_g\|_\infty\leq 1}\langle \mathbf{z}^g, \mathbf{v}_g\rangle$. By choosing a positive smoothness parameter μ, $\|\mathbf{z}^g\|_1$ can be approximated by the differentiable strongly convex function $b_\mu(\mathbf{z}^g) = \max_{\|\mathbf{v}_g\|_\infty\leq 1}\langle \mathbf{z}^g, \mathbf{v}_g\rangle - \mu/2\|\mathbf{v}_g\|_2^2$. Clearly, $\lim_{\mu\to 0} b_\mu(\mathbf{z}^g) = \|\mathbf{z}^g\|_1$. In fact, $b_\mu(\mathbf{z}^g)$ has a closed form expression: $b_\mu(\mathbf{z}^g) = \langle \mathbf{z}^g, \mathbf{v}_g^*\rangle - \mu/2\|\mathbf{v}_g^*\|_2^2$, where $\mathbf{v}_g^* = \mathcal{S}(\mu^{-1}\mathbf{z}^g)$ and $\mathcal{S}(.)$ is the elementwise shrinkage operator defined as $\mathcal{S}(\mathbf{a}_i) = \min(1,\max(-1,\mathbf{a}_i))$. By introducing this smooth approximation into (2), we obtain (6), where $\mathbf{E}_g$ is the linear operator that extracts $\mathbf{z}^g$ from $\mathbf{z}$, i.e., $\mathbf{z}^g = \mathbf{E}_g\mathbf{z}$. To apply the APG method [47] on (6), we require the proximal gradient of $F(\mathbf{z})$, which is computed as $\nabla F_\mu(\mathbf{z}) = 2\mathbf{D}^T(\mathbf{D}\mathbf{z}-\mathbf{x}) + 2\lambda\sum_{g\in\mathcal{G}} b_\mu(\mathbf{E}_g\mathbf{z})\mathbf{E}_g^T\mathcal{S}(\mu^{-1}\mathbf{E}_g\mathbf{z})$. In addition, F_μ is Lipschitz continuous with constant L, which can be calculated as in [37]. Note that APG could not be directly applied to (2), since the proximal gradient in that case is not computable in closed form. The overall APG-based method for this optimization stage is described in Algorithm 1

$$\min_{\mathbf{z}} \quad F_\mu(\mathbf{z}) = \|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2^2 + \lambda\sum_{g\in\mathcal{G}} b_\mu^2(\mathbf{E}_g\mathbf{z}). \tag{6}$$

Algorithm 1: Context-Aware Exclusive Sparse Representation

Input : $\mathbf{x}$, $\mathbf{D}$, G groups, L, λ, and μ
Output: $\mathbf{z}$

Initialize $\mathbf{z}_0$, $\mathbf{q}_0$, $\alpha_0 \leftarrow 0$, $t \leftarrow 0$
while *not converged* **do**
  $\mathbf{p}_t = (1-\alpha_t)\mathbf{z}_t + \alpha_t \mathbf{q}_t$; $\mathbf{q}_{t+1} = \mathbf{q}_t - 1/2\alpha_t L \nabla F_\mu(\mathbf{p}_t)$
  $\mathbf{z}_{t+1} = (1-\alpha_t)\mathbf{z}_t + \alpha_t \mathbf{q}_{t+1}$, $\alpha_{t+1} = 2/t+1$; $t \leftarrow t+1$
end

1) Time Complexity: We denote CEST as the tracker that applies Algorithm 1 to represent all n particles. The time complexity of CEST is therefore $O(nd^2\epsilon^{-0.5})$, where the number of APG iterations is $O(\epsilon^{-0.5})$ and the complexity of each APG iteration is $O(d^2)$. We denote CEST* as the tracker that employs the pruning scheme in Section IV-A before applying Algorithm 1 on the remaining K particles. Since $K < n$, the complexity of particle pruning (including particle scoring and sorting) is dominated by the complexity of solving (6) for the remaining K particles. Therefore, the complexity of CEST* is $O(Kd^2\epsilon^{-0.5})$. Based on our experiments where $K = 100$, $s = 10^5$, and $\epsilon = 10^{-6}$, CEST* is approximately n/K times faster than CEST. This speedup grows linearly with the number of particles. Since increasing n usually leads to improved tracking performance, CEST* can produce better tracking results without much added computational overhead.

V. Experimental Results

In this section, we present experimental results that validate the effectiveness of our CEST method.

A. Datasets and Baseline Trackers

To evaluate CEST, we compile a set of 15 challenging tracking sequences that are publicly available online. The videos are recorded in indoor and outdoor environments and include challenging appearance variations due to changes in pose, illumination, scale, and partial occlusion. We compare CEST to nine recent and state-of-the-art trackers denoted as: VTD [19], L_1 [1], MTT [7], IVT [17], MIL [22], OAB [20], SDG [29], tracking-learning-detection (TLD) tracker [48], and context tracker (CT) [42]. We implemented these trackers using publicly available source codes or binaries provided by the authors. They are initialized using their default parameters.

B. Implementation Details

We evaluate CEST for both single object tracking and multi-object tracking. For single object tracking, $\mathbf{D}_C$ is obtained from background information. For multiobject tracking, $\mathbf{D}_C$ includes context information from the background and other objects. Of course, the context information in [16], [41], and [42] could also be adopted, such as, distracters and supporters in [42]. The distracters are regions which have similar appearance as the target and consistently co-occur with high confidence score, and the supporters, on the other hand, are local key-points around the target with consistent co-occurrence and motion correlation in a short time interval. Both of them play an important role in visual tracking [42]. In our experiments, the initial position of the target is selected manually, and we shift the initial bounding box by 1–3 pixels in each dimension, thus, resulting in $m_F = 13$ target templates $\mathbf{D}_F$ (similar to L_1 tracker [1]). Also, we initialize $\mathbf{D}_C$ to image patches randomly sampled at a sufficient distance from the initial tracking result, and obtain $m_C = 12$ templates for context information. The template size d is set to half the size of the target in the first frame. Usually, d is in the order of several thousands of pixels, and the number of occlusion templates $\mathbf{D}_O$ is $m_O = d$. Note that m_F, m_C, and m_O are user-defined parameters. In object tracking, the $\mathbf{D}_F$ and $\mathbf{D}_C$ are updated over time by the strategy in Section III-C.

As shown in (2), our CEST models $\mathbf{D}$ as a concatenation of G groups of dictionary templates, indexed by the set $\mathcal{G}$. In each group g, $\mathbf{D}_g = \left[\mathbf{D}_F^g\ \mathbf{D}_O^g\ \mathbf{D}_C^g\right]$, and $\mathbf{D}_F^g$, $\mathbf{D}_O^g$, and $\mathbf{D}_C^g$ are a subset of $\mathbf{D}_F$, $\mathbf{D}_O$, and $\mathbf{D}_C$, respectively. To construct this kind of $\mathcal{G}$, at each time instance, the elements in $\mathbf{D}_F$, $\mathbf{D}_C$, and $\mathbf{D}_O$ are clustered into k_1, k_2, and k_3 clusters, respectively. Then, the set $\mathcal{G}$ is constructed by all possible combinations of these clusters. In other words, each $\mathbf{D}_g$ comprises one cluster from each base-group. As a result, there are $G = k_1 \times k_2 \times k_3$ groups in $\mathcal{G}$. The effect of these parameters (k_1, k_2, and k_3) on tracking performance is discussed in Section V-C3.

As in the L_1 tracker [1], we model $p\left(\vec{\mathbf{s}}_t|\vec{\mathbf{s}}_{t-1}\right) \sim \mathcal{N}(\vec{\mathbf{0}}, \mathrm{diag}(\vec{\sigma}))$, where $\vec{\sigma} = [0.005,\ 0.0005,\ 0.0005,\ 0.005,\ 2,\ 2]^T$. We set the number of particles $n = 600$ and surviving particles $K = 100$. In Algorithm 1, we set $\lambda = 1$ and $\mu = 0.01$. Each tracker uses the same parameters for all sequences. All our experiments are done using MATLAB on a 2.66 GHZ Intel Core2 Duo PC with 18 GB RAM. Next, we first do parameter analysis for our proposed method and show how the parameters affect the performance and how to decide their values in Section V-C. The computational cost is discussed in Section V-D. We present qualitative and quantitative results of the tracking methods in Sections V-E and V-F. The videos are available in the supplementary material.

C. Parameter Analysis

Several parameters play important roles in the proposed tracking algorithm, such as, the λ in (2), the th in Section III-C, and the number of groups in $\mathcal{G}$ in (2). In this section, we show how to determine their values and their effects on tracking performance.

1) Effect of λ: The parameter λ in (2) is to balance the reconstruction error and the exclusive sparse term. To show the effect of λ, we parameterize it by a discrete set Λ, where $\Lambda = \{1e^{-3},\ 0.01,\ 0.1,\ 0.5,\ 1.0,\ 2.0,\ 5.0,\ 10.0\}$. We evaluate these values on ten videos with about 4000 frames. For each $\lambda \in \Lambda$, we compute the average overlap score from all frames. For different λ, we obtain the corresponding results as shown in Fig. 4(a). Overall, the proposed algorithm is robust to different λs as long the value is within reasonable ranges. From on these results, we can set $\lambda = 1$ in (2) due to its best performance.

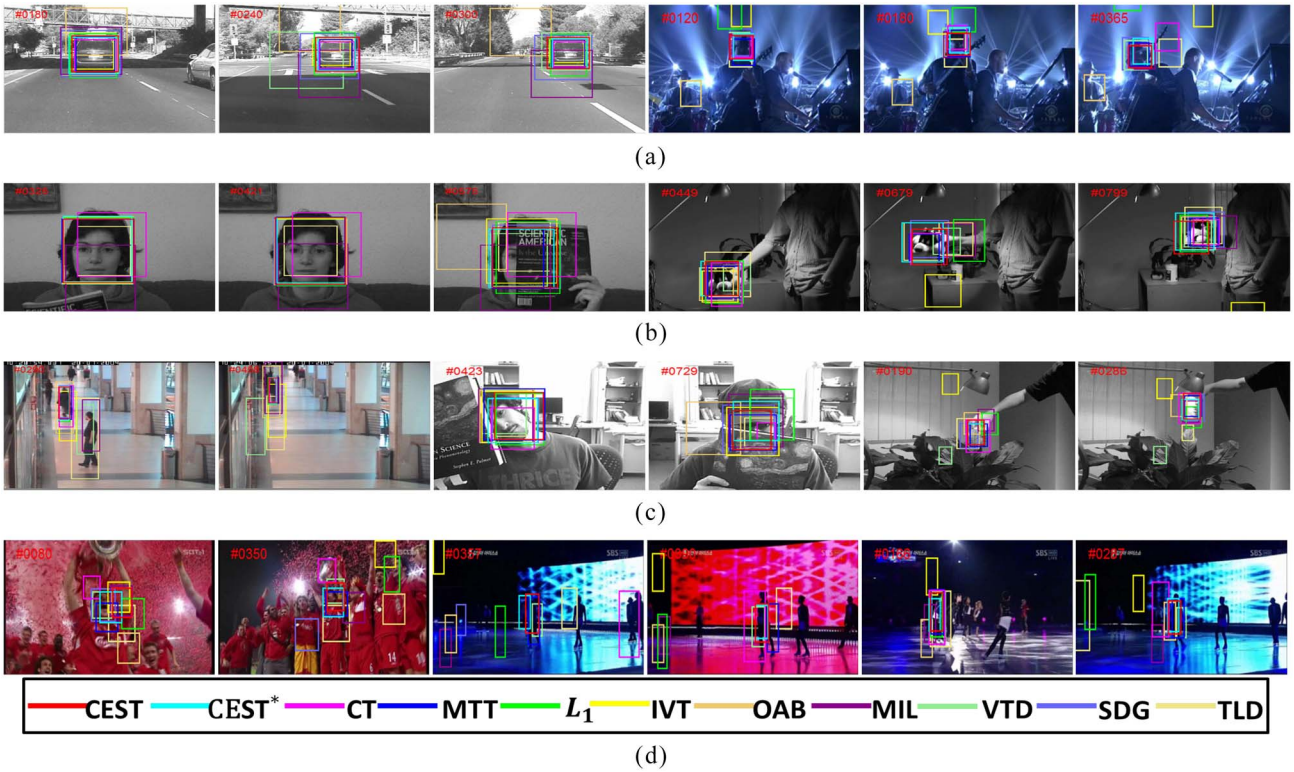


Fig. 3. Single object tracking results of 11 trackers (denoted in different colors) on ten video sequences. Frame numbers are overlayed in red. See text for details. For better viewing, please see original color pdf file. (a) Tracking results on sequences *Car4* and *Shaking* with illumination and pose variation. (b) Tracking results on sequences *Faceocc* and *Sylv* with occlusion and pose variation, respectively. (c) Tracking results on sequences *Onelsr*, *Faceocc2*, and *Coke11* with occlusion and pose variation. (d) Tracking results on sequences *Soccer*, *Skating1*, and *Skating2* with background clutter, abrupt motion, and illumination change, respectively.

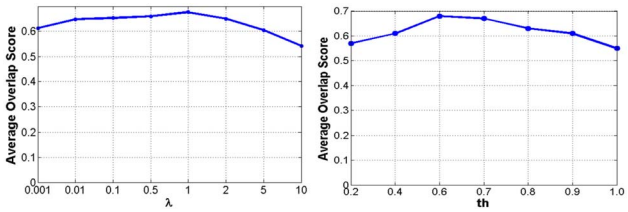


Fig. 4. Effects of (a) λ and (b) th on visual tracking performance.

2) Effect of th: The parameter th in Section III-C decides the template updating. A fixed appearance template is not sufficient to handle changes in appearance due to occlusion or changes in illumination and pose. Also, if the templates are updated too often, small errors are introduced each time a template is updated, errors accumulate, and the tracker may drift from the target. Therefore, the parameter th is closely related to the tracking performance. To analyze the effect of th on tracking performance, we use different th on seven videos with about 2500 frames. To simplify this problem, we assume that th can be parameterized by a discrete set {0.2,0.4,0.6,0.7,0.8,0.9,1.0}. Experimental results, shown in Fig. 4(b), indicate that we can set the value of th to 0.6 because it achieves the best performance.

3) Effect of $\mathcal{G}$: The parameter $\mathcal{G}$ in (2) is about the group construction for $\mathbf{D}$ to obtain the exclusive sparse representation $\mathbf{z}$. As discussed in Section V-B, the set $\mathcal{G}$ is constructed based on the combinations of the k_1, k_2, and k_3 clusters ($G = k_1 \times k_2 \times k_3$). To show the effect of G, we parameterize it by a discrete set as shown in Table I. We evaluate these values on five videos with about 2000 frames. For each setting, we compute the average overlap score from all frames, and the corresponding results are as shown in Table I. Overall, the proposed algorithm is quite stable to different settings. Moreover, small G can reduce the computational. From on these results, we can see that it is a good trade-off to set the value of G to $4 \times 4 \times 6$.

TABLE I
EFFECTS OF $\mathcal{G}$ ON VISUAL TRACKING PERFORMANCE

G	$2 \times 2 \times 10$	$3 \times 3 \times 10$	$4 \times 4 \times 6$	$4 \times 4 \times 10$	$5 \times 5 \times 10$
Performance	67.9	67.8	68.9	69.2	68.6

D. Computational Cost

Tracking algorithms based on sparse representations and particle filters [1], [7] have been demonstrated to perform well in visual tracking. However, the run-time of sparse trackers grows proportionally as the number of particles and templates in the dictionary. Table II shows the average per-frame runtime of state-of-the-art algorithms based on sparse representation [1], [7] and the proposed algorithms. Clearly, the L_1 tracker is much slower than CEST and CEST* for any (n, d), with CEST* being about n/K times faster than CEST. The MTT tracker [7] makes use of the correlation

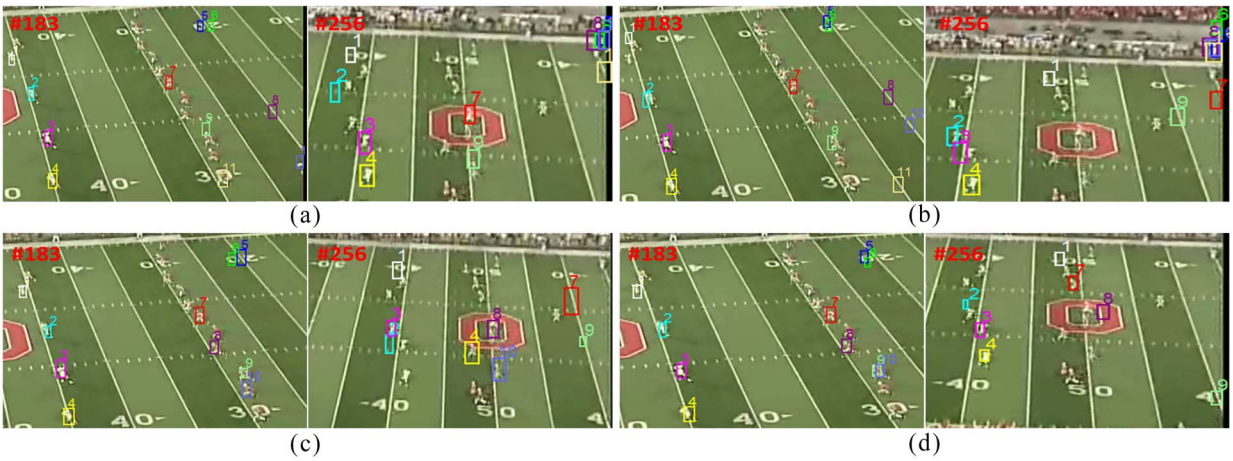


Fig. 5. Multiobject tracking results of four related trackers (a) CEST, (b) MTT, (c) L_1, and (d) SDG on one sports video sequence.

TABLE II
AVERAGE PER-FRAME RUNTIME (IN SECONDS) OF FOUR TRACKERS WITH TEMPLATE SIZE $d = 16 \times 16$ AND VARYING NUMBER OF PARTICLES n

n \ d	16×16			
	L_1	CEST	CEST*	MTT
100	2.075	1.54	1.89	0.32
200	3.915	1.91	1.55	0.33
300	5.798	2.68	1.49	0.73
400	7.902	3.51	1.61	1.21
500	11.394	4.17	1.66	1.61
600	12.057	4.77	1.63	2.24
700	13.580	5.90	2.00	2.96
800	15.897	7.54	1.68	3.58

among particles to improve robustness and reduce computational load. The runtime of CEST* (as stated in Section IV) is constant with respect to n. Since increasing n usually leads to improved tracking performance, CEST* can produce better tracking results without much added computational overhead.

E. Qualitative Comparison

In Figs. 3 and 5, we show tracking results of single object and multiple objects, respectively. In Fig. 3, we evaluate 11 trackers on ten video sequences, and the results are categorized according to the tracking challenge faced in each sequence. In Fig. 5, we show multiobject tracking on five sports video sequences by using the contextual information from other tracking targets.

1) Illumination and Pose Variation: For the *Car4*, and *Shaking* sequences, the tracked object is subject to changes in illumination and pose. In the *Car4* sequence in the left of Fig. 3(a), OAB, SDG, TLD, and VTD start to drift from the target at frame 186, while MIL drifts at frame 200 and finally loses the target at frame 300. CT experiences some drift, while MTT, IVT, and L_1 track the target quite well. The target is successfully tracked throughout the entire sequence by CEST and CEST*. In the *Shaking* sequence [refer to the right of Fig. 3(a)], the stage lighting condition is drastically changed, and the pose of the object is severely varied due to head *Shaking*. OAB, IVT, TLD, L_1, and CT fail to track the object when these dramatic changes occur. VTD, MIL, SDG, and MTT track the object quite well except for some errors around frames 60 and 260, respectively. CEST and CEST* successfully track the object due to the use of robust sparse representation and contextual information from background.

2) Occlusion: In *Faceocc*, a moving face is tracked. Some results are shown in the left of Fig. 3(b). Because occlusion is minor, most of the methods track the face accurately except OAB, MIL and CT, which experience some drift.

3) Pose Variation: Results on the *Sylv* sequence are shown in the right of Fig. 3(b). Here, the stuffed animal undergoes significant pose and scale changes. IVT fails around frame 600 due to the pose change. The rest of the trackers track the target throughout the sequence, with MIL, VTD, OAB, and L_1 veering off the target at certain instances.

4) Occlusion and Pose Variation: In the *Onelsr* sequence [refer to the left of Fig. 3(c)], the walking woman is partially occluded by a walking man. IVT, MIL, OAB, CT, TLD, and VTD lose the target woman, start tracking the man when partial occlusion occurs around frame 200, and some of them are unable to recover from this failure. CEST, CEST*, MTT, SDG, and L_1 track the woman quite well. Results on the *Faceocc2* sequence are shown in the middle of Fig. 3(c). Most trackers start drifting from the manąŕs face when it is almost fully occluded by the book. Because the CEST and CEST* make use of the background information with the exclusive sparse learning and avoid drifting, they can handle the appearance changes in this sequence very well and continue tracking the target during and after the occlusion. The *Coke11* sequence contains frequent occlusions and fast motion, which cause motion blur. The CEST, CEST*, L_1, SDG, MTT, OAB, and MIL can track the target almost throughout the entire sequence. The other trackers fail due to pose change and occlusion as shown in the right of Fig. 3(c).

5) Background Clutter: In the *Soccer* sequence [refer to the left of Fig. 3(d)], the background is cluttered. CEST and CEST* accurately track the player's face despite scale and pose changes as well as occlusion/noise from the confetti raining around him. All other trackers, except VTD and MTT, fail to track the object reliably. The TLD tracker can not track the

TABLE III
AVERAGE CENTER LOCATION ERROR AND STANDARD DEVIATION (IN PIXELS) OF 11 TRACKERS ON TEN VIDEO SEQUENCES. THE BEST AND SECOND BEST RESULTS ARE SHOWN IN RED AND BLUE FONTS, RESPECTIVELY

	CEST	CEST*	CT	MTT	L_1	IVT	OAB	MIL	VTD	SDG	TLD
car4	1.9 ± 1.2	3.0 ± 1.8	9.2 ± 17.5	2.2 ± 1.2	8.5 ± 5.0	6.4 ± 2.8	88.1 ± 60.6	53.7 ± 40.3	27.0 ± 31.3	59.4 ± 50.3	6.8 ± 5.6
coke11	3.5 ± 3.7	3.5 ± 2.6	5.8 ± 6.1	3.2 ± 3.2	12.1 ± 9.9	58.5 ± 27.4	11.3 ± 5.8	13.7 ± 8.3	62.6 ± 25.2	5.2 ± 3.4	11.5 ± 10.3
faceocc	6.9 ± 3.7	9.4 ± 4.2	31.2 ± 14.2	7.7 ± 5.8	7.0 ± 3.5	9.7 ± 6.1	17.2 ± 27.4	34.3 ± 19.2	8.7 ± 9.9	4.3 ± 3.0	14.8 ± 9.1
faceocc2	8.0 ± 8.1	6.6 ± 4.1	8.3 ± 8.7	8.5 ± 6.8	15 ± 19	7.6 ± 4.5	20 ± 20	10 ± 5.5	12 ± 13	11 ± 9.0	13.2 ± 3.7
onelsr	3.0 ± 2.1	3.1 ± 2.9	9.1 ± 25.6	3.3 ± 1.2	4.7 ± 4.0	24.0 ± 16.7	12.5 ± 7.3	23.8 ± 27.1	44.3 ± 35.2	4.2 ± 2.5	49.5 ± 35.1
shaking	6.4 ± 4.7	4.6 ± 2.2	33.9 ± 51.5	8.4 ± 3.6	37.8 ± 22.8	52.3 ± 19.8	100 ± 25	7.9 ± 5.3	4.0 ± 1.9	8.1 ± 3.3	21.0 ± 15.4
skating1	4.4 ± 2.3	4.5 ± 2.8	35.7 ± 53.4	7.4 ± 6.7	20.1 ± 38.2	75.0 ± 48.7	39.3 ± 54	49 ± 51	5.1 ± 3.7	28 ± 56	36 ± 28.2
skating11	3.0 ± 1.9	3.7 ± 1.7	61.2 ± 51	3.4 ± 2.0	31 ± 52	57 ± 40	39 ± 30	44 ± 38	6.2 ± 4.1	128 ± 69	52.6 ± 35.6
soccer	15 ± 11	11 ± 5.2	72 ± 47	14 ± 7.0	58 ± 46	97 ± 52	65 ± 43	46 ± 27	11 ± 6.8	42 ± 26	29.8 ± 16.1
sylv	5.3 ± 5.9	4.6 ± 5.3	6.5 ± 3.3	4.8 ± 3.0	15 ± 18	39 ± 56	11 ± 14	15 ± 18	7.5 ± 9.9	6.6 ± 3.3	5.9 ± 4.2

TABLE IV
AVERAGE OVERLAP RATE BASED ON [49] AND STANDARD DEVIATION OF 11 TRACKERS ON TEN VIDEO SEQUENCES. THE BEST AND SECOND BEST RESULTS ARE SHOWN IN RED AND BLUE FONTS, RESPECTIVELY

	CEST	CEST*	CT	MTT	L_1	IVT	OAB	MIL	VTD	SDG	TLD
car4	0.81 ± 0.08	0.78 ± 0.08	0.64 ± 0.23	0.78 ± 0.09	0.62 ± 0.09	0.73 ± 0.07	0.22 ± 0.32	0.27 ± 0.32	0.47 ± 0.34	0.30 ± 0.33	0.56 ± 0.25
coke11	0.78 ± 0.14	0.75 ± 0.14	0.57 ± 0.15	0.74 ± 0.15	0.52 ± 0.23	0.43 ± 0.36	0.19 ± 0.25	0.21 ± 0.26	0.22 ± 0.28	0.31 ± 0.33	0.43 ± 0.23
faceocc	0.70 ± 0.17	0.73 ± 0.13	0.66 ± 0.19	0.71 ± 0.14	0.66 ± 0.23	0.78 ± 0.09	0.58 ± 0.27	0.67 ± 0.12	0.68 ± 0.21	0.70 ± 0.18	0.53 ± 0.12
faceocc2	0.71 ± 0.17	0.74 ± 0.13	0.72 ± 0.19	0.70 ± 0.14	0.67 ± 0.23	0.73 ± 0.09	0.59 ± 0.26	0.72 ± 0.09	0.69 ± 0.21	0.73 ± 0.16	0.57 ± 0.13
onelsr	0.80 ± 0.08	0.76 ± 0.09	0.65 ± 0.19	0.78 ± 0.09	0.77 ± 0.09	0.44 ± 0.26	0.47 ± 0.19	0.35 ± 0.28	0.34 ± 0.38	0.78 ± 0.07	0.27 ± 0.29
shaking	0.66 ± 0.15	0.67 ± 0.12	0.56 ± 0.29	0.64 ± 0.14	0.41 ± 0.34	0.39 ± 0.36	0.18 ± 0.22	0.58 ± 0.18	0.65 ± 0.21	0.68 ± 0.14	0.44 ± 0.25
skating1	0.66 ± 0.13	0.60 ± 0.16	0.51 ± 0.27	0.61 ± 0.22	0.53 ± 0.26	0.39 ± 0.37	0.35 ± 0.24	0.42 ± 0.29	0.59 ± 0.23	0.61 ± 0.25	0.29 ± 0.26
skating11	0.69 ± 0.13	0.63 ± 0.15	0.39 ± 0.34	0.67 ± 0.18	0.51 ± 0.32	0.41 ± 0.38	0.26 ± 0.25	0.39 ± 0.32	0.55 ± 0.25	0.42 ± 0.37	0.37 ± 0.24
soccer	0.46 ± 0.29	0.49 ± 0.24	0.37 ± 0.35	0.50 ± 0.28	0.36 ± 0.35	0.43 ± 0.37	0.21 ± 0.24	0.36 ± 0.33	0.46 ± 0.28	0.43 ± 0.36	0.34 ± 0.27
sylv	0.74 ± 0.14	0.75 ± 0.14	0.70 ± 0.11	0.73 ± 0.10	0.58 ± 0.28	0.47 ± 0.28	0.67 ± 0.23	0.58 ± 0.28	0.73 ± 0.19	0.73 ± 0.09	0.70 ± 0.01

TABLE V
AVERAGE CENTER LOCATION ERRORS (STANDARD DEVIATION) AND AVERAGE OVERLAP RATES (STANDARD DEVIATION) OF FOUR RELATED TRACKERS FOR MULTIOBJECT TRACKING ON FIVE VIDEO SEQUENCES

Tracker	Average Center Location Error (Standard Deviation)					Average Overlap Rate (Standard Deviation)				
	video1	video2	video3	video4	video5	video1	video2	video3	video4	video5
CEST	46 (67)	50 (54)	51 (71)	11 (14)	31 (54)	0.26 (0.27)	0.22 (0.27)	0.28 (0.32)	0.35 (0.27)	0.25 (0.28)
MTT	55 (59)	36 (45)	68 (76)	17 (28)	28 (51)	0.23 (0.29)	0.27 (0.30)	0.26 (0.32)	0.33 (0.29)	0.26 (0.28)
L_1	66 (73)	67 (73)	64 (70)	23 (27)	44 (68)	0.18 (0.26)	0.17 (0.27)	0.26 (0.33)	0.21 (0.30)	0.19 (0.25)
SDG	60 (77)	37 (47)	55 (69)	31 (47)	37 (60)	0.20 (0.26)	0.21 (0.26)	0.24 (0.29)	0.24 (0.28)	0.20 (0.25)

target again when it fails to detect. These sequences demonstrate how CEST achieves state-of-the-art performance despite dramatic pose change and occlusion.

6) Abrupt Motion and Illumination Change: The *Skating1* and *Skating2* sequences contain abrupt object motion, severe illumination and scale changes, and viewpoint changes and occlusions, which cause most of the trackers to fail. CEST, CEST*, MTT, and VTD handle these changes well, as shown in Fig. 3(d). Note that CEST performs slightly better than MTT and VTD (e.g., frame 357), which are the most recent tracker that copes with abrupt motion and appearance change.

7) Multiobject Tracking: To demonstrate that context in the case of CEST can include not only the background of a single-target but other tracking targets, we test all trackers on multiobject tracking. To fairly compare our CEST with other related single-target trackers (e.g., MTT, L_1, and SDG), we run multiple single-target trackers independently. In Fig. 5, two sample results of four trackers (CEST, MTT, L_1, and SDG) on the same video clip are shown. From the results, we can see that it is difficult for the four methods to track players in sports video. Due to fast camera motion, occlusion, low-resolution image capture, varying viewpoints and illumination changes, and many trackers are prone to drifting. However, our CEST achieves the best performance because it adopts exclusive sparse learning to consider the context information (background information and object templates) to avoid drifting.

F. Quantitative Comparison

To give a quantitative comparison among the trackers, we evaluate them using the center location error as well as the overlapping rate [49], and the results (mean and standard deviation) are shown in Tables III and IV for single object tracking, and in Table V for multiobject tracking. Overall, the proposed tracker performs well against the other state-of-the-art algorithms.

Now, we compare CEST and CEST* with the L_1, MTT, and SDG trackers, which are the most related trackers to ours and has shown good performance. L_1 is a tracker to adopt sparse learning representing particles. MTT only considers foreground template and adopts multitask learning to consider the correlations among particles to improve L_1. Our proposed CEST is different from L_1 and MTT, because our method adopts exclusive sparse learning to consider the context information (background or objects). SDG also uses background information for tracking. Different from SDG, our CEST adopts exclusive sparse to decide only a few elements from $\mathbf{D}_F$, $\mathbf{D}_O$, or $\mathbf{D}_C$ to represent particles. However, in L_1, SDG, and MTT, elements from all of $\mathbf{D}_F$, $\mathbf{D}_O$, and $\mathbf{D}_C$ can be used to represent particles together. Based on the results in Tables III–V, CEST and CEST* outperform three of the trackers (L_1, MTT, SDG). This is primarily due to the use of context information, which makes CEST less prone to tracker drift. Moreover, CEST and CEST* have quite comparable performance. This demonstrates that the particle pruning stage in CEST* has minor impact

$$\min_{\mathbf{r},\mathbf{z}_g,c_g\forall g} f(\mathbf{r},\mathbf{z},c_g) = \|\mathbf{r}\|_2 + \lambda\sum_{g\in\mathcal{G}} c_g^2 \quad \text{such that} \quad \mathbf{r} = \mathbf{x} - \mathbf{D}\mathbf{z};\ \ \|\mathbf{z}^g\|_1 \le c_g \ \ \forall g \in \mathcal{G} \tag{9}$$

$$\begin{aligned} g(\mathbf{u}_0, u_g) &= \inf_{\mathbf{r},\mathbf{z},c_g\forall g} \|\mathbf{r}\|_2 + \lambda\sum_{g\in\mathcal{G}} c_g^2 + \mathbf{u}_0^T\left(\mathbf{x} - \sum_{g\in\mathcal{G}}\mathbf{D}_g\mathbf{z}^g - \mathbf{r}\right) + \sum_{g\in\mathcal{G}} u_g\left(\|\mathbf{z}^g\|_1 - c_g\right) \\ &= \mathbf{u}_0^T\mathbf{x} - \max_{\mathbf{r}}(\mathbf{u}_0^T\mathbf{r} - \|\mathbf{r}\|_2) + \sum_{g\in\mathcal{G}}\min_{c_g}(\lambda c_g^2 - u_g c_g) - \max_{\mathbf{z}^g}\left(\left(\mathbf{D}_g^T\mathbf{u}_0\right)^T\mathbf{z}^g - u_g\|\mathbf{z}^g\|_1\right) \\ &= \mathbf{u}_0^T\mathbf{x} - \frac{1}{4\lambda}\sum_{g\in\mathcal{G}} u_g^2 \quad \text{such that} \quad \|\mathbf{u}_0\|_2 \le 1;\ \ \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty \le u_g \ \ \forall g \in \mathcal{G}. \end{aligned} \tag{11}$$

on tracking accuracy, even though it makes CEST* much faster.

VI. Conclusion

In this paper, we formulate particle filter-based tracking as an exclusive sparse representation problem that exploits context information. Particle representations are learned using an efficient APG method. By using a fast particle pruning method based on an approximate dual formulation, the runtime of the proposed tracker is constant with respect to the number of particles. We show that the popular L_1 tracker [1] is a special case of our formulation. Also, we extensively analyze the performance of our tracker on challenging real-world video sequences and show that it outperforms nine state-of-the-art trackers.

Appendix

In CEST, the representation problem to be solved for a particle $\mathbf{x}$ is defined in

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda\sum_{g\in\mathcal{G}} \left\|\mathbf{z}^g\right\|_1^2 \quad \left[\text{where} \quad \mathbf{D}\mathbf{z} = \sum_{g\in\mathcal{G}}\mathbf{D}_g\mathbf{z}^g\right]. \tag{7}$$

In order to derive the dual of (7), we first recast it as the equivalent problem in

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda\sum_{g\in\mathcal{G}} c_g^2 \quad \text{such that} \quad \|\mathbf{z}^g\|_1 \le c_g \ \ \forall g. \tag{8}$$

By adding redundant variables and assuming that the reconstruction error term is small (i.e., $\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \le \alpha \le 1$), we approximate (8) with its upper bound in (9), as shown at the top of the page. By forming the Lagrange function of (9) and taking its minimum with respect to the primal variables $\mathbf{r}$, $\mathbf{z}$, $c_g\forall g$, we obtain its dual problem in (10), where $g(\mathbf{u}_0, u_g)$ is the dual function expressed in (11) and $(\mathbf{u}_0, u_g\forall g \in \mathcal{G})$ are the dual variables corresponding to the primal equality and inequality constraints, respectively

$$\min_{\mathbf{u}_0,u_g\ge 0\forall g} g(\mathbf{u}_0, u_g). \tag{10}$$

In the last step of (11), as shown at the top of the page, we make use of the closed form expression for the solution to an unconstrained convex quadratic program and the conjugate function of a vector norm. In (12), we define the conjugate function of the ℓ_p norm and its relationship with its dual ℓ_q norm, such that $1/p+1/q = 1$. So, for instance, the dual norm of the ℓ_1 norm is the ℓ_∞ norm, while the ℓ_2 norm is its own dual norm

$$h_*(\mathbf{y}) \coloneqq \sup_{\mathbf{x}}\left(\mathbf{y}^T\mathbf{x} - \alpha\,\|\mathbf{x}\|_p\right) = \begin{cases} 0 & \text{if } \|\mathbf{y}\|_q \le \alpha \\ \infty & \text{otherwise.} \end{cases} \tag{12}$$

Therefore, the dual problem of (9) is written in

$$\begin{aligned} \max_{\mathbf{u}_0,u_g\ge 0\ \forall g} g(\mathbf{u}_0, u_g) &= \mathbf{u}_0^T\mathbf{x} - \frac{1}{4\lambda}\sum_{g\in\mathcal{G}} u_g^2 \\ \text{such that} \quad \|\mathbf{u}_0\|_2 &\le 1;\ \ \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty \le u_g \ \ \forall g \in \mathcal{G}. \end{aligned} \tag{13}$$

Note that the nonnegativity constraint of u_g is satisfied by the inequality in (13). Also, it is easy to see that for any feasible $\mathbf{u}_0$, the best choice of u_g (i.e., the one that maximizes the objective) is $u_g = \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty$. Therefore, the dual problem is equivalent to (14). As stated in this paper, we sample a set of s feasible vectors $\mathbf{u}_0$ to form the set $\mathcal{U}$, which is compiled offline and is used for all particles in all frames. Then, the optimal objective $\max_{\|\mathbf{u}_0\|_2\le 1} g(\mathbf{u}_0)$ is approximated with $\max_{\mathbf{u}_0\in\mathcal{U}} g(\mathbf{u}_0)$. A theoretical study of how good this approximation is (i.e., how close the approximation is to the actual optimal value) is kept for future work

$$\begin{aligned} \max_{\mathbf{u}_0} \quad g(\mathbf{u}_0) &= \mathbf{u}_0^T\mathbf{x} - \frac{1}{4\lambda}\sum_{g\in\mathcal{G}} \|\mathbf{D}_g^T\mathbf{u}_0\|_\infty^2 \\ \text{such that} \quad \|\mathbf{u}_0\|_2 &\le 1. \end{aligned} \tag{14}$$

References

[1] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[2] Y. Alper, J. Omar, and S. Mubarak, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–13, 2006.

[3] S. Salti, A. Cavallaro, and L. D. Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4334–4348, Oct. 2012.

[4] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, 2013, pp. 2411–2418.

[5] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l_1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1257–1264.

[6] B. Liu, J. Huang, L. Yang, and C. Kulikowski, "Robust visual tracking with local sparse appearance model and K-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1313–1320.

[7] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 2042–2049.

[8] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 1830–1837.

[9] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking with compressed sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1305–1312.

[10] T. Zhang, C. Jia, C. Xu, Y. Ma, and N. Ahuja, "Partial occlusion handling for visual tracking via robust part matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, 2014, pp. 1258–1265.

[11] Z. Husz, A. Wallace, and P. Green, "Tracking with a hierarchical partitioned particle filter and movement modelling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1571–1584, Dec. 2011.

[12] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 470–484.

[13] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 864–877.

[14] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.

[15] M. J. Black and A. D. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 63–84, 1998.

[16] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, Jul. 2009.

[17] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 125–141, 2008.

[18] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, 2006, pp. 798–805.

[19] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 1269–1276.

[20] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., 2006, pp. 1–10.

[21] S. Avidan, "Ensemble tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, 2005, pp. 494–501.

[22] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 983–990.

[23] R. T. Collins and Y. Liu, "On-line selection of discriminative tracking features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Madison, WI, USA, 2003, pp. 346–352.

[24] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. Eur. Conf. Comput. Vis.*, Marseille, France, 2008, pp. 678–691.

[25] Z. Yin and R. Collins, "Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, 2008, pp. 1–8.

[26] 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, 2013.

[27] B. Liu *et al.*, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 1–14.

[28] T. Zhang, B. Ghanem, C. Xu, and N. Ahuja, "Object tracking by occlusion detection via structured sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Portland, OR, USA, 2013, pp. 1033–1040.

[29] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 1838–1845.

[30] J. D. Rincon, D. Makris, C. Urunuela, and J.-C. Nebel, "Tracking human position and lower body parts using Kalman and particle filters constrained by human biomechanics," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 26–37, Feb. 2011.

[31] N. Widynski, S. Dubuisson, and I. Bloch, "Integration of fuzzy spatial information in tracking based on particle filtering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 635–649, Jun. 2011.

[32] D. Wang, H. Lu, and M. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.

[33] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, "Visual tracking via discriminative sparse similarity map," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1872–1881, Apr. 2014.

[34] D. Wang and H. Lu, "Visual tracking via probability continuous outlier model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, 2014, pp. 3478–3485.

[35] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, 2013.

[36] Y. Zhou, R. Jin, and S. C. H. Hoi, "Exclusive lasso for multi-task feature selection," *J. Mach. Learn. Res.*, vol. 9, no. 1, pp. 988–995, 2010.

[37] X. Chen, X. Yuan, Q. Chen, S. Yan, and T.-S. Chua, "Multi-label visual classification with label exclusive context," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 834–841.

[38] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 1271–1278.

[39] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin Markov network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 975–982.

[40] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Minneapolis, MN, USA, 2007, pp. 1–8.

[41] H. Grabner, J. Matas, L. V. Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 1285–1292.

[42] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1177–1184.

[43] T. Zhang, B. Ghanem, and N. Ahuja, "Robust multi-object tracking via cross-domain contextual information for sports video analysis," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, 2012, pp. 985–988.

[44] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2011.

[45] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 480–493.

[46] S. P. Boyd and L. Vandenberghe, "Duality," in *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[47] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM J. Optim.*, vol. 1, no. 1, pp. 1–20, 2008.

[48] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 49–56.

[49] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

Tianzhu Zhang (M'08) received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011.

He is a Visiting Research Scientist with Advanced Digital Sciences Center, Singapore, and an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision and multimedia, such as action recognition, object classification, and object tracking.

Bernard Ghanem (M'04) received the Ph.D. degree from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2010.

He is currently an Assistant Professor with the CEMSE Division and a member of the Visual Computing Center at King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. He was a Senior Research Scientist at the University of Illinois Urbana-Champaign (UIUC), Singapore, where he still holds an adjunct position. He heads projects that develop algorithms in computer vision, machine learning, and convex optimization geared toward real-world applications, including semantic video analysis in sports and automated surveillance, content-based image retrieval, large-scale object recognition, and 2-D/3-D scene understanding

Si Liu (M'08) received the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012.

She is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences. She was a Research Fellow with the Learning and Vision Group, National University of Singapore, Singapore. Her current research interests include computer vision and multimedia.

Changsheng Xu (F'13) received the Ph.D. degree from Tsinghua University, Beijing, China, in 1996.

He is a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and an Executive Director of China-Singapore Institute of Digital Media, Singapore. His current research interests include multimedia content analysis, pattern recognition, and computer vision.

Dr. Xu is an Associate Editor of *ACM Transactions on Multimedia Computing, Communications and Applications* and the IEEE TRANSACTIONS ON MULTIMEDIA. He served as a Program Chair of *ACM Multimedia* 2009.

Narendra Ahuja (F'92) received the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 1979.

He is the Donald Biggar Willet Professor with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Prof. Ahuja is on the Editorial Boards of several journals. He was the Founding Director of the International Institute of Information Technology, Hyderabad, Hyderabad, India, where he continues to serve as a Director International. He is a fellow of the American Association for Artificial Intelligence, the International Association for Pattern Recognition, the Association for Computing Machinery, the American Association for the Advancement of Science, and the International Society for Optical Engineering.