Visual Tracking via Sparse and Local Linear Coding

Guofeng Wang, Xueying Qin, Fan Zhong, Yue Liu, Hongbo Li, Qunsheng Peng,
and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— The state search is an important component of any object tracking algorithm. Numerous algorithms have been proposed, but stochastic sampling methods (e.g., particle filters) are arguably one of the most effective approaches. However, the discretization of the state space complicates the search for the precise object location. In this paper, we propose a novel tracking algorithm that extends the state space of particle observations from discrete to continuous. The solution is determined accurately via iterative linear coding between two convex hulls. The algorithm is modeled by an optimal function, which can be efficiently solved by either convex sparse coding or locality constrained linear coding. The algorithm is also very flexible and can be combined with many generic object representations. Thus, we first use sparse representation to achieve an efficient searching mechanism of the algorithm and demonstrate its accuracy. Next, two other object representation models, i.e., least soft-threshold squares and adaptive structural local sparse appearance, are implemented with improved accuracy to demonstrate the flexibility of our algorithm. Qualitative and quantitative experimental results demonstrate that the proposed tracking algorithm performs favorably against the state-of-the-art methods in dynamic scenes.**

***Index Terms*— State space search, convex sparse coding, locality-constrained linear coding, visual tracking.**

I. INTRODUCTION

VISUAL tracking is a classic problem in computer vision, and numerous algorithms have been developed for a wide range of visual tracking applications. Broadly speaking, the main components of a visual tracking method are object representation, a motion model and a search mechanism. Object representation refers to the description of the appearance of the target; the motion model is used to describe how the object moves between frames with state prediction; and the search mechanism is used to determine the object location. In some approaches, some of these tasks may be performed implicitly. While most algorithms focus on the object representation model, in this paper, we mainly focus on the searching mechanism, which is also of crucial importance for the accuracy and speed of visual tracking.

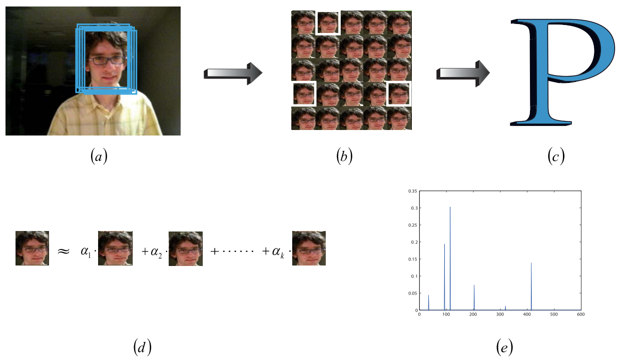


Fig. 1. Linear coding for object location. (a) Observed images corresponding to the drawn particles. (b) Cropped images. (c) Dictionary P formed by the observations. (d) The target object appearance in the current frame is sparsely represented by the observations corresponding to the drawn particles. (e) Coefficients of the proposed linear representation.

State searches for object tracking have been based on gradient descent [1], [2], stochastic sampling [3]–[5], and dense sampling [6]–[8], among other methods. Although states can be estimated by gradient descent methods efficiently, frequently only local optimal solutions are computed. To address this problem, stochastic or dense sampling algorithms have been exploited in recent tracking algorithms. However, either of these approaches requires drawing hundreds of samples in the state space to alleviate the local optimum issue and consequently entails solving numerous optimization problems [4], [5]. Despite the demonstrated success of these sampling methods for visual tracking, the discretized state space makes it difficult to search the object state precisely due to the omission of the sampling points; in addition, the use of dense sampled particles may also increase the computational complexity significantly [4], [5], [9]–[11].

In this paper, we propose a tracking algorithm based on a novel stochastic sampling algorithm for an effective and efficient state search. The target appearance of a frame is modeled by a linear combination of the observations corresponding to particles drawn stochastically in an image (Figure 1). Using this formulation, we demonstrate that the discretized state space can be modeled by a continuous space as for the space of the corresponding observations, which produces greater

Manuscript received September 15, 2014; revised February 25, 2015 and May 12, 2015; accepted May 26, 2015. Date of publication June 30, 2015; date of current version July 24, 2015. This work was supported in part by the 973 Program of China under Grant 2015CB352502, and in part by the 863 Program of China under Grant 2015AA016405, in part by the National Natural Science Foundation of China under Grant 61173070 and Grant 61202149. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Janusz Konrad. *(Corresponding author: Xueying Qin.)*

G. Wang, X. Qin, and F. Zhong are with the School of Computer Science and Technology, Shandong University, Jinan 250100, China, and also with the Engineering Research Center of Digital Media Technology, Ministry of Education, Jinan 250100, China (e-mail: wangguofeng525@gmail.com; qxy@sdu.edu.cn; zhongfan@sdu.edu.cn).

Y. Liu and H. Li are with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yliu_amss@126.com; hli@mmrc.iss.ac.cn).

Q. Peng is with the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University, Hangzhou 310027, China (e-mail: peng@cad.zju.edu.cn).

M.-H. Yang is with the Department of Electrical Engineering and Computer Science, University of California, Merced, CA 95344 USA (e-mail: mhyang@ucmerced.edu).

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/TIP.2015.2445291

precision than particle filters. In addition, unlike particle filters, the proposed method does not need to evaluate each particle individually, which may be time consuming [4], [5], [9]–[11]. We directly find the object state in an iterative fashion, which can be determined efficiently via iterative linear coding between two convex hulls including the space of particle observations and object template. For concreteness, two specific linear coding methods, i.e., convex sparse coding [12] and locality-constrained linear coding [13], are exploited for object tracking. Although the proposed algorithm is mainly concerned with the search mechanism, it can be incorporated with many generic object representations [5], [14]. In this work, we first use sparse representation to illustrate the algorithm and demonstrate its accuracy. Next, two other object representation models, i.e., least soft-threshold squares [14] and adaptive structural local sparse appearance [15], are implemented with improved accuracy, confirming the flexibility of our algorithm.

Online updating of the object template is an important component for robust visual tracking. Although straightforward online update methods can alleviate the tracking drift problems, we present an adaptive method based on reconstruction errors between two frames. The templates are updated only when the object appearance undergoes a significant change, and then a set of basis of the previous tracked objects are learned instead of directly using the previous tracked objects themselves. We demonstrate that this update method significantly facilitates robust tracking.

The main contributions of this work are summarized as follows:

- A novel state search algorithm is proposed in which the target appearance is modeled by a linear sparse combination of image observations corresponding to particles drawn by stochastic sampling. Using this formulation, the state space can be modeled by a continuous space from which the optimal state can be determined by an effective and efficient algorithm in a few iterations.
- The proposed convex sparse coding and locality-constrained linear coding algorithms greatly facilitate accurate state search than particle filters.
- An adaptive object template update method is proposed that helps alleviate the tracking drift problem compared to existing approaches.

II. Related Work

A thorough review of the rich literature is presented in [16], [17]. In this section, we discuss the methods most related to this work.

Existing tracking methods can be broadly categorized into generative and discriminative approaches. Generative tracking methods locate an object position by searching for the image region corresponding to a state that can be reconstructed by the current model with minimal error. Typical target objects are modeled by color histograms [1], covariance matrices [18], subspace models [4], and sparse representations [5], [9]–[11], [15], [19], [20]. Mei and Ling [5] use sparse representations of holistic templates from a foreground object and trivial background patterns to determine the best target image region with minimal reconstruction errors. Because this algorithm involves solving one ℓ_1 minimization problem for each particle, the time complexity is significant. Several algorithms have subsequently been developed to reduce the time complexity by feature selection [10], compressive sensing to reduce the dimension [19], and efficient optimization algorithm [9]. However, these algorithms still require considerable computational load for real-time applications. Other popular generative methods include multi-task sparse learning tracker [20], adaptive structured local sparse tracker [15], and sparsity-based collaborative tracker [11].

Discriminative approaches pose the tracking problem as a classification task with a local search based on prior object location to separate a foreground region from the background [6]–[8], [21]–[24]. In [21], an optical flow approach with a support vector machine classifier is proposed for object tracking. In [22], the most discriminative features are selected online to best separate target object pixels from the background. In [6], an online boosting method for tracking is developed to select discriminative Haar-like features for foreground and background separation, in which the state is determined based on dense local sampling. In [7], a multiple instance learning method is proposed in which samples are considered within positive and negative bags, which can address the ambiguity of samples. Other popular discriminative methods include tracking-learning-detection [24], struck tracker [23], and compressive tracker [8].

For visual tracking, online updating of generative models [4], [5], [18], [25] or discriminative classifiers [6], [7], [22] are effective for handling object appearance changes. Generative model updates are mainly based on adding the most recent tracking result to the models and discarding the old ones. For example, Ross *et al.* [4] use an incremental subspace learning method to represent the most recent object appearance in the frame without determining whether the newly observed image is occluded. Kwon and Lee [25] utilize the sparse principal component analysis method to update an object model in every frame. However, frequent updating may adversely cause tracking drift problems. Discriminative classifiers are updated based on online learning algorithms. For example, Grabner and Bischof [6] use the online boosting algorithm to update the classifier with the most recent features [6]. Babenko *et al.* [7] developed an online multiple instance learning algorithm to update the classifier.

III. General Particle-Based Tracking Model

Visual tracking can be cast as a Bayesian inference problem related to the state space and observation space. The state space describes the parameters of the target object, e.g. affine state space [4], [5], [9], while the observation space represents the object's appearance and is employed to determine the optimal state of an object. Thus, there exists a mapping between the state space and observation space in every frame, as illustrated in Figure 2.

In the tracking process, a particle filter is commonly used to sample particles, thereby discretizing the state space and,

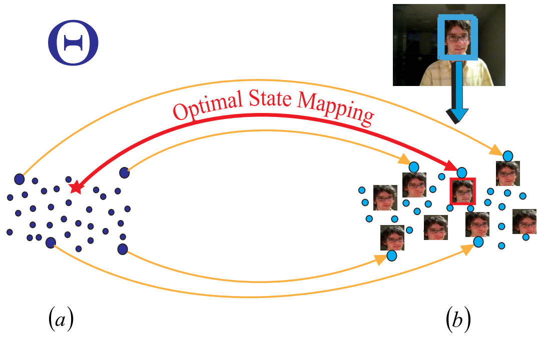


Fig. 2. Mapping between the state space and observation space in every frame. The orange and red arrow lines indicate the mapping from the state space to the observation space, where the red arrow line indicates the optimal state mapping. (a) The state space constructed by Θ. (b) The observation space constructed by particle observations.

correspondingly, discretizing the observation space as well. In the observation space, each particle has a corresponding vectorized gray-scale image observation, $\mathbf{p}_i \in \mathbf{R}^d$. These observations form a dictionary: $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_n] \in \mathbf{R}^{d\times n}$. To determine the optimal state of the object in the current frame, we search for the state in which the corresponding appearance is most similar to the object template model. Consequently, it is very important to measure the similarity between each particle's observation and the object templates. The maximum a posteriori probability (MAP) estimation is then usually used to determine the optimal state of a particle and to finally determine the object's location in the current frame. Therefore, the object tracking process mentioned above relies on the state space and observation space models.

A. State Space Model

Many state space models can be used in tracking, with the affine state space typically used in a particle filter. The affine state describing the parameters of an object is typically denoted by $\Theta_t = (x_t, y_t, \eta_t, s_t, \beta_t, \theta_t)$, which comprises the x and y translations, rotation angle, scale, aspect ratio, and skew direction at time t, respectively. A Gaussian function is usually used to model the motion distribution in the state space, and the affine parameters are assumed to be independent, with the result that $p(\Theta_t|\Theta_{t-1}) = N(\Theta_t; \Theta_{t-1}, \Lambda)$, where Λ is the variance matrix of these parameters. Particles in the state space are usually distributed around the state of the last frame according to the Gaussian function and are then mapped onto the observation space of the current frame (Figure 2). Finally, the optimal state is identified as the state whose corresponding appearance is the most similar to the object template model in the observation space.

B. Observation Space Model of Particle Filter

Object tracking involves identifying the target in the observation space that is visually similar to the templates. For particle filter methods, the particle in the observation space that is the most similar to the templates is the best choice for the target. Thus, the object should be represented efficiently to determine which particle should be selected as the target. Furthermore, the similarity measurement between each sampled particle and the object template in the observation space plays a very important role.

We reformulate this mechanism as a new functional energy minimization problem that is equivalent to the traditional particle filter (MAP) as follows:

$$\begin{aligned}\{\hat{\alpha}, \hat{\mathbf{x}}_t\} &= \arg\min_{\alpha,\mathbf{x}_t} \ \|\mathbf{x}_t - \mathbf{P}\alpha\|_2^2 + D(\mathbf{x}_t, \mathcal{T}),\\ &\quad \text{s.t. } \mathbf{x}_t \in \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}, \quad Card(\alpha) = 1,\\ &\qquad \forall i, \quad \alpha_i \geq 0, \quad \sum_i \alpha_i = 1, \end{aligned} \tag{1}$$

where α is a nonnegative coefficient with respect to the dictionary; $Card(\alpha) = 1$ is the cardinality constraint requiring that only one element of α is nonzero; $\mathbf{x}_t$ is the object observation in the current frame; $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ is the set of observations associated with particles in the current frame; $\mathcal{T}$ is the object template set, indicating one or more object templates; and $D(\cdot\,,\cdot)$ is the metric for measuring the distance between the observation and object templates, for which the Euclidean distance is usually used. Figure 3(a) illustrates Equation 1 geometrically.

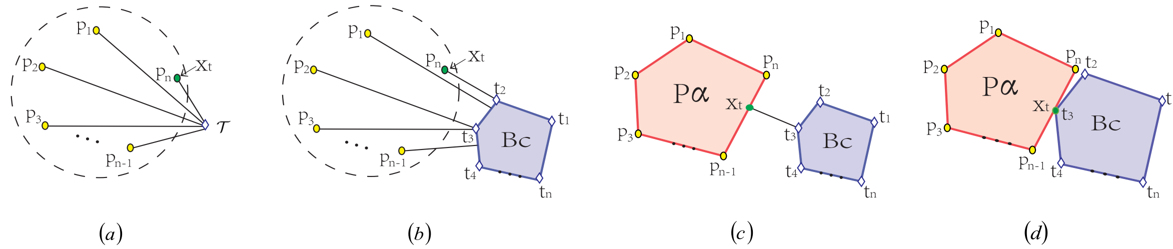
In Equation 1, the cost function consists of two terms: the particle term, which measures the similarity of the target and the particles, and the template term, which measures the similarity of the target and the templates. As the constraint, $Card(\alpha) = 1$ indicates that the current object, $\mathbf{x}_t$, must be represented by one of the particles; thus, the first term should be exactly equal to zero in this case. After resolving Equation 1, target $\hat{\mathbf{x}}_t$ should be very similar to some particle and the templates, in which similarity is measured by a measurement function. In fact, different metrics lead to different results. Therefore, from the particle term, target $\hat{\mathbf{x}}_t$ should actually be represented by $\mathbf{P}\alpha$, and from the template term, target $\hat{\mathbf{x}}_t$ should be approximately represented by a function of the templates $\mathcal{T}$. The function $D(\mathbf{x}_t, \mathcal{T})$ of the templates can be realized by many different representations, e.g., subspace representation [4] or sparse representation [5], [9], among others [14]. From this point of view, the particle filter requires the target to be represented by some particle and the templates.

Equation 1 is equivalent to the traditional MAP-based particle filter; in fact, it implies a new insight that the target appearance can be a linear sparse combination of the particle observations. As shown later, this linear property is very important in visual tracking and is used to define a new state search algorithm that is more effective and efficient than particle filters.

C. Sparse Representation Model of Templates

As mentioned above, the template term $D(\mathbf{x}_t, \mathcal{T})$ in Equation 1 implies an object representation model. This model can be realized by many different representations, including the subspace representation model [4] and the sparse representation model [5], [9].

The sparse representation model [5] is effective in representing object appearance, particularly in environments with occlusion and noise. Therefore, we first introduce this model into Equation 1 to illustrate the algorithm. Next, two other object representation models, i.e., least



(*a*) (*b*) (*c*) (*d*)

Fig. 3. Illustration from particle filter to linear coding for visual tracking. (a) The general particle filter model, in which each particle $\mathbf{p}_i$ needs to compute the distance to the template $\mathcal{T}$. (b) The distance metric between each particle $\mathbf{p}_i$ and templates' continuous space **Bc** (light blue color) in the sense of ℓ_1 norm. (c) The distance metric between object's continuous space $\mathbf{P}\alpha$ (light red color) and templates' continuous space **Bc** in the sense of ℓ_1 norm. (d) The object's continuous space $\mathbf{P}\alpha$ in touch with the templates' continuous space **Bc** in the sense of ℓ_1 norm.

soft-threshold squares and adaptive structural local sparse appearance, are implemented with improved accuracy.

Suppose that $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_n] \in \mathbf{R}^{d\times n} (d \gg n)$ and $\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, \cdots, \mathbf{i}_d] \in \mathbf{R}^{d\times d}$ are a set of object templates and trivial templates, respectively. With the constraint of non-negativity, the basis can be represented as

$$\mathbf{B} = [\mathbf{T}, \mathbf{I}, -\mathbf{I}] \in \mathbf{R}^{d\times(n+2d)}. \qquad (2)$$

Thus, the term $D(\mathbf{x}_t, \mathcal{T})$ becomes

$$D(\mathbf{x}_t, \mathcal{T}) = \parallel \mathbf{x}_t - \mathbf{Bc} \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1, \quad s.t.\ \mathbf{c} \geq 0, \qquad (3)$$

where **B** is a dictionary comprising the object templates and trivial templates and, with ℓ_1 regularization, **c** should satisfy the nonnegative and sparse constraints, $\mathbf{c}^\top = [\beta, \mathbf{e}^+, \mathbf{e}^-] \in \mathbf{R}_+^{n+2d}$. This representation aims to identify a linear sparse combination of the basis that best represents the object observation in the current frame.

Replacing term $D(\mathbf{x}_t, \mathcal{T})$ in Equation 1 with Equation 3, we obtain

$$\begin{aligned} \{\hat{\alpha}, \hat{\mathbf{x}}_t\} = \arg\min_{\alpha, \mathbf{x}_t} \ & \|\mathbf{x}_t - \mathbf{P}\alpha\|_2^2 + \parallel \mathbf{x}_t - \mathbf{Bc} \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1, \\ \text{s.t. } & \mathbf{x}_t \in \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}, \quad Card(\alpha) = 1, \\ & \forall i, \quad \alpha_i \geq 0, \quad \sum_i \alpha_i = 1, \quad \mathbf{c} \geq 0. \end{aligned} \qquad (4)$$

The geometric explanation of Equation 4 is illustrated in Figure 3(b), where $\mathcal{T}$ in Figure 3(a) is replaced by a special term, **Bc**. To obtain target $\hat{\mathbf{x}}_t$ in this formulation, we must measure the distance between each particle's observation $\mathbf{p}_i$ and the object template space **Bc**, as shown in Figure 3(b), which leads to high computational overhead if each calculation entails solving an ℓ_1 optimization problem [5].

Moreover, in Equation 4, the representation of target $\hat{\mathbf{x}}_t$ in the particle term is still discretized as $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$. As shown in the next section, this discretization can be relaxed to become continuous through the sparse linear representation of **P**. The solution can then be determined efficiently via iterative linear coding between two convex hulls $\mathbf{P}\alpha$ and **Bc**, instead of solving each individual ℓ_1.

IV. Visual Tracking by Sparse and Local Linear Coding

Intuitively, representation of an object in continuous space should be more precise than in discrete space. In Equation 4, target $\mathbf{x}_t$ belongs to one of the particle's observations $\mathbf{p}_i$ due to the constraint $Card(\alpha) = 1$. However, this constraint is too strict. The constraint would be more flexible if target $\mathbf{x}_t$ can be represented by the linear combination of **P**. We argue that this characteristic is very important for visual tracking and produces a novel tracking algorithm based on iterative linear coding. Specifically, two linear codings are proposed in the next sub-sections: convex sparse coding (CSC) and locality-constrained linear coding (LLC).

A. CSC-Based Visual Tracking

Equation 4 is equal to the original ℓ_1 tracker [5], in which the particles' state space is in a discrete space. We will demonstrate how this discrete state space can be extended to a continuous space and how the calculation is greatly accelerated by this extension.

We relax the constraint $Card(\alpha)$ in Equation 4 by placing an ℓ_1-norm regularization on α, which indicates that the coefficient of α can contain a small number of nonzero elements rather than only one, as follows:

$$\begin{aligned} \{\hat{\alpha}, \hat{\mathbf{c}}, \hat{\mathbf{x}}_t\} = \arg\min_{\alpha, \mathbf{c}, \mathbf{x}_t} \ & \parallel \mathbf{x}_t - \mathbf{P}\alpha \parallel_2^2 + \parallel \mathbf{x}_t - \mathbf{Bc} \parallel_2^2 \\ & + \mu \parallel \alpha \parallel_1 + \lambda \parallel \mathbf{c} \parallel_1, \\ s.t.\ \forall i, \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i = 1, \quad \mathbf{c} \geq 0, \end{aligned} \qquad (5)$$

where $\parallel \cdot \parallel_1$ denotes the ℓ_1 norm that guarantees the sparsity of α and **c**. This equation indicates that the object in the current frame is not only sparsely represented by its ambient particle observations but can also be expected to be sparsely represented by the object templates and trivial templates. More specifically, this type of representation is in a continuous space when α is relaxed to a positive real number. Therefore, the efficiency of the resolving process is also a key problem.

In Equation 5, the first and second terms can be considered the two edges of a triangle. The third edge of the triangle is $\parallel \mathbf{P}\alpha - \mathbf{Bc} \parallel_2^2$. Therefore, the minimum of Equation 5 is reached when $\parallel \mathbf{x}_t - \mathbf{P}\alpha \parallel_2^2 + \parallel \mathbf{x}_t - \mathbf{Bc} \parallel_2^2 = \parallel \mathbf{P}\alpha - \mathbf{Bc} \parallel_2^2$, in which case, Equation 5 can be formulated as

$$\begin{aligned} \{\hat{\alpha}, \hat{\mathbf{c}}\} = \arg\min_{\alpha, \mathbf{c}} \ & \parallel \mathbf{P}\alpha - \mathbf{Bc} \parallel_2^2 + \mu \parallel \alpha \parallel_1 + \lambda \parallel \mathbf{c} \parallel_1, \\ s.t.\ \forall i, \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i = 1, \quad \mathbf{c} \geq 0. \end{aligned} \qquad (6)$$

Equation 6 can be considered as the Euclidean distance between two convex hulls $\mathbf{P}\alpha$ and **Bc**; see **Proposition 1** in the Appendix. The geometrical explanation of Equation 6 is illustrated in Figure 3(c-d).

From Figure 3, we can observe that the strategy of the particle filter is that all of the particles (vertices of the space $\mathbf{P}\alpha$) must calculate the distance to the continuous space $\mathbf{Bc}$ (light blue color). With our method, particles are placed into a continuous and convex space $\mathbf{P}\alpha$ (light red color in Figures 3(c-d)); hence, the solution can be searched more efficiently by iteratively updating the result between the two convex hulls $\mathbf{P}\alpha$ and $\mathbf{Bc}$ until the minimum distance is attained, as illustrated in 3(c). Note that, in theory, the spaces $\mathbf{P}\alpha$ and $\mathbf{Bc}$ may be in contact (Figure 3(d)), with the result that $\mathbf{P}\alpha = \mathbf{Bc}$.

Although the formulation is only slightly revised in Equations 4 to 6, the tracking problem has changed considerably and requires a different solution strategy. In fact, in the particle filter, the mean square error (MSE) estimation is also adopted to represent the target in the current frame by a combination of all the particles. However, in MSE, each particle's observation $\mathbf{p}_i$ must be compared with the template space $\mathbf{Bc}$, which still must be computed for each particle, leading to high computational load.

Equation 6 can be solved efficiently by alternately updating α and $\mathbf{c}$. Initially, we let $\mathbf{P}\alpha = \mathbf{x}^*_{t-1}$, initializing the iterations with the previous target. The following standard ℓ_1 problem is then solved to obtain $\hat{\mathbf{c}}$:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \parallel \mathbf{x}^*_{t-1} - \mathbf{Bc} \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1 . \tag{7}$$

Various methods have been developed to solve this ℓ_1 problem, e.g., Lasso [12]. Given $\hat{\mathbf{c}}$, we then fix $\hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{c}}$ to solve the CSC problem of Equation 6:

$$\hat{\alpha} = \arg\min_{\alpha} \parallel \hat{\mathbf{y}} - \mathbf{P}\alpha \parallel_2^2 + \mu \parallel \alpha \parallel_1 . \tag{8}$$

These two procedures are executed alternately to update $\hat{\alpha}$ and $\hat{\mathbf{c}}$ and form an iterative process that gradually approximates the minima of Equation 6. In our experiments, this process typically converges in only a few iterations, demonstrating that only a few sparse coding procedures are involved. Consequently, the time complexity is much lower than that of the particle filter-based tracker.

The solved coefficient vector $\hat{\alpha}$ actually measures the correlation of each particle with the tracked object. To obtain the optimal state of the object in the current frame, $\hat{\alpha}$ is used as the weights to combine the states (affine parameters) related to the elements of $\mathbf{P}$, and the resulting optimal state is then used to locate the object in the target frame by transforming the tracking window from the source frame; Finally, the object target $\mathbf{x}^*_t$ is cropped out.

B. LLC-Based Visual Tracking

In practice, the CSC method generally performs well in many cases, but the solution of CSC cannot guarantee the locality of the selected particles [13], [26]; as shown in Figure 4(a), $\mathbf{p}_3$ and $\mathbf{p}_n$ (purple color) may be selected to represent the object in the current frame, but intuitively, $\mathbf{p}_{n-1}$ and $\mathbf{p}_n$ (Figure 4(b) purple color) are more suitable for object tracking because these two selected particles are closer to the object template space $\mathbf{Bc}$. Recently, Yu *et al.* [26] proposed Local Coordinate Coding (LCC), a modification of CSC that explicitly encourages local coding because theoretically, under certain assumptions, locality is more essential than sparsity. They presented the novel LLC coding [13], which can be viewed as a rapid implementation of LCC. Using LLC, the solution encourages the found particles to be not only sparse but also local to the object template for object representation, as in Figure 4(b). The relaxation of Equation 4 used in LLC can be written as

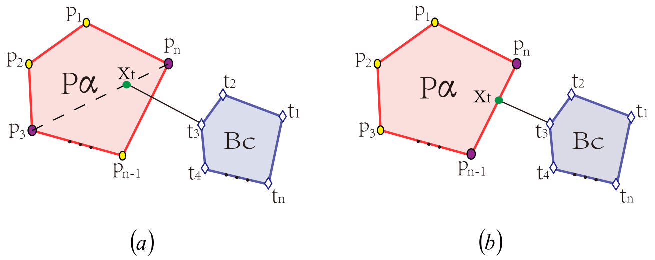


Fig. 4. Illustration of convex sparse coding and locality-constrained linear coding for object tracking. (a) Convex sparse coding, in which particles $\mathbf{p}_3$ and $\mathbf{p}_n$ (purple color) are selected for object tracking. (b) Locality-constrained linear coding, in which particles $\mathbf{p}_{n-1}$ and $\mathbf{p}_n$ (purple color) are selected for object tracking.

$$\begin{aligned}\{\hat{\alpha}, \hat{\mathbf{c}}, \hat{\mathbf{x}}_t\} = \arg\min_{\alpha,\mathbf{c},\mathbf{x}_t} \ & \parallel \mathbf{x}_t - \mathbf{P}\alpha \parallel_2^2 + \parallel \mathbf{x}_t - \mathbf{Bc} \parallel_2^2 \\ & + \mu \parallel \mathbf{w} \odot \alpha \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1 \\ s.t.\ \forall i, \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i = 1, \quad \mathbf{c} \geq 0,\end{aligned} \tag{9}$$

where $\odot$ denotes the element-wise multiplication and $\mathbf{w}$ indicates whether the particle $\mathbf{p}_i$ is close to the template space $\mathbf{Bc}$ (can be computed by K-NN); if close, then its correspondence coefficient of $\mathbf{w}$ is 1 and is 0 otherwise. $\mathbf{w}$ guarantees α is not sparse in the ℓ_1 norm sense but in the sense that the solution has only a few significant values ($\mathbf{p}_{n-1}$ and $\mathbf{p}_n$ in Figure 4(b)). LLC can be computed efficiently using analytical means [13], which can enable real-time object tracking.

Similar to Equation 5, Equation 9 can be formulated further as

$$\begin{aligned}\{\hat{\alpha}, \hat{\mathbf{c}}\} = \arg\min_{\alpha,\mathbf{c}} \ & \parallel \mathbf{P}\alpha - \mathbf{Bc} \parallel_2^2 + \mu \parallel \mathbf{w} \odot \alpha \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1, \\ & s.t.\ \forall i, \alpha_i \geq 0, \sum_i \alpha_i = 1, \ \mathbf{c} \geq 0.\end{aligned} \tag{10}$$

Equation 10 is considered as the metric distance between the two spaces $\mathbf{P}\alpha$ and $\mathbf{Bc}$ in the sense of LLC, as illustrated in Figure 4(b). From Figure 4(b), we can observe that with LLC, the particles identified for object representation are all close to the space $\mathbf{Bc}$, i.e., LLC is more suitable for object tracking.

Similar to Equation 6, Equation 10 can also be solved by alternately updating α and $\mathbf{c}$. Updating $\mathbf{c}$ is the standard ℓ_1 problem. Next, given $\hat{\mathbf{c}}$, we can fix $\hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{c}}$, and thus to update α, we can solve the following equation:

$$\hat{\alpha} = \arg\min_{\alpha} \parallel \hat{\mathbf{y}} - \mathbf{P}\alpha \parallel_2^2 + \mu \parallel \mathbf{w} \odot \alpha \parallel_2^2 . \tag{11}$$

This equation can be solved efficiently [13]. More details about our tracking method are provided in Algorithm 1.

Algorithm 1 Linear Coding for Visual Tracking

```
Input: Location in the first frame
Output: Location in every frame
For each frame t = 1 : T, T is the total number of frames.
    Sample the particles around the previous object location in the current frame.
    Every particle crops out a warp image, normalized to the standard template, and then all of the warped images are used to construct a dictionary P.
    Use Equation 6 or 10 to solve linear coding for object localization.
    The nonzero of the coefficients of particles are used to locate the object in the current frame.
    Update the object template model.
end for.
```

C. Object Template Updating

The appearance of an object typically changes dynamically for a variety of reasons, such as motion, occlusion, background, and illumination change. Constant object templates $\mathbf{T}$ obviously cannot address these cases effectively. A particle filter typically uses a heuristic strategy to update the object templates and thus is not able to deal effectively with the drifting problem. Consequently, online updating of the object template model is also very difficult, with a fundamental problem being when and how to update the template model.

An updating opportunity must be carefully selected and can be naturally related to changes in the object observation. The faster an object observation changes, the more frequently the templates must be updated. Therefore, a proper metric for measuring change is essential. Recall that in section IV, the tracked object without an occlusion part $\tilde{\mathbf{y}}_t$ is represented as $\mathbf{T}\hat{\beta}$; hence, if the object observation is not changed much, then $\tilde{\mathbf{y}}_t$ must be close to $\mathbf{x}^*_{t-1}$, or else they may differ greatly. Therefore, variation of the object can be measured by the difference between $\tilde{\mathbf{y}}_t$ and $\mathbf{x}^*_{t-1}$.

$$error = \| \tilde{\mathbf{x}}_t - \mathbf{x}^*_{t-1} \|_2^2 \, . \tag{12}$$

Our strategy to control the updating dynamically is to update the object template model only if the error is greater than a threshold τ, indicating that the object observation has undergone significant change.

To update the object template model, we can either use the previous tracked object directly as the new template model or use some learning-based methods instead of only the previous frame, such as principal component analysis (PCA) [4] and dictionary learning (DL). With the development of sparse representation, DL is powerful for representing objects [27], [28]. Thus, in this paper, we adopt the dictionary learning strategy for learning the basis and then use it to update the object templates. The initialization of the template set $\mathbf{T}$ is created by manually selecting the first template and then creating the rest of the templates by perturbing one pixel in four possible directions at the corner points of the first.

V. Other Object Representation Models

We have stated our searching mechanism in a continuous space for visual object tracking, which is very flexible and can be applied to many other object representation models. In this section, we introduce two more object representation models combined with our LLC searching mechanism, which can further improve the accuracy of the algorithm. Specifically, a least soft-threshold squares (LSS) object representation model [14] and an adaptive structural local sparse appearance (ASLA) model [15] are adopted.

A. Least Soft-Threshold Squares Model

As shown in [14], the object representation of the LSS model assumes that the tracked object is generated by a PCA subspace with i.i.d Gaussian-Laplacian noise, such that the object is represented by

$$\mathbf{x}_t = \mathbf{U}\mathbf{z} + \mathbf{n} + \mathbf{s}, \tag{13}$$

where $\mathbf{U}$ represents the basis vectors of PCA, $\mathbf{z}$ indicates the coefficients of basis vectors, $\mathbf{n}$ is the Gaussian noise component, and $\mathbf{s}$ is the Laplacian noise component. Thus, the term $D(\mathbf{x}_t, \mathcal{T})$ can be represented as

$$D(\mathbf{x}_t, \mathcal{T}) = \| \mathbf{x}_t - \mathbf{U}\mathbf{z} - \mathbf{s} \|_2^2 + \lambda \| \mathbf{s} \|_1, \tag{14}$$

Combined with our LLC searching mechanism, the whole tracking model can be illustrated as

$$\begin{aligned} \{\hat{\alpha}, \hat{\mathbf{z}}, \hat{\mathbf{s}}\} = \arg\min_{\alpha,\mathbf{z},\mathbf{s}} \quad & \| \mathbf{P}\alpha - \mathbf{U}\mathbf{z} - \mathbf{s} \|_2^2 + \mu \| \mathbf{w} \odot \alpha \|_2^2 \\ & + \lambda \| \mathbf{s} \|_1, \\ s.t. \ \forall i, \quad & \alpha_i \geq 0, \ \sum_i \alpha_i = 1. \end{aligned} \tag{15}$$

Similar to Equation 10, Equation 15 can also be considered as the metric distance between two spaces $\mathbf{P}\alpha$ and $\mathbf{U}\mathbf{z}$ in the sense of LLC. Equation 15 can be solved by alternately updating coefficients α and $\mathbf{z}, \mathbf{s}$. $\mathbf{z}$ and $\mathbf{s}$ can be solved by least soft-threshold squares; we refer readers to [14] for more details.

B. Adaptive Structural Local Sparse Appearance Model

As demonstrated in the benchmark proposed by [29], the local sparse representation is important for tracking regarding performance improvement compared with the holistic sparse representation (e.g., LSS model). In addition, ASLA also achieves high ranks in the benchmark. Therefore, we combined our LLC searching mechanism with the ASLA object representation model to further demonstrate the advantage of our algorithm.

The ASLA model involves the following steps: first a particle is evaluated by computing its local sparse coefficients; next, these coefficients are obtained by a new alignment-pooling technology; finally, the maximal sum of the gathered coefficients of the particle is chosen as the tracked object. Suppose that $\mathbf{v}_i$ is the i-th local patch coefficient of a particle and $\mathbf{V}$ is the square matrix in which each column comprises $\mathbf{v}_i$; the pooled feature $\mathbf{f}$ of each particle is then defined as

$$\mathbf{f} = diag(\mathbf{V}). \tag{16}$$

The score of each particle is defined as

$$s = \sum_{k=1}^{N} \mathbf{f}_k, \tag{17}$$

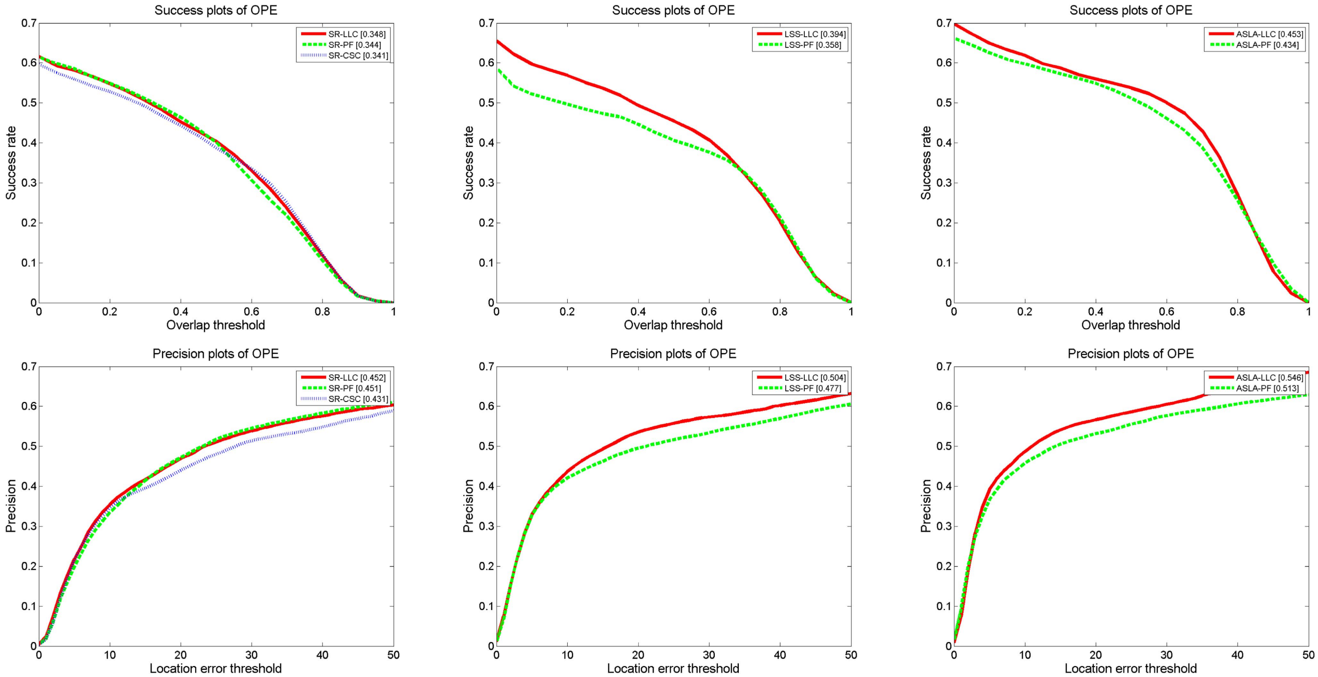


Fig. 5. The OPE evaluation the object appearance of SR, LSS, ASLA between particle filter and proposed convex sparse coding and locality-constrained linear coding methods.

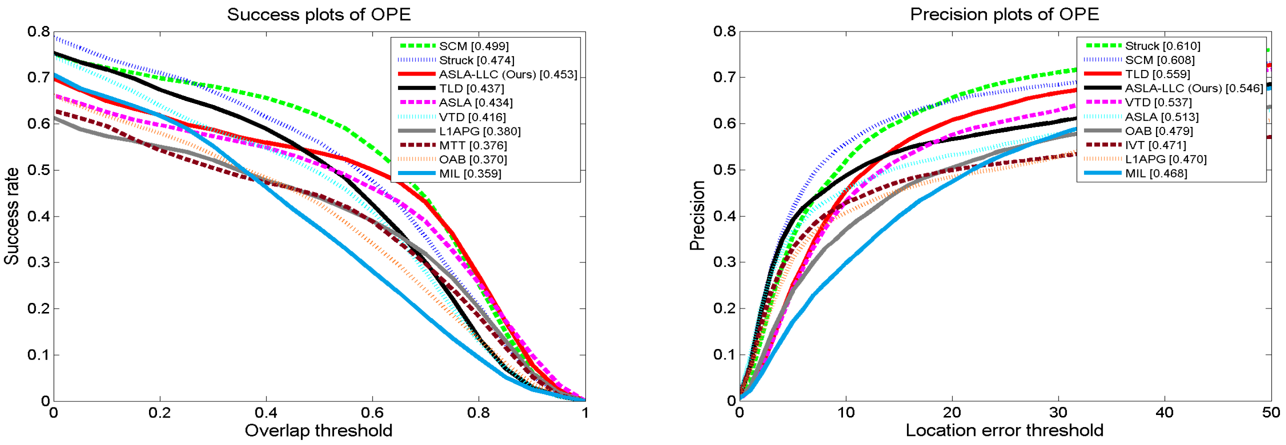


Fig. 6. The OPE evaluation of the proposed ASLA-LSS method compared with state-of-the-art methods.

where N is the dimension of feature vector $\mathbf{f}$. We refer readers to [15] for more details about this process.

As shown above, this process actually captures nonlinear mapping between an object and templates. Consequently, it is difficult to directly write the analysis function between the object $\mathbf{x}_t$ and templates $\mathbf{T}$; thus, we use a symbol φ as the *nonlinear* mapping function for this processing, defined as $s = \varphi(\mathbf{x}_t)$. Therefore, the term $D(\mathbf{x}_t, \mathcal{T})$ can be simply defined as

$$D(\mathbf{x}_t, \mathcal{T}) = -\varphi(\mathbf{x}_t). \tag{18}$$

Our strategy of combining ASLA and LLC is that when computing the *nonlinear* function φ, its reconstructed object $\hat{\mathbf{y}} = \varphi^{-1}(s)$ is only *part* of the object. For example, if the dimensions of object $\mathbf{x}_t$ is 32 × 32, then the dimensions of reconstructed object $\hat{\mathbf{y}}$ may be only 24 × 24; thus, it is more effective than using a holistic object when capturing an occlusion because local parches are used. The local patch can be chosen for reconstructing object $\hat{\mathbf{y}}$ by vector $\mathbf{f}$; if the patch of $\mathbf{x}_t$ can be represented by its corresponding templates $\mathbf{T}$'

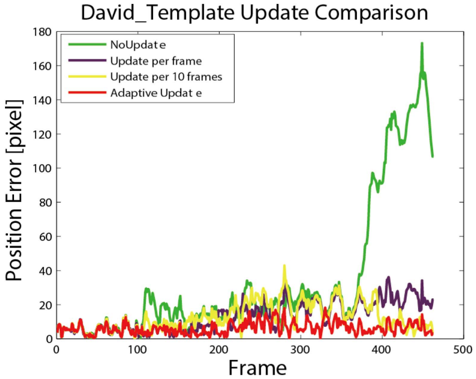


Fig. 7. Position error by using different updating frequencies.

patches well, then the value of its corresponding location of $\mathbf{f}$ is high, and vice versa. Thus, using this property, we can obtain only part of the object with a high score of $\mathbf{f}$ instead of a holistic object for more effective object tracking.

VI. Experiments

To examine the effectiveness of the proposed tracking method, we tested it on an online benchmark [29], which includes 51 challenge sequences, and compared it with 29 other state-of-the-art algorithms proposed by [29]. For conciseness, we use the following abbreviations: our proposed sparse representation-based CSC model as SR-CSC, the sparse representation-based LLC model as SR-LLC, the LSS-based LLC model as LSS-LLC, and the ASLA-based LLC model as ASLA-LLC. Similarly, the following abbreviations are used: the origin sparse representation-based particle filter (PF) model as SR-PF, LSS-based PF model as LSS-PF, and ASLA-based PF model as ASLA-PF. We implement SR-PF, SR-CSC and SR-LLC in C++ and LSS-PF, LSS-LLC, ASLA-PF, ASLA-LLC in Matlab, where the CSC code uses a SPAMS package[1] [28], and LLC in [13]. All algorithms are executed on a personal computer with an Intel i5 3.2 GHz CPU and 8 GB of RAM.

The parameters are set as follows: for the SR-CSC and SR-LLC models, Λ is set to $\{5, 5, 0.0005, 0.02, 0.002, 0.0005\}$, the regularization constants μ and λ in Equation 6 are set to 0.06, μ and λ in Equation 10 are set to 0.0001 and 0.06, respectively, τ is set to 0.5, and the number of object templates is set to 10; for the LSS-LLC model and ASLA-LLC model, the parameters are the same as in the original LSS model [14] and ASLA model [15].

A. Performance Analysis

In our method, for the SR-CSC and SR-LLC trackers, 600 particles are sampled for each frame, and the cropped image of each particle is resized to 20×20 pixels; hence, the matrix $\mathbf{P}$ is 400×600. We only require three to five passes of the linear coding iterations for each target, whereas the SR-PF tracker requires 600 sparse representation processes. Our test achieves a frame rate of 30-40 fps with our unoptimized code for both SR-CSC and SR-LLC trackers, whereas the PF tracker can achieve only approximately 4.1 fps with all the same parameters setting; therefore, our method is approximately 10 times faster than the PF tracker. For the accuracy test, we set the same parameters for the SR-PF, SR-CSC and SR-LLC methods, and as shown in Figure 5(a), they can reach accuracies of 0.348, 0.344, and 0.341, respectively, in the success plot and 0.452, 0.451, 0.431, respectively, in the precision plot. Thus, our SR-LLC tracker is not only faster but also more accurate than the SR-PF tracker, and our SR-CSC tracker is much faster than the SR-PF tracker, although somewhat less accurate.

For the LSS-LLC tracker and the ASLA-LLC tracker, 600 particles are also sampled for each frame, and each particle is resized to 32×32 pixels, as is the case for the original LSS-PF and ASLA-PF trackers. As shown in Figure 5, the accuracies of LSS-LLC and ASLA-LLC are both higher than those of the LSS-PF and ASLA-PF trackers. For example, LSS-LLC and LSS-PF can reach accuracies of 0.394 and 0.358, respectively, in the success plot and 0.504 and 0.477, respectively, in the precision plot. In addition, ASLA-LLC and ASLA-PF can reach accuracies of 0.453 and 0.434, respectively, in the success plot and 0.546 and 0.513, respectively, in the precision plot. Thus, with a more complex object representation model, our LLC algorithm can be greatly improved to be more effective than the PF algorithm, further demonstrating that our proposed method can be combined with many generic object representation models. Regarding speed, the object representations in the original LSS-PF and ASLA-PF do not cost much time; thus, our LLC algorithm in these two tests do not significantly improve speed performance because it is only approximately 2-3 fps faster than the original trackers.

The dynamic object template update strategy improves the SR-based tracking performance. We examine the algorithm on the sequence of *David*. When David's appearance changes rapidly, our adaptive strategy should update every 1-2 frames. As shown in Figure 7, the adaptive updating strategy (in red color) performs much better than the case of never updating (in green color), updating in every frame (in purple color), and updating every 10 frames. All other strategies may cause drift of the tracking results.

B. Evaluation on Benchmark

The benchmark proposed by [29] contains 51 annotated sequences, which represents an up-to-date tracking evaluation criteria. These sequences are tagged with 11 attributes that evaluate different challenges of the sequences, e.g., illumination variation, occlusion, deformation, etc. In addition, the benchmark also provides the results of 29 trackers. Thus, we use the online available tracking results that contain all 29 trackers and the tool provided by [29] to compute the evaluation plots. We use our ASLA-LLC tracker for comparison because it has the highest score of our proposed methods.

In [29], the evaluation is based on two different metrics: the precision plot and the success plot. The precision plot evaluates the center location error, which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths, with the distance accepted by a given threshold of the ground truth (20 pixels in the paper). Another evaluation metric is the bounding box overlap, which is defined as $score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}$, where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. To further evaluate the precision, in the success plot, the ranking is based on the Area Under the Curve (AUC) instead of using a specific threshold. For more details about the benchmark, we refer readers to the original paper [29].

For comparison, we run the One-Pass Evaluation (OPE) [29] on the benchmark. As shown in Figure 6, the proposed ASLA-LLC method is among the three top-performing trackers using the measurement of success plot. Although the proposed ASLA-LLC method is not the top tracker, we believe

[1] http://www.di.ens.fr/willow/SPAMS/downloads.html

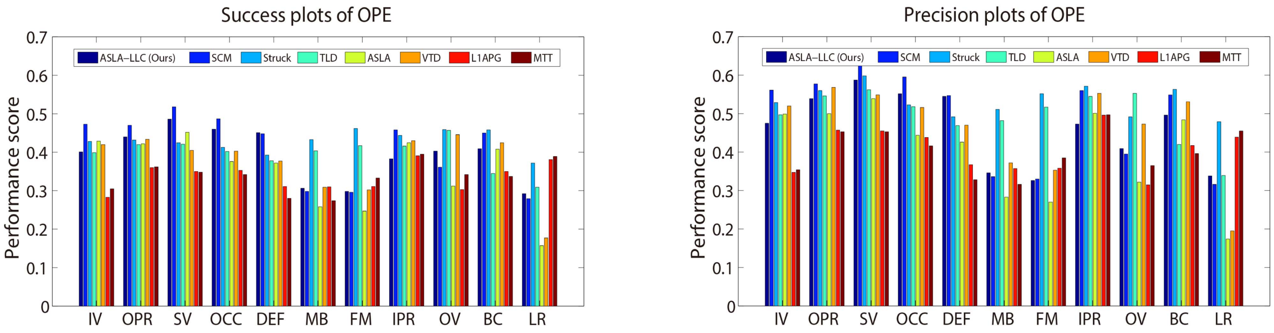


Fig. 8. Average performance ranking scores of test sequences in OPE. Left: performance ranking score based on success plots; right: performance ranking score based on precision plots. Each subset of sequences corresponds to an attribute, such as illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC), low resolution (LR).

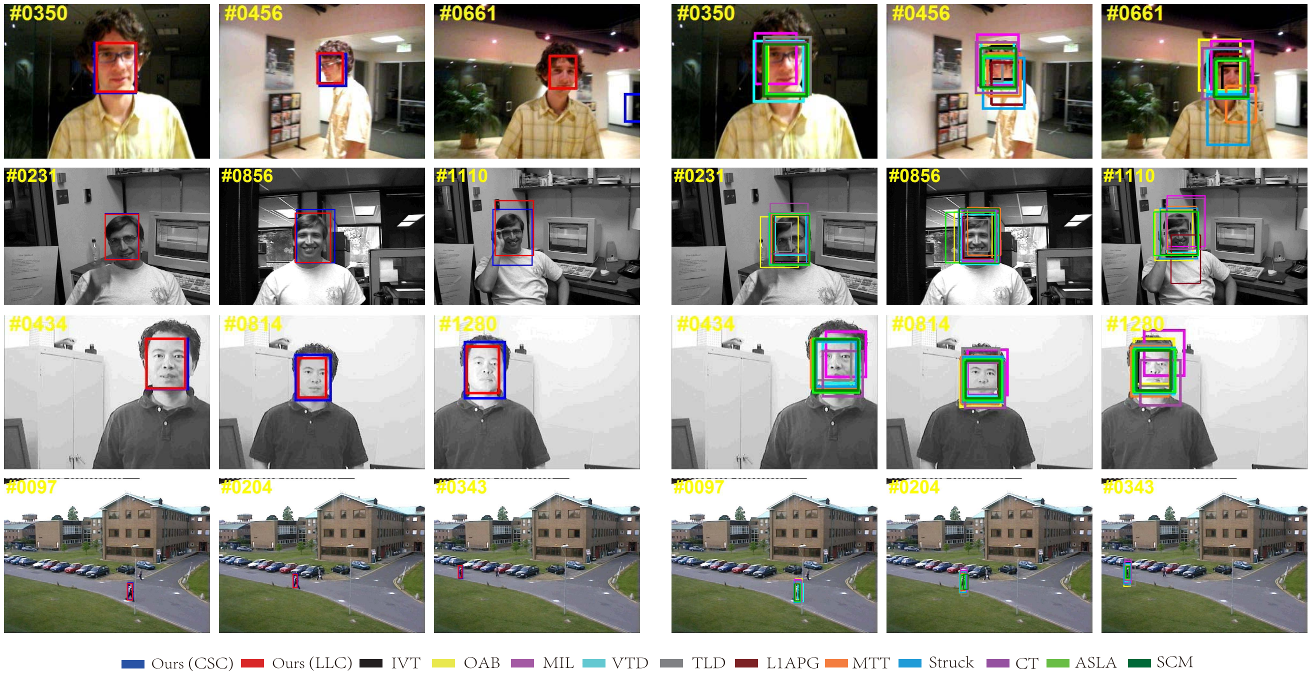


Fig. 9. Some tracking results for target appearance change on sequences of *David*, *Dudek*, *Mhyang* and *Walking*.

that with a more complex object representation model, we can achieve a higher tracking score because our proposed method can be combined with many generic object representations. Regarding the performance of the ASLA-LLC method, our ASLA-LLC is more robust to deformation, scale variation, occlusion etc., as shown in Figure 8.

We also present the results of qualitative comparisons with popular and top-ranked trackers, such as incremental visual tracking (IVT) [4], online AdaBoost (OAB) [6], multiple instance learning (MIL) [7], visual tracking decomposition (VTD) [25], tracking-learning-detection tracker (TLD) [24], real-time L1 tracker (L1APG) [9], multi-task sparse learning tracker (MTT) [20], structured output tracker (Struck) [23], real-time compressive tracker (CT) [8], adaptive structured local sparse tracker (ASLA) [15] and sparsity-based collaborative tracker (SCM) [11]. The tracking results are presented in Figure 9 and Figure 13-16. Figure 9 presents some tracking results in the sequences with target appearance change on the sequences of *David*, *Dudek*, *Mhyang* and *Walking*. Figure 13 demonstrates how our proposed method performs when the target undergoes different types of illumination and scale variation on the sequences of *Car4*, *Fish* and *Singer1*. Figure 14 demonstrates how the proposed method performs when the target undergoes heavy occlusion or partial occlusion for the sequences of *Faceocc1*, *Faceocc2*, *Girl* and *Walking2*. Figure 15 presents the tracking results for the sequences with in-plane and out-of-plane rotation on the sequences of *David2*, *Dog1* and *Sylvester*. Figure 16 also presents the tracking results for the sequences with abrupt motion or background clutter for the sequences of *Coke* and *MountainBike*.

C. Discussion

As shown in Figure 5, compared with the PF method, the object representation models combined with our LLC method

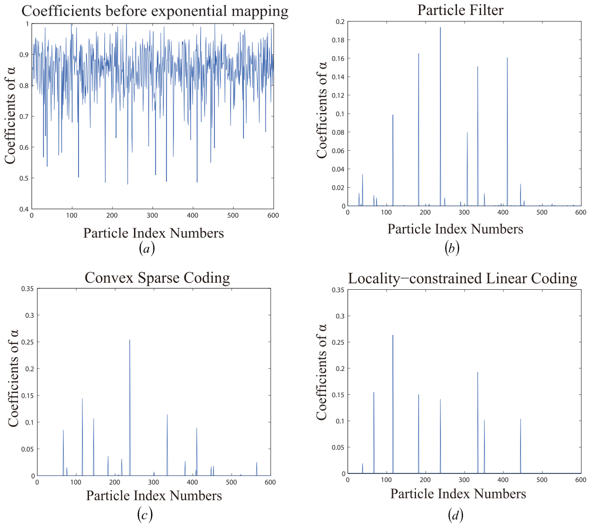


Fig. 10. Comparison of the particle filter method, the convex sparse coding method and the locality-constrained linear coding method for coefficients of α. Coefficients generated by (a) the particle filter before exponential mapping, (b) the particle filter method, (c) the convex sparse coding method, and (d) the locality-constrained linear coding method.

result in higher tracking accuracy. This result indicates that the proposed LLC method is superior to the PF method regarding the accuracy of the tracker; in addition, the proposed method can be combined with many generic object representation models, such as SR, LSS, and ASLA. Based on the experiments, we also believe that by combing a more complex object representation model combined with our LLC method in the future, we can achieve much higher accuracy of the tracking result.

To understand the high performance of the proposed methods, we compared the coefficients generated by the PF, CSC, and LLC methods (Figure 10). PF measures the similarity of particles and templates and then uses an exponential function (e.g., $e^{-\lambda D^2}$) in which the coefficients actually become very sparse to generate weights for the particles to improve accuracy (Figure 10(b)), i.e., only very sparse particles are suitable for representing the target. Figure 10(a) shows the coefficients generated by the PF before exponential mapping; because these coefficients are very dense, they cannot be used for tracking. Our CSC and LLC methods use a sparse structure of the particle coefficients (Figures 10(c-d)) directly, which matches this tracking approach very well. However, PF must enumerate the distance between each particle and the object templates (Figure 3(a-b)) and therefore does not fully explore the continuity property of the state space.

We also compared the CSC and the LLC approaches; as shown in Figure 4(a), the coefficients generated by CSC are not locally sparse; Thus, CSC may lead to greater errors in object representation and is therefore not optimal for object tracking. As shown in Figure 4(b), LLC enforces local sparsity on the coefficients, and thus particles selected by LLC are much more suitable to represent the object. In Figure 10, we can also observe that the nonzero coefficients of α generated by LLC are also nonzero in the PF, which demonstrates that both LLC and PF facilitate the selection of particles that are similar to the object template. By contrast, CSC does not facilitate the selection of such particles because the resolution procedure for sparse coding cannot ensure it; that is, the selected particles are not guaranteed to be similar to the object template. The experiments also demonstrated that LLC is more accurate than CSC. As shown in Figure 11, the output bounding boxes of the target from the two trackers are similar for many sequences, whereas the LLC tracker is more accurate for some challenging sequences.

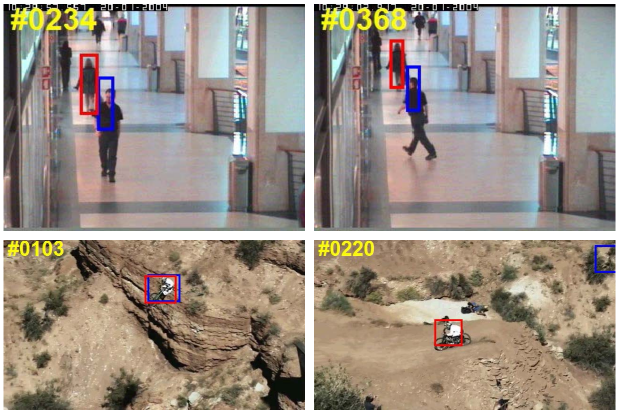


Fig. 11. Demonstration of the improvement of the tracking accuracy of the LLC tracker (red) compared to the CSC tracker (blue).

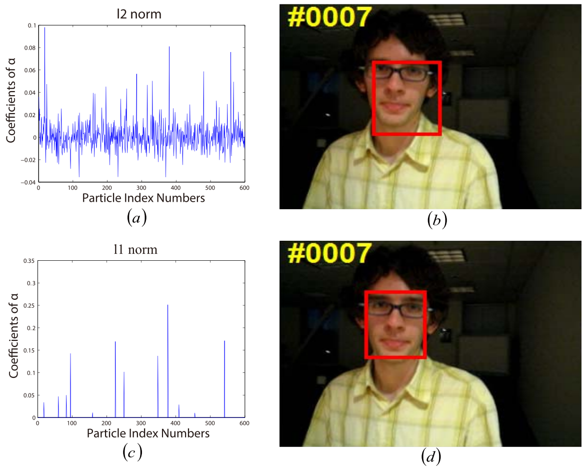


Fig. 12. Comparison of different norms on the 7-th frame of the video of *David*: (a) α values resolved by the ℓ_2 norm. (b) The tracked result corresponding to (a). (c) α values resolved by the ℓ_1 norm. (d) The tracked result corresponding to (c).

Finally, we examine whether the object observation in the current frame must be sparsely represented by its ambient particles. Typically, we place the ℓ_2 norm on the α term of Equation 6 for the video of *David*. The solution of α with the ℓ_2 norm, which is shown in Figure 12(a), is very dense, indicating that most of the particles are used to represent the target. Because many particles actually differ from the target, the tracking result can easily drift, as shown in Figure 12(b). By contrast, the solution of α with the sparse constraint, which is shown in Figure 12(c), is very sparse and, consequently, only a few particles are selected to represent the target; Thus, the result is very stable, as shown in Figure 12(d).

We note that template-based representations with the ℓ_1-norm sparsity were recently demonstrated to not neces-

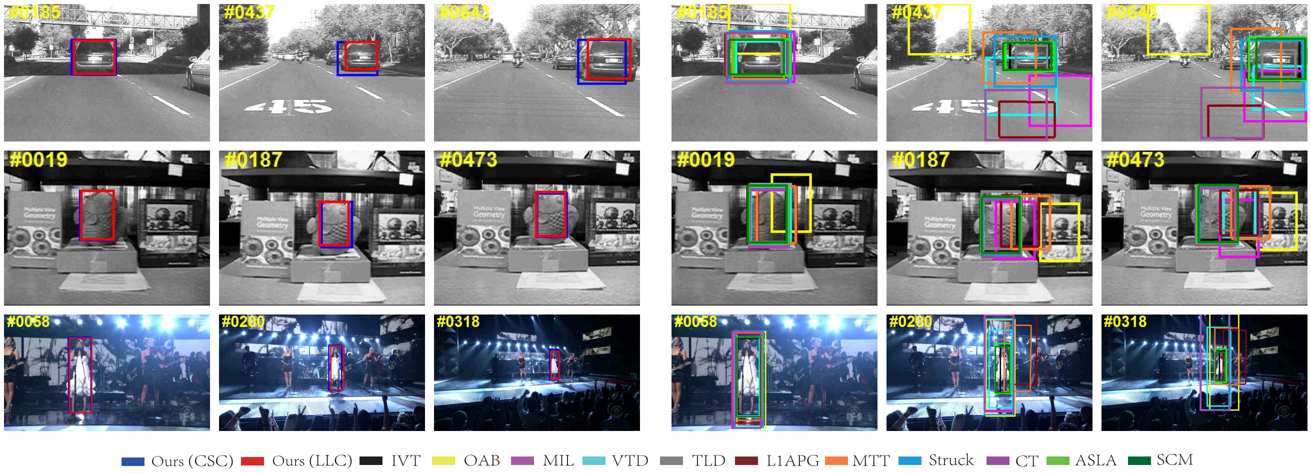


Fig. 13. Some tracking results for illumination and scale variation on sequences of *Car4*, *Fish* and *Singer1*.

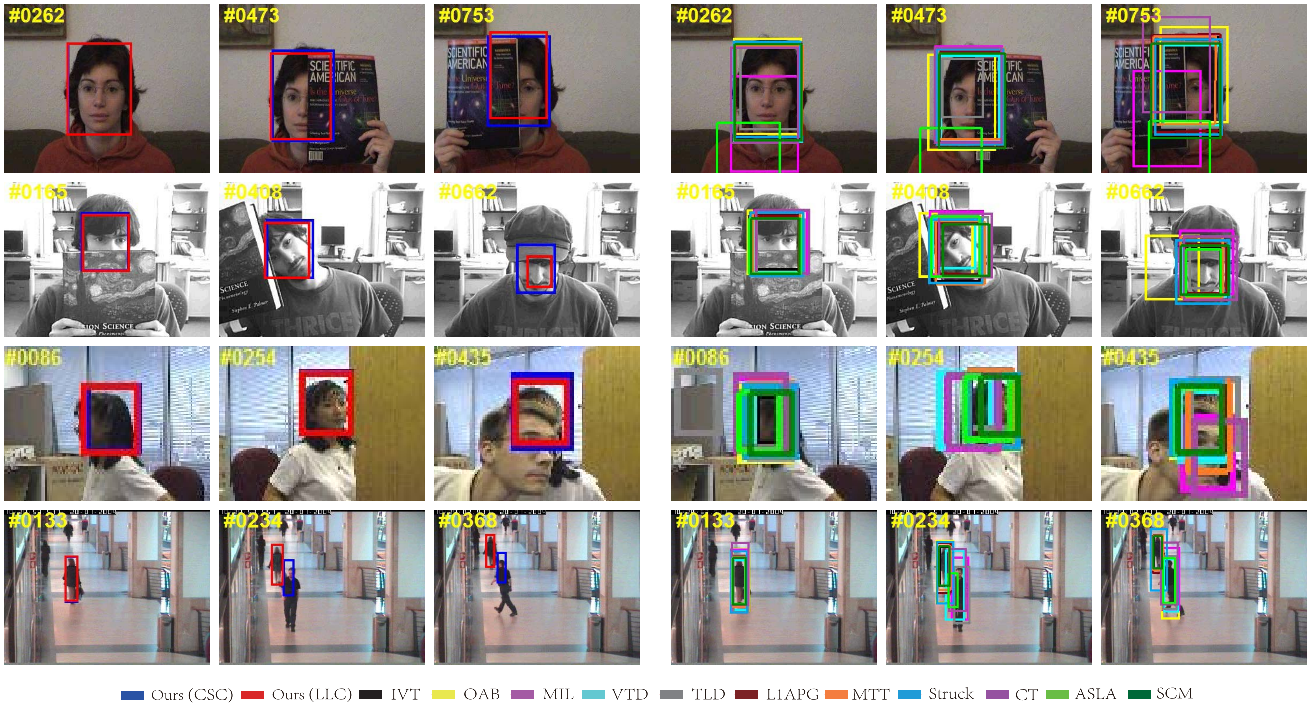


Fig. 14. Some tracking results for heavy occlusion or partial occlusion on sequences of *Faceocc1*, *Faceocc2*, *Girl* and *Walking2*.

sarily improve image classification [30], [31] or visual tracking [32]. However, the sparsity term in this work, as shown in Equation 1, is enforced on particles rather than templates. As discussed in Section III-B, in the particle term, the object observation in the current frame must be sparsely represented by its ambient particles. However, this property does not necessarily hold in the template term. Thus, either sparsity or non-sparsity can be used in the template term. We also note that numerous methods on the distance measurement between two spaces have been proposed for other clustering problems [33], [34]. However, these metrics are less effective for visual tracking without the sparsity term on the particles.

VII. Conclusion and Future Work

In this paper, we presented a novel search mechanism for visual tracking based on sparse and local linear coding. In the particle filter, the state space of the target is discretized by the distributed particles, resulting in discretization of the corresponding observation space of the target as well. Of course, representing and searching the target in a continuous space is more robust and efficient. We used a new representation model to code an object's appearance by linear combination of the particles, which becomes a continuous space, i.e., a convex hull. With the intrinsic sparsity constraints of the CSC and LLC representation models, this reconstruction of the appearance space results in an elegant and efficient tracking method that successfully uses only a few linear coding iterations. To cope with this appearance change, an adaptive updating strategy of the object template was also proposed.

Additionally, we demonstrated that PF actually uses an exponential function to ensure the weights of the particles are sparse. The two proposed representation models, CSC and LLC, are naturally sparse. Experiments demonstrated that the proposed methods operate more accurately and

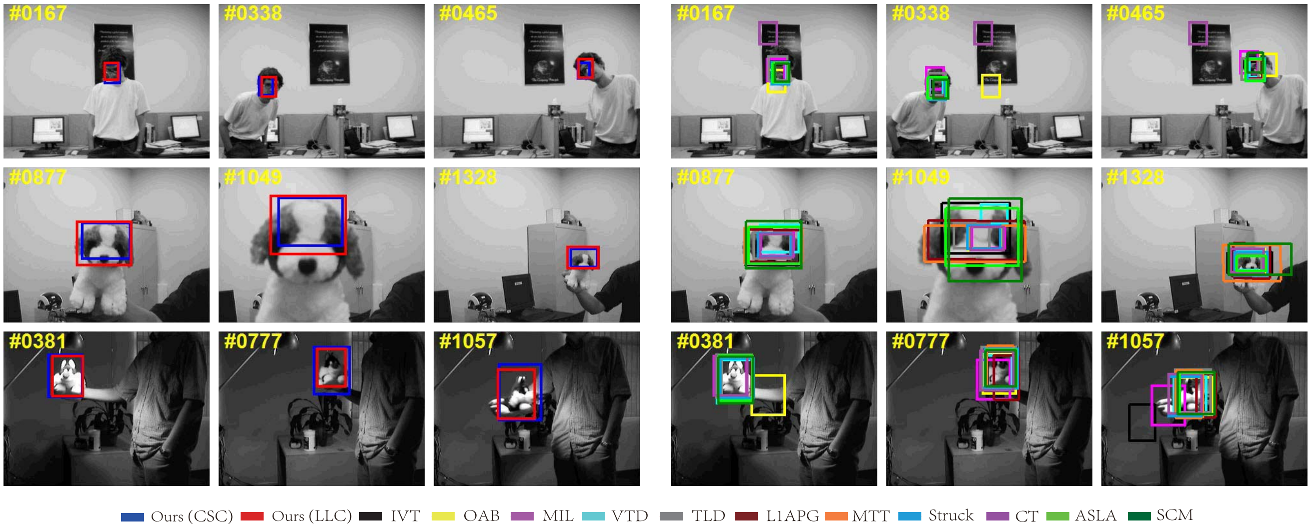


Fig. 15. Some tracking results for in plane and out of plane rotation on sequences of *David2*, *Dog1* and *Sylvester*.

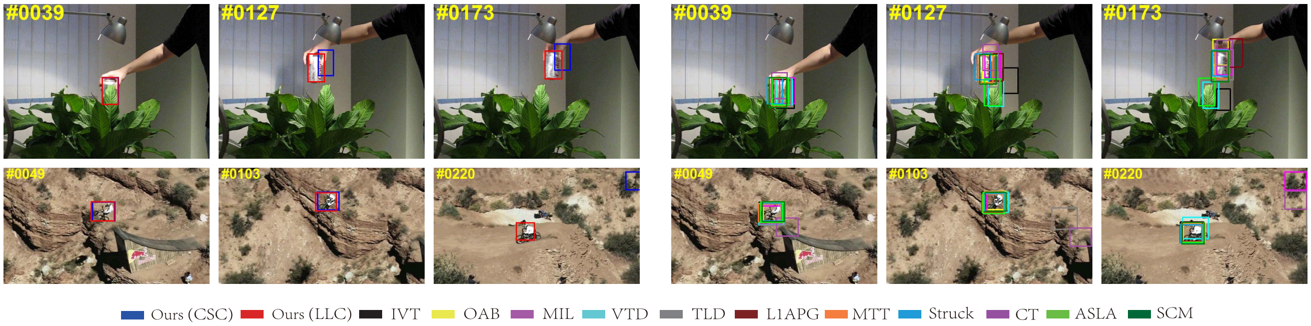


Fig. 16. Some tracking results for abrupt motion or background clutter on sequences of *Coke*, *MountainBike*.

effectively than a PF-based tracker. Moreover, LLC has much better locality than CSC, which leads to a more precise representation of the target appearance. Experiments also demonstrated that LLC is much more precise than CSC.

The proposed algorithm is formulated as an optimization problem of a function that consists of two terms, which correspond to the representation of the target on the space of both particle observations and object template. Therefore, our algorithm can be incorporated with many generic object representations. While sparse representation of CSC and LLC involves searching for the target in a continuous space efficiently, various object representation models, i.e., least soft-threshold squares and adaptive structural local sparse appearance, are alternative approaches. Experimental results demonstrated that these models can further improve tracking accuracy, verifying the flexibility of our algorithm.

In future work, we may consider exploring other object representations and improving the accuracy further by considering more challenging cases.

Appendix

Definition 1 (Convex Hull): For a collection of r (r>1) n-dimensional vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$, if there exists any nonnegative numbers λ_1, λ_2, $\cdots$, λ_r, combined with $\lambda_1 + \lambda_2 + \cdots + \lambda_r = 1$, then the set $\sum_{i=1}^{r} \lambda_i \mathbf{u}_i$ is the convex hull.

Proposition 1: $\mathbf{P}\alpha$ and $\mathbf{Bc}$ are convex hulls, and Equation 6 is the minimum Euclidian distance between these two convex hulls.

Proof: Equation 6 is equivalent to

$$\begin{aligned} &\min\{\| \mathbf{P}\alpha - \mathbf{Bc} \|_2^2 + \lambda_1 \sum_i |\alpha_i| + \lambda_2 \sum_i |c_i|\} \\ &\quad = \min_{\mu \geq 0}\{\min\{\| \mathbf{P}\alpha - \mathbf{Bc} \|_2^2\} + \lambda_1 + \lambda_2 \mu\}, \\ &s.t.\ \forall i, \alpha_i \geq 0, \sum_i \alpha_i = 1, c_i \geq 0, \sum_i c_i = \mu. \end{aligned} \tag{19}$$

For the space $\mathbf{P}\alpha$, $\alpha_i \geq 0, \sum_i \alpha_i = 1$; hence, according to Definition 1, the space $\mathbf{P}\alpha$ is a convex hull.

For the space $\mathbf{Bc}$, for any nonnegative μ, the coefficient of $\mathbf{c}$ is subject to $c_i \geq 0, \sum_i c_i = \mu$; hence, there exists a collection of n + 2d d-dimensional vectors $\mu\mathbf{b}_1$, $\mu\mathbf{b}_2, \cdots$, $\mu\mathbf{b}_{n+2d}$, and a collection of n + 2d nonnegative numbers $\frac{c_1}{\mu}$, $\frac{c_2}{\mu}, \ldots$, $\frac{c_{n+2d}}{\mu}$ and $\frac{c_1}{\mu} + \frac{c_2}{\mu} + \cdots + \frac{c_{n+2d}}{\mu} = 1$, and thus the space $\mathbf{Bc}$ is also a convex hull.

In Equation 19, for a given μ, the minimization of Equation 19 is dependent on the $\min\{\| \mathbf{P}\alpha - \mathbf{Bc} \|_2^2\}$, which may reach zero, i.e., the spaces $\mathbf{P}\alpha$ and $\mathbf{Bc}$ are in contact or just the $\min\{\| \mathbf{P}\alpha - \mathbf{Bc} \|_2^2\}$, which is the Euclidian distance between convex hulls $\mathbf{P}\alpha$ and $\mathbf{Bc}$.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 142–149.

[2] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

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

[4] 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.

[5] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.

[6] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 260–267.

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

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

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

[10] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 624–637.

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

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[13] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

[14] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2371–2378.

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

[16] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006, Art. ID 13.

[17] K. Cannons, "A review of visual tracking," Dept. Comput. Sci. Eng., Univ. York, York, U.K., Tech. Rep. CSE-2008-07, 2008.

[18] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 728–735.

[19] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1305–1312.

[20] 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.*, Jun. 2012, pp. 2042–2049.

[21] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[22] R. T. Collins and Y. Liu, "On-line selection of discriminative tracking features," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 346–352.

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

[24] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[25] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.

[26] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, Inc., 2009.

[27] 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.

[28] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

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

[30] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 553–560.

[31] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.

[32] X. Li, C. Shen, Q. Shi, A. Dick, and A. van den Hengel, "Non-sparse linear representations for visual tracking with online reservoir metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1760–1767.

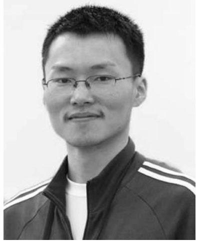
[33] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. I-11–I-18.

[34] A. W. Fitzgibbon and A. Zisserman, "Joint manifold distance: A new approach to appearance based clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. I-26–I-33.

Guofeng Wang received the B.S. degree in electrical engineering from Central South University, Changsha, China, in 2009. He is currently pursuing the Ph.D. degree in computer science with Shandong University, Jinan, China. His research interests include object tracking, pose estimation, and video analysis.

Xueying Qin received the B.S. degree from Peking University, in 1988, the M.S. degree from Zhejiang University, in 1991, and the Ph.D. degree from Hiroshima University, Japan, in 2001.

She was an Associate Researcher with the State Key Lab of CAD and CG, Zhejiang University, from 2003 to 2008. She is currently a Professor with the School of Computer Science and Technology, Shandong University, since 2008. Her main research interests are augmented reality, computer vision, video-based rendering, and robotics. Her main goal is to build an augmented world by vision-based methods with photorealistic quality in highly dynamic environments. She is a Senior Member of the China Computer Federation.



Fan Zhong received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2005 and 2010, respectively. He is currently a Lecturer with the School of Computer Science and Technology, Shandong University, Jinan, China. His research interests include image and video editing, and augmented reality.

Yue Liu received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2010. He is currently pursuing the Ph.D. degree in mathematics with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include mathematics mechanization, geometric invariants, and applications in computer vision.

Hongbo Li received the B.Sc., M.Sc., and Ph.D. degrees from Peking University, in 1988, 1991, and 1994, respectively, and received the Full Professorship of the Academy of Mathematics and Systems Science (AMSS) in 1998. He is currently a Professor of Mathematics and Computer Science of AMSS with the Chinese Academy of Sciences. He is also the Director of the Key Laboratory of Mathematics Mechanization, and the Advanced Manufacturing Division, National Center for Mathematics and Interdisciplinary Sciences with the Chinese Academy of Sciences. His research field includes mathematics mechanization, geometric algebra, differential geometry, and computer vision.

Qunsheng Peng received the B.S. degree from Beijing Mechanical College, Beijing, China, in 1970, and the Ph.D. degree from the School of Computing Studies, University of East Anglia, in 1983.

He is currently a Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include realistic image synthesis, virtual reality, scientific visualization, and biological computing. He is the Chairman of the Professional Committee of CAD and Graphics, the China Computer Federation, and serves as a member of the editorial boards of several international and domestic journals.

Ming-Hsuan Yang received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, IL, USA, in 2000. He is currently an Associate Professor of Electrical Engineering and Computer Science with the University of California, Merced. His research interests include computer vision, pattern recognition, artificial intelligence, robotics, and machine learning. He served as the Program Chair of the 2014 Asian Conference on Computer Vision, and the Area Chair of the IEEE International Conference on Computer Vision, the IEEE Conference on Computer Vision and Pattern Recognition, the European Conference on Computer Vision, and the Asian Conference on Computer Vision. He served as an Associate Editor of the IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and is currently an Associate Editor of the *International Journal of Computer Vision*, *Image and Vision Computing*, and the *Journal of Artificial Intelligence Research*. He was a recipient of the NSF CAREER Award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a Senior Member of the Association for Computing Machinery.