Inverse Sparse Group Lasso Model for Robust Object Tracking

Yun Zhou, Jianghong Han, Xiaohui Yuan, *Senior Member, IEEE*, Zhenchun Wei, and Richang Hong, *Member, IEEE*

***Abstract*—Sparse representation has been applied to visual tracking. The visual tracking models based on sparse representation use a template set as dictionary atoms to reconstruct candidate samples without considering similarity among atoms. In this paper, we present a robust tracking method based on the inverse sparse group lasso model. Our method exploits both the group structure of similar candidate samples and the local structure between templates and samples. Unlike the conventional sparse representation, the templates are encoded by the candidate samples, and similar samples are selected to reconstruct the template at the group level, which facilitates inter-group sparsity. Every sample group achieves the intra-group sparsity so that the information between the related dictionary atoms is taken into account. Moreover, the local structure between templates and samples is considered to build the reconstruction model, which ensures that the computed coefficients similarity is consistent with the similarity between templates and samples. A gradient descent-based optimization method is employed and a sparse mapping table is obtained using the coefficient matrix and hash-distance weight matrix. Experiments were conducted with publicly available datasets and a comparison study was performed against 20 state-of-the-art methods. Both qualitative and quantitative results are reported. The proposed method demonstrated improved robustness and accuracy and exhibited comparable computational complexity.**

***Index Terms*—Computer vision, hash distance, sparse coding, sparse group lasso, visual tracking.**

I. Introduction

OBJECT tracking, one of the important and challenging research subjects in the field of computer vision, has been receiving a great amount of attention and investment of researchers. The object tracking algorithms extract image low-level information from continuous video or image sequence and detect object to get the parameters of the target object, such as location, speed, trace, and other information. It is widely used in applications such as behavior analysis, motion recognition, and traffic control, etc. Although a large number of tracking algorithms have been developed in recent years [1]–[8], it remains a challenging problem that demands further study for robust methods to overcome many practical issues such as occlusion, illumination variation, and background clutter as shown in Fig. 1.

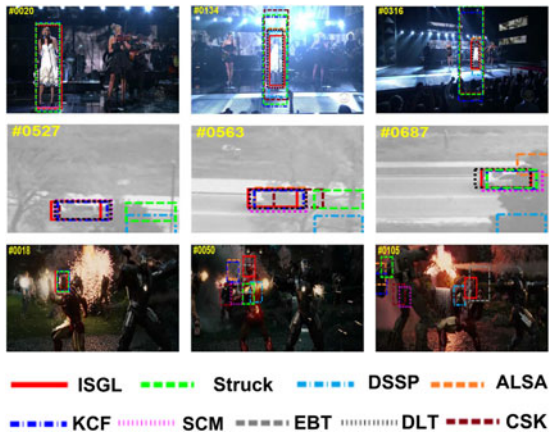


Fig. 1. Challenging factors for object tracking including illumination(*Singer1*), occlusions(*Suv*), and background clutter(*Ironman*). Rectangles in several colors are used to show the tracking results of the Struck [3], SCM [4], DSSP [9], KCF [10], EBT [11], DLT [12], CSK [13], ALSA [14], and our method.

A tracking method usually consists of three components: a motion model, an observation model, and an update strategy. The motion model, e.g. particle filter [1], aims at describing the motion states of an object over time and predicts the possible position based on a set of possible regions. The purpose of an observation model is to calculate the likelihood of the possible region, and is updated frequently to adapt to the changes of the object as well as the background. An update strategy is to fine-tune the parameters of the observation model according to the video and the target objects.

In this paper, we propose a novel tracking method based on inverse sparse group lasso (ISGL) model. With the inverse sparse structure, templates are encoded with a small number of samples. Group sparsity is used to explore the commonality among atoms and inter-group and intra-group sparsity are ensured by the group sparsity regularization, which selects a set of relevant

Manuscript received September 13, 2016; revised December 19, 2016 and February 21, 2017; accepted March 25, 2017. Date of publication March 30, 2017; date of current version July 15, 2017. This work was supported in part by the International S&T Cooperation Program of China under Grant 2014DFB10060, in part by the National Science Foundation of China under Grant 61472116, and in part by the Anhui Fund for Distinguished Young Scholars under Grant 1508085J04. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shu-Ching Chen. *(Corresponding author: Zhenchun Wei.)*

Y. Zhou, J. Han, Z. Wei, and R. Hong are with the School of Computer and Information, Hefei University of Technology, Hefei 230009, China (e-mail: zhouyun_hfut@163.com; hanjh@hfut.edu.cn; weizc@hfut.edu.cn; hongrc.hfut@gmail.com).

X. Yuan is with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203 USA, and also with the College of Information Engineering, China University of Geosciences, Wuhan 430074, China (e-mail: xiaohui.yuan@unt.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/TMM.2017.2689918

atoms and hence reduces the reconstruction error. In addition, the local structure of templates is considered to keep the computed similarity coefficients in accordance with the similarity between the templates and the atoms. Using sparse coefficient matrix and distance weight matrix, a discriminative sparse mapping table is constructed, in which the weight matrix is measured by hash distance. The tracking problem is hence transformed into an optimization problem with constraint. Throughout the tracking process, the templates are updated to handle occlusion and recover from drifts.

The contributions of this work is three-fold: 1) A novel robust tracking method based on an ISGL model, which is an inverse sparse reconstruction structure with group sparsity constraint. Templates are reconstructed from candidate atoms at the group level, and the templates are represented efficiently with few errors. 2) A local structure of templates is used to keep the computed coefficients similarity in accordance with the similarity between templates and samples, which improves the robustness and accuracy. 3) A discriminative sparse mapping table is constructed from the sparse coefficient matrix and hash distance weight matrix to refine the sparse coefficients for improved target discrimination.

In the rest of this paper, we review the background and related work in Section II. Section III presents the inverse sparse representation model. Section IV discusses the details of our proposed ISGL algorithm. The experimental results are presented in Section V followed by a conclusion in Section VI.

II. Background and Related Work

A. Particle Filter Framework

Particle Filter is a Bayesian sequential sampling technique and has been applied for object tracking. Assume that $\mathbf{Z}$ is the state of tracked object and $\mathbf{O}$ is the observation. $\mathbf{Z}_{1:\tau} = (\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_\tau)$ represents the available state vectors up to time τ and $\mathbf{O}_{1:\tau}$ denotes their corresponding observations. Six affine motion transformation parameters are used to describe the state of the object including the location translations, rotation angle, scale, aspect ratio, and skew. These parameters are mutual independent and follow Gaussian distributions. The posteriori probability of the object state at time τ is estimated by the following rule:

$$p(\mathbf{Z}_\tau|\mathbf{O}_{1:\tau}) = p(\mathbf{O}_\tau|\mathbf{Z}_\tau) \int p(\mathbf{Z}_\tau|\mathbf{Z}_{\tau-1}) p(\mathbf{Z}_{\tau-1}|\mathbf{O}_{1:\tau-1}) d\mathbf{Z}_{\tau-1} \tag{1}$$

where the dynamic motion model $p(\mathbf{Z}_\tau|\mathbf{Z}_{\tau-1})$ predicts the conditional transition probability of the target states between the two adjacent frames and $p(\mathbf{O}_\tau|\mathbf{Z}_\tau)$ denotes the observation likelihood. The particle filter is used to select N samples $\mathbf{Z}_\tau^i$, $i = 1, \ldots, N$ to simulate the posterior probability distribution $p(\mathbf{Z}_\tau|\mathbf{O}_{1:\tau})$. The optimal state at time τ can be estimated by the maximum a posteriori probability (MAP) over these samples, where $\mathbf{Z}_\tau^i$ is the state of i-th sample at time τ

$$\widehat{\mathbf{Z}}_\tau = \arg_{\mathbf{Z}_\tau^i} \max p(\mathbf{O}_\tau|\mathbf{Z}_\tau^i) p(\mathbf{Z}_\tau^i|\mathbf{Z}_{\tau-1}). \tag{2}$$

B. Sparse Representation

1) Preliminaries and Notations: In this paper, vectors are presented in lower-case, bold font, and matrices are in upper-case, bold font, for instance, a vector $\mathbf{a} = (a_1, a_2, \ldots, a_k)$, where a_i is the i-th element in vector $\mathbf{a}$, and a matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k]$, where $\mathbf{a}_i$ indicates the i-th column in matrix $\mathbf{A}$. a_{ij} denotes the element in the i-th row and j-th column in matrix $\mathbf{A}$. $\mathbf{I}$ denotes the identity matrix. The symbol $\odot$ is a Hadamard product operator, which multiplies the corresponding elements between two matrixes.

2) General Lasso Method: Least Absolute Shrinkage and Selection Operator (Lasso) method achieves a reconstruction by minimizing the following loss function [15]:

$$\arg\min_{\mathbf{b}} (\|\mathbf{a} - \mathbf{D}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_\rho) \tag{3}$$

where $\mathbf{a} \in \mathbb{R}^{e\times 1}$ represents the given data which is reconstructed, $\mathbf{D} \in \mathbb{R}^{e\times k}$ represents k dictionary atoms, and $\mathbf{b} \in \mathbb{R}^{k\times 1}$ is the sparse coefficients or codes of data $\mathbf{a}$, $\lambda > 0$ is a weight parameter. When $\rho = 0$, $\|\mathbf{b}\|_\rho$ is the ℓ_0-norm of $\mathbf{b}$, which means the total number of nonzero elements in vector $\mathbf{b}$. Unfortunately, the solution of ℓ_0-norm is NP-hard even though it is a perfect sparsity constraint. The ℓ_1-norm constraint results in many zero elements and its outcome is close to that of the ℓ_0-norm. Moreover, it is convex, which allows a solution to be computed rather easily. Thus ℓ_1-norm is widely adopted as an approximate constraint of ℓ_0-norm in practices.

3) Sparse Model in Object Tracking: In object tracking, given a video (or an image sequence) and the initial position of the target, we track the target in the following video frames. Suppose that a candidate sample set $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbb{R}^{e\times N}$ is extracted from the current frame τ, while the template set $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_M] \in \mathbb{R}^{e\times M}$ consists of object templates and background templates. The best candidate sample can be selected as the tracking result in the current frame.

General Sparse Model: In the sparse representation, the template set $\mathbf{T}$ is treated as dictionary $\mathbf{D}$, $\mathbf{y} \in \mathbb{R}^{e\times 1}$ represents a candidate sample region. A few atoms are selected from $\mathbf{D}$ to reconstruct $\mathbf{y}$ following the ℓ_1-norm constraint, and the coefficient vector $\mathbf{b}$ is obtained as follows:

$$\arg\min_{\mathbf{b}} (\|\mathbf{y} - \mathbf{T}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1), \text{s.t. } \mathbf{b} \geq 0. \tag{4}$$

The sparse coefficient for each sample in $\mathbf{Y}$ is computed based on $\mathbf{D}$, which yields a coefficient matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N] \in \mathbb{R}^{M\times N}$. The optimal tracking result is chosen from $\mathbf{Y}$ according to the reconstruction error. In this model, the optimization process is performed for each candidate, which results in a high computation cost.

Inverse sparse model: To reduce the computation cost, we consider the method of inverse sparse representation. Conversely, dictionary $\mathbf{D}$ is composed of a candidate set $\mathbf{Y}$. Taking one object template $\mathbf{t}$ as an example, it is reconstructed from $\mathbf{Y}$ within a ℓ_1-norm constraint

$$\arg\min_{\mathbf{b}} (\|\mathbf{t} - \mathbf{Y}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1), \text{s.t. } \mathbf{b} \geq 0. \tag{5}$$

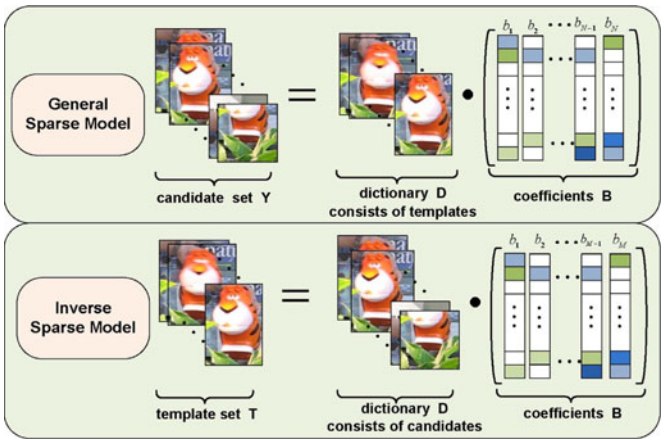


Fig. 2. General sparse model and inverse sparse model. In the sparse model, the candidate sample set is reconstructed by the templates. In the inverse sparse model, the template set is reconstructed by the candidate samples in turn.

The candidate with the maximum value in the coefficient vector $\mathbf{b}$ is selected as the optimal tracking result. To assure the tracking reliability, all templates in set $\mathbf{T}$ are reconstructed by the candidate set $\mathbf{Y}$. An optimal candidate is identified from the coefficient vectors $\mathbf{b}_i$, $i = 1, 2, \ldots, M$. Since the number of the selected templates M is much smaller than that of samples N, the complexity of optimization is greatly reduced. The difference between the two models is shown in Fig. 2.

The general lasso tends to select atoms based on the strength of individual atom. The underlying commonality among dictionary atoms is usually ignored. This often results in selecting more atoms than necessary to represent the given data, and noisy atoms could be introduced.

4) Sparse Group Lasso Model: In recent years, the group property in the sparse representation has attracted many research interests [16], [17]. The group sparsity consists of two aspects: inter-group sparsity and intra-group sparsity. The inter-group sparsity refers to the sparsity between different groups, while the intra-group sparsity is the sparsity within a group of instances.

If a dictionary is divided into disjointed groups according to the similarity, the given data can be sparsely represented by a set of groups, rather than atoms, which achieves the inter-group sparsity. In addition, all atoms in these selected groups have a strong correlation with the given data. Suppose a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k] = [\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \ldots, \mathbf{D}^{(\mathcal{G})}] \in \mathbb{R}^{e \times k}$ with k atoms in $\mathbf{D}$ is divided into $\mathcal{G}$ groups. For ease of notation, $\mathbf{d}_i$ denotes the i-th atom of the dictionary $\mathbf{D}$, $\mathbf{D}^{(g)}$ is used to represent the atoms within the g-th group, $\mathbf{b}^{(g)}$ is the corresponding coefficient of that group. The coefficient vector $\mathbf{b}$ of the group lasso model [18] is computed as follows:

$$\arg\min_{\mathbf{b}} \left(\left\| \mathbf{a} - \sum_{g=1}^{\mathcal{G}} \mathbf{D}^{(g)} \mathbf{b}^{(g)} \right\|_2^2 + \lambda \sum_{g=1}^{\mathcal{G}} \sqrt{n_g} \left\| \mathbf{b}^{(g)} \right\|_2 \right) \quad (6)$$

where λ is parameter to control the weight of the regularization term, and n_g is the number of atoms in group $\mathbf{D}^{(g)}$. When a group is selected, all the atoms in this group are used in reconstruction. Thus, the group lasso considers the inter-group sparsity without taking into account the intra-group sparsity.

To take into account both the inter-group sparsity and intra-group sparsity, Friedman *et al.* [19] presented the group sparse lasso by adding an additional ℓ_1-norm term

$$\arg\min_{\mathbf{b}} \left(\frac{1}{2} \left\| \mathbf{a} - \sum_{g=1}^{\mathcal{G}} \mathbf{D}^{(g)} \mathbf{b}^{(g)} \right\|_2^2 + \lambda_1 \sum_{g=1}^{\mathcal{G}} \sqrt{n_g} \left\| \mathbf{b}^{(g)} \right\|_2 + \lambda_2 \|\mathbf{b}\|_1 \right). \quad (7)$$

The second term expresses the inter-group sparsity, while the third term represents the intra-group sparsity. The parameters λ_1 and λ_2 balance the two sparsity constraints ($\lambda_1 = 0$ gives the lasso fit, $\lambda_2 = 0$ gives the group-lasso fit). Fig. 3 illustrates the idea of the general lasso, the group lasso and the sparse group lasso model, where $\mathbf{a}$ is reconstructed with $\mathbf{D}$ and $\mathbf{b}$ is the sparse coefficient. In the general lasso model, the dictionary atoms are treated to be individual and the coefficient $\mathbf{b}$ generates the element-sparsity through the whole column, as shown in Fig. 3(a). In the group lasso model, $\mathbf{D}$ and $\mathbf{b}$ are divided into groups. The groups of zeros exist in $\mathbf{b}$ due to the inter-group sparsity, while the atoms in a group with non-zero values are chosen, as shown in Fig. 3(b). In the sparse group lasso model, besides the existing inter-group sparsity, parts of atoms are selected in non-zero groups due to intra-group sparsity, as shown in Fig. 3(c).

C. Related Work and Problem Context

1) Generative and Discriminative Tracking: Visual tracking algorithms, in general, can be divided into two main categories: generative and discriminative trackers. Generative methods [2] focus on how to represent the appearance of objects and search the most similar one in the candidate regions with minimal reconstruction error. In [20], the incremental visual tracker (IVT) learns a subspace model that cope with the appearance changes. MTT [21] method formulates object tracking in a particle filter framework as a multi-task sparse learning problem. Yang *et al.* [22] propose a tracking method from the perspective of mid-level vision with structural information captured in super-pixels (SPT). In [4], the SCM tracker develops a sparse discriminative classifier and sparse generative model within the collaborative appearance model. ASLA[14] exploits both partial and spatial information with an alignment-pooling method to represent the targets. In [23], Laura *et al.* use the distribution fields (DFs) to represent the targets and images, and searching for targets in an image is achieved with a gradient descent method.

Discriminative methods [24] aim to build online classifiers to distinguish the target region from the background. The Struck method [3] adopts the kernelized structured output support vector machine to avoid the labeling ambiguity when updating the classifier during tracking. The multiple instance learning (MIL) is applied to an online setting for object tracking [25]. DLT [26] trains a stacked deionising autoencoder offline to extracting the robust image features in visual tracking. The EBT [11] learns the trajectory of the target and the reliability of each tracker jointly in the ensemble. CSK [13] tracker exploits the circulant

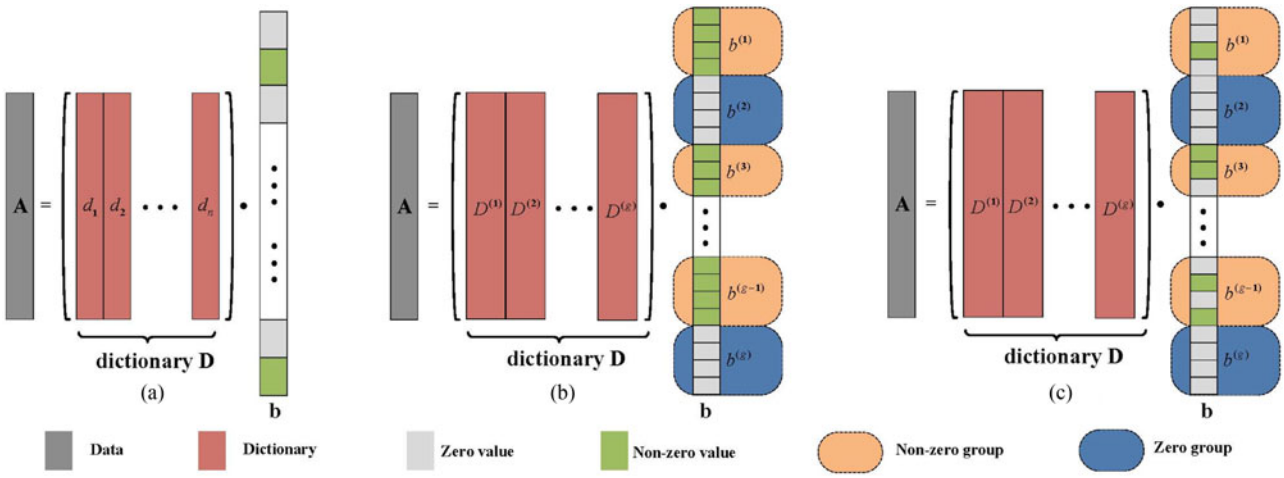


Fig. 3. Three sparse models. (a) General lasso model. $\mathbf{b}$ consists of zeros and non-zero values. (b) Group lasso model. $\mathbf{b}$ consists of groups of zeros and groups of non-zero values. (c) Sparse Group lasso model. $\mathbf{b}$ consists of group of zeros and groups of non-zero values.

structure of adjacent image patches in a kernel space. KCF [10] uses the properties of a circular structure in the frequency domain and improves processing speed and accuracy by using the multi-channel features and Gauss kernel function.

2) Object Tracking Using Sparse Representation: The sparse representation has been applied to face recognition [27], image classification [28], image super-resolution [29], and other computer vision applications [30]–[32], which yielded good performance. Extensive research has been dedicated to object tracking with favorable experimental performance. Mei and Ling [33] proposed a visual tracking algorithm by framing the tracking problem as a sparse representation of candidates. Each candidate sample is reconstructed from dictionary atoms composed of target and trivial templates. This leads to a sparse coefficient vector, i.e., coefficients of trivial templates are close to zeros. The candidate sample with the smallest reconstruction error is used.

The computational cost restricts the application of the aforementioned methods in real-time tracking applications [34]. To improve the efficiency, Bao *et al.* [35] proposed an efficient method of accelerated proximal gradient descent (APG) to boost the speed of ℓ_1 tracking process. Liu and Sun [36] used the sparsity-induced similarity to construct the tracker. Templates are represented by the candidates, and the coefficients imply the similarity between the candidates and the templates. This method performs one optimization for each template. Zhuang *et al.* [9] formulated the tracking problem as finding the candidate that scores highest in the evaluation model based on a discriminative sparse similarity map (DSSP).

However, uncorrelated atoms are usually selected and the underlying commonality shared among the dictionary atoms are ignored in the aforementioned methods. To take advantage of the commonality among atoms, we propose a robust tracking method based on the inverse sparse group model. Group sparse tracking exploits the dual group structure of both candidate samples and dictionary templates and formulates the sparse representation at group level to ensure data with similar appearance are encoded jointly [37]. In this paper, we combine the inverse and group lasso sparse representation structure to reduce computational time and reconstruction error. Positive atoms are selected based on the strength of groups of similar ones rather than the individual column. Moreover, a local structure information is embedded in this joint framework, the underlying relationship between templates and atoms are fully utilized. In addition, an adaptive updating mechanism is developed to handle heavy occlusion and recover from drifts.

III. Inverse Sparse Group Lasso (ISGL) Model

According the related sparse representation models which are reviewed in the preceding section, we introduce our proposed inverse sparse group lasso (ISGL) framework in Sections III-A and III-B, the optimization scheme is presented in Section III-C.

A. Inverse Sparse Group Lasso

Based on the above analysis and the existing researches [19], [38]–[40], our objective function of the general inverse sparse group model is as follows:

$$\arg\min_{\mathbf{b}}\left(\frac{1}{2}\left\|\mathbf{t}-\sum_{g=1}^{\mathcal{G}}\mathbf{Y}^{(g)}\mathbf{b}^{(g)}\right\|_2^2+\lambda_1\sum_{g=1}^{\mathcal{G}}\left\|\mathbf{b}^{(g)}\right\|_2+\lambda_2\|\mathbf{b}\|_1\right) \tag{8}$$

where $\mathbf{t}\in\mathbb{R}^{e\times 1}$ represents a template. Given a candidate sample set $\mathbf{Y}=[\mathbf{Y}^{(1)},\ldots,\mathbf{Y}^{(\mathcal{G})}]=[\mathbf{y}_1,\ldots,\mathbf{y}_N]\in\mathbb{R}^{e\times N}$, $\mathbf{b}=[\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(\mathcal{G})}]\in\mathbb{R}^{N\times 1}$ is the sparse coefficient vector, and $\mathbf{b}^{(g)}$ is the corresponding coefficient of a group $\mathbf{Y}^{(g)}$. Again, λ_1 and λ_2 are the weights of the two sparsity constraints.

B. Inverse Sparse Group Lasso Model

According to the sparse representation, if two samples are similar in an original space, they are also close after projection into the new space spanned by the dictionary [41], [42]. This process is shown in Fig. 4. The local structure between template and atoms is conducive to reconstruct a more accurate template, which means a template is represented by more similar atoms.

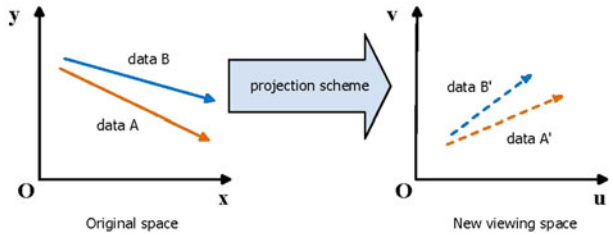


Fig. 4. Local structure between two vectors. Vectors A and B are close in the original space xOy, and they are also similar in the new space uOv.

According to Laplacian Eigenmaps, we expect to preserve the local structure in the new space to achieve a better reconstruction. We add a Laplacian constraint to ensure the local structure as follows:

$$\sum_{i=1}^{N} \|\mathbf{b}-\mathbf{u}_i\|_2^2 \cdot p_i. \tag{9}$$

Given a template $\mathbf{t}$, the k nearest neighbors in a candidate sample set $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_N]$ are identified together with a similarity vector $\mathbf{p}=(p_1,\ldots,p_N)^\top$, where $p_i=e^{-\|\mathbf{t}-\mathbf{y}_i\|^2/\sigma}$ represents the similarity between atom $\mathbf{y}_i$ and template $\mathbf{t}$. The constraint $\sum_{i=1}^{N}\|\mathbf{b}-\mathbf{u}_i\|_2^2\cdot p_i$ is used to measure the similarity between the template and the atom in the new space, where $\mathbf{u}_i$ is the projection vector of the i-th atom $\mathbf{y}_i$ and $\mathbf{b}$ is the projection of template $\mathbf{t}$ in the new space. It is represented as follows:

$$\sum_{i=1}^{N} \|\mathbf{b}-\mathbf{u}_i\|_2^2 \cdot p_i = \mathbf{b}^\top D\mathbf{b} - 2\mathbf{b}^\top\mathbf{U}\mathbf{p} + \sum_{i=1}^{N}\mathbf{u}_i{}^\top p_i\mathbf{u}_i \tag{10}$$

where $D=\sum_{i=1}^{N}p_i$ and $\mathbf{U}=[\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_N]$.

By combining (8) and (10), the objective function becomes

$$\arg\min_{\mathbf{b}}\left(\frac{1}{2}\left\|\mathbf{t}-\sum_{g=1}^{\mathcal{G}}\mathbf{Y}^{(g)}\mathbf{b}^{(g)}\right\|_2^2+\lambda_1\sum_{g=1}^{\mathcal{G}}\left\|\mathbf{b}^{(g)}\right\|_2 + \lambda_2\|\mathbf{b}\|_1+\frac{1}{2}\alpha\sum_{i=1}^{N}\|\mathbf{b}-\mathbf{u}_i\|_2^2\cdot p_i\right) \tag{11}$$

where α is the weight of the local structure constraint.

According to [39], the sum of the first and fourth terms in (11) is equivalent to

$$\arg\min_{\mathbf{b}}\frac{1}{2}(\mathbf{b}^\top\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{Y}}\mathbf{b}-2\mathbf{b}^\top\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{t}}) \tag{12}$$

where $\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{Y}}=\mathbf{Y}^\top\mathbf{Y}+\alpha D\mathbf{I}$ and $\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{t}}=\mathbf{Y}^\top\mathbf{t}+\alpha\mathbf{U}\mathbf{p}$. $\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{Y}}$ is a positive definite matrix, and we can get $\widetilde{\mathbf{Y}}$ through the Cholesky decomposition. Thus we can simplify the objective function as follows:

$$\arg\min_{\mathbf{b}}\left(\frac{1}{2}\left\|\widetilde{\mathbf{t}}-\sum_{g=1}^{\mathcal{G}}\widetilde{\mathbf{Y}}^{(g)}\mathbf{b}^{(g)}\right\|_2^2+\lambda_1\sum_{g=1}^{\mathcal{G}}\left\|\mathbf{b}^{(g)}\right\|_2+\lambda_2\|\mathbf{b}\|_1\right). \tag{13}$$

C. Optimization Scheme

Given the separability of (13), the optimization problem can be divided into intra-group sparse function optimization and inter-group sparse function optimization. We, hence, solve this problem by using block coordinate decent as follows.

1) Intra-group Optimization: The coefficients $\mathbf{b}$ are put into groups and a group is updated while the coefficients of other groups are fixed. Suppose that in group g, we have $\mathbf{b}^{(g)}=(b_1^{(g)},b_2^{(g)},\ldots,b_{n_g}^{(g)})^\top$ and $\widetilde{\mathbf{Y}}^{(g)}=[\widetilde{\mathbf{y}}_1^{(g)},\widetilde{\mathbf{y}}_2^{(g)},\ldots,\widetilde{\mathbf{y}}_{n_g}^{(g)}]$. Then $\mathbf{b}^{(g)}$ is obtained by solving the following optimization problem:

$$\arg\min_{\mathbf{b}}\left(\frac{1}{2}\left\|\mathbf{r}_g-\sum_{j=1}^{n_g}\widetilde{\mathbf{y}}_j^{(g)}b_j^{(g)}\right\|_2^2+\lambda_1\sum_{g=1}^{\mathcal{G}}\left\|\mathbf{b}^{(g)}\right\|_2+\lambda_2\|\mathbf{b}\|_1\right) \tag{14}$$

where $\mathbf{r}_g=\widetilde{\mathbf{t}}-\sum_{k\neq g}\widetilde{\mathbf{Y}}^{(k)}\mathbf{b}^{(k)}$ is the residual.

Check whether all elements in $\mathbf{b}^{(g)}$ are zero, that is, whether there are any candidate sample in $\widetilde{\mathbf{Y}}^{(g)}$ chosen to reconstruct $\widetilde{\mathbf{t}}$. Following the idea in [19] and [38], the necessary and sufficient condition of vector $\mathbf{b}^{(g)}=0$ is that the solution of equation $(\widetilde{\mathbf{y}}_j^{(g)})^\top\mathbf{r}_g=\lambda_1\cdot v_j+\lambda_2\cdot w_j$ fulfills $|v_j|\le 1$ and $\|\mathbf{w}\|_2\le 1$, where $\mathbf{v}=(v_1,v_2,\ldots,v_{n_g})$ and $\mathbf{w}=(w_1,w_2,\ldots,w_{n_g})$. Hence we determine this by minimizing the following function of $\mathbf{v}$:

$$J(\mathbf{v})=(1/{\lambda_2}^2)\sum_{j=1}^{n_g}\left((\widetilde{\mathbf{y}}_j^{(g)})^\top\mathbf{r}_g-\lambda_1\cdot v_j\right)^2=\|\mathbf{w}\|_2^2 \tag{15}$$

with respect to $|v_j|\le 1$. Suppose that $Q_j^{(g)}=(\widetilde{\mathbf{y}}_j^{(g)})^\top\mathbf{r}_g/\lambda_1$, the minimizer is

$$\hat{v}_j=\begin{cases}Q_j^{(g)}, \left|Q_j^{(g)}\right|\le 1\\ \mathrm{sign}(Q_j^{(g)}), \left|Q_j^{(g)}\right|>1\end{cases}. \tag{16}$$

We compute $J(\overset{\wedge}{\mathbf{v}})$ following (15) and (16). In case that $J(\overset{\wedge}{\mathbf{v}})$ is less than or equal to one, we have $\mathbf{b}^{(g)}=0$ and the optimization proceeds to calculate the coefficient of the next group.

2) Inter-group Optimization: In case that $J(\overset{\wedge}{\mathbf{v}})$ is greater than one, it means that elements in $\mathbf{b}^{(g)}$ are not all zeros. We need to know which element $b_j^{(g)}$ is zero or nonzero. When $b_j^{(g)}$ equals zero , the corresponding $v_j=\mathrm{sign}(b_j^{(g)})$ and $w_j=b_j^{(g)}/\|\mathbf{b}^{(g)}\|_2$ satisfy the criteria that $|v_j|\le 1$ and $w_j=0$, respectively. Otherwise, we calculate $b_j^{(g)}$ according to the following objective function:

$$\arg\min_{b_j^{(g)}}\frac{1}{2}\left\|\widetilde{\mathbf{t}}-\sum_{k}^{n_g}\widetilde{\mathbf{y}}_j^{(g)}b_j^{(g)}\right\|_2^2+\lambda_1\left\|\mathbf{b}^{(g)}\right\|_2+\lambda_2\left\|\mathbf{b}^{(g)}\right\|_1. \tag{17}$$

The first and the second term are differentiable convex functions, and the third item is a separable penalty term. We employ the fast iterative shrinkage-thresholding algorithm (FISTA) [43] to solve this optimization problem. The detail procedure is shown in Algorithm 1. In this way, we get a sparse coefficient

Algorithm 1: Sparse Group Lasso Algorithm

Input: template $\mathbf{t}$, dictionary $\mathbf{Y}$, atom group set $\{1, 2, \ldots, \mathcal{G}\}$, parameter α, λ_1 and λ_2
Output: coefficient vector $\mathbf{b}$

1: Calculate $\widetilde{\mathbf{t}}$ and $\widetilde{\mathbf{Y}}$, obtain the objective function in (13).
2: **for** $g = 1 \rightarrow \mathcal{G}$ **do**
3: Calculate $\mathbf{r}_g$ and $\hat{v}_j$, obtain $J(\hat{\mathbf{v}})$ according to (15).
4: **if** $J(\hat{\mathbf{v}}) \leq 1$ **then**
5: $\mathbf{b}^{(g)} \leftarrow 0$ and go to step 18.
6: **else**
7: Go to step 9.
8: **end if**
9: **for** $j = 1 \rightarrow n_g$ **do**
10: Calculate $|v_j|$ and w_j.
11: **if** $|v_j| \leq 1$ and $w_j = 0$ **then**
12: $b_j^{(g)} = 0$ and go to step 16.
13: **else**
14: Calculate $b_j^{(g)}$ according to (17) and the FISTA algorithm.
15: **end if**
16: $j \leftarrow j + 1$.
17: **end for**
18: $g \leftarrow g + 1$.
19: **end for**
20: **return** $\mathbf{b}$

matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_M] \in \mathbb{R}^{N \times M}$ for every template in set $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_M] \in \mathbb{R}^{e \times M}$.

IV. ISGL-Based Object Tracking Method

To have a complete presentation of our proposed method, we first give an introduction of the motion model based particle filter and then present the observation model by employing the inverse sparse group lasso model. Our template updating mechanism is discussed in the end. The flowchart of our proposed tracking algorithm is shown in Fig. 5.

A. Motion Model

Within the particle filter framework, the motion model follows Gaussian distribution

$$p(\mathbf{Z}_\tau | \mathbf{Z}_{\tau-1}) = \mathcal{N}(\mathbf{Z}_\tau; \mathbf{Z}_{\tau-1}, \Psi) \tag{18}$$

where $\mathbf{Z}_\tau$ is the state of target at frame τ, $\mathbf{Z}_{\tau-1}$ is the state of target at frame $\tau - 1$, Ψ is the covariance matrix with the elements on the diagonal line being the standard deviations for location, scale, rotation and so on. For example, the standard deviation for scale σ_θ dictates how the proposed ISGL algorithm accounts for scale changes.

For each video, the frames are converted into grayscale images. A set of candidate regions $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$ are sampled in the current frame based on the location of the target in the previous frame, which are used as the dictionary atoms. The atoms are put into $\mathcal{G}$ groups based on their similarity. In addition, M templates $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_M]$ in the previous frames are extracted. Following the method in [4], a template set $\mathbf{T}$ includes p object templates $\mathbf{T}_{pos}$ and q background templates $\mathbf{T}_{neg}$, which are the positive templates (target templates) and the negative templates (non-target templates), respectively. The candidate and template image regions are resized into a fixed sized (32×32 pixels) and reformatted into one-dimensional vectors.

B. Observation Model

An observation model is used to calculate the likelihood of each candidate sample to be the tracking result in the current frame. We construct a sparse mapping table to measure the likelihood. A good candidate is usually more similar to several positive templates, which results in larger reconstruction coefficients given the positive template set. On the other hand, a candidate with larger coefficients in the negative template set indicates a poor choice.

Putting the candidate set $\mathbf{Y}$ and template set $\mathbf{T}$ into the proposed ISGL model in Section III-C, the sparse coefficient matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_p, \mathbf{b}_{p+1}, \ldots, \mathbf{b}_{p+n}] \in \mathbb{R}^{N \times M}$ is calculated following Algorithm 1. The sub-matrix $[\mathbf{b}_1, \ldots, \mathbf{b}_p]$ consists of coefficients corresponding to the positive template set, and the sub-matrix $[\mathbf{b}_{p+1}, \ldots, \mathbf{b}_{p+n}]$ consists of coefficients corresponding to the negative template set. For a candidate sample $\mathbf{y}_i$, the confidence is proportional to the element in set $(b_{i1}, \ldots, b_{ip})$ and inversely proportional to the element in set $(b_{i(p+1)}, \ldots, b_{i(p+n)})$.

To improve the discrimination of samples in the positive and negative templates, we construct a distance weight matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$ using hash distance [9], the matrix is defined as follows:

$$\mathbf{W}(ij) \propto \exp(-H(\mathbf{t}_i, \mathbf{y}_j)) \tag{19}$$

where $H(\mathbf{t}_i, \mathbf{y}_j)$ represents the hash distance between the template $\mathbf{t}_i$ and the candidate $\mathbf{y}_i$. We employ the perceptual image hashing method [44] based on discrete wavelet transformation (DWT) since it compactly captures significant image characteristics. It becomes apparent that the corresponding weight increased gradually with decreasing the distance between sample and template.

We combine the coefficient matrix $\mathbf{B}$ with the distance weight matrix $\mathbf{W}$, and get a sparse mapping table $\mathbf{X}$ as follows, in which $\mathbf{X} = \mathbf{B}^\top \odot \mathbf{W} \in \mathbb{R}^{M \times N}$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{p1} & \cdots & x_{pN} \\ x_{(p+1)1} & \cdots & x_{(p+1)N} \\ \vdots & \ddots & \vdots \\ x_{(p+n)1} & \cdots & x_{(p+n)N} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1,pos} & \cdots & \mathbf{x}_{N,pos} \\ \mathbf{x}_{1,neg} & \cdots & \mathbf{x}_{N,neg} \end{bmatrix} \tag{20}$$

where $\mathbf{x}_{i,pos}$ and $\mathbf{x}_{i,neg}$ are vectors which represent the discrimination features of candidate samples in the positive and negative templates respectively, where $\mathbf{x}_{i,pos} = (x_{i1}, \ldots, x_{pi})^\top$ and $\mathbf{x}_{i,neg} = (x_{(p+1)i}, \ldots, x_{(p+n)i})^\top$.

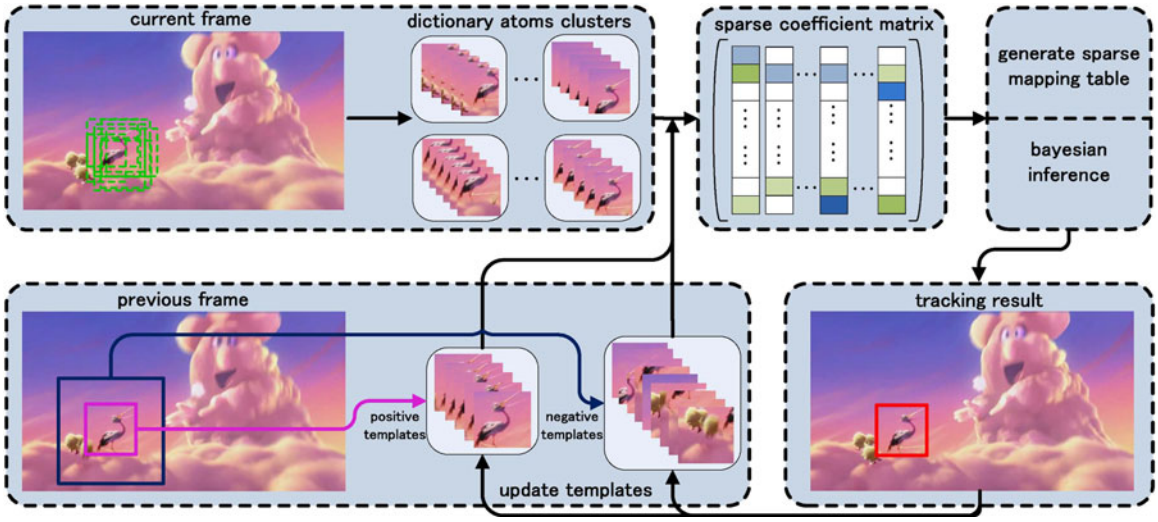


Fig. 5. Flowchart of the proposed tracking algorithm based on the inverse group sparse structure. It includes the process of dictionary atoms clustering, the process of positive templates and negative templates extraction, the process of templates reconstruction, the process of sparse mapping table generation, the following steps are to choose the best tracking result and update templates respectively.

Algorithm 2: ISGL-Based Object Tracking Method

Input: frame at time τ, previous target state $\mathbf{Z}^*_{\tau-1}$, template set $\mathbf{T}_{\tau-1}$, dictionary $\mathbf{Y}$, parameter α, λ_1 and λ_2

Output: current target state $\mathbf{Z}^*_\tau$, updated template set $\mathbf{T}_\tau$

1: Obtain $\mathcal{G}$ dictionary clusters of dictionary $\mathbf{Y}$ by K-means algorithm.
2: Compute the coefficient vector $\mathbf{b}$ for each template set by **Algorithm 1**, get the coefficient matrix $\mathbf{B}$.
3: Calculate the weight matrix $\mathbf{W}$ according to (19), get the Sparse mapping table $\mathbf{X}$ according to $\mathbf{X} = \mathbf{B}^\top \odot \mathbf{W}$.
4: Calculate the discrimination for each candidate, and choose the optimal target state $\mathbf{Z}^*_\tau$.
5: Update template set $\mathbf{T}_{\tau-1}$ to $\mathbf{T}_\tau$.
6: **return** $\mathbf{Z}^*_\tau$, $\mathbf{T}_\tau$

The discrimination score for the candidate sample $\mathbf{y}_i$ can be calculated by the following equation:

$$dis_i = \sum \mathbf{x}_{i,pos} - \sum \mathbf{x}_{i,neg} \tag{21}$$

where dis_i is the observation discrimination score of $\mathbf{y}_i$, $\mathbf{dis} = (dis_1, dis_2, \ldots, dis_N)$ indicates the scores for all candidates.

A candidate sample with a larger positive score or a lower negative score is more likely to be the target object, that is, a good target candidate has a relatively high discriminative score, whereas a bad candidate has a low one. The likelihood of the sample $\mathbf{y}_i$ being the target at state $\mathbf{Z}_\tau$ is computed within the Bayesian framework for our observation model $p(\mathbf{O}_i \mid \mathbf{Z}_\tau) \propto dis_i$. With the motion model and observation model, the maximum a posterior (MAP) criterion is used to select the best target observation by maximize $p(\mathbf{O}_i \mid \mathbf{Z}_\tau)$.

C. Update Strategy

The template set $\mathbf{T}$ consists of positive templates (target templates) and negative templates (background templates), so the positive and negative templates are updated respectively. In updating the positive templates, we modified the method by Zhuang *et al.* [9] by employing hash distance to measure the image similarity.

To update the negative templates, when the significant part of the target object is occluded, error occurs in the tracking result. At this time, target information is contained in the negative templates if update continuously. To solve this problem, we control the update of negative templates based on a threshold ψ. Denote that the discrimination score of tracking result at frame τ is $\widehat{dis}_\tau$, which is obtained by the score set $\mathbf{dis}_\tau$ at frame τ. V_τ denotes the variance of the discrimination score set $(\widehat{dis}_{\tau-4}, \widehat{dis}_{\tau-3}, \widehat{dis}_{\tau-2}, \widehat{dis}_{\tau-1}, \widehat{dis}_\tau)$. In a stable tracking, the change of discrimination is smooth. When interference occurs in tracking, $\widehat{dis}_\tau$ changes drastically. So if V_τ is greater than a threshold ψ, the tracking of the current frame is interfered and the negative templates remain unchanged; otherwise, they are updated. Algorithm 2 summarizes our ISGL tracking method.

V. Experimental Results

A. Experimental Setup and Evaluation Metrics

The proposed tracking method is implemented with Matlab R2012a, and evaluated in a PC with Intel i5 CPU (3.20 GHz) and 16 GB memory. The parameters are empirically determined and fixed for each test sequences as follows. We randomly selected 10 positive templates and 140 negative templates, and 300-600 candidate samples according to different experiment sequences, the size of the warped image is 32×32. In the optimization process, the number of atoms cluster $\mathcal{G}$ is 6, both λ_1 and λ_2 are 0.04, α is 0.03, the threshold η of positive template update is 0.35, and the threshold ψ of negative template update is 0.15×10^{-6}.

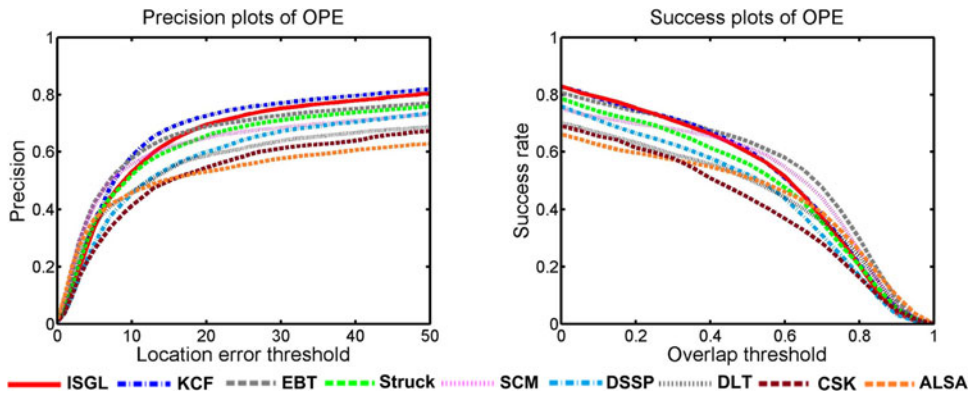


Fig. 6. Precision and success rate of OPE for the top 10 trackers. The trackers are ranked by the performance score of precision and success. The precision is the value at location error threshold of 20 pixels, and the success rate is the area-under-curve.

TABLE I
OVERALL PERFORMANCE SCORE OF THE TOP NINE TRACKERS IN TERMS OF PRECISION AND SUCCESS RATE

Overall performance	**ISGL**	**KCF**	**EBT**	**SCM**	**Struck**	**DSSP**	**DLT**	**ALSA**	**CSK**
Precision	0.695	**0.726**	0.689	0.649	0.656	0.599	0.587	0.545	0.532
Success rate	0.502	0.505	**0.532**	0.499	0.474	0.440	0.436	0.434	0.398

The three best performances are indicated by different fonts.

The parameters are decided with cross validation and a detailed analysis is presented in Section V-D.

To evaluate our tracking algorithm, we conduct experiments with OTB dataset [45] against 20 tracking methods: Struck [3], SCM [4], DSSP [9], CXT [12], CSK [13], ASLA [14], IVT [20], MTT [21], SPT [22], DFT [23], MIL [25], L1APG [35], LOT [46], IST [47], WLCS [48], CT [49], LSS [50], KCF [10], EBT [11] and DLT [26]. The 50 sequences in OTB datasets are tagged with 11 attributes including fast motion (FM), background clutters (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out of plane rotation (OPR), out of view (OV) and scale variation (SV), which cover the most challenging factors in visual tracking.

The performance of our tracker is quantitatively evaluated by precision and success under the one pass evaluation (OPE) criterion. The OPE criterion used the ground truth object location in the first frame for evaluation. The precision plot demonstrates the percentage of frames which the distance between the tracked location and the ground-truth is within a given threshold, and the tracker are ranked by the precision score while the threshold equal to 20 pixels. Meanwhile, the success plot is calculated by the overlap ratio [51], $score = area(R_T \cap R_G)/area(R_T \cup R_G)$, where R_T is the tracking result area, and R_G is the ground truth area. The success plot means the percentage of frames where the overlap ratio is greater than a threshold $\phi \in [0, 1]$. The area under curve (AUC) [45] is applied to rank the performance in success plot. For clarity, we only present the top 9 trackers in each plot.

B. Quantitative Analysis

1) Overall Performance: The overall performance of the top 9 trackers in terms of success rate and precision is illustrated in Fig. 6 with respect to the location errors and overlaps. In addition, the performance score is shown in Table I. Overall, ISGL method performs favorably based on the OPE criterion.

TABLE II
SCORE OF PRECISION PLOT IN DIFFERENT ATTRIBUTES

Attribute	*LR*	*SV*	*MB*	*DEF*	*OV*	*BC*
KCF	0.379	0.680	**0.589**	**0.702**	**0.649**	**0.752**
ISGL	**0.690**	**0.701**	0.587	0.643	0.589	0.653
EBT	0.411	0.696	0.504	0.585	0.569	0.652
Struck	0.545	0.639	0.551	0.521	0.539	0.585
SCM	0.305	0.672	0.339	0.586	0.429	0.578
DSSP	0.458	0.560	0.474	0.581	0.206	0.567
DLT	0.396	0.590	0.453	0.563	0.444	0.495
CSK	0.411	0.503	0.342	0.476	0.379	0.585
ALSA	0.156	0.552	0.278	0.445	0.333	0.496

The three best performances are indicated by different fonts.

In the precision plot in Fig. 6, the ISGL tracker achieves an average precision score of 0.695, which outperforms sparse representation based trackers including SCM, DSSP and ASLA. The underlying reason for the performance improvement is that the ISGL method integrates both group sparse constraint and local structure constraint.

Note that the ISGL method employs the gray feature and achieves comparable performance with the two best trackers (EBT and KCF) that use much complicated HOG features. As depicted in the Success rate plot in Fig. 6, ISGL outperforms KCF and EBT in most overlap rates, which demonstrate the superior robustness of our proposed method.

2) Attribute-Based Evaluation: The attributes (or cases) are representative for analyzing the performance of the trackers in handling different challenges. Tables II and III summarize the

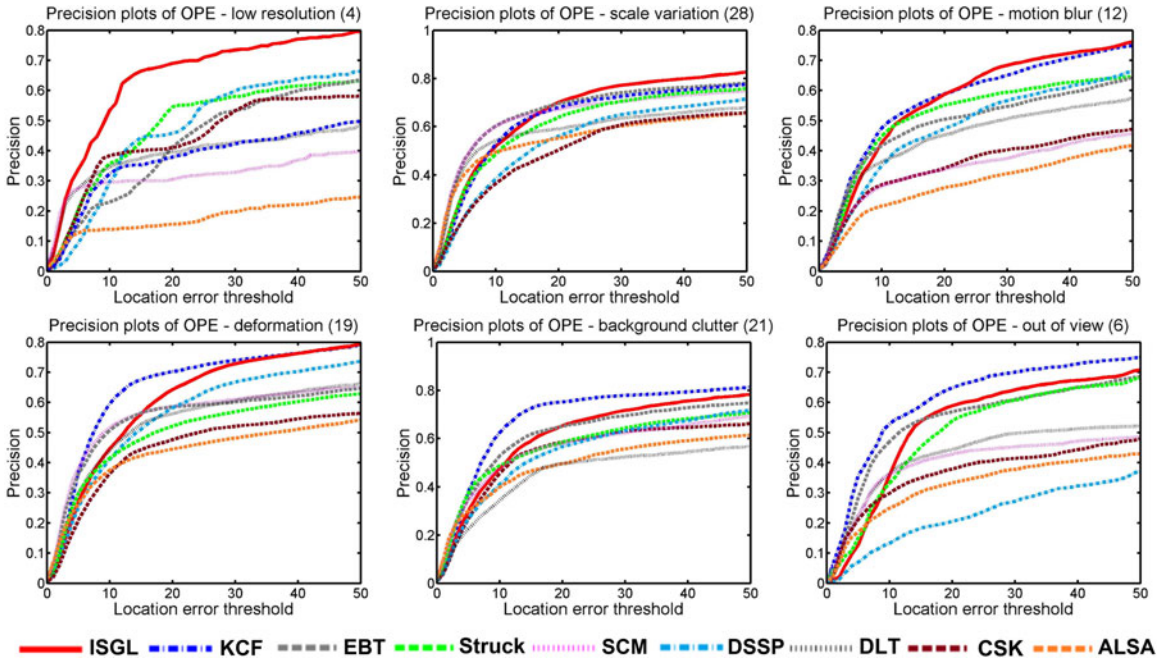


Fig. 7. Precision of sequences with different attributes in terms of the OPE criterion. The number of sequences in every attribute is marked in the panel title.

TABLE III
SCORE OF SUCCESS PLOT IN DIFFERENT ATTRIBUTES

Attribute	*LR*	*SV*	*MB*	*DEF*	*OV*	*BC*
EBT	0.334	**0.529**	0.424	0.443	0.498	0.496
KCF	0.310	0.427	**0.465**	**0.511**	**0.550**	**0.533**
ISGL	**0.474**	0.491	0.448	0.466	0.490	0.467
SCM	0.279	0.518	0.298	0.448	0.361	0.450
Struck	0.372	0.425	0.433	0.393	0.459	0.458
DSSP	0.331	0.388	0.374	0.440	0.201	0.417
DLT	0.346	0.455	0.363	0.394	0.367	0.339
ALSA	0.157	0.452	0.258	0.372	0.312	0.408
CSK	0.350	0.350	0.305	0.343	0.349	0.421

The three best performances are indicated by different fonts.

tracking results in terms of success and precision plots. Fig. 7 illustrates the precisions of the top 9 performers in six scenarios. It is shown that ISGL demonstrates much superior performance to the other methods in scenarios including low resolution, scale variation, and motion blur. The precision of ISGL is improved by an average of 31.1% and 27.9% in contrast to KCF and EBT (the best performers in overall performance), respectively, in the low resolution scenario. It performs highly competitively in the other scenarios.

Fig. 8 illustrates the success rate with respect to overlap ratio. As the overlap ratio increases, the success rate of all methods decreases. Among the compared methods, the ISGL method exhibits much improved performance, particularly in the low resolution case. This is partially due to the inclusion of group sparsity in the target evaluation as well as the inverse sparse representation. In comparison with the CSK method, which relies on the gray feature as used in ISGL method, ISGL clearly demonstrates the advantage in robustness.

In Comparison with the sparse representation based trackers (SCM, DSSP and ASLA), our ISGL method exhibits greater performance in all cases. As shown in Figs. 7 and 8, our approach demonstrates outstanding results in both precision and success rate.

C. Qualitative Analysis

1) Heavy Occlusion: Occlusion is one of the most frequently encountered problems in object tracking, the tracking results in sequence *Faceocc1* and *Jogging1* are illustrated in Fig. 9(a) and 9(b). Our method enhanced adaptability to object tracking when there exist heavy occlusions. It is noteworthy that our proposed method locates the target successfully despite it is completely obscured by a pillar for a period (#80) in sequence *Jogging1*. As our template update scheme selectively updates the templates with adaptive learning, incorrect targets are abandoned to prevent an inappropriate template. The target detected in the initial frames is kept as a reference. Hence, the influence from the occluded object is suppressed and the template sets are not severely affected in the updating process.

2) Motion Blur: Fig. 9(c) and 9(d) show the tracking result in sequences *Deer* and *Jumping* with the almost illegible appearance. Our proposed ISGL method outperforms the other tracking methods, most of which exhibit a drift away and fail to locate the target accurately. Our tracker exploits the group sparsity constraint, which leads to an accurate representation and increases the tracking performance.

3) Illumination Variation: Fig. 9(e) and 9(f) present the performance of different tracking methods in the presence of dramatic light changes. Since we represent the templates with a local structure constraint that reduces the reconstruction error, the relationship between the template and the corresponding samples is reinforced. A robust observation model is obtained that enables adaptation to illumination changes.

4) Scale Variation: The targets in sequences *CarScale* and *Walking2* experience greater scale changes which are shown in

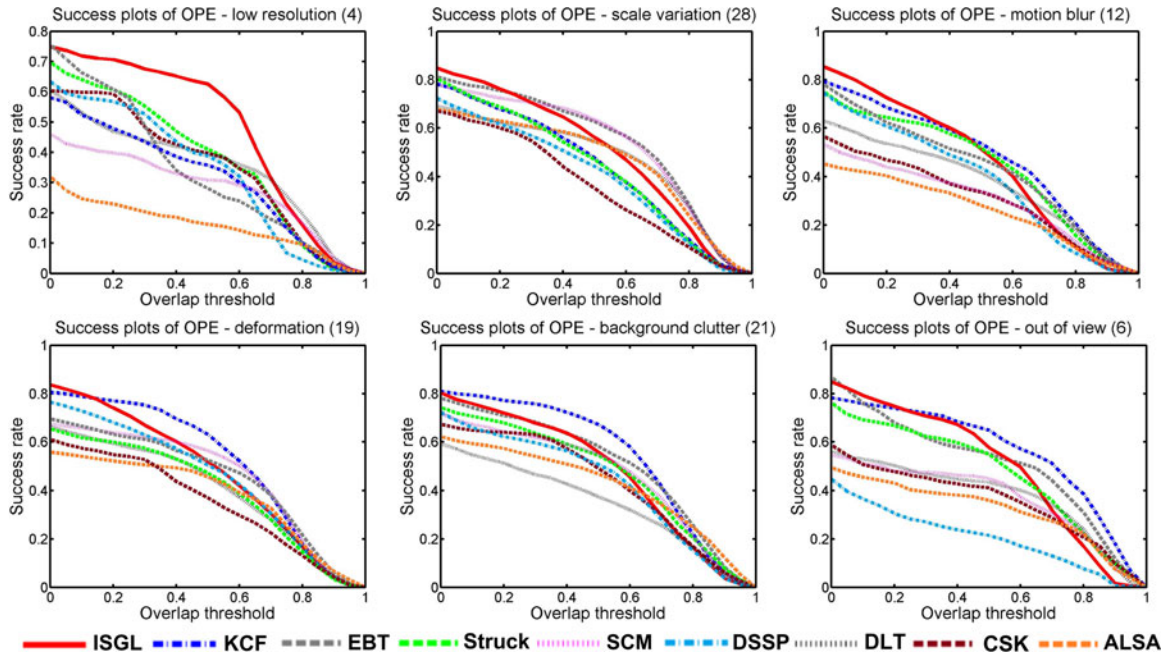


Fig. 8. Success rate of sequences with different attributes in terms of the OPE criterion. The number of sequences in every attribute is marked in the panel title.

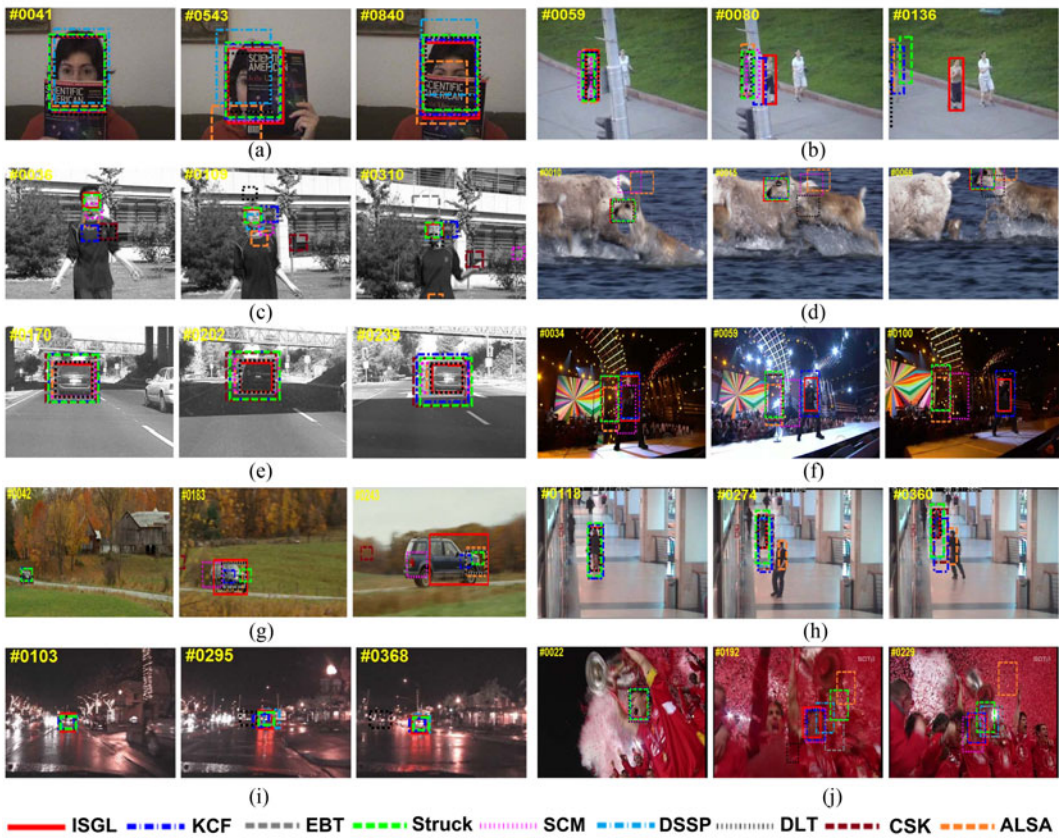


Fig. 9. Sampled results of the top 10 tracking algorithms in ten challenging sequences *Faceocc1*, *Jogging1*, *Jumping*, *Deer*, *Car4*, *Singer2*, *CarScale*, *Walking2*, *CarDark*, and *Soccer*. The targets in these sequences undergo heavy occlusion, motion blur, illumination variation, scale variation, and background clutter, respectively.

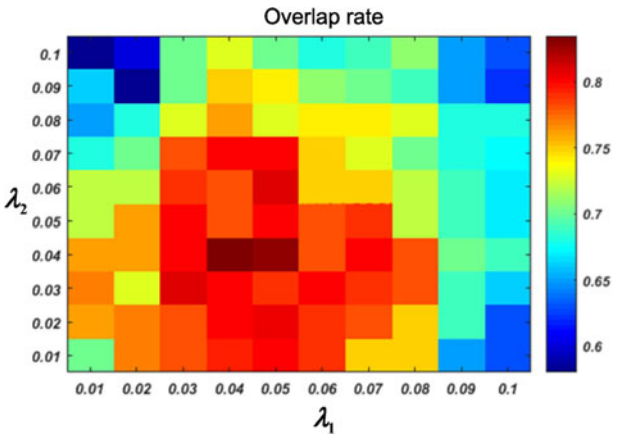


Fig. 10. Overlap rate using different λ_1 and λ_2 values.

Fig. 9(g) and 9(h). As the application of affine transformation in our method, different scaling particles are generated to be the candidate samples, our method select an optimal sample in the candidate sets with the same scale. By contrast, the methods in [10] and [11] fail to adapt to the scale changes without the affine transformation strategy.

5) Background Clutter: The sequences *Soccer* and *CarDark* in Fig. 9(i) and 9(j) are challenging due to complex background and poor illumination and contrast, which make it difficult to distinguish the target from the clutter. The templates of foreground and background allow the tracker to compare the similarity (in terms of the discrimination score) to the interested object. In addition, a discriminative mapping table is obtained by combining the sparse coefficient matrix and distance weight matrix, which enables improved precision.

D. Analysis of Parameters

Cross validation is used to choose the parameters using the overlap rate [51]. We randomly selected 11 sequences with different attributes from OTB database, which were used as our validation datasets.

1) Intra-group and Inter-group Sparsity (λ_1 and λ_2): The intra-group and inter-group sparsity are decided by λ_1 and λ_2, respectively, as shown in (11). When λ_1 is small, more similar samples are selected, which produces a reconstruction with redundant samples. When λ_1 is large, relevant samples could be missed because the coefficients are over-sparse. Similarly, λ_2 adjusts the sparsity between groups. The average overlap rate performance on validation datasets with various λ_1 and λ_2 are shown in Fig. 10. The overlap rate is color-coded with red indicating higher values and blue indicating lower values. It is clear that the best performance is achieved at $\lambda_1 = 0.04$ and $\lambda_2 = 0.04$, which were used in the rest of our experiments.

2) Local Structure Constraint (α): In our objective function as shown in (11), α is the weight for the local structure constraint. A small α could omit the contribution from the difference between the template and the atom, which results in a degraded performance in terms of accuracy. Fig. 11 illustrates the average overlap rate of our ISGL method on the validation datasets with different α values. The greatest overlap rate is achieved with α at 0.03.

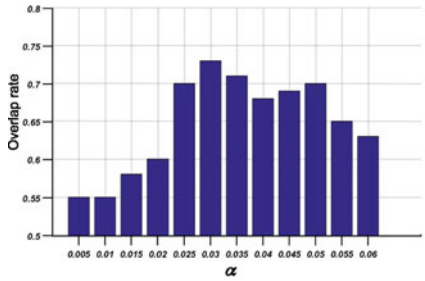


Fig. 11. Overlap rate using different α values.

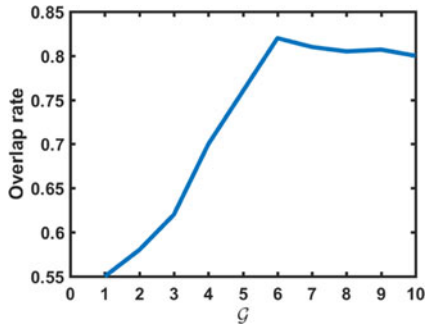


Fig. 12. Overlap rate with respect to the number of clusters.

3) Number of Clusters ($\mathcal{G}$): An appropriate number of clusters presents the plausible relevance of the samples. A large or a small $\mathcal{G}$ results in a less representative group formation and hence degraded tracking performance. In the extreme case where a cluster consists of one sample, the group property is completely disregarded. Fig. 12 depicts the overlap rate with respect to the number of clusters on the validation datasets. With a small number of clusters, the performance in terms of overlap rate is very low. As $\mathcal{G}$ increases from 1, the overlap rate increases as well as shown in Fig. 12. When $\mathcal{G}$ reaches 6, the overlap rate is maximized, which yields the best performance. There is a slight drop as we further increase $\mathcal{G}$.

E. Analysis of ISGL With Different Configuration

In this section, we evaluate the effectiveness of different components to detect their contributions. We propose three variants of ISGL for comparisons: the ISGL_WOLS method, the ISGL_WOWM method and the SGL method. The ISGL_WOLS method exploits the inverse sparse group lasso model without considering the local structure. The ISGL_WOWM method is the ISGL algorithm without a refined weight matrix $\mathbf{W}$. The SGL method utilizes sparse group lasso model instead of the inverse sparse group lasso model. In addition, we include the DSSP algorithm, which represents the method only exploits inverse sparse representation model. The performance of DSSP method, our method and its variants is illustrated in Fig. 13. The results demonstrate that without considering the local structure, the score of precision rate reduces by 3.1% and the AUC score of success rate reduces by 1.3%. Besides that, Fig. 13 depicts the precision and success

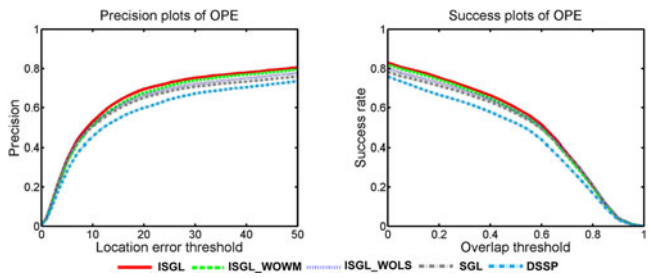


Fig. 13. Precision plots and success plots of OPE for ISGL with different configuration. The DSSP serves as the baseline performance.

curves of the ISGL method with and without a distance weight matrix (denoted with ISGL and ISGL_WOWM, respectively) with respect to various thresholds. As shown in the two plots, ISGL outperformed ISGL_WOWM consistently. The improvement is due to the refined coefficients by leveraging the weight matrix. Meanwhile, the SGL method achieve a precision score of 0.652 and an AUC score of 0.478, which implies that the inverse sparse model improves the overall performance by 4% and 2.4% in terms of precision and success, respectively.

By making comparisons between DSSP and the variants of ISGL method, we notice that all variants perform better than the DSSP method. These results show that our proposed inverse sparse group lasso model and the local structure information played important parts in the ISGL algorithm for a robust visual tracking.

F. Computational Cost

The most time-consuming aspect of our ISGL algorithm is in the process of computing the sparse coefficients. The per-frame complexity of ISGL is $O(eM)$, where M is the number of template and e is the dimension of the feature vector. Among the methods in our comparison study, SCM and DSSP are the ones that employ sparse representation, and both exhibit satisfactory performance as shown in Fig. 6. The complexity of DSSP [9] is $O(eM)$ and the complexity of SCM [4] is $O(eN)$, where N is the number of samples. In practice, the number of templates is less than the number of samples albeit they are usually in the same order. Hence, the complexity of these methods is comparable. The average frame per second (FPS) of SCM, MTT and L1APG are 0.4, 1.0 and 2.4, respectively, whereas our method processes 2.5 frames per second in average. This is superior to the best performing and existing sparse tracker SCM.

VI. Conclusion

In this paper, we propose a robust tracking method based on inverse sparse group lasso model, which combines the inverse sparse representation and the sparse group lasso model. In our tracker, we integrate the inter-group and the intra-group sparsity constraints that enable effective tracking of objects in complex environments. Local structure between the templates and samples ensure improved robustness and accuracy. In order to improve the discrimination of coefficients on positive and negative templates, hash distance is adopted to construct a sparse mapping table for the selection of the optimal tracking results. Moreover, the adaptive updating strategy reduces drifts and accounts for varying appearance in dynamic scenes. The experimental results demonstrate that, under the disturbance such as rotation, occlusion, scale change, and rapid movement, our algorithm achieved greater accuracy and robustness compared to the state-of-the-art methods.

References

[1] V. Kilic, M. Barnard, W. Wang, and J. Kittler, "Audio assisted robust visual tracking with adaptive particle filtering," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 186–200, Feb. 2015.

[2] X. Zhou, L. Xie, Q. Huang, S. J. Cox, and Y. Zhang, "Tennis ball tracking using a two-layered data association approach," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 145–156, Feb. 2015.

[3] 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.

[4] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[5] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 14–21.

[6] Y. Wu, H. Ling, E. Blasch, L. Bai, and G. Chen, *Visual Tracking Based on Log-Euclidean Riemannian Sparse Representation*. Berlin, Germany: Springer-Verlag, 2011, pp. 738–747.

[7] X. Yuan, L. Kong, D. Feng, and Z. Wei, "Automatic feature point detection and tracking of human action in time-of-flight videos," *IEEE/CAA J. Automat. Sinica*, to be published.

[8] B. Ma *et al.*, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[9] 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.

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

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

[12] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1177–1184.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

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

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Mathematics*, vol. 32, no. 2, pp. 407–451, 2004.

[16] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," *Adv. Neural Inf. Process. Syst.*, vol. 22, no. 11, pp. 82–89, 2009.

[17] Y. T. Chi, M. Ali, M. Rushdi, and J. Ho, "Affine-constrained group sparse coding and its application to image-based classifications," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 681–688.

[18] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc.*, vol. 68, no. 1, pp. 49–67, 2006.

[19] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *Statistics*, 2010.

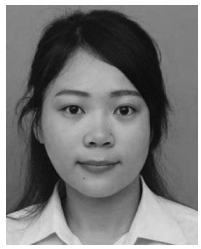
[20] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 125–141, 2008.

[21] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2042–2049.

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

[23] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1910–1917.

[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] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 983–990.
[26] N. Wang and D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.
[27] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[28] S. Gao, L. T. Chia, I. W. H. Tsang, and Z. Ren, "Concurrent single-label image classification and annotation via efficient multi-layer group sparse coding," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 762–771, Apr. 2014.
[29] Z. Zhu, F. Guo, H. Yu, and C. Chen, "Fast single image super-resolution via self-example learning and sparse representation," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2178–2190, Dec. 2014.
[30] F. Shao, K. Li, W. Lin, G. Jiang, and Q. Dai, "Learning blind quality evaluator for stereoscopic images using joint sparse representation," *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 2104–2114, Oct. 2016.
[31] L. Zhao, Q. Hu, and W. Wang, "Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1936–1948, Nov. 2015.
[32] K. Li, J. Yang, and J. Jiang, "Nonrigid structure from motion via sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1401–1413, Aug. 2015.
[33] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep.-Oct. 2009, pp. 1436–1443.
[34] R. D. Lascio, P. Foggia, G. Percannella, A. Saggese, and M. Vento, "A real time algorithm for people tracking using contextual reasoning," *Comput. Vis. Image Understand.*, vol. 117, no. 8, pp. 892–908, 2013.
[35] 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 Recog.*, Jun. 2012, pp. 1830–1837.
[36] H. Liu and F. Sun, "Visual tracking using sparsity induced similarity," in *Proc. IEEE Int. Conf. Pattern Recog.*, Aug. 2010, pp. 1702–1705.
[37] F. Li, H. Lu, D. Wang, Y. Wu, and K. Zhang, "Dual group structured tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1697–1708, Sep. 2016.
[38] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Statist.*, vol. 22, no. 2, pp. 231–245, 2013.
[39] X. Zhu, Z. Huang, J. Cui, and H. T. Shen, "Video-to-shot tag propagation by graph sparse group lasso," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 633–646, Apr. 2013.
[40] Y. Yang and J. Jiang, "Hybrid sampling-based clustering ensemble with global and local constitutions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 952–965, May 2016.
[41] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 2399–2434, 2006.
[42] Y. Yang, Y. T. Zhuang, F. Wu, and Y. H. Pan, "Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 437–446, Apr. 2008.
[43] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[44] M. K. Mıhçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Proc. ACM Conf. Comput. Commun. Security*, 2001, pp. 13–21.
[45] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2411–2418.
[46] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1940–1947.
[47] D. Wang, H. Lu, Z. Xiao, and M. H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, Sep. 2015.
[48] D. Wang, H. Lu, and C. Bo, "Visual tracking via weighted local cosine similarity," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1838–1850, Sep. 2015.
[49] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.
[50] D. Wang, H. Lu, and M. H. Yang, "Least soft-threshold squares tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2371–2378.
[51] 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.



Yun Zhou received the B.S. degree in electronic and information engineering from the School of Physics and Electronic Information Engineering, Anhui Normal University, Wuhu, China, the M.S. degree in signal and information processing from the School of Computers and Information, Hefei University of Technology, Hefei, China, in 2010 and 2013, respectively, and is currently working toward the Ph.D. degree in the School of Computer and Information, Hefei University of Technology.

Her current research interests include computer vision and pattern recognition.

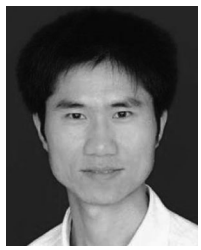
Jianghong Han received the B.S. degree in computer application technology from the Hefei University of Technology, Hefei, China, in 1982.

He is currently a Professor with the School of Computer and Information, Hefei University of Technology. His research interests include computer control, communication and information systems, machine learning, and computer vision.

Xiaohui Yuan (S'01–M'05–SM'16) received the B.S. degree in electrical engineering from the Hefei University of Technology, Hefei, China, in 1996, and the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 2004.

He is currently an Associate Professor with the Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA. His research interests include computer vision, data mining, machine learning, and artificial intelligence. His research findings have been reported in more than 100 peer-reviewed papers.

Prof. Yuan was the recipient of Ralph E. Powe Junior Faculty Enhancement Award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013.



Zhenchun Wei received the B.S. and Ph.D. degrees in computer application technology from the Hefei University of Technology, Hefei, China, in 2000 and 2007, respectively.

He is currently an Associate Professor with the School of Computer and Information, Hefei University of Technology. His research interests include internet of things, wireless sensor networks, distributed system, and machine learning.

Richang Hong (M'14) received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008.

He was a Research Fellow with the School of Computing, National University of Singapore, Singapore, from September 2008 to December 2010. He is currently a Professor with Hefei University of Technology. He has coauthored more than 70 publications in the areas of his research interests, which include multimedia content analysis and social media.

Prof. Hong is a Member of the ACM and an Executive Committee Member of the ACM SIGMM China Chapter. He was the Associate Editor of *Information Sciences and Signal Processing*, and the Technical Program Chair of the MMM 2016. He was the recipient of the Best Paper Award in the ACM Multimedia 2010, the Best Paper Award in the ACM ICMR 2015, and the Honorable Mention of the IEEE TRANSACTIONS ON MULTIMEDIA Best Paper Award.