Robust Visual Tracking with Dual Group Structure

Fu Li, Huchuan Lu and Dong Wang

Dalian University of Technology, Dalian, China
lifu.dlut@gmail.com, {lhchuan, wdice}@dlut.edu.cn

Abstract. The "sparse representation"-based tracking framework generally considers the testing candidates and dictionary atoms individually, thus failing to model the structured information within data. In this paper, we present a robust tracking framework by exploiting the *dual group structure* of both candidate samples and dictionary templates, and formulate the sparse representation at group level. The similar samples are encoded simultaneously by a few atom groups, which induces the inter-group sparsity, and also each group enjoys different internal sparsity. In this way, not only the potential commonality shared by the related candidates is taken into account, but also the individual differences between samples are reflected. Then we provide an effective optimization method to solve our formulation by two stages: thresholding and computing with the accelerated proximal gradient method. Finally, we embed the dual group structure model into the particle filter framework for visual tracking. Extensive experimental results demonstrate that our tracker achieves favorable performance against the state-of-the-art tracking methods.

1 Introduction

As one of the most active research topics in recent years, sparse representation (SR) has been widely investigated in numerous practical fields (such as face recognition [1], image restoration [2], object tracking [3] and so on), and achieves quite satisfactory performance. The fundamental assumption of the SR framework is that testing samples from each class reside on a low-dimensional linear subspace which is spanned by the training samples belonging to the given class. Therefore, every testing sample can be approximately represented by a set of training samples (dubbed dictionary) with the sparse constraint.

Tracking can generally be categorized into generative methods (e.g., [4], [5], [6]) and discriminative methods (e.g., [7], [8], [9]). As a generative tracking model, SR has been extensively studied in the past several years (e.g., [10], [11], [12]). In the "sparse representation"-based tracking framework, the samples collected from the first frame and subsequent tracking results are often directly used as the dictionary atoms, and then the ℓ_1 minimization problem attempts to seek for a sparse representation of the testing sample by selecting a few columns of the dictionary. However, treating the dictionary atoms individually suffers several disadvantages [13], [14]. Due to the ignorance of the underlying commonality

shared by dictionary atoms, it tends to make selection based on the strength of individual column rather than the strength of groups of similar atoms. Thus, in the SR framework, it is prone to select only one atom from the group and does not care which one is selected.

In the tracking problem, the dictionary enjoys the group property due to the temporal continuity and appearance similarity. For one thing, the tracked object usually undergoes small motions between two consecutive frames. For another, as time proceeds, the target may share similar appearance with that in some previous time. By clustering the dictionary atoms into several groups, we can represent a candidate with a few groups, rather than individual atoms. The active groups include semblable templates with the testing candidate, and representation based on multiple templates contributes to more robust tracking.

Many of current tracking algorithms are based on the Particle Filter (PF) framework, in which hundreds of particles are drawn based on the state of the previous frame and used to depict the appearance of the tracked object. In these methods, the SR method is straightforwardly applied to each testing sample for obtaining the sparse coefficients, in which both dictionary atoms and candidate samples are treated to be individual. Although this manner is simple and easy to be implemented, it is not very satisfactory as it completely ignores the structured information within dictionary atoms and within candidate samples, and it is also computationally expensive by verifying a large number of candidate samples individually. To consider the potential relationships among candidates, Zhang *et al.* [5] design a SR-based tracker based on the $\ell_{2,1}$-norm, which sparsely codes all candidate samples simultaneously. The regularizer based on $\ell_{2,1}$-norm encourages all samples to share the same sparsity pattern and exploits the underlying relationships among different candidate samples. However, since there always exist obvious differences between candidates, this compulsive constraint may lead to undesirable results. Because of the densely sampling strategy, the appearance of some candidates may be very similar, and therefore we can divide candidate samples into some disjointed groups. To reveal the common characteristics among samples in a group, they are encoded jointly and represented by the same atom groups. In addition, we adopt the ℓ_1 norm to account for the variation between individuals, thereby inducing sparsity within group. Finally, the subgroup with the minimum reconstruction error is selected and the weighted sum over all particles in this selected group is regarded as the final state, which would lead to more robust and stable results than only picking one candidate.

In this paper, we propose a novel tracking formulation exploiting the group structure of both candidate samples and dictionary atoms, which we name *Dual Group Structure*. The structured information within candidate samples considers the potential commonality shared by the related samples, ensuring that data with similar appearance are encoded jointly and bringing large gains in terms of computational efficiency. The structural information within dictionary atoms encourages the grouping effect of coefficients, leading to the selection of a group of atoms which come from the same set rather than an individual atom. Moreover, the sparse-inducing regularizer yields sparsity at both the group and atom

level, that is, not only a few groups of atoms are active at a time, but also each group enjoys internal sparsity. In this way, samples from the same class will share group properties, but will not necessarily share the full active sets as they are not identical. The solution to our model can be achieved by using two basic stages: thresholding and computing. Finally, we embed the proposed dual group structure model into the particle filter framework for visual tracking, and adopt challenging image sequences to evaluate the proposed tracker. The experimental results demonstrate the effectiveness of the proposed tracking algorithm in comparison with other competing trackers.

Contributions: The contributions of this work can be summarized into three folds. **(1)** We exploit the underlying structured information of similar candidates and similar dictionary atoms, and formulate the sparse representation process at the group level. Each sample group is represented by a few atom groups, and inside each atom group only a few members are active at a time. By using this manner, it not only makes full use of the commonality shared by data from the same group, but also takes the differences between individuals into consideration. **(2)** We provide an efficient optimization procedure by using the Accelerated Proximal Gradient (APG) method. The solution process includes a matrix thresholding and a vector thresholding, naturally yielding to the desired inter-group and intra-group sparsity pattern. **(3)** We design a generative tracker based on the proposed dual group structure model. Numerous experiments show that the proposed tracking algorithm achieves favorable performance against many state-of-the-art trackers.

2 Related Work

Group Sparse Coding: In recent years, the group property in the SR framework (often called group sparsity) has drawn interesting attentions (such as [15], [16], [17] and so on), where dictionary atoms are often divided into several disjointed groups. Given these group memberships, the task is to seek for a solution where a query sample is represented by only a small set of the groups, rather than a few atoms. Yuan and Lin [13] first propose the group lasso criterion for this problem, which exploits the sum of ℓ_2-norm to set most of group coefficients to be exactly zeros. While the group lasso method can provide a sparse set of groups, it fails to consider the sparsity property within each group. To model both sparsity of groups and within each group, Friedman *et al.* [18] present the sparse group lasso by adding an additional ℓ_1-norm regularization term. This model achieves the effect of promoting group selection while at the same time leading to overall sparse feature selection. Based on this theory, several works focus on the practical applications to computer vision. Elhamifar and Vidal [19] cast the face classification problem as a structured sparse recovery problem, the goal of which is to approximate the testing sample by using the minimum number of blocks from the dictionary. Zhang *et al.* [20] utilize the group sparsity properties in feature selection for the image annotation task, which leverages both sparsity and clustering priors to prune the features. Liu *et al.* [21] use a

dynamic group sparsity scheme to exploit temporal and spatial relationship for object tracking, which can be solved by a two stage optimization approach. However, all these approaches sparsely code the testing samples individually and do not consider the latent similarities among different samples, thus will causing heavy computational load and the loss of structured information among data.

Simultaneous Sparse Coding: Another line of group coding, dubbed simultaneous sparse coding, offers a solution that involves the potential relation among samples by coding all testing data jointly. Based on the assumption that features or data within a group are expected to share the same sparsity pattern in their representations, a mixed $\ell_{2,1}$-norm is employed to make all the column vectors of the coefficient matrix look alike. Mairal *et al.* [2] jointly decompose groups of similar patches on the dictionary and combine the non-local means and sparse coding approaches to image restoration within a unified framework. Zhang *et al.* [5] employ mixed norms to enforce the joint sparsity and learn particle representations together to improve the tracking performance. Chi *et al.* [16] propose the affine-constrained group sparse coding and extend the sparse representation framework to classification problems with multiple inputs. However, these methods treat the dictionary atoms individually, and thus cannot lead to sparsity in group level. Furthermore, the constraint of forcing these similar yet not identical samples to have the same representations is relatively strong.

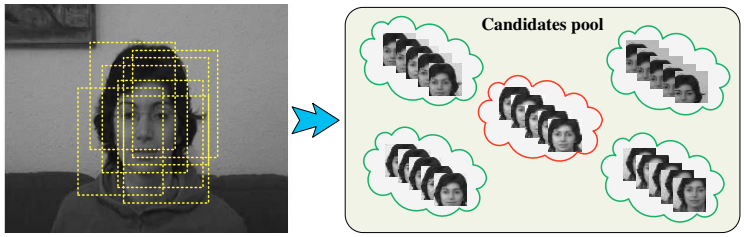
Our Work: The proposed formulation takes advantage of the structural constraints of both dictionary atoms and testing samples. On one hand, instead of considering the atoms as singletons, we divide the atoms into groups, with a few of groups active at a time. On the other hand, multiple similar samples are encoded simultaneously, requesting that they all share the same active set. Besides the common characteristics, the sparsity regularizer within each group is added to account for the intrinsic differences between individuals.

3 Problem Formulation

Given a set of observed samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{m \times n}$, where each column $\mathbf{x}_i$ can be the vectorized image or extracted feature vector, the task is to encode these samples by a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ (the column vector $\mathbf{d}_i$ denotes the i-th atom of the dictionary $\mathbf{D}$). By solving some optimization problems, the coefficient matrix $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n] \in \mathbb{R}^{k \times n}$ can be obtained as the encodings of $\mathbf{X}$, with one column corresponding to one sample. Recently, many sparse-inducing regularizers have been proposed in the literature (e.g., [15], [22]), and most of them are based on the sparse-promoting property of the ℓ_1 norm.

In order to achieve group sparsity, we can suppose that the k atoms are divided into $\mathcal{G}$ groups (classes). For ease of notation, we use a matrix $\mathbf{D}^{(g)}$ to represent the set of atoms within the g-th group, and adopt a matrix $\mathbf{S}^{(g)}$ to stand for the corresponding coefficients. Then for each individual sample $\mathbf{x}_i$, the sparse group lasso criterion [18] is formulated as follows:

$$\min_{\mathbf{s}_i} \frac{1}{2}\|\mathbf{x}_i - \sum_{g=1}^{\mathcal{G}} \mathbf{D}^{(g)}\mathbf{s}_i^{(g)}\|_2^2 + \lambda_1 \sum_{g=1}^{\mathcal{G}} \|\mathbf{s}_i^{(g)}\|_2 + \lambda_2 \sum_{g=1}^{\mathcal{G}} \|\mathbf{s}_i^{(g)}\|_1, \tag{1}$$



(a) (b)

Fig. 1. (a) The dense sampling strategy in particle filter framework. (b) The collected candidate samples. We can see they are of strong correlations, and thus can be divided into several groups.

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the ℓ_2-norm and ℓ_1-norm respectively, parameters λ_1 and λ_2 control the balance between the two regularization terms.

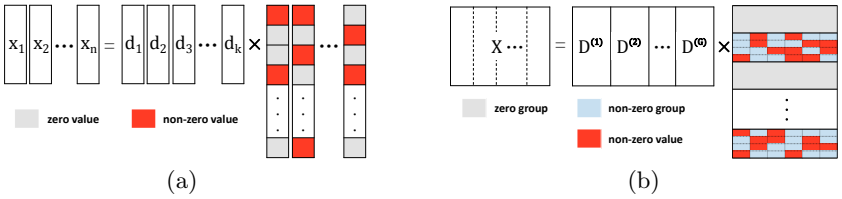
It can be seen from equation (1) that the sparse group lasso method takes the structured information within the dictionary by using an ℓ_2-norm constraint on the coefficients of each group. However, it fails to model the relationships among data samples (i.e., to exploit the structured information of similar samples in $\mathbf{X}$), which is not a good candidate method to solve many vision problems (such as visual tracking). In the "particle filter"-based tracking framework, the candidates are usually densely sampled according to the object's state in the last frame. Thus, these candidate samples are of strong correlations (i.e., have sufficient structured information), as shown in Figure 1.

In order to exploit the structured information of similar samples in $\mathbf{X}$, we also classify the n data samples into $\mathcal{L}$ groups (classes) based on some criterion. For example, if data is image patch, each group may be the set of patches in a particular image; if instances are human faces, then each group may consist of facial images from one person under different illumination, pose and expression conditions. Likewise, we denote $\mathbf{X}^{(l)}$ as the submatrix correlated to the l-th group and sparsely code one group data jointly. Thus, we can define our objective function as

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X}^{(l)} - \sum_{g=1}^{\mathcal{G}} \mathbf{D}^{(g)}\mathbf{S}^{(g)}\|_F^2 + \lambda_1 \sum_{g=1}^{\mathcal{G}} \|\mathbf{S}^{(g)}\|_F + \lambda_2 \sum_{g=1}^{\mathcal{G}} \|\mathbf{S}^{(g)}\|_1, \qquad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm of matrix. The sum of the F-norm regularizer induces the sparsity in group level, while ℓ_1-norm encourages sparsity in an individual level. It means that samples in $\mathbf{X}^{(l)}$ share group properties as they are from the same class, but will not share the active sets since they are not identical.

We thereby obtain a collaborative sparse model, with the cooperation to identify the class labels by all samples, and the freedom at the individual level inside the group to adapt to each particular image. The objective function in equation (2) is the sum of three convex functions, and thus, the optimization



(a) (b)

Fig. 2. (a) Sparse representation where both testing samples and dictionary atoms are treated to be individual, thus there are no relations among the learned coefficients. (b) Dual group structure model where the coefficient matrix enjoys group sparsity and in-group sparsity. Not only a few of groups are selected, but also in each group, minority of elements admit non-zero values.

problem (2) is a convex one. In Section 3.1, we will provide an effective solution to this problem. The sparsity patterns of SR and our model are illustrated in Figure 2.

Here, we note that $\sum_{g=1}^{\mathcal{G}} \|\mathbf{S}^{(g)}\|_1 = \sum_{i=1}^{n} \|\mathbf{s}_i\|_1$. When $\lambda_1 = 0$, the grouping effect is neglected, then equation (2) reduces to the original sparse representation (lasso) problem [23]. If each individual sample is treated as a group, the optimization problem in equation (2) reduces to the sparse group lasso problem [18]. In addition, when $\lambda_2 = 0$ and all dictionary atoms are treated as a single group, the equation (2) turns into the multi-task learning problem with the mixed $\ell_{1,1}$-norm [5]. Therefore, we can conclude that all three above-mentioned problems can be viewed as special cases of the proposed formulation.

3.1 Theoretical Calculation

Since equation (2) with the ℓ_1-regularization is non-differentiable at zero, the standard unconstrained optimization methods cannot be applied directly. In the following, we develop an optimization method based upon coordinate descent to solve this problem. The formulation can be separable with respect to $\mathbf{S}^{(g)}$, and thus we can update $\mathbf{S}^{(g)}$ individually by fixing other group coefficients. For each subproblem, the solution can be obtained from two stages: thresholding and computing.

Formally, the submatrix $\mathbf{S}^{(g)}$ is obtained by solving the following optimization problem:

$$\mathbf{S}^{(g)} = \arg\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{R} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda_1\|\mathbf{Z}\|_F + \lambda_2\|\mathbf{Z}\|_1, \tag{3}$$

where $\mathbf{R} = \mathbf{X}^{(l)} - \sum_{j \neq g} \mathbf{D}^{(j)}\mathbf{S}^{(j)}$ is the residual.

Thresholding: We first check whether the elements of $\mathbf{Z}$ are all zeros, which means the corresponding atom group is not activated. Let $f = \frac{1}{2}\|\mathbf{R} - \mathbf{D}\mathbf{Z}\|_F^2$, then the subgradient of the function $\frac{1}{2}\|\mathbf{R} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda_1\|\mathbf{Z}\|_F + \lambda_2\|\mathbf{Z}\|_1$ with respect to each Z_{ij} is:

$$\nabla f + \lambda_1 P_{ij} + \lambda_2 T_{ij}, \quad \forall\, i, j \tag{4}$$

where matrices $\mathbf{P}$ and $\mathbf{T}$ are the subgradient matrices of F-norm and ℓ_1-norm of $\mathbf{Z}$ respectively. The element $P_{ij} = Z_{ij}/\|\mathbf{Z}\|_F$ if $\mathbf{Z}$ is not a null matrix, and otherwise $\mathbf{P}$ is a matrix satisfying $\|\mathbf{P}\|_F \leq 1$. Similarly, the element $T_{ij} = sign(Z_{ij})$ if $Z_{ij} \neq 0$, and $T_{ij} \in [-1, 1]$ if $Z_{ij} = 0$. Moreover, when $\mathbf{Z} = \mathbf{0}$, we can obtain that the first term $\nabla f = -A_{ij}$, where $\mathbf{A} = \mathbf{D}^\top \mathbf{R}$.

To achieve group sparsity, we focus on the case where $\mathbf{Z}$ is a null matrix, then a necessary and sufficient condition for $\mathbf{Z}$ to be zero is that the system of equations,

$$A_{ij} = \lambda_1 P_{ij} + \lambda_2 T_{ij}, \quad \forall\, i, j \tag{5}$$

has a solution with $\|\mathbf{P}\|_F \leq 1$ and $T_{ij} \in [-1, 1]$. With some mathematical manipulations, we can determine this by minimizing the function of $\mathbf{T}$:

$$J(\mathbf{T}) = (1/{\lambda_1}^2) \sum_{i,j} (A_{ij} - \lambda_2 T_{ij})^2 = \sum_{i,j} P_{ij}^2 \tag{6}$$

with respect to $T_{ij} \in [-1, 1]$ and then check if $J(\hat{\mathbf{T}}) \leq 1$. The minimizer is easily seen to be

$$\hat{T}_{ij} = \begin{cases} \frac{A_{ij}}{\lambda_2}, & |\frac{A_{ij}}{\lambda_2}| \leq 1 \\ sign(\frac{A_{ij}}{\lambda_2}), & |\frac{A_{ij}}{\lambda_2}| > 1 \end{cases} \tag{7}$$

and we can compute $J(\hat{\mathbf{T}})$ by equation (6). If $J(\hat{\mathbf{T}}) \leq 1$, then we directly set $\mathbf{Z} = \mathbf{0}$ and proceed to solve for the next submatrix.

Computing: Now in the case where $J(\hat{\mathbf{T}}) > 1$, we can see that equation (3) is actually the sum of a convex differential function (the first two terms) and a separable penalty, and hence we can employ the APG method to efficiently solve this convex optimization problem. As compared to traditional projected gradient methods, the APG method achieves an $\mathcal{O}(\frac{1}{t^2})$ residual from the optimal solution after t iterations with quadratic convergence [24]. Specifically, APG proceeds the iterative update between the current coefficient matrix $\mathbf{Z}_t$ and an aggregation matrix $\mathbf{V}_t$. Each APG iteration consists of two steps: (1) a generalized gradient mapping step that updates $\mathbf{Z}_t$ with $\mathbf{V}_t$ fixed, where in general an analytic solution is needed to ensure the materialization of APG, and (2) an updating step that promotes $\mathbf{V}_t$ by linearly combining $\mathbf{Z}_{t+1}$ and $\mathbf{Z}_t$.

(1) Gradient Mapping: Given the current estimate $\mathbf{V}_t$, we obtain $\mathbf{Z}_{t+1}$ by solving the following equation:

$$\mathbf{Z}_{t+1} = \arg\min_{\mathbf{Y}} \frac{1}{2}\|\mathbf{Y} - \mathbf{H}\|_F^2 + \tilde{\lambda}\|\mathbf{Y}\|_1, \tag{8}$$

where $\tilde{\lambda} = \eta\lambda_2$ and η is a small step parameter. Denote $g = \frac{1}{2}\|\mathbf{R} - \mathbf{D}\mathbf{V}_t\|_F^2 + \lambda_1\|\mathbf{V}_t\|_F$, then

$$\begin{aligned} \mathbf{H} &= \mathbf{V}_t - \eta\nabla g_t \\ &= \mathbf{V}_t - \eta\mathbf{D}^\top(\mathbf{D}\mathbf{V}_t - \mathbf{R}) - \eta\lambda_1 \frac{\mathbf{V}_t}{\|\mathbf{V}_t\|_F}. \end{aligned} \tag{9}$$

We decouple equation (8) into several disjoint subproblems, one for each row vector $\mathbf{z}^i$:

$$\mathbf{z}_{t+1}^i = \arg\min_{\mathbf{y}^i} \frac{1}{2}\|\mathbf{y}^i - \mathbf{h}^i\|_F^2 + \tilde{\lambda}\|\mathbf{y}^i\|_1. \tag{10}$$

Each subproblem is a variant of the projection problem unto the ℓ_1 ball. The optimization can be solved by a soft-thresholding method and the solution is obtained as $\mathbf{z}_{t+1}^i = \mathcal{S}_{\tilde{\lambda}}(\mathbf{h}^i)$, where $\mathcal{S}_{\tilde{\lambda}}$ is the soft-thresholding operator defined as $\mathcal{S}_{\tilde{\lambda}}(a) = sign(a)\max(0, |a| - \tilde{\lambda})$. Note that the $\max(\cdot)$ operator induces the sparsity within group, and samples in the same group enjoy different sparsity patterns.

(2) Updating: We update $\mathbf{V}_t$ as follows:

$$\mathbf{V}_{t+1} = \mathbf{Z}_{t+1} + \alpha_{t+1}(\frac{1}{\alpha_t} - 1)(\mathbf{Z}_{t+1} - \mathbf{Z}_t), \tag{11}$$

where α_t is conventionally set to $\frac{2}{t+3}$. We summarize the algorithm of the APG computing stage in Algorithm 2.

Suppose the number of samples in group g is ρ_g. The computational complexity in thresholding step concentrates on the multiplication of matrices, i.e., the calculation of matrix $\mathbf{A}$, and thus the complexity is $\mathcal{O}(mn\rho_g)$. While in the second stage, the computational complexity is dominated by the gradient computation in equation (9) and the soft-thresholding operation in equation (10). Similarly, the complexity of equation (9) is $\mathcal{O}(mn\rho_g)$, while that of equation (10) is $\mathcal{O}(n\rho_g)$. Thus the total complexity of one iteration is $\mathcal{O}((2m+1)n\rho_g)$, linear with respect to the group size ρ_g, therefore the solution can be obtained efficiently.

Our overall algorithm is summarized in Algorithm 1. The convergence is achieved when the relative change in solution falls below a predefined tolerance after several cyclic iterations.

3.2 Noise Handling

In the noisy scenarios, samples are often corrupted by noise or partially occluded. To deal with the unknown corruption, a set of trivial templates are added after the dictionary as in the previous works [1], [3]. Then the occluded part is modeled as sparsely additive noises that can take on large values anywhere in the representation.

We employ the identity matrix $\mathbf{I}$ as the trivial atoms, and the corresponding coefficient matrix is denoted as $\mathbf{S}^{(I)}$. The nonzero entries of $\mathbf{S}^{(I)}$ indicate the pixels in sample that are corrupted or occluded. We regard all trivial templates as an atom group and along with other groups solve equation (2) to obtain the coefficients. In this way, a set of occluded samples can be represented by both the related dictionary group of the same class and the trivial group.

Algorithm 1: Learning dual group regularized sparse codes

Input: sample matrix $\mathbf{X}$, dictionary $\mathbf{D}$, sample group set $\{1, 2, \ldots, \mathcal{L}\}$, atom group set $\{1, 2, \ldots, \mathcal{G}\}$, regularization parameters λ_1 and λ_2, learning step η.
Output: coefficient matrix $\mathbf{S}$.

1. Initialize $\mathbf{S} = \mathbf{0}$.
2. **For** $l = 1$ to $\mathcal{L}$
3. Initiativly set $g = 1$.
4. Calculate $\mathbf{R} = \mathbf{X}^{(l)} - \sum_{j \neq g} \mathbf{D}^{(j)}\mathbf{S}^{(j)}$, $\mathbf{A} = \mathbf{D}^\top \mathbf{R}$.
5. Compute $J(\hat{\mathbf{T}})$ according to equation (6) and equation (7).
6. Check whether $J(\hat{\mathbf{T}}) \leq 1$. If so, set $\mathbf{S}^{(g)} = \mathbf{0}$ and proceed to step 7 for the next group directly. Otherwise go to step 6.
7. Compute $\mathbf{S}^{(g)}$ using Algorithm 2.
8. If $g == \mathcal{G}$, reset $g = 1$, else update $g = g + 1$.
9. Iterate the cyclic optimization for $g = 1, 2, \ldots, \mathcal{G}, 1, 2, \ldots$ until convergence.
10. **End**
11. **Return** coefficient matrix $\mathbf{S}$.

Algorithm 2: learning coefficient with the APG method

Input: residual matrix $\mathbf{R}$, dictionary $\mathbf{D}$, warm start $\mathbf{Z}$, learning steps η and $\tilde{\lambda}$.
Output: Coefficient matrix $\mathbf{Z}$.

1. Initialize $t = 0$, $\alpha_t = 1$.
2. If $\mathbf{Z}$ is null matrix, set $\mathbf{Z}_0 = \mathbf{1}$, else $\mathbf{Z}_0 = \mathbf{Z}$. $\mathbf{V}_0 = \mathbf{Z}_0$.
3. **While** not converged **do**:
4. Compute $\mathbf{H}$ according to equation (9).
5. Solve the subproblem equation (10) to obtain $\mathbf{Z}_{t+1}$.
6. Set $\alpha_{t+1} = \frac{2}{t+3}$.
7. Update $\mathbf{V}_{t+1}$ by equation (11).
8. $t = t + 1$.
9. **End**
10. **Return** coefficient matrix $\mathbf{Z}$.

3.3 Visual Tracking

For object tracking task, the tracking results are usually directly used as the dictionary atoms. Since the target object often undergoes various pose changes in the tracking process, the dictionary covers diversity of the appearance variations of the target. Therefore, these dictionary atoms enjoy the group structure of the consecutive tracking results or the similar appearance at different time. We cluster these atoms using K-means method. By sparse group coding, the correct target sample is reconstructed by sparse grouped templates.

When the new frame arrives, large amount of candidates are drawn around the target location in the previous frame. Individually treating them could be computationally expensive. To explore the structural information of positions and features among candidates, we also divide these samples into several disjoint-

ed groups according to their coordinates and appearance. Denote $\boldsymbol{\phi} = [x, y, \mathbf{q}^\top]^\top$ as the extracted feature from a candidate, where x and y are the coordinates, $\mathbf{q}$ is a response vector such as intensity, color or gradients, then the candidates can be clustered by K-means or spectral clustering method. We can also add some weights on the coordinates to adjust their contributions.

For each candidate group $\mathbf{X}^{(l)}$, we compute the corresponding coefficient matrix using equation (2), and obtain the reconstruction error only by the best dictionary atom subset:

$$e(l) = \min_g \|\mathbf{X}^{(l)} - \mathbf{D}^{(g)}\mathbf{S}^{(g)}\|_F^2. \tag{12}$$

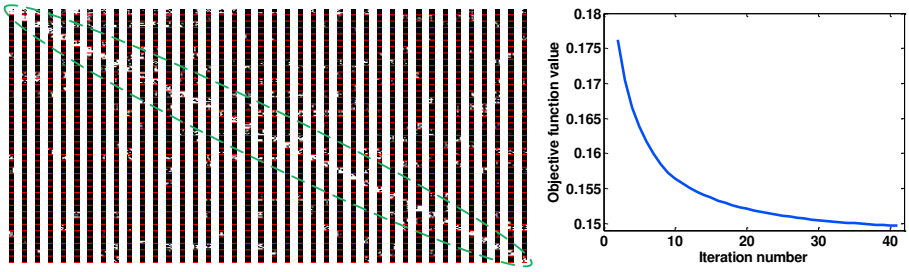
The best group with the minimum error e is then picked out. The weighted sum over all particles in this group is taken as the final target location, where the weight of each particle is inversely proportional to the reconstruction error. We update the dictionary by replacing the old templates with the new coming tracking results and re-cluster it every five frames. The occlusion handling strategy in [12] is adopted to prevent the blocked part from being updated into the dictionary.

4 Experiments

4.1 Sparsity Pattern

To demonstrate the effectiveness of the proposed method, we first conduct experiments on the ORL face database [25] and examine the sparsity pattern of the learned coefficient matrix. The ORL database contains 400 frontal face images of 40 subjects under different pose and expression conditions. Each face image is scaled to 48×48 pixels and normalized in the preprocessing. A subset with half numbers per individual is collected to form the training set. Images with the same labels are clustered into one atom group and these groups are arranged in the order from label 1 to 40. Note that in the testing stage, the image is not labeled one by one, instead we treat the testing faces of one subject as a whole, and estimate the belonging of this group.

The sparsity patterns of all 40 subjects are illustrated in Figure 3(a) ordered by the true label, with the red line indicating the group splitting line in each pattern. In the ideal condition, the testing group is represented by only the atom group of the same class, thus resulting in that the non-zero values concentrate on the diagonal line from the overall point of view. For example, in the coefficient relating to the first person, the elements corresponding to the first atom group enjoy most large non-zero values, and thus this atom group admits the minimum reconstruction error and shares the same label with the testing group. Due to the presence of noise, there may exist non-zero entries in other groups. In addition, the convergence curve is shown in Figure 3(b). We can see that our algorithm could reach convergence smoothly after several iterations without vibrations.



(a) (b)

Fig. 3. (a) Illustration of the coefficient matrix of all 40 subjects in ORL face database. The white entries indicate the non-zero values while the black ones represent zero elements. (b) Convergence curve.

4.2 Tracking Experiments

In the implementation, each target is initialized manually by a bounding box in the first frame. We resize each image region to 32×32 pixels for post-processing. The parameters λ_1, λ_2 and η are set to 0.01, 0.01 and 0.1 in all experiments. As a trade-off between effectiveness and speed, 600 particles are adopted and our tracker is incrementally updated every 5 frames. The number of atom groups and candidate groups are set to 5 and 10, respectively.

We evaluate the performance of the proposed method on sixteen challenging sequences from [26] and our own. The challenges of these videos include partial occlusion, illumination variation, pose change, deformation and scale change. The proposed tracker is compared with twelve state-of-the-art algorithms including the MIL [7], IVT [27], TLD [28], VTD [29], MTT [5], CT [9], ℓ_1-APG [10], NDLT [30], LSHT [31], SCM [32], STK [8], PBT [33] methods. Both qualitative and quantitative comparative results are presented below.

Table 1. Success rate. The top two results are shown in red and blue fonts respectively.

	MIL	IVT	TLD	VTD	MTT	CT	ℓ_1-APG	NDLT	LSHT	SCM	STK	PBT	OURS
Car4	0.27	**1.00**	**0.86**	**1.00**	0.38	0.27	**1.00**	**1.00**	0.27	**1.00**	0.28	0.39	**1.00**
David2	0.33	0.88	0.96	**0.99**	**1.00**	0.01	**1.00**	0.96	**0.99**	0.91	**1.00**	**0.99**	0.98
David3	0.35	0.61	0.18	0.53	0.56	0.32	0.05	0.67	0.75	0.58	**0.68**	0.57	**0.98**
Faceocc1	0.76	**1.00**	0.78	**0.97**	**1.00**	0.66	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
Skater	0.94	0.88	0.48	**0.98**	**1.00**	0.95	0.81	0.41	0.16	0.87	0.44	**1.00**	**1.00**
Crossing	**0.99**	0.23	0.52	0.42	0.23	**0.99**	0.25	0.82	0.44	**1.00**	0.96	**0.99**	**1.00**
Jogging	0.16	0.19	0.86	0.16	0.16	0.15	0.19	0.17	0.15	**0.89**	0.18	0.17	**1.00**
Seq1	0.34	0.21	0.94	0.45	0.31	0.23	**0.99**	0.21	0.94	**0.99**	0.68	0.45	**1.00**
Singer1	0.25	0.94	0.46	**0.95**	0.35	0.25	**1.00**	0.48	0.25	**1.00**	0.26	0.23	**1.00**
Stone	0.28	0.51	0.23	0.62	**1.00**	0.21	0.83	0.14	0.29	0.95	0.60	0.41	**0.97**
Leno	0.53	**1.00**	0.82	**1.00**	0.98	0.97	**1.00**	**1.00**	0.79	**0.99**	0.78	0.92	**1.00**
Toystory	0.72	0.94	0.27	**0.99**	0.39	0.39	0.37	0.37	0.36	0.28	0.38	0.75	**1.00**
Walking	0.55	**0.99**	0.39	0.84	**0.99**	0.53	**0.99**	0.97	0.55	0.95	0.64	0.55	**0.98**
Walking2	0.39	**0.99**	0.34	0.40	**0.99**	0.39	0.97	0.41	0.39	**1.00**	0.46	0.42	0.96
Mountainbike	0.58	**1.00**	0.26	**1.00**	0.95	0.17	0.83	**1.00**	**0.99**	0.98	0.86	**1.00**	0.97
Dog1	0.65	0.86	0.68	0.71	0.79	0.65	**0.99**	0.88	0.65	0.85	0.65	0.65	**1.00**

Table 2. Average center error (in pixels). The best two results are shown in red and blue fonts respectively.

	MIL	IVT	TLD	VTD	MTT	CT	ℓ_1-APG	NDLT	LSHT	SCM	STK	PBT	OURS
Car4	60.1	**2.9**	-	12.3	37.2	234.1	16.4	6.0	27.2	3.5	27.6	12.4	**3.1**
David2	10.9	1.4	5.0	2.9	**1.3**	76.7	1.4	2.7	2.9	3.8	1.5	1.9	**1.1**
David3	**38.2**	52.9	173.0	61.9	65.5	69.5	233.4	52.8	50.9	64.1	54.6	66.3	**5.5**
Faceocc1	32.3	9.3	17.6	11.1	14.1	30.7	6.8	6.3	4.6	**3.2**	16.2	9.1	**4.3**
Skater	11.1	14.9	-	15.2	**8.8**	13.4	12.4	32.3	118.4	17.5	22.5	8.9	**5.9**
Crossing	3.2	2.8	24.3	26.1	56.5	3.6	54.5	4.1	29.2	**1.3**	2.8	**2.5**	3.2
Jogging	136.6	130.2	**6.4**	118.0	153.5	130.6	42.2	52.1	150.5	12.1	129.7	114.6	**6.2**
SeqI	48.8	111.7	7.9	45.1	87.8	52.4	5.5	87.6	13.4	**1.8**	17.1	34.3	**4.7**
Singer1	15.1	8.5	32.7	4.1	41.2	19.3	**3.1**	7.6	27.8	**3.7**	21.9	26.2	**3.7**
Stone	32.3	26.9	14.1	26.1	**2.2**	30.3	3.1	41.8	5.6	**2.8**	3.3	4.9	3.6
Leno	28.1	6.2	24.0	9.5	17.2	13.1	**5.9**	12.9	37.5	6.9	37.1	14.3	**4.8**
Toystory	34.5	17.4	-	**10.4**	59.4	47.2	68.8	56.3	67.4	68.2	64.8	29.4	**10.1**
Walking	3.4	**1.8**	10.2	5.8	2.9	6.9	1.9	**1.8**	5.5	2.4	4.6	8.3	**1.4**
Walking2	60.6	3.1	-	46.2	3.6	58.5	4.4	28.8	41.4	**2.1**	11.1	14.2	**2.8**
Mountainbike	73.0	7.4	-	9.8	7.3	214.3	25.5	**6.5**	**7.2**	10.5	8.6	9.1	8.2
Dog1	7.8	**3.5**	4.2	11.0	3.8	7.0	3.7	3.7	6.8	7.1	5.7	6.3	**3.2**

Quantitative Comparison: We first evaluate quantitatively the performance of the trackers mentioned above with the success rate criterion, which is defined as the ratio of the successfully tracked frames. Given the tracking bounding box B_T and the ground truth B_G, if the PASCAL VOC score $\frac{B_T \cap B_G}{B_T \cup B_G}$ is larger than 0.5, then tracking in the frame is regarded as successful. Table 1 presents the success rate results, where a bigger value means better performance. We also utilize the center location error between the tracking results and ground truth to assess these trackers. Table 2 shows the average center error in pixels, where the smaller the value is, the better the tracker performs. From Table 1 and 2, we can see that our tracker performs favorably against other state-of-the-art methods in terms of both criteria.

Qualitative Comparison:

Pose Change: The *Toystory* sequence is challenging for large pose deformation and dusky background. The target toy exhibits different moves and the other one also causes distraction to mislead the tracker. Most of other methods lose the target after frame *#147*. Since the dictionary in our model captures various target poses and the group structure exploits the temporal information and appearance similarity, our algorithm could track the target successfully throughout the whole sequence. In addition, the weighted sum of all promising candidates contributes to the tracking robustness. In sequences *Skater* and *Dog1*, the appearance of the target changes dramatically due to the pose variation, resulting in great difficulty for tracking. For all that, our tracker is able to catch the target accurately all through. The PAT and SCM methods also do well in some cases as they employ the part-based representations. The target faces in sequences *David2* and *Leno* experience out-of-plane rotation, causing the trackers to fail easily. But our method is able to locate the target all through.

Partial Occlusion: In the *David3*, *Jogging*, *Walking2* and *SeqI* sequences, the target is completely occluded by other similar objects or obstacles, making the tracker easy to drift. We can see that our method performs better than other

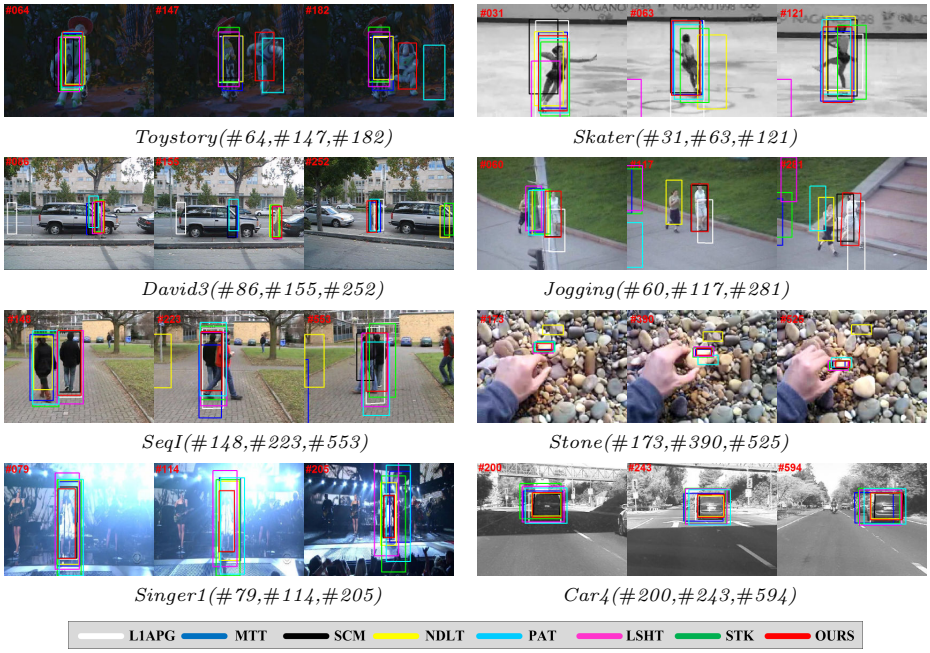


Fig. 4. Sampled tracking results on challenging image sequences. This figure demonstrates the results of seven state-of-the-art tracking methods and the proposed method. More results can be found in the supplementary material.

trackers in these cases, since we introduce the trivial template group to account for the occlusion and use the atom group structure to take advantage of the previous similar appearance information. In sequence *Faceocc1*, the tracked face is blocked by a book from different directions. Because the object undergoes little pose variation, majority of algorithms could track the target generally, yet our method achieves a relatively smaller center error.

Illumination Change and Background Clutter: In sequence *Car4*, the car passes under a bridge which blocks out the light, and in sequence *Crossing*, the human is crossing the sidewalk from the black shadow. While in sequence *Singer1*, the strength of the stage lighting increases all at once. Both the targets in these sequences experience severe illumination change, causing the image pixels to change a lot. Our algorithm is capable of handling this challenge due to the use of dual group structure, and locates the target more stably and accurately than others. The target in the *Stone* sequence is easy to be distracted by other stones of different shapes and colors with cluttered background. Likewise, our method could achieve favorable performance.

5 Conclusion

We exploit the dual group structure information of both dictionary atoms and testing candidate samples, and formulate the sparse representation at a group

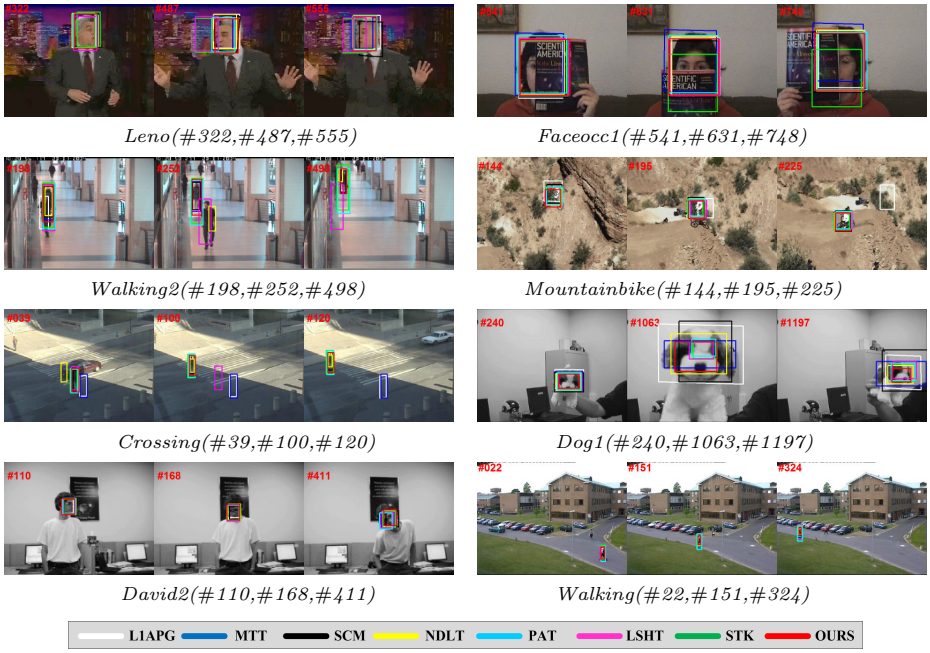


Fig. 5. Sampled tracking results on challenging image sequences. This figure demonstrates the results of seven state-of-the-art tracking methods and the proposed method. More results can be found in the supplementary material.

level. The commonalities shared by samples and the individual characteristics among data are both taken into account through inter-group sparsity and intra-group sparsity. In this way, the temporal continuity and appearance similarity of tracking results can be made full use of. The objective function is solved efficiently by two stages, thresholding and computing using the accelerated proximal gradient method. Then we devise a generative tracker based on the dual group structure model. Instead of selecting only one best candidate, we estimate the target location with the weighted sum over a set of related particles, which leads to a more stable and robust tracker. Numerous experiments on visual tracking are conducted with a wide variety of challenging factors, including partial occlusion, pose variation, illumination change and background clutter. Experimental results demonstrate that our tracker performs favorably against state-of-the-art methods.

Acknowledgement: This work was supported by the Joint Foundation of China Education Ministry and China Mobile Communication Corporation under Grant MCM20122071, and in part by the Fundamental Research Funds for the Central Universities under Grant DUT14YQ101 and the Natural Science Foundation of China under Grant 61472060.

References

1. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 210–227
2. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV. (2009)
3. Mei, X., Ling, H.: Robust visual tracking using ℓ_1 minimization. In: ICCV. (2009)
4. Wang, D., Lu, H., Chen, Y.: Incremental MPCA for color object tracking. In: ICPR. (2010)
5. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR. (2012)
6. Zhuang, B., Lu, H., Xiao, Z., Wang, D.: Visual tracking via discriminative sparse similarity map. IEEE Transactions on Image Processing **23** (2014) 1872–1881
7. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. (2009)
8. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV. (2011)
9. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV. (2012)
10. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR. (2012)
11. Jia, X., Lu, H., Yang, M.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR. (2012)
12. Wang, D., Lu, H., Yang, M.H.: Online object tracking with sparse prototypes. IEEE Transactions on Image Processing **22** (2013) 314–325
13. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B **68** (2006) 49–67
14. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B **67** (2005) 301–320
15. Bengio, S., Pereira, F.C.N., Singer, Y., Strelow, D.: Group sparse coding. In: NIPS. (2009)
16. Chi, Y.T., Ali, M., Rushdi, M., Ho, J.: Affine-constrained group sparse coding and its application to image-based classifications. In: ICCV. (2013)
17. Chi, Y.T., Ali, M., Rajwade, A., Ho, J.: Block and group regularized sparse modeling for dictionary learning. In: CVPR. (2013)
18. Friedman, J., Hastie, T., Tibshirani, R.: A note on the group lasso and sparse group lasso. CoRR (2010)
19. Elhamifar, E., Vidal, R.: Structured sparse recovery via convex optimization. CoRR (2011)
20. Zhang, S., Huang, J., Huang, Y., Yu, Y., Li, H., Metaxas, D.N.: Automatic image annotation using group sparsity. In: CVPR. (2010)
21. Liu, B., Yang, L., Huang, J., Meer, P., Gong, L., Kulikowski, C.: Robust and fast collaborative tracking with two stage sparse optimization. In: ECCV. (2010)
22. Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. In: AISTATS. (2010)
23. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B **58** (1994) 267–288
24. Nesterov, Y.: Gradient methods for minimizing composite functions. Math. Program. **140** (2013) 125–161

25. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: ACV. (1994)
26. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR. (2013)
27. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77** (2008) 125–141
28. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 1409–1422
29. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR. (2010)
30. Wang, N., Wang, J., Yeung, D.Y.: Online robust non-negative dictionary learning for visual tracking. In: ICCV. (2013)
31. He, S., Yang, Q., Lau, R.W., Wang, J., Yang, M.H.: Visual tracking via locality sensitive histograms. In: CVPR. (2013)
32. Zhong, W., Lu, H., Yang, M.: Robust object tracking via sparse collaborative appearance model. IEEE Transactions on Image Processing **23** (2014) 2356–2368
33. Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A.: Part-based visual tracking with online latent structural learning. In: CVPR. (2013)