DOI:10.11992/tis.201507073
网络出版地址:http://www.cnki.net/kcms/detail/23.1538.tp.201509030.1456.002.html

Efficient tracker based on sparse coding with Euclidean local structure-based constraint

WANG Hongyuan[1], ZHANG Ji[1], CHEN Fuhua[2]
(1. School of Information Science and Engineering, Changzhou University, Changzhou, Jiangsu, China 213164; 2. Department of Natural Science and Mathematics, West Liberty University, West Virginia, United States 26074)

Abstract:Sparse coding (SC) based visual tracking (l_1-tracker) is gaining increasing attention, and many related algorithms are developed. In these algorithms, each candidate region is sparsely represented as a set of target templates. However, the structure connecting these candidate regions is usually ignored. Lu proposed an NLSSC-tracker with non-local self-similarity sparse coding to address this issue, which has a high computational cost. In this study, we propose an Euclidean local-structure constraint based sparse coding tracker with a smoothed Euclidean local structure. With this tracker, the optimization procedure is transformed to a small-scale l_1-optimization problem, significantly reducing the computational cost. Extensive experimental results on visual tracking demonstrate the effectiveness and efficiency of the proposed algorithm.

Keywords:euclidean local-structure constraint; l_1-tracker; sparse coding; target tracking
CLC Number:TP18; TP301.6 **Document Code**:A **Article ID**:1673-4785(2016)01-0136-12

Citation:WANG Hongyuan, ZHANG Ji, CHEN Fuhua. Efficient tracker based on sparse coding with Euclidean local structure-based constraint[J]. CAAI Transactions on Intelligent Systems, 2016, 11(1): 136-147.

Recently, visual target tracking was widely used in security surveillance, navigation, human-computer interaction, and other applications[1-2]. In a video sequence, targets for tracking often change dynamically and uncertainly because of disturbance phenomena such as occlusion, noisy and varying illumination, and object appearance. Many tracking algorithms were proposed in the last twenty years that can be divided into two categories: generative tracking and discriminant tracking algorithms[1-2]. Generative algorithms (e.g., eigen tracker, mean-shift tracker, incremental tracker, covariance tracker[2]) adopt appearance models to express the target observations, whereas discriminant algorithms (e.g., TLD[3], ensemble tracking[4], and MILTrack[5]) view tracking as a classification problem, thus attempting to distinguish the target from the backgrounds. Here, we present a new generative algorithm.

Received Date:2015-07-31. **Online Pulication**:2015-09-30.
Foundation Item: National Natural Foundation of China under Grant (61572085,61502058).
Corresponding Author:Hongyuan Wang. E-mail: hywang@cczu.edu.cn.

Based on sparse coding (SC; also referred to as sparse sensing or compressive sensing)[6-7], Mei proposed an l_1-tracker for generative tracking[8-9], addressing occlusion, corruption, and some other challenging issues. However, this tracker incurs a very high computational cost to achieve efficient tracking (see section 2.1 and Fig.1 for details), and the local structures of similar regions are ignored, which may cause the instability and even failure of the l_1-tracker. Indeed, the sparse coefficients, for representing six similar regions (CR_1-CR_6) under ten template regions (T_1-T_{10}) with original l_1-tracker, are diversified (Fig. 3). Considering CR_1 and CR_4, for example, we can see that although the latter is almost the partial occlusion version of the former, their sparse representations are very different. Tracking CR_4 (the woman's face) may fail, because the tracker is likely to incorrectly consider the region T_8(the book) as its target.

Contrary to expectations, Xu proved that a sparse algorithm cannot be stable and that similar signals may not exhibit similar sparse coefficients[10]. Thus, a

trade-off occurs between sparsity and stability when designing a learning algorithm. In addition, instability in the l_1-optimization problem affects the performance of the l_1-tracker.

Lu developed a NLSSS-tracker (NLSSST) based on SC applying a non-local self-similarity constraint by introducing the geometrical information of the set of candidates as a smoothing term to alleviate the instability of the l_1-tracker[11]. However, its low efficiency (even slower than the original l_1-tracker, Table 4) restricts its applicability in real-time tracking. In this study, motivated by the robustness of the l_1-tracker and stability of NLSSST, we propose a novel tracker, called ELSS-tracker (ELSST), that is both robust and efficient. The main contributions of this study are as follows:

1) An efficient tracker, i.e., ELSST, is developed by considering the local structure of the set of target candidates. In contrast to the Lu5s[11] and Mei5s-tracker[8-9], our tracker is more stable and sparse.

2) The proposed tracker shows excellent performance in tracking different video sequences with regard to scale, occlusion, pose variations, background clutter, and illumination changes.

The rest of this study is organized as follows: l_1- and NLSSS-tracker are introduced in section 2; in section 3, we analyze the disadvantages of these two trackers and propose our tracker; experimental results with our tracker and four comparison algorithms are reported in section 4; the conclusion and future work are summarized in section 5.

1 Related works

1.1 Sparse coding and the l_1-tracker

Sparse coding is an attractive signal reconstruction method proposed by Candes[6-7] that reconstructs a signal $y \in \mathrm{R}^{m\times 1}$ with an over-complete dictionary $\boldsymbol{D} \in \mathrm{R}^{m\times(n+2m)}$ with a sparse coefficient vector$c \in \mathrm{R}^{n\times 1}$. The SC formulation can be written as the l_0-norm-constrained optimization problem as follows:

$$\min_c \| \boldsymbol{y} - \boldsymbol{Dc} \|_F^2 + \alpha \| \boldsymbol{c} \|_0 \qquad (1)$$

which is NP-hard, where $\| \cdot \|_F$denotes the vector's Frobenius norm (i.e., l_2-norm), and $\| \cdot \|_0$counts the number of non-zero elements of the vector. Candes proved that the l_1-norm $\| \cdot \|_1$ is the tightest upper bound of the l_0-norm $\| \cdot \|_0$, and thus, Eq.(1) can be rewritten as the following l_1-optimization problem[6-7]:

$$\min_c \| \boldsymbol{y} - \boldsymbol{Dc} \|_F^2 + \alpha \| \boldsymbol{c} \|_1 \qquad (2)$$

Based on SC, Mei presented a nice l_1-tracker for robust tracking[8-9] (Fig. 1). Considering that the target is located in the latest frame, the l_1-tracker is initialized in the new arrival frame and N candidate regions are generated with Bayesian inference (Fig. 1a, b). With n templates learned from previous tracking and $2m$ trivial templates (m positive ones and m negative ones, where m is the dimension of $1D$ stretched image, Fig. 1c), Eq.(2) can be solved (Fig. 1d,e,f). With positive and negative trivial templates, Mei added a non-negative constraint $c \geqslant 0$ in Eq.(2), with which the reconstruction errors of all candidate regions with SC coefficients can be used to determine the weights for each candidate, and the object in the new arrival frame can be located with the sum of the weighted candidates. The dictionaries updating strategies can be seen in[8-9].

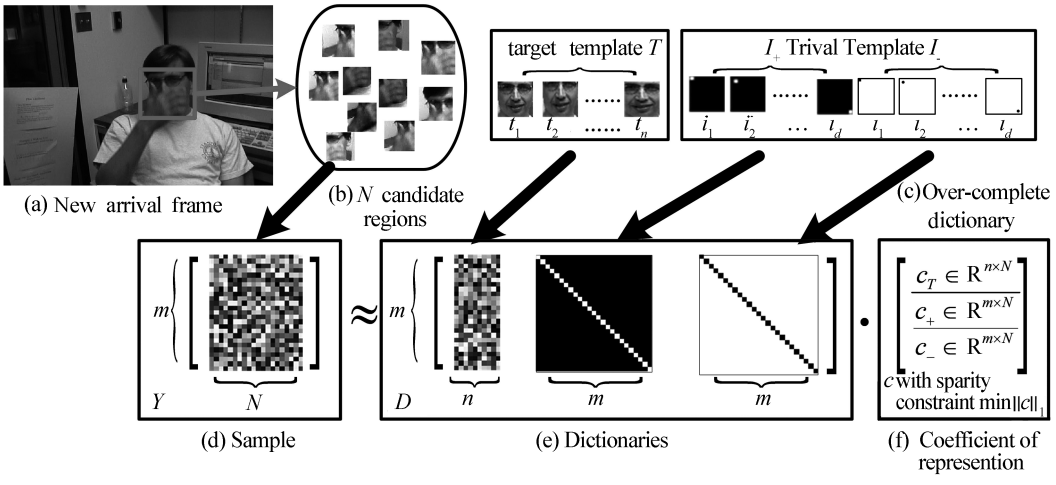


Fig.1　Original l_1-tracker algorithm

1.2 Non-local self-similarity based sparse coding for tracking (NLSSST)

Recently, Xu indicated the trade-off between sparsity and stability in sparse regularized algorithms[10]. Moreover, Yang pointed out the same A-optimization issue in pattern classification[12]. Based on the fact that lots of similar regions exist in all N candidates generated by Bayesian inference, Lu proposed his tracker with the non-local self-similarity constraint as

$$\sum_{i=1}^{n} \left\| c_i - \sum_{j=1}^{K} w_{ji} c_j \right\|^2 = \| \boldsymbol{C} - \boldsymbol{CW} \|_F^2 \quad (3)$$

where c_i and c_j are the sparse coefficients corresponding to the candidate regions y_i and y_j, respectively, and w_{ji} is the weight assigned to c_j. Given N m-dimensional candidates $\boldsymbol{Y} = [y_1 \cdots y_N] \in \mathrm{R}^{m \times N}$, the first K-closest candidate point aroundy_j is denoted by $N_K(y_j)$, and the weight $w_{ji} = \frac{1}{s_j} \mathrm{e}^{-\frac{\| N_K(y_j) - N_K(y_i) \|^2}{h}}$, where h is a parameter enforcing similarity, and s_j is the normalization factor. The weight w_{ji} measures the similarity between the K-neighborhood of y_j and y_i. Lu's algorithm actually attempts to solve the following:

$$\min_C \| \boldsymbol{Y} - \boldsymbol{DC} \|_F^2 + \alpha \| \boldsymbol{C} \|_1 + \beta \| \boldsymbol{C} - \boldsymbol{CW} \|_1 \quad (4)$$

Taking the solution of the l_1-tracker from Eq.(2) as the initial coefficients c_0, Eq.(4) can be solved through iterative computations[11]. However, the high computational cost of the original l_1-tracker and iterative procedure for maintaining the neighborhood constraints of sparse coefficients make NLSSST inefficient in achieving real-timing tracking. In contrast to Fig. 1, the schematic diagram of NLSSST presented in Fig. 2, includes an additional neighborhood constraint between y_i and $N_K(y_i)$.

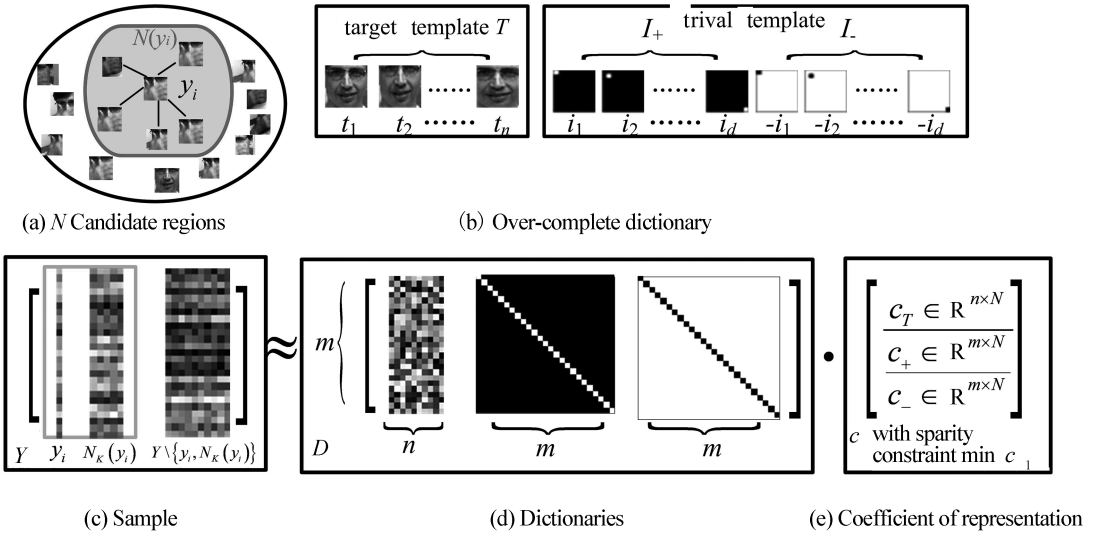


Fig. 2 Lu's NLSSST Algorithm

2 Euclidean local structure-based sparse coding for tracking (ELSST)

To circumvent the heavy computation burden of the l_1-tracker and NLSSST (Table 4), we propose an efficient tracker, called ELSST, that considers the local Euclidean structures of the candidates.

2.1 Original euclidean local structure constraint sparse coding (Original ELSSC)

It is evident from Eq. (4) that NLSSST attempts to solve a double l_1-norm problem. However, it is well known that the l_2-norm is much more commonly used for measuring the distance between two vectors and is much easier to optimize than the l_1-norm. Thus, we take the former to measure the relationships between the sparse coefficient vectors, which are close to each other, i.e., the Euclidean local-structure constraint, and the latter l_1-norm of $\boldsymbol{C}$ to maintain the sparsity of the optimization as follows:

$$\min_C \| \boldsymbol{Y} - \boldsymbol{DC} \|_F^2 + \alpha \| \boldsymbol{C} \|_1 + \beta \| \boldsymbol{C} - \boldsymbol{CW} \|_F^2 \quad (5)$$

Table 1　Optimization for ELS constraint based SC(ELSSC)

Input: Given N data points $\boldsymbol{Y}=[\ y_1\cdots y_N]\in\mathbf{R}^{m\times N}$, over-complete dictionary $\boldsymbol{D}\in\mathbf{R}^{m\times(n+2m)}$

Output: Sparse matrix $\boldsymbol{C}=[c_1\cdots c_N]\in\mathbf{R}^{(n+2m)\times N}$

Parameters: Maxiumn iteration number $J=10$, neighborhood size $K=5$, all-zero vector c_0, $\alpha=0.01$, $\beta=0.5$, $\gamma=0.001$

1: For each point y_i, compute the nearest K neighborhoods $N_K(y_i)$ and weights w_{ji}

2: Compute the SVD-decomposition of $D = \boldsymbol{U\Sigma V}^{\mathrm{T}}$, where $\boldsymbol{V}\in\mathbf{R}^{(n+2m)\times(n+2m)}$

3: compute $\| \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} \|$, and set $\| \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} \| +2\beta$ randomly

4: For $i=1:N$

5: For $t=1:J$

6: If $\| c_i^{(t)} - c_i^{(t-1)} \|_2 < \tau$, break inner iteration

7: Compute $\theta_i^{(t)} = \sum_j w_{ji}c_j^{(t-1)}$, $v_i^{(t)} = \frac{1}{\gamma}[\boldsymbol{D}^{\mathrm{T}}y_i + 2\beta\theta_i^{(t-1)} + (\gamma - 2\beta)c_i^{t-1} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}_i c^{t-1}]$, and $x_i^{(t)} = \boldsymbol{V}v_i^{(t)}\in\mathbf{R}^{(n+2m)\times 1}$

8: Represent $x_i^{(t)}$ with sparse coefficient vector $c_i^{(t)}$, i.e., optimize $\frac{\gamma}{2}\| x_i^{(t)} - Vc_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1$

9: End

10: End

Equation (5) is the objective function of our Euclidean local structure constraint-based SC and can be solved through iterative computation. In particular, at the t-th iteration, for a single candidate y_i in Y, Eq. (5) can be written as follows:

$$\min_{c_i^{(t)}} f(c_i^{(t)}) = \min_{c_i^{(t)}} \| \boldsymbol{y}_i - \boldsymbol{D}c_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1 + \beta \| c_i^{(t)} - \theta_i^{(t-1)} \|_2^2 \tag{6}$$

where $\theta_i^{(t)} = \sum_j w_{ji}c_j^{(t-1)}$. At the t-th iteration for the optimization of c_i, c_j, $i \neq j$ is fixed. Therefore, we can regard $\theta_i^{(t-1)}$ as a constant. To solve Eq. (6), we introduce the following surrogate function as presented in [11]:

$$\psi(c_i, c_0) = \frac{\lambda}{2}\| c_i^{(t)} - c_0 \|_2^2 - \frac{1}{2}\| \boldsymbol{D}c_i^{(t)} - \boldsymbol{D}c_0 \|_2^2 \tag{7}$$

where λ is convex. According to Daubechies[13], when $\lambda\boldsymbol{I} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}$ is a strictly positive definite matrix, $\psi(c_i, c_0)$ is strictly convex for any c_0 with respect to c_i. Hence, in our experiments, the constant λ is set accordingly ($\lambda = \gamma - 2\beta$; Table 1). Once the over-complete dictionary $\boldsymbol{D}$ is fixed, we can derive the following convex objective function from Eq. (7):

$$\begin{aligned} f(c_i^{(t)}) = {} & \frac{1}{2}\| y_i - \boldsymbol{D}c_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1 + \beta \| c_i^{(t)} - \theta_i^{(t-1)} \|_2^2 + \frac{\gamma - 2\beta}{2}\| c_i^{(t)} - c_0 \|_2^2 - \frac{1}{2}\| \boldsymbol{D}c_i^{(t)} - \boldsymbol{D}c_0 \|_2^2 = \\ & \frac{1}{2}\| y_i \|_2^2 - \langle y_i, \boldsymbol{D}c_i^{(t)} \rangle + \alpha \| c_i^{(t)} \|_1 - 2\beta\langle c_i^{(t)}, \theta_i^{(t-1)} \rangle + \frac{\gamma}{2}\| c_i^{(t)} \|_2^2 - (\gamma - 2\beta)\langle c_i^{(t)}, c_0 \rangle + \frac{\gamma - 2\beta}{2}\| c_0 \|_2^2 + \\ & \langle \boldsymbol{D}c_i^{(t)}, \boldsymbol{D}c_0 \rangle - \frac{1}{2}\| \boldsymbol{D}c_0 \|_2^2 + \beta \| \theta_i^{(t-1)} \|_2^2 = \\ & \frac{\gamma}{2}\| c_i^{(t)} \|_2^2 - \langle y_i, \boldsymbol{D}c_i^{(t)} \rangle - 2\beta\langle c_i^{(t)}, \theta_i^{(t-1)} \rangle - (\gamma - 2\beta)\langle c_i^{(t)}, c_0 \rangle + \langle \boldsymbol{D}c_i^{(t)}, \boldsymbol{D}c_0 \rangle + \alpha \| c_i^{(t)} \|_1 + \\ & \left(\frac{1}{2}\| y_i \|_2^2 + \frac{\gamma - 2\beta}{2}\| c_0 \|_2^2 + \beta \| \theta_i^{(t-1)} \|_2^2\right) = \\ & \frac{\gamma}{2}\| c_i^{(t)} - v_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1 + R \end{aligned} \tag{8}$$

where

$$v_i^{(t)} = \frac{1}{\gamma}[\boldsymbol{D}^{\mathrm{T}}y_i + 2\beta\theta_i^{(t-1)} + (\gamma - 2\beta)c_i^{t-1} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}c^{t-1}]$$

and

$$R = \frac{1}{2}\| y_i \|_2^2 + \frac{\gamma - 2\beta}{2}\| c_0 \| - \frac{1}{2}\| \boldsymbol{D}c_0 \|_2^2 + \beta \| \theta_i^{(t-1)} \|_2^2 - \frac{\gamma}{2}\| v_i^{(t)} \|_2^2$$

are fixed at the t-th iteration. Thus, we can simplify

Eq. (8) as follows:

$$f(c_i^{(t)}) = \frac{\gamma}{2} \| c_i^{(t)} - v_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1 \quad (9)$$

To solve Eq. (9) using SVD, we decompose the over-complete dictionary $\boldsymbol{D} \in \mathrm{R}^{m\times(n+2m)}$ as $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{U} \in \mathrm{R}^{m\times m}$, $\boldsymbol{\Sigma} \in \mathrm{R}^{m\times(n+2m)}$ and $\boldsymbol{V} \in \mathrm{R}^{(n+2m)\times(n+2m)}$. Since $\boldsymbol{V}$ is an orthogonal matrix, Eq. (9) can be re-written as

$$f(c_i^{(t)}) = \frac{\gamma}{2} \| x_i^{(t)} - \boldsymbol{V}c_i^{(t)} \|_2^2 + \alpha \| c_i^{(t)} \|_1 \quad (10)$$

where $x_i^{(t)} = \boldsymbol{V}v_i^{(t)}$. Consequently, we can transform the optimization problem with the Euclidean local structure constraint in Eq. (6) to a pure l_1-optimization problem in Eq. (10), i.e., to represent the given signal $x_i^{(t)}$ with sparse coefficients $c_i^{(t)}$ under the new dictionary $\boldsymbol{V} \in \mathrm{R}^{(n+2m)\times(n+2m)}$. The procedure of Euclidean local-structure constraint based sparse coding (ELSSC) is summarized in Table 1 and is very different from the optimization procedure followed for NLSSSC[11], even though the difference between their objective functions seems very small (Eqs. (4) and (5), respectively).

2.2 Improved euclidean local structure constraint sparse coding (Improved ELSSC)

If m in Eq. (10) is large, it is time-consuming to obtain the optimization result c_i, as that in l_1-optimization and NLSSSC. Fortunately, in terms of SVD and the structure of $\boldsymbol{D}$ (Figs. 1 and 2), we have

$$\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = [\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'^{\mathrm{T}}, \boldsymbol{I}, -\boldsymbol{I}] = [\boldsymbol{T}, \boldsymbol{I}, -\boldsymbol{I}] \quad (11)$$

where $\boldsymbol{I}$ denotes the m-ordered identity matrix. $\boldsymbol{\Sigma}'$ is the first n rows of $\boldsymbol{\Sigma}$, $\boldsymbol{V}'$ consists of the first n rows and the first n columns of $\boldsymbol{V}$, and $m \gg n$. As a result, when constructing the dictionary $\boldsymbol{V}$ in Eq. (10), only the first n rows and first n columns of $\boldsymbol{V}$ must be prepared, whereas the remaining parts of $\boldsymbol{V}$ are not considered to make any contribution to the target templates $\boldsymbol{T}$. Thus, the large scale optimization in Eq. (10) can be reduced to a much smaller one as follows:

$$f(c'^{(t)}_i) = \frac{\gamma}{2} \| x'^{(t)}_i - \boldsymbol{V}''c'^{(t)}_i \|_2^2 + \alpha \| c'^{(t)}_i \|_1, \quad (12)$$

where $\boldsymbol{V}'' = [\boldsymbol{V}'\ \boldsymbol{I}'\ -\boldsymbol{I}''] \in \mathrm{R}^{n\times 3n}$, $\boldsymbol{I}'$ denotes the n-ordered identity matrix, and x_i' is the first n rows of x_i in Eq. (10).

2.3 Original and improved ELSSC-tracker

Based on the above algorithm, our tracker can be obtained with the framework of the original l_1-tracker[8-9] (Table 2). We need to iteratively solve the large-scale l_1-optimization problem in Eq. (10) twice, up to three times for each candidate in the algorithm, and more than five times in NLSSST. The initial sparse coefficients c_0 are considered as all-zero vectors and iteratively solve the problem without any l_1-optimization issues, as in Table 1 in [11]. Nevertheless, we find that, in NLSSST, it is more effective and accurate to initialize c_0 as the solution of the l_1-optimization problem. Therefore, the computation complexity of our tracker is of the same order of magnitude as that of the l_1-tracker and NLSSST. When we resize all $n = 10$ targets and $N = 200$ candidate regions to 40×40, i.e., $m = 1\,600$ (Figs. 1 and 2), then the over-complete dictionary $\boldsymbol{D}$ is $1\,600 \times 3\,210$ and the orthogonal matrix $\boldsymbol{V}$ is $3\,210 \times 3\,210$ in Eq. (10). It is very difficult to solve the corresponding l_1-optimization problem with such a $\boldsymbol{D}$ (in l_1-tracker and NLSSST) or $\boldsymbol{V}$ (in our ELSST).

With the improved ELSSC, $\boldsymbol{\Sigma}'$ is the first ten rows of $\boldsymbol{\Sigma}$, and $\boldsymbol{V}'$ consists of the first ten rows and first ten columns of $\boldsymbol{V}$. Thus, each iteration of each candidate region in ELSST can be reduced from the large-scale l_1-optimization problem to a much smaller one because of the much smaller scale $\boldsymbol{V}' \in \mathbf{R}^{10\times 10}$. To overcome the problem of occlusions in tracking, the analogous trivial templates are used to construct the new dictionary $\boldsymbol{V}'' \in \mathbf{R}^{10\times 30}$, i.e., a ten-ordered identity matrix and ten-ordered negative identity matrix.

Table 2　Euclidean Local Structure based Tracking (ELSS-tracker)

Input: Given a video stream for tracking, location of the target l_1 in frame #1

Output: Tracking results of each frame

1: Set s = 1, select 10 template regions extremely near the target in #1, then resize and stretch them to be $\boldsymbol{T} \in \mathbf{R}^{1\,600\times 10}$

2: While not reach the end of the video sequence, s←s+1

3: Pick N = 200 candidate regions around the latest target location l_{s-1} in frame #s, and stretch to be $\boldsymbol{Y} \in \mathbf{R}^{1\,600\times 200}$

4: Construct $\boldsymbol{D}$ with $\boldsymbol{T}$, positive and negative identity matries, likewise in Fig.1 and Fig2

5: Solve ELSSC-optimization with $\boldsymbol{Y}$ and $\boldsymbol{D}$ in Tab.1, and denote the optimization result as $c_i^{(s)}$

6: Compute the reconstruct errors $e_i^{(s)} = \| x_i^{(s)} - Vc_i^{(s)} \|_2^2$ and the normalized weight $w_i^{(s)} = w_i^{(s)} / \sum_i w_i^{(s)}$, where $w_i^{(s)} = \exp(-e_i^{(s)}/\alpha)$

7: Locate the object for tracking with the weighted sum of all 200 candidate regions and $w_i^{(s)}$ in frame #s

8: Select 10 regions that extremely nearby the object as the new target templates $\boldsymbol{T}$

9: End

3　Experiments

3.1　Experimental setting

In order to evaluate the proposed tracker, experiments on 12 video sequences were conducted, including Surfer, Dudek, Faceocc2, Animal, Girl, Stone, Car, Cup, Face, Juice, Singer, Sunshade, Bike, Car Dark, and Jumping[17-19]. These sequences covered almost all challenges in tracking, including occlusion (even heavy occlusion), motion blur, rotation, scale variation, illumination variation, and complex background. For comparison, we used four state-of-the-art algorithms with the same initial positions and the same representations of the targets. They were the incremental learning-based tracker (IVT, a common discriminant tracker)[14], the covariance-based tracker (CovTrack, a generative tracker on Lie-group)[15], the l_1-tracker (a generative tracking method)[8-9], and the NLSSST[11]. All the experiments were run on a computer with a 2.67 GHz CPU and a 2 GB memory.

The main parameters used in our experiments are set as follows: the number of candidate regions $N = 200$, the number of template regions is $n = 10$, and the candidates and targets are resized to 40×40.

3.2　Experimental results for sparsity and stability

The stability and sparsity of the original sparse coding, the NLSSSC, and the original and improved ELSSC were verified. The experiments were designed with the Face sequence in the VOT 2013 benchmark dataset[18] as follows: six similar regions were represented ($CR_1, \cdots, CR_6$, their means and standard derivations illustrate the similarity) sparsely with template $\boldsymbol{T} = [T_1, \cdots, T_{10}] \in \mathrm{R}^{1\,600\times 10}$ from two regions apart from each other (the red region and the green one). Evidently, $\boldsymbol{T}$ is over-completed, and the entire dictionary $\boldsymbol{D} \in \mathrm{R}^{1\,600\times 3\,210}$ is constructed likewise in Figs. 1 and 2.

The sparse coefficients of $CR_1, \cdots, CR_6$ generated with the l_1-, the NLSSSC-, the original ELSSC-, and the improved ELSSC-optimization are plotted in Fig. 3. In particular, six similar regions have very different representation coefficients, when using the original l_1-optimization problem, which ignores the structure information between regions. The results of the other three algorithms are much more stable, because of preservation of the structural information. If two regions are similar to each other, they also have similar sparse coefficients. This improves the robustness of tracking; otherwise, the tracker may degenerate or even fail to track. CR_4 for example, with l_1-optimization, can be represented by T_2, T_8, T_6, T_7, and T_1, and the tracker may fail to track the top of the book. Meanwhile, experimental results show that, NLSSSC and our two ELSSC are sparser than the original l_1-optimization problem.

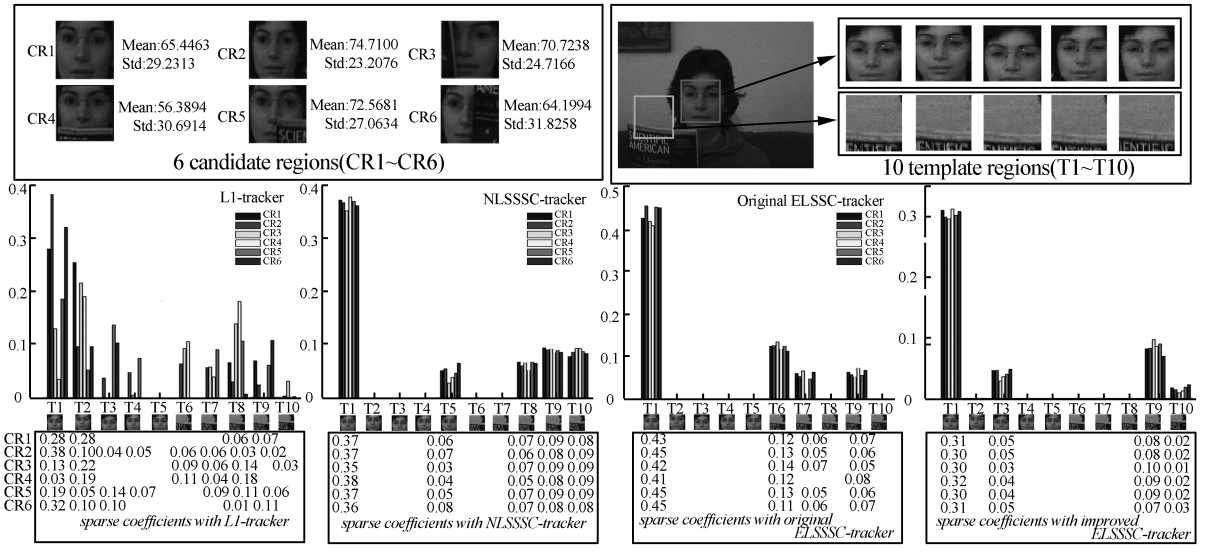


Fig. 3 Comparisions of sparsity and stability with the original l_1-, NLSSSC-, and our ELSSC-optimization. The sparse coefficients only are accurated to the second decimal place.

3.3 Experimental results for visual target tracking

We evaluate the investigated algorithms comparatively, using the center location errors, the average success rates, and the average frames per second. The results are shown in Figs. 4&5 and in Tables 3&4. The templates of NLSSST, the original ELSST, and the improved ELSST are shown in Fig. 4(g-o). Overall, our original and improved trackers outperform the other state-of-the-art algorithms.

For occlusion, five algorithms, except IVT, function satisfactorily, especially at #206, #366 of the Dudek sequence in Fig. 4 (b) (the head in tracking is covered by the hand and glasses), #143, #265, #496 of the Faceocc2 sequence in Fig. 4 (c) (the head in tracking is covered by the book), #85, #108, #433 of the Girl sequence in Fig .4 (e) (the head in tracking turns right, turns back, and blocks someone else), and #56, #104, #301 of the Face sequence in Fig. 4 (i) (the head in tracking is also covered by the book). After the target recovers from occlusion, these five trackers can seek it quickly. IVT works poorly, even loses the target in #10 of the Girl sequence (Fig. 5 (e)), because the number of positive and negative samples is limited (considering the learning efficiency), and the incremental updating of the classifier in IVT is less effective. CovTracking has a large size of candidates (based on the definition of integral image, the feature extraction of these candidates is so fast, that its cost can be ignored), which makes it robust for occlusion, scale variation, and blur. NLSSST and our original and improved trackers all work well, when the targets are occluded; our two trackers work even better.

For motion blur, our two trackers work better than IVT and the original l_1-tracker. Moreover, CovTracking also reveals its ability to handle blur (e.g., #4, #9, and #38 in Fig. 4(d,o). In the former sequence, the animal runs and jumps fast (motion blur) with a lot of water splashing (occlusion), while in the latter, the man ropes skipping and the camera cannot take the clear face of the man. IVT and l_1-tracker fail both from #4 in Fig. 4(d), and never recover after that. Our original and improved ELSS lost the target in #31 and #41, then recovered in #33 and #44 (Fig. 4(d)). In #12 to #21 and #44 to #71, the improved ELSST works better than original ELSST, CovTracking, l_1-tracker, and NLSSST.

For rotation and scale variation, our trackers also perform robustly (Figs. 4(a,c,e,g,j) and 5(a,c,e,g,j). When the surfer falls forward and backward, the girl turns left and right, moves towards and away from the camera, the man turns left and right, the car turns

over, and the juice bottle becomes bigger and smaller in Surfer, Girl, Faceocc2, Car, and Juice sequence, respectively, five trackers except IVT perform well, especially the NLSSS-tracker and our two ELSSC-trackers.

In a complex background and with high illumination variance (Fig. 4(f)), there are many similar stones to track. The l_1-tracker and our two trackers work better than other three trackers. Cov-tracker fails, because it extracts edge information of targets as one dimension of features, and in this sequences, edge of targets are ambiguous and hard to be distinct. Similar results are obtained from Fig. 4(h,l,m).

Table 3 summarizes the average success rates. Given the tracking results R_T and the ground-truth R_G, we use the detection criterion in the PASCAL VOC challenge[16], i.e.,

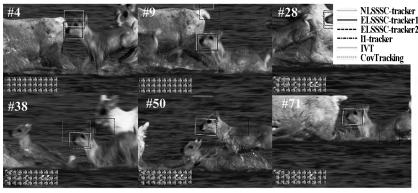
$$\text{score} = \frac{\text{area}(R_T \cap R_G)}{\text{area}(R_T \cup R_G)}$$

to evaluate the success rate. In general, from the above analysis, we find that our original and improved ELSSC-trackers perform almost the same, and the former is slightly better, especially in the Dudek, Faceocc2, Surfer, Stone, CarDark, and Jumping sequences (Fig. 5(a,b,c,f,n,o). However, we also find from Table 4, which summarizes the average frames per second, that the improved ELSST works much faster than the original ELSST and almost all the other trackers; IVT is faster than the improved ELSST when dealing with Surfer and Dudek sequences, but its success rate is much worse than that of the improved ELSST. It is sensitive under the phenomena of occlusion, rotation, and target motion blur. The original l_1-tracker performs well in most frames, but it is also time-consuming and fails to track sometimes; Cov-Tracking is suitable for occlusion and rotation, but fails when facing a complex background.

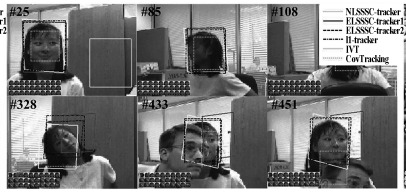
(a) Sequence surfer (375 frames)

(b) Sequence dudek (1145 frames)

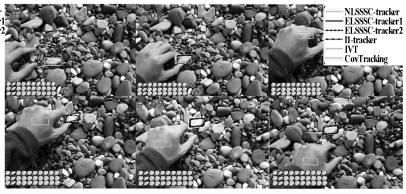
(c) Sequence faceocc2 (819 frames)



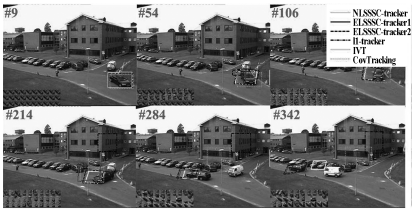
(d) Sequence animal (71 frames)



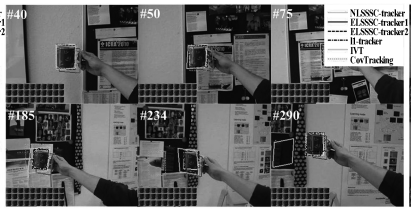
(e) Sequence girl (500 frames)



(f) Sequence S tone (593 frames)

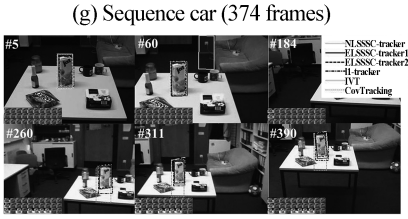


(g) Sequence car (374 frames)

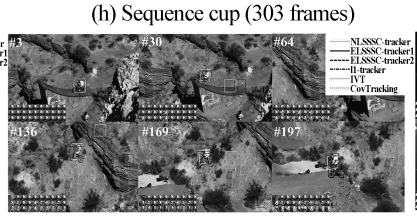


(h) Sequence cup (303 frames)

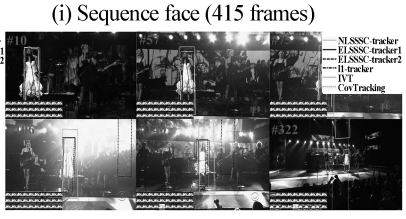
(i) Sequence face (415 frames)



(j) Sequence juice (404 frames)



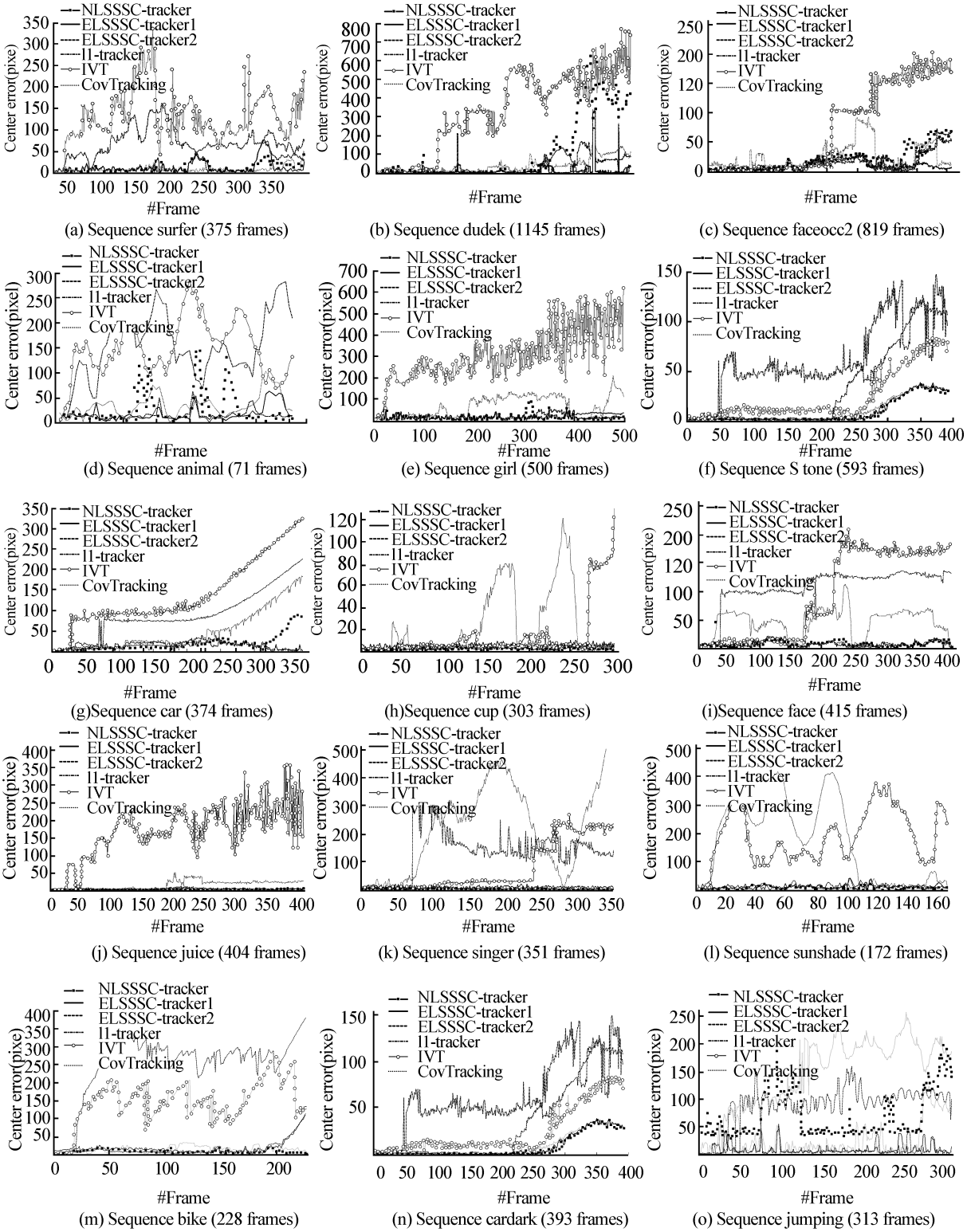
(k) Sequence singer (351 frames)



(l) Sequence sunshade (172 frames)

(m) Sequence bike (228 frames) (n) Sequence cardark (393 frames) (o) Sequence jumping (313 frames)

Fig. 4 Some tracking results



(a) Sequence surfer (375 frames) (b) Sequence dudek (1145 frames) (c) Sequence faceocc2 (819 frames)

(d) Sequence animal (71 frames) (e) Sequence girl (500 frames) (f) Sequence S tone (593 frames)

(g)Sequence car (374 frames) (h)Sequence cup (303 frames) (i)Sequence face (415 frames)

(j) Sequence juice (404 frames) (k) Sequence singer (351 frames) (l) Sequence sunshade (172 frames)

(m) Sequence bike (228 frames) (n) Sequence cardark (393 frames) (o) Sequence jumping (313 frames)

Fig. 5 Quantitative evaluation in terms of center location error (in pixel)

Table 3　Average Success Rates

Video	IVT	CovTrack	l_1-tracker	NLSSST	ELSST1	ELSST2
Sufer	0.051 5	**0.477 0**	0.038 8	0.464 6	**0.466 7**	0.405 2
Dudek	0.201 1	0.421 6	0.621 5	0.652 8	**0.672 6**	**0.660 4**
Faceocc2	0.455 3	0.391 8	**0.608 4**	0.457 9	**0.574 7**	0.464 1
Animal	0.021 8	0.270 1	0.033 6	0.369 2	**0.407 8**	**0.411 7**
Girl	0.022 8	0.217 1	**0.486 9**	**0.485 3**	0.400 6	0.469 3
Stone	0.097 4	0.111 4	0.583 4	0.410 9	**0.661 1**	**0.657 2**
Car	0.060 7	0.185 8	0.095 6	**0.341 8**	0.327 8	**0.382 5**
Cup	**0.630 0**	0.376 9	0.559 8	**0.573 8**	0.523 8	0.563 7
Face	0.334 1	0.280 6	0.047 9	0.524 8	**0.549 6**	**0.582 7**
Juice	0.074 3	0.421 8	0.511 1	**0.529 9**	0.518 6	**0.583 5**
Singer	0.332 6	0.136 1	0.118 4	**0.579 0**	0.478 1	**0.565 1**
Sunshade	0.048 1	0.180 3	**0.525 7**	**0.534 8**	0.474 3	0.494 8
Bike	0.057 6	0.372 1	0.045 1	**0.443 8**	0.360 8	**0.391 7**
CarDark	0.083 1	0.308 7	0.079 0	0.011 0	**0.420 8**	**0.373 7**
Jumping	0.057 7	0.275 5	0.071 1	0.084 7	**0.453 0**	**0.450 5**

The best two results are shown in bold. Our original and improved algorithms are shown in the last two columns, respectively.

Table 4　Average Frames per Second

Video	IVT	CovTrack	l_1-tracker	NLSSST	ELSST1	ELSST2
Sufer	**2.864 9**	1.570 7	0.035 8	0.014 1	0.015 6	**2.346 9**
Dudek	**3.321 1**	1.245 4	0.038 8	0.017 1	0.017 9	**3.245 4**
Faceocc2	**2.788 6**	1.127 8	0.018 0	0.010 7	0.014 2	**3.127 8**
Animal	**1.897 9**	1.253 4	0.031 2	0.015 0	0.007 1	**3.253 4**
Girl	**1.654 8**	1.220 9	0.037 6	0.016 7	0.009 8	**3.220 9**
Stone	1.290 3	**1.889 0**	0.027 1	0.014 4	0.014 6	**4.113 8**
Car	**3.684 1**	2.850 2	0.062 1	0.052 5	0.036 5	**6.225 3**
Cup	**7.817 5**	3.547 9	0.079 8	0.067 7	0.053 8	**6.394 9**
Face	**6.742 2**	2.896 1	0.054 3	0.041 7	0.054 6	**6.168 1**
Juice	**7.048 9**	3.929 7	0.063 5	0.066 5	0.058 6	**5.473 8**
Singer	**6.095 9**	2.802 6	0.019 5	0.068 3	0.048 1	**6.189 1**
Sunshade	**7.302 7**	2.790 5	0.071 3	0.058 7	0.078 1	**6.060 1**
Bike	**6.974 7**	2.719 2	0.016 3	0.032 0	0.021 0	**5.840 0**
CarDark	**3.704 1**	1.395 1	0.022 6	0.055 2	0.029 5	**2.422 1**
Jumping	**7.329 6**	2.608 0	0.052 0	0.047 6	0.057 9	**3.551 9**

The best two results are shown in bold. Our original and improved algorithms are shown in the last two columns, respectively.

4 Conclusions

In this study, to deal with sparsity and instability in the l_1-optimization problem[10-12] and the high time complexity of the NLSSSC-tracker [11], we propose a novel efficient tracker, i.e., the Euclidean local-structure constraint based sparse coding (ELSSC). Our new algorithm is a l_1-tracker with a reconstructed over-complete dictionary, which is different from that in the original l_1-tracker and NLSSSC-tracker. Moreover, we simplify the large-scale l_1-optimization problem in our tracker to a much smaller one in our improved ELSSC-tracker.

Compared with the original l_1-tracker, our ELSSC-tracker introduces the structure information among the candidate regions generated by the Bayesian inference to the l_1-tracker, similar to that in the NLSSSC-tracker. With our derivation, the optimization procedure of our tracker (Eq.(10)) can be solved as that in the l_1-optimization but very differently from that in the NLSSSC. Furthermore, our improved tracker is much more efficient than the l_1-tracker and NLSSSC-tracker. Our experiments demonstrate the sparsity, stability, and efficiency of our tracker.

References

[1] ZHANG Shengping, YAO Hongxun, SUN Xin, et al. Sparse coding based visual tracking: review and experimental comparison[J]. Pattern recognition, 2013, 46(7): 1772-1788.

[2] YILMAZ A, JAVED O, SHAH M. Object tracking: a survey[J]. ACM computing surveys (CSUR), 2006, 38(4): 1-45.

[3] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking-learning-detection[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(7): 1409-1422.

[4] AVIDAN S. Ensemble tracking[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(2): 261-271.

[5] BABENKO B, YANG M H, BELONGIE S. Visual tracking with online multiple instance learning[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Miami, USA, 2009: 983-990.

[6] CANDÈS E J, WAKIN M B. An introduction to compressive sampling[J]. IEEE, signal processing magazine, 2008, 25(2): 21-30.

[7] CANDÈS E J, ROMBERG J, TAO J. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information[J]. IEEE transactions on information theory, 2006, 52(2): 489-509.

[8] MEI Xue, LING Haibin, WU Yi, et al. Minimum error bounded efficient *l*1 tracker with occlusion detection[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Colorado, USA, 2011:1257-1264.

[9] MEI Xue, LING Haibin. Robust visual tracking and vehicle classification via sparse representation[J]. IEEE transactions on pattern analysis and machine intelligence, 2011, 33(11): 2259-2272.

[10] XU Huan, CARAMANIS C, MANNOR S. Sparse algorithms are not stable: a no-free-lunch theorem[J]. IEEE transactions on pattern analysis and machine intelligence, 2011, 34(1): 187-193.

[11] LU Xiaoqiang, YUAN Yuan, LU Pingkun, et al. Robust visual tracking with discriminative sparse learning[J]. Pattern recognition, 2013, 46(7): 1762-1771.

[12] YANG Jian, ZHANG Lei, XU Yong, et al. Beyond sparsity: the role of L_1-optimizer in pattern classification[J]. Pattern recognition, 2012, 45(3): 1104-1118.

[13] DAUBECHIES I, DEFRISE M, DE MOL C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint[J]. Communications on pure and applied mathematics, 2004, 57(11): 1413-1457.

[14] ROSS D A, LIM J, LIN R S, et al. Incremental learning for robust visual tracking[J]. International journal of computer vision, 2008, 77(1-3): 125-141.

[15] PORIKLI F, TUZEL O, MEER P. Covariance tracking using model update based on lie algebra[C]//Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York, USA, 2006: 728-735.

[16] EVERINGHAM M, VAN GOOL L, WILLIAMS C K I, et al. The pascal visual object classes (VOC) challenge[J]. International journal of computer vision, 2010, 88(2): 303-338.

[17] WU Yi, LIM J, YANG M H. Online object tracking: A benchmark[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, USA, 2013: 2411-2418.

[18] KRISTAN M, PflUGFELDER R, LEONARDIS A, et al.

The visual object tracking VOT2013 challenge results [C]//Proceedings of IEEE International Conference on Computer Vision Workshops (ICCVW). Sydney, Australia, 2013: 98-111.

[19] SONG Shuran, XIAO Jianxiong. Tracking revisited using RGBD camera: unified benchmark and baselines [C]// Proceedings of IEEE International Conference on Computer Vision (ICCV). Sydney, Australia, 2013: 233-240.

Author introduction

Hongyuan WANG, male, was born in 1960, Professor of Changzhou University. His research interest is image processing and recognition, artificial intelligence. He has published over 20 papers in international journals and conferences.

Ji ZHANG, male, was born in 1981, Lecturer of Changzhou University. His research interest is image processing and recognition. He has published five papers in international journals and conferences.

Fuhua CHEN, male, was born in 1966, Assistant Professor of West Liberty University. His research interest is variation image segmentation and inverse problems. His current research also involves object tracking and person re-identification. He has published over ten papers in international journals cited by SCI or EI.

2016 年 IEEE 云计算与智能系统国际会议
The 4th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS 2016)

August 17-19, 2016, Beijing, China

Research and application on cloud computing and intelligent systems have been extensively developed in recent years. We have witnessed numerous successes in various critical sectors of our society. As always, CCIS provides a forum for researchers and practitioners to exchange ideas and present their latest outputs, discuss challenging issues, and share experiences mainly in the field of cloud computing and intelligence systems. The past CCIS conferences were held in Beijing (2011), Hangzhou (2012), and Shenzhen & Hong Kong (2014), respectively. CCIS 2016, the 4th event of this exciting conference series, will be held at Beijing, August 17-19, 2016. The theme of CCIS 2016 is embracing challenges of cloud computing and intelligent technology at the age of big data. The conference will feature a comprehensive technical program, including a number of world class keynote speeches, frontline panel discussions, and session presentations. It is a great opportunity for learning, education, and information exchanging for all participants.

The conference proceedings will be published by IEEE Press (EI indexed). Selected high quality papers from CCIS 2016 will be invited to a number of prestigious Special Issues after a solid extension. The tentative Special Issues are with IEEE Transactions on Systems, Man, and Cybernetics: Systems, Journal of Network and Computer Applications, Nurocomputing, and China Communications.

Important Dates:

Paper submission deadline: April 15, 2016

Acceptance notification: May 25, 2016

Camera-ready submission: July 1, 2016

Conference date: August 17-19, 2016

Contacts:

Information and queries should be sent to Mr. Lu, Beijing University of Posts and Telecommunications, No 10, Xitucheng Road, Haidian District, Beijing 100876, PR China. Tel: +86-10-62281360, Email: ccis2016@163.com

Website: http://ccis2016.caai.cn/