Visual Tracking via Nonnegative Regularization Multiple Locality Coding

Fanghui Liu, Tao Zhou and Jie Yang
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
{lfhsgre,zhou.tao,jieyang}@sjtu.edu.cn

Irene Y.H. Gu
Dept. of Signals and Systems
Chalmers University of Technology
irenegu@chalmers.se

Abstract

This paper presents a novel object tracking method based on approximated Locality-constrained Linear Coding (LLC). Rather than using a non-negativity constraint on encoding coefficients to guarantee these elements nonnegative, in this paper, the non-negativity constraint is substituted for a conventional ℓ_2 norm regularization term in approximated LLC to obtain the similar nonnegative effect. And we provide a detailed and adequate explanation in theoretical analysis to clarify the rationality of this replacement. Instead of specifying fixed K nearest neighbors to construct the local dictionary, a series of different dictionaries with pre-defined numbers of nearest neighbors are selected. Weights of these various dictionaries are also learned from approximated LLC in the similar framework. In order to alleviate tracking drifts, we propose a simple and efficient occlusion detection method. The occlusion detection criterion mainly depends on whether negative templates are selected to represent the severe occluded target. Both qualitative and quantitative evaluations on several challenging sequences show that the proposed tracking algorithm achieves favorable performance compared with other state-of-the-art methods.

1. Introduction

Visual tracking is an indispensable part of computer vision with wide ranging applications, such as video surveillance, vehicle navigation and medical imaging [5, 21]. While much effort [1, 4, 28, 18] has been made on object tracking in the last few years, it seems difficult to find lasting solutions to enduring problems due to intrinsic factors (e.g. shape deformation and pose variation) and extrinsic factors (e.g. occlusions and varying illumination).

Recently, sparse representation [20, 23] has been successfully applied in visual tracking, early stemming from the ℓ_1 tracker proposed by Mei *et al.* [12] and their improved version [13]. The ℓ_1 method represents a target by a sparse linear combination of the target templates and trivial templates, and then solves it using a ℓ_1-regularized least squares method. The accelerated proximal gradient approach [3] is used to solve ℓ_1 norm minimization efficiently. However, because of adopting the global sparse appearance model, these methods are less effective in handling heavy occlusions. Different from them, a local sparse appearance model [7, 19] is introduced to enhance target representation and tracking robustness. Local patches inside a possible target candidate are sparsely represented with local patches in the dictionary templates. Joint sparse appearance model [6, 24, 26] exploits the intrinsic relationship among particles to represent the target jointly.

Locality-constrained Linear Coding (LLC) [17] is proposed to represent local appearance in tracking framework [10] because of its excellent performance (eg. similarity in feature space and close-form solution). It utilizes the sparse histograms of sparse coefficients and local optimal search scheme for object tracking. However, this method just uses a static local dictionary and does not provide additional constraint on encoding coefficients, leading to tracking drift easily. The spatial layout information is embedded in coding stage on appearance model [11].

Motivated by the previous work, we aim to develop a more robust approximated LLC tracker, especially when the non-negativity constraint on encoding coefficients is taken into consideration. The main contributions of this paper are as follows. (1) A ℓ_2 norm regularization term is introduced into approximated LLC, instead of the non-negativity constraint, to guarantee encoding coefficients nonnegative by choosing a regularization parameter. (2) Rather than using a static local sparse dictionary, a series of local dictionaries with pre-defined different numbers of nearest neighbors are provided. Weights of a linear combination of these dictionaries are learned from approximated LLC, and that is similar with solving encoding coefficients in the same framework. (3) To mitigate drifting problem, the occlusion detection criterion depends on whether negative templates are used to represent the target. Once negative templates are selected to reconstruct the target, we conclude that the target suffers from severe occlusion. Experiments on some public sequences compared with several prevalent tracking methods show the effectiveness and robustness of the proposed tracking algorithm.

The remainder of the paper is organized as follows. Section 2 gives the relevant related work of the proposed method. Section 3 gives the details of the proposed method. Section 4 shows the experimental results from the proposed method, with comparisons to eight existing state-of-the-art methods. Finally, conclusion is given in Section 5.

2. Preliminaries

2.1. Particle Filter in Tracking Framework

The implicit rationale behind particle filter [8, 27] is to estimate the posterior distribution $p(\mathbf{x}_{\mathrm{t}}|\mathbf{z}_{1:t})$ approximately by a finite set of random sampling particles. Given some observed image patches at t-th frame $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_t\}$, the state of the target $\mathbf{x}_t$ can be estimated recursively.

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \mathrm{d}\mathbf{x}_{t-1} \quad (1)$$

Let $\mathbf{x}_t = [l_x, l_y, \theta, s, \alpha, \phi]^T$, where $l_x, l_y, \theta, s, \alpha, \phi$ denote translations in the direction of x,y, rotation angle, scale, aspect ratio, and skew respectively. $p(\mathbf{x}_{\mathrm{t}}|\mathbf{x}_{t-1}) \sim \mathcal{N}(\mu, \sigma^2)$, referred to as the motion model, denotes state transition between two consecutive frames. The observation model $p(\mathbf{z}_t|\mathbf{x}_t)$ reflects the similarity between a target candidate and the target templates. Thus, the optimal state at t-th frame is obtained by maximizing the observation model:

$$\mathbf{x}_t^* = \underset{x}{\operatorname{argmax}}\, p(\mathbf{z}_t|\mathbf{x}_t) \quad (2)$$

In our method, the observation model is formulated from the reconstruction error by the multiple local dictionaries using approximated LLC.

2.2. Locality-constrained Linear Coding

LLC applies locality constraint to select similar basis of local image descriptors from a codebook, and learns a linear combination weight of these basis $\mathbf{B}$ to reconstruct each descriptor $\mathbf{x}_i$.

$$\begin{aligned} &\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}_i\mathbf{c}_i\|^2 + \lambda\|\mathbf{d}_i \odot \mathbf{c}_i\|^2 \\ &s.t.\ \mathbf{1}^{\mathrm{T}} c_i = 1, \forall i \end{aligned} \quad (3)$$

where $\odot$ denotes the element-wise multiplication, and $\mathbf{d}_i = \exp(\frac{\mathrm{dist}(\mathbf{x}_i, \mathbf{B})}{\sigma})$. $\mathrm{dist}(\mathbf{x}_i, \mathbf{B})$ represents the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{B}$. σ is a scale factor parameter that controls the weight decay speed.

Approximated LLC method [17], as the simplified edition of LLC, takes local sparsity into consideration. Instead of using the metric measure $\mathrm{dist}(\mathbf{x}_i, \mathbf{B})$, we can simply select K nearest neighbors of $\mathbf{x}_i$ as the local dictionary $\mathbf{B}_i$. In visual tracking process [10, 11], the candidate $\mathbf{y}_i$ is represented by the linear combination of several local basis vectors as the dictionary $\mathbf{B}_i$ with encoding coefficients $\mathbf{c}_i$ sparsely.

$$\min_{\mathbf{c}_i} \|\mathbf{y}_i - \mathbf{B}_i\mathbf{c}_i\|^2 \ \ s.t.\ \ \mathbf{1}^{\mathrm{T}}\mathbf{c}_i = 1, \forall i \quad (4)$$

where $\mathbf{1}$ denotes a vector with all ones and the constraint $\mathbf{1}^{\mathrm{T}}\mathbf{c}_i = 1$ ensures shift-invariant. The closed-form solution of (4) is $\mathbf{c}_i = [(\mathbf{B}_i^{\mathrm{T}} - \mathbf{1}\mathbf{y}_i^{\mathrm{T}})(\mathbf{B}_i - \mathbf{y}_i\mathbf{1}^{\mathrm{T}})]^{-1}\mathbf{1}$.

3. Proposed visual tracking algorithm

Some observed vectorization image patches($\in R^M$) at t-th frame $\mathbf{Y}_{1:N} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\} \in R^{M \times N}$ are sampled based on particle filter framework. In order to better capitalize on the distinction between the foreground and the background to locate the target, plenty of negative templates are collected. The positive and negative template sets are defined as $\mathbf{T}^{pos} = [\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_p]$ and $\mathbf{T}^{neg} = [\mathbf{T}_{p+1}, \mathbf{T}_{p+2}, ..., \mathbf{T}_{p+n}]$, where p and n denote the number of positive and negative template sets respectively. Generally, the tracking result in the first frame is manually chosen as a rectangle box. Define that $\mathcal{I}(x, y)$ is the center of the rectangle box, and the initial positive templates are sampled from an inner circular area that satisfies $\|\mathcal{I}_i - \mathcal{I}(x, y)\| < r$, where $\mathcal{I}_i$ is the center of the i-th sampled patch. Similarity, negative templates are sampled from the annular region $r < \|\mathcal{I}_j - \mathcal{I}(x, y)\| < s$, where $\mathcal{I}_j$ is the center of the j-th sampled image, r and s are the inner and outer radius of the annular region respectively. At the beginning of our tracking process, the number of positive templates and negative templates are set to $p = 50$, $n = 150$ respectively.

3.1. Modification on approximated LLC

3.1.1 Non-negativity constraint on approximated LLC

It might be tempting to agree that approximated LLC guarantees local sparsity and good reconstruction in visual tracking algorithm. However, (4) overlooks some potential information on coefficients. The experiment in Fig.1 shows the two tracking results reconstructed by different positive templates with encoding coefficients values in approximated LLC, where Template Index from #1 to #60 represents the positive template set, and the negative template set is from #61 to #210. As shown in Fig.1, the green tracking box denotes a bad tracking result without non-negativity constraint. It is represented by by a linear combination of eight templates (#6, #7, #23, #32, #45, #47, #52 and #53) with their corresponding coefficient vectors in green curves. The red one, as a good tracking result, is indicated by eight positive templates from #53 to #60 in red. Although the good or bad tracking results are both dictated by the positive template set, the bad result suffers severe drifts and contains much background information. In this case, the only

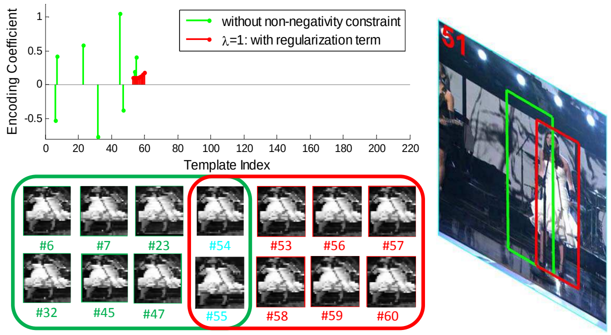

Figure 1. Illustration of tracking result in the Sequence *Singer1*. The green curves means without non-negativity constraint on encoding coefficients, leading to a bad tracking result represented by eight templates (#6, #7, #23, #32, #45, #47, #52 and #53) in the left-hand. And an added ℓ_2 norm regularization term leads to the good result (in red) by eight templates (#53, #54, #55, #56, #57, #58, #59 and #60) in the right-hand.

difference between these two tracking results is that some selected positive templates (#6, #32 and #47) are with negative coefficients. In other words, positive templates with negative coefficients lead to an unreliable tracking result. It seems to be a little specious and negative coefficients lose the significance of data representation on the target.

Just as the nonnegative matrix factorization(NMF) [9, 22], the target appearance is modeled as nonnegative linear combinations of a set of nonnegative bases that implicitly captures structure information of the target. Thus, for every candidate $\mathbf{y}_i$, its K nearest neighbors from the template set $\mathbf{T} = [\mathbf{T}^{pos}, \mathbf{T}^{neg}]$ is obtained to construct the local dictionary $\mathbf{B}_i$. The corresponding encoding vector is obtained by:

$$\min_{\mathbf{c}_i} \|\mathbf{y}_i - \mathbf{B}_i\mathbf{c}_i\|^2 \;\; s.t. \;\; \mathbf{c}_i \geq 0, \quad \mathbf{1}^{\mathrm{T}}\mathbf{c}_i = 1, \forall i \tag{5}$$

3.1.2 Various KNN for Multiple Local Dictionaries

In most case, the number of nearest neighbors K is specified manually with a fixed value. As such, this parameter is not automatically adjusted and only one local dictionary $\mathbf{B}_i$ corresponding to $\mathbf{y}_i$ is selected. To sufficiently capitalize on the intrinsic information of each $\mathbf{y}_i$ and templates set $\mathbf{T}$, we can produce various local dictionaries with different numbers of nearest neighbors. Their corresponding coefficient vectors $\mathbf{c}_i^j$ are obtained by:

$$\begin{cases} \min\limits_{\mathbf{c}_i^1} \|\mathbf{y}_i - \mathbf{B}_i^1\mathbf{c}_i^1\|^2 & s.t. \;\; \mathbf{c}_i^1 \geq 0, \quad \mathbf{1}^{\mathrm{T}}\mathbf{c}_i^1 = 1, \forall i; \\ \min\limits_{\mathbf{c}_i^2} \|\mathbf{y}_i - \mathbf{B}_i^2\mathbf{c}_i^2\|^2 & s.t. \;\; \mathbf{c}_i^2 \geq 0, \quad \mathbf{1}^{\mathrm{T}}\mathbf{c}_i^2 = 1, \forall i; \\ \ldots\ldots & \\ \min\limits_{\mathbf{c}_i^j} \|\mathbf{y}_i - \mathbf{B}_i^j\mathbf{c}_i^j\|^2 & s.t. \;\; \mathbf{c}_i^j \geq 0, \quad \mathbf{1}^{\mathrm{T}}\mathbf{c}_i^j = 1, \forall i; \\ \ldots\ldots & \\ \min\limits_{\mathbf{c}_i^m} \|\mathbf{y}_i - \mathbf{B}_i^m\mathbf{c}_i^m\|^2 & s.t. \;\; \mathbf{c}_i^m \geq 0, \quad \mathbf{1}^{\mathrm{T}}\mathbf{c}_i^m = 1, \forall i. \end{cases} \tag{6}$$

where m is the number of various dictionaries $(\mathbf{B}_i^1, \mathbf{B}_i^2, ..., \mathbf{B}_i^m)$. And $\mathbf{B}_i^j \in \mathbb{R}^{M \times k_j}$ is constructed by the candidate $\mathbf{y}_i$'s k_j nearest neighbors from the templates set $\mathbf{T}$. After obtaining their corresponding encoding vectors $\mathbf{c}_i^j$, how to combine these $\mathbf{c}_i^j$ to the final encoding vector $\mathbf{c}_i$ is an overriding concern in our tracking framework. Let $\mathbf{w} = [w_1, w_2, ..., w_m]^T$ be a weight vector, it implies $w_1 + w_2 + ... + w_m = 1$ and $w_j \geq 0$. The weight vector $\mathbf{w}$ is obtained by the following objective function:

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{y}_i - \sum_{j=1}^{m} w_j(\mathbf{B}_i^j\mathbf{c}_i^j)\|^2 \\ &s.t. \;\; \mathbf{1}^{\mathrm{T}}\mathbf{w} = 1, \quad w_j \geq 0 \quad \forall j \end{aligned} \tag{7}$$

Let $\mathbf{D} = [\mathbf{B}_i^1\mathbf{c}_i^1, \mathbf{B}_i^2\mathbf{c}_i^2, ..., \mathbf{B}_i^m\mathbf{c}_i^m]$, (7) is transformed into:

$$\min_{\mathbf{w}} \|\mathbf{y}_i - \mathbf{D}\mathbf{w}\|^2 \;\; s.t. \;\; \mathbf{1}^{\mathrm{T}}\mathbf{w} = 1, \; w_j \geq 0 \quad \forall j \tag{8}$$

This objective function is also an approximate LLC problem with non-negativity constraint, similar to (5) and (6).

3.2. Solving for these Objective Functions

3.2.1 Substitute Non-negativity constraint to ℓ_2 norm

In the above analysis, $\mathbf{c}_i$ and $\mathbf{w}$ are both obtained by solving the approximated LLC problem with non-negativity constraint. There are many optimal iteration methods for the constraint linear quadratic programming problem, such as interior point method, the accelerated proximal gradient method (APG) [3], ADMM [15]. However, the non-negativity constraint destroys the structure of analytic solution in approximated LLC due to its non-differentiable character.

To tackle this problem , we introduce a ℓ_2 norm regularization term to replace the non-negativity constraint. Therefore, (5) and (8) are rewritten as:

$$\min_{\mathbf{c}_i} \|\mathbf{y}_i - \mathbf{B}_i\mathbf{c}_i\|^2 + \lambda\|\mathbf{c}_i\|^2 \;\; s.t. \; \mathbf{1}^{\mathrm{T}}\mathbf{c}_i = 1, \forall i \tag{9}$$

$$\min_{\mathbf{w}} \|\mathbf{y}_i - \mathbf{D}\mathbf{w}\|^2 + \beta\|\mathbf{w}\|^2 \;\; s.t. \; \mathbf{1}^{\mathrm{T}}\mathbf{w} = 1, \forall j \tag{10}$$

The solution of the objective function in (9) is $\mathbf{c}_i = [(\mathbf{B}_i^{\mathrm{T}} - \mathbf{1}\mathbf{y}_i^{\mathrm{T}})(\mathbf{B}_i - \mathbf{y}_i\mathbf{1}^{\mathrm{T}}) + \lambda\mathbf{I}]^{-1}\mathbf{1}$. And the solution of $\mathbf{w}$ is in the similar fashion. In this case, the structure of closed-form in coefficient vector $\mathbf{c}_i$ and weight vector $\mathbf{w}$ is preserved. We will illustrate these elements in $\mathbf{c}_i$ and $\mathbf{w}$ still remain nonnegative in the following sections. The rationale for this replacement is described in more details.

3.2.2 Theoretical Analysis of this Replacement

For the sake of mathematical convenience and easy to use in the subsequent description, we construct a local dictionary $\mathbf{B}_i$ selected from $\mathbf{y}_i$'s K nearest neighbors (temporary not consider multiple dictionaries $\mathbf{B}_i^j$ with different nearest neighbors).

Define $\mathbf{A} = (\mathbf{B}_i - \mathbf{y}_i\mathbf{1}^{\mathrm{T}})$, $\mathbf{F} = (\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{I})$, the solution of (9) is rewritten as $\mathbf{c}_i = \mathbf{F}^{-1}\mathbf{1}$, where $\mathbf{F} \in \mathbb{R}^{K\times K}$ (and $\mathbf{F}^{-1}$) is a positive definite matrix. We denote it as $\mathbf{F} \succ 0$.

Theorem 1. *if $\boldsymbol{F}^{-1} \succ 0$ and $\boldsymbol{F}^{-1}$ is a strictly diagonally dominant matrix* [1]*, $\boldsymbol{c}_i = \boldsymbol{F}^{-1}\boldsymbol{1}$ is nonnegative.*

Proof. Because $\mathbf{F}^{-1}$ is a positive definite matrix, elements in the dominant diagonal of $\mathbf{F}^{-1}$ are all with positive values ($(\mathbf{F}^{-1})_{ii} > 0, \;\; i = 1, 2, ..., K$).

On the other hand, considering that $\mathbf{F}^{-1}$ is a strictly diagonally dominant matrix, for each row of $\mathbf{F}^{-1}$, satisfies:

$$|(\mathbf{F}^{-1})_{jj}| > \sum_{i=1, i\neq j}^{K} |(\mathbf{F}^{-1})_{ji}| \quad \forall j \tag{11}$$

Noticing the above equation and we have

$$\begin{aligned}\mathbf{F}^{-1}\mathbf{1} &= \sum_{i=1}^{K}(\mathbf{F}^{-1})_{ji} = \sum_{i=1,i\neq j}^{K}(\mathbf{F}^{-1})_{ji} + (\mathbf{F}^{-1})_{jj}\\ &> \sum_{i=1,i\neq j}^{K}(\mathbf{F}^{-1})_{ji} + \sum_{i=1,i\neq j}^{K}|(\mathbf{F}^{-1})_{ji}|\\ &\geq 0\end{aligned} \tag{12}$$

Thus $\mathbf{c}_i = \mathbf{F}^{-1}\mathbf{1} > 0$.

In the following, *the proposition that $\mathbf{c}_i$ is nonnegative is converted to how to guarantee $\boldsymbol{F}^{-1}$ as a strictly diagonally dominant matrix.* An intuitive idea is to make $\mathbf{F}$ a strictly diagonally dominant matrix by choosing λ value ($\mathbf{F} = \mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{I}$). And then seek for the relationship between $\mathbf{F}$ and $\mathbf{F}^{-1}$ in terms of strictly diagonally dominant character. It is easy to choose λ, and the lower bound of λ is given:

$$\lambda \geq \max\{\sum_{j\neq 1}^{K}|(\mathbf{A}^T\mathbf{A})_{1j}|, \sum_{j\neq 2}^{K}|(\mathbf{A}^T\mathbf{A})_{2j}|, ..., \sum_{j\neq K}^{K}|(\mathbf{A}^T\mathbf{A})_{Kj}|\} + \epsilon \tag{13}$$

where ϵ is an arbitrarily small positive constant. Because $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is a positive semi-definite matrix, then $\mathbf{F} \in R^{K\times K}$ is not only a positive definite matrix but also strictly diagonally dominant matrix by choosing λ to satisfy (13).

In this condition, *the proposition that $\mathbf{c}_i$ is nonnegative is transferred to prove a conclusion that $\boldsymbol{F}^{-1}$ is a strictly diagonally dominant matrix if and only if $\boldsymbol{F}$ is a strictly diagonally dominant matrix.*

However, $\mathbf{F} \succ 0$ and its strictly diagonally dominant property can not guarantee $\mathbf{F}^{-1}$ is a diagonally dominant matrix in theory. This proposition is tenable only when $K \leq 2$, and a counterexample ($K = 3$ means the dimension of $\mathbf{F}$ is 3) is constructed as shown in below:

$$\mathbf{F}^{-1} = \begin{pmatrix} 5 & 2 & -2 \\ 2 & 5 & 2 \\ -2 & 2 & 5 \end{pmatrix}^{-1} = \begin{pmatrix} 0.43 & -0.29 & 0.29 \\ -0.29 & 0.43 & -0.29 \\ 0.29 & -0.29 & 0.43 \end{pmatrix} \tag{14}$$

In the above example, it is clear that $\mathbf{F}^{-1}$ is not a strictly diagonally dominant matrix despite that $\mathbf{F}$ is.

3.2.3 Rationality of this Replacement in Image Data

In the above theoretical analysis, it seems to be unreasonable when the non-negativity constraint is replaced by ℓ_2 norm regularization term on $\mathbf{c}_i$. However, in our practical application, some implicit information in $\mathbf{B}_i$, $\mathbf{A}$, and $\mathbf{F}$ is overlooked. For example, these elements in these matrices are nonnegative; $\mathbf{B}_i$, $\mathbf{y}_i$ have been normalized, whose values are from 0 to 1.

First, we need to analyse the property of $\mathbf{F}$ and $\mathbf{F}^{-1}$ with respect to a growth tendency to λ. Ostroski $et\ al.$ discusses the upper bound of these elements in a strictly diagonally dominant matrix's inverse matrix [14]. Let $\mathbf{F} = [f]_{ij}$, and

$$\mu_i = \frac{1}{|f_{ii}|}\sum_{i=1,j\neq i}^{K}|f_{ij}|, \;\; 0 \leq \mu_i < 1, \;\; i = 1, 2, ..., K. \tag{15}$$

Then elements in the dominant diagonal of $\mathbf{F}^{-1}$ satisfies:

$$\frac{1}{|f_{jj}|(1+\mu_j)} \leq (\mathbf{F}^{-1})_{jj} \leq \frac{1}{|f_{jj}|(1-\mu_j)} \tag{16}$$

As with λ increases, μ_i tends to decrease. And the value range of $(\mathbf{F}^{-1})_{jj}$ will reduce. When λ tends to a sufficient constant, $\sum_{i=1,j\neq i}^{K}|f_{ij}|$ pales in importance compared with $|f_{ii}|$ (means $\mu_i \to 0$). $(\mathbf{F}^{-1})_{jj}$ approximates to $1/|f_{jj}|$. $\mathbf{F}^{-1}$ is a diagonal matrix approximately. In this case, as a diagonal matrix, $\mathbf{F}^{-1}$ is definitely a diagonally dominant matrix. In a word, by choosing an appropriate λ, $\mathbf{F}^{-1}$ approximates to a diagonally dominant matrix to ensure $\mathbf{c}_i$ nonnegative.

Now we should be confronted with how to select the proper λ. That λ is selected to a extremely large value (e.g.,10^4) would make no sense to our practice application. The relatively proper λ is mainly determined by our image data.

Considering each column of $\mathbf{B}_i$ and $\mathbf{y}_i\mathbf{1}^{\mathrm{T}}$ ($\mathbf{y}_i \in \mathbb{R}^{1024}$ represents a 32×32 image patch), elements in these matrices are quite small. Empirical statistic experiments[2] in Fig.2 show that the mean gray value of each positive template is roughly 0.029, and those of negative templates seem to be relatively chaotic, from 0 to 0.3. Thus the mean gray

[1] $\mathbf{A} = [a_{ij}]$, if $|a_{ii}| > \sum_{j=1,j\neq i}^{n}|a_{ij}|$, then $\mathbf{A}$ is called as a strictly diagonally dominant matrix.

[2] We analyse 21 sequences in VTB, and datasets available at http://visualtracking.net

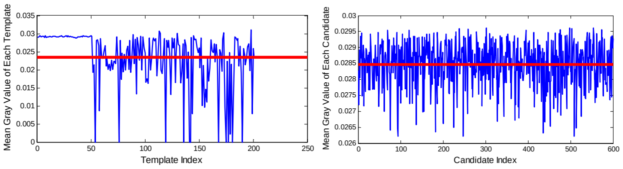

Figure 2. Illustration of the mean gray value of each template in the left-hand figure and candidate in the right-hand figure. The red line represents the mean gray value of all templates and candidates.

value of $\mathbf{B}_i$ selected from positive templates $\mathbf{T}^{pos}$ is approximately equal to 0.029. For candidates as shown in the right-hand of Fig.2, their values range from 0.026 to 0.03. In this condition, each element a_{ij} in $\mathbf{A} = \mathbf{B}_i - \mathbf{y}_i\mathbf{1}^T \in (-0.001, 0.003)$. The upper bound of each element in $\mathbf{A}^T\mathbf{A}$ is estimated to $0.003 \times 0.003 \times 1024 \approx 0.01$.

After the upper bound of elements in $\mathbf{A}^T\mathbf{A}$ are obtained, λ is easily solved.

$$\sum_{i=1, i\neq j}^{K} |f_{ij}| \leq 0.01 * (K-1) \leq \lambda \quad \forall j \tag{17}$$

In sum, the lower bound of the proper λ is obtained in (17). The lower bound of the proper β is in a similar fashion. Specially, with respect to different nearest neighbors $k_1 = 5, k_2 = 8, k_3 = 10$ in our experiment, $\lambda = 1$ is chosen. This value not only satisfies (17) but also is much larger than 0.01. In this condition, $\mathbf{F}$ can be approximated to a diagonal matrix, accordingly, $\mathbf{F}^{-1}$ can be regarded as a strictly diagonal dominant matrix. Moreover, because the **Theorem 1** is a sufficient and unnecessary condition, relatively smaller λ could also guarantee these elements in $\mathbf{c}_i$ non-negative. We will analyze the tracking results with different λ values in Section 5.

3.3. Encoding Vectors and Confidence Measure

From the above analysis, it is reasonable to substitute the non-negativity constraint to ℓ_2 norm regularization term in our image data. Therefore $\mathbf{c}_i^j \in \mathbb{R}^{k_j}$ is obtained by (9) with its corresponding local dictionaries $\mathbf{B}_i^j \in \mathbb{R}^{M\times k_j}$. Noting that the selected templates composed of $\mathbf{B}_i^j$ need to be recorded. The location information of these templates is denoted as an indicator vector $\mathbf{U} = [u_1, u_2, ..., u_{k_j}]$. It means that the first adopted template is in the u_1-th of the whole templates set, and until the k_j adopted template is in the u_{k_j}-th of templates set.

Subsequently, let $\mathbf{d}_i^j \in \mathbb{R}^{p+n}$ be the uniform coefficient vector, whose dimension is equal to the number of templates. Then elements in $\mathbf{c}_i^j$ are assigned to $\mathbf{d}_i^j$. The allocation rule by the indicator vector is following:

$$\mathbf{d}_i^j(u_1) := \mathbf{c}_i^j(1); \;\; \mathbf{d}_i^j(u_2) := \mathbf{c}_i^j(2); ...... \;\; \mathbf{d}_i^j(u_{k_j}) := \mathbf{c}_i^j(k_j) \tag{18}$$

where the remaining elements in $\mathbf{d}_i^j$ are filled with zero.

After weight vector $\mathbf{w}$ and uniform coefficient vector $\mathbf{d}_i^j$ are obtained, the output encoding vector $\mathbf{d}_i = \mathbf{w}^T\mathbf{d}_i^j$. The algorithm for solving the output encoding vector $\mathbf{d}_i$ is described in **Algorithm 1**.

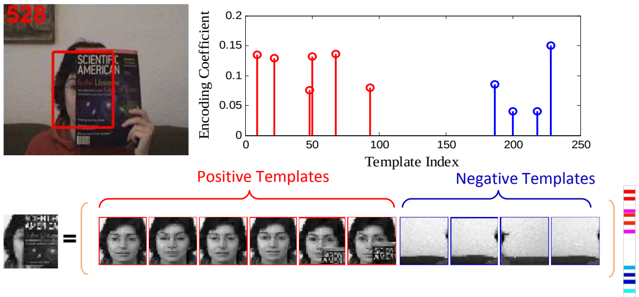

Figure 3. Illustration of an occluded target is represented by positive templates and negative templates.

Algorithm 1. Algorithm for solving $\mathbf{d}_i$.

Input: the candidate $\mathbf{y}_i$, dictionaries with different nearest neighbors: $\mathbf{B}_i^1$, $\mathbf{B}_i^2$,...,$\mathbf{B}_i^m$, iteration times T

Output: the output encoding coefficient $\mathbf{d}_i$

```
1. Initialization: $w_1 = w_2 = ... = w_m = 1/m$
2. for $j = 1, 2, ..., m$ do
   2.1 $\mathbf{c}_i^j$ is obtained by (9).
   2.2. the uniform coefficient vector $\mathbf{d}_i^j$ is obtained by (18).
2. end for
3. for $k = 1, 2, ..., T$ do
   3.1. by fixing $\mathbf{w}$, $\mathbf{d}_i = \sum_{j=1}^{m} w_j \mathbf{d}_i^j$
   3.2. by fixing $\mathbf{d}_i$, $\mathbf{w}$ is solved by Eq.(10).
3. end for
```

$\mathbf{d}_i$ is divided into two parts: $\mathbf{d}_i = [\mathbf{d}_i^{pos}, \mathbf{d}_i^{neg}]$ with respect to $\mathbf{T}^{pos}$ and $\mathbf{T}^{neg}$. Thus, we formulate the confidence value h_i of its corresponding candidate $\mathbf{y}_i$:

$$h_i = \frac{1}{exp(-\alpha(\varepsilon_i^{pos} - \varepsilon_i^{neg}))} \tag{19}$$

where $\varepsilon_i^{pos} = \|\mathbf{y}_i - \mathbf{T}^{pos}\mathbf{d}_i^{pos}\|^2$ is the reconstruction error of the candidate $\mathbf{y}_i$ with the positive template set, and $\mathbf{d}_i^{pos}$ is corresponding coefficients. Similarly, $\varepsilon_i^{neg} = \|\mathbf{y}_i - \mathbf{T}^{neg}\mathbf{d}_i^{neg}\|^2$ is the reconstruction error of the candidate $\mathbf{y}_i$ with the negative template set, and $\mathbf{d}_i^{neg}$ is related coefficients. The parameter α is a normalization factor, fixed to 2.5 in our experiments. The optimal state x_t^* of frame t is the candidate with the highest probability in $\mathbf{H} = [h_1, h_2, ..., h_N]$.

3.4. Occlusion Detection and Model Update

In tracking processing, under the condition of no or slight occlusion, the target should be entirely represented by positive templates. If the object suffers severe occlusion, it is reconstructed by not only the positive template set but also the negative template set.

Based on this, we propose a heuristic method to detect relatively larger occlusion. The occlusion detection criterion is mainly involved with whether negative templates are used to represent the target. If several negative templates are used to reconstruct the target, it means that the target suffers from severe occlusions in high probability, and vice versa. The target is regarded as suffering from severe occlusions when more than one negative template are utilized to reconstruct the target, just to decrease error detection rate. For the sake of mathematical convenience, we denote the number of negative templates representing the target as $LEN(neg_*)$.

The result of the occluded target reconstructed by template sets is shown in Fig.3. For example, at #528th frame, the severely occluded target is represented by six positive templates and four negative templates. Four negative templates are used to represent the target, which means that the target is heavily occluded. Therefore, the experimental results verify the validity of occlusion detection criterion.

Normally when an occlusion is detected, the positive template set should not be updated while the negative template set is updated regularly every 5 frames. If the reconstruction error with the positive template set ε_*^{pos} is smaller than a pre-defined threshold, such as 0.1, we conclude that the current tracking result is a good candidate to represent the target in the following sequence. Thus the good tracking result is added into the positive template set. Along with these good tracking results continuously added in, the size of positive template set becomes larger. To avoid higher computational complexity, a template in positive template set closest to the newly added good tracking result is substituted when the number of positive template reaches 100.

The proposed tracker use approximated LLC for finding encoding vectors based on particle filter framework. The flowchart of the tracking algorithm is summarized in **Algorithm 2**.

Algorithm 2. Algorithm for Our proposed Tracker.

```
1. Initialization: Extract templates $\mathbf{T}^{pos}$, $\mathbf{T}^{neg}$ in the 1st frame.
2. for t = 2 to the end of the sequence
  2.1. $N$ particles $\mathbf{Y}_{1:N}$ are sampled.
  2.2. Construct different dictionaries with different
         $(k_1, k_2, ..., k_m)$ nearest neighbors: $\mathbf{B}_i^1$, $\mathbf{B}_i^2$,...,$\mathbf{B}_i^m$.
  2.3. for $i = 1 : N$
    2.3.1 Solve the output encoding vector $\mathbf{d}_i$ by Algorithm 1.
    2.3.2 Calculate confidence value $h_i$ of $\mathbf{y}_i$ by (19).
  2.3. end for
  2.4. Chose the optimal state $\mathbf{x}_t^*$ by the highest confidence value.
  2.5. Update: for every 5 frames
      2.5.1 Update negative templates
      2.5.2 if $LEN(neg_*) \leq 2$
         2.5.2.1 incremental update the positive template set.
      2.5.2 end if
  2.5. end for
2. end for
```

3.5. Further Analysis on Regularization term

Based on the above analysis, non-negative property of $\mathbf{c}_i$ are involved with choosing an appropriate λ.

From machine learning view, the objective function in (9) can be viewed as a loss function $\|\mathbf{y}_i - \mathbf{B}_i\mathbf{c}_i\|^2$ and its regularization term $\lambda\|\mathbf{c}_i\|^2$. Only considering the regularization term and its shift-invariant constraint:

$$\min \ \lambda\|\mathbf{c}_i\|^2 \quad s.t. \mathbf{1}^{\mathrm{T}}\mathbf{c}_i = 1, \forall i \tag{20}$$

This is an average inequality and the minimum is following:

$$\begin{aligned} \mathbf{c}_i^{\mathrm{T}}\mathbf{c}_i &= \sum_{j=1}^{K} c_{i(j)}^2 = c_{i(1)}^2 + c_{i(2)}^2 + ... + c_{i(K)}^2 \\ &\geq (\frac{c_{i(1)} + c_{i(2)} + ... + c_{i(K)}}{K})^2 = \frac{1}{K^2} \end{aligned} \tag{21}$$

The inequality achieves the minimal value if and only if $c_{i(1)} = c_{i(2)} = ... = c_{i(K)} = \frac{1}{K}$. It illustrates that these nonnegative elements in $\mathbf{c}_i$ tend to be approximately equal by adjusting the regularization parameter λ. The larger regularization parameter is, the more obvious average effect shown in Eq.(21) on encoding coefficients have. On the other hand, by adding this ℓ_2 norm regularization term, the model effectively avoids over-fitting. As with λ increases, the bias of the model increases and the variance falls down. The disparity and diversity of the output $\mathbf{c}_i$ compared with its expectation would decrease. In other words, $\mathbf{c}_i$ is relatively more stable.

Experiments about the benefit of the additional ℓ_2 norm regularization term $\lambda\|\mathbf{c}_i\|^2$ seem clearly to be the litmus test for our discussions as shown in Fig.1. The red one with $\lambda = 1$ in Fig.1 shows that the target is approximately equally reconstructed by the local dictionary composed of eight positive templates. These eight positive templates takes the same and equal effect on representing the target. We think this phenomenon that the target should be equally represented by several positive templates have distinct physical meanings.

In sum, the added ℓ_2 norm regularization term leads to many advantages more than avoidance of over-fitting. Especially for high-dimension data (e.g., our image data), in approximated LLC, the non-negativity constraint is entirely substituted by the ℓ_2 norm regularization with adjusting an appropriate regularization parameter λ. And this not only fits our computer vision applications but also will be well done in other fields.

4. Experiments

Setup: The proposed tracker was implemented in MATLAB with a PC with Intel Xeon E5506 CPU (2.13 GHz) with 24 GB memory. The following parameters were used for our tests: each observation (i.e. patch of image) was

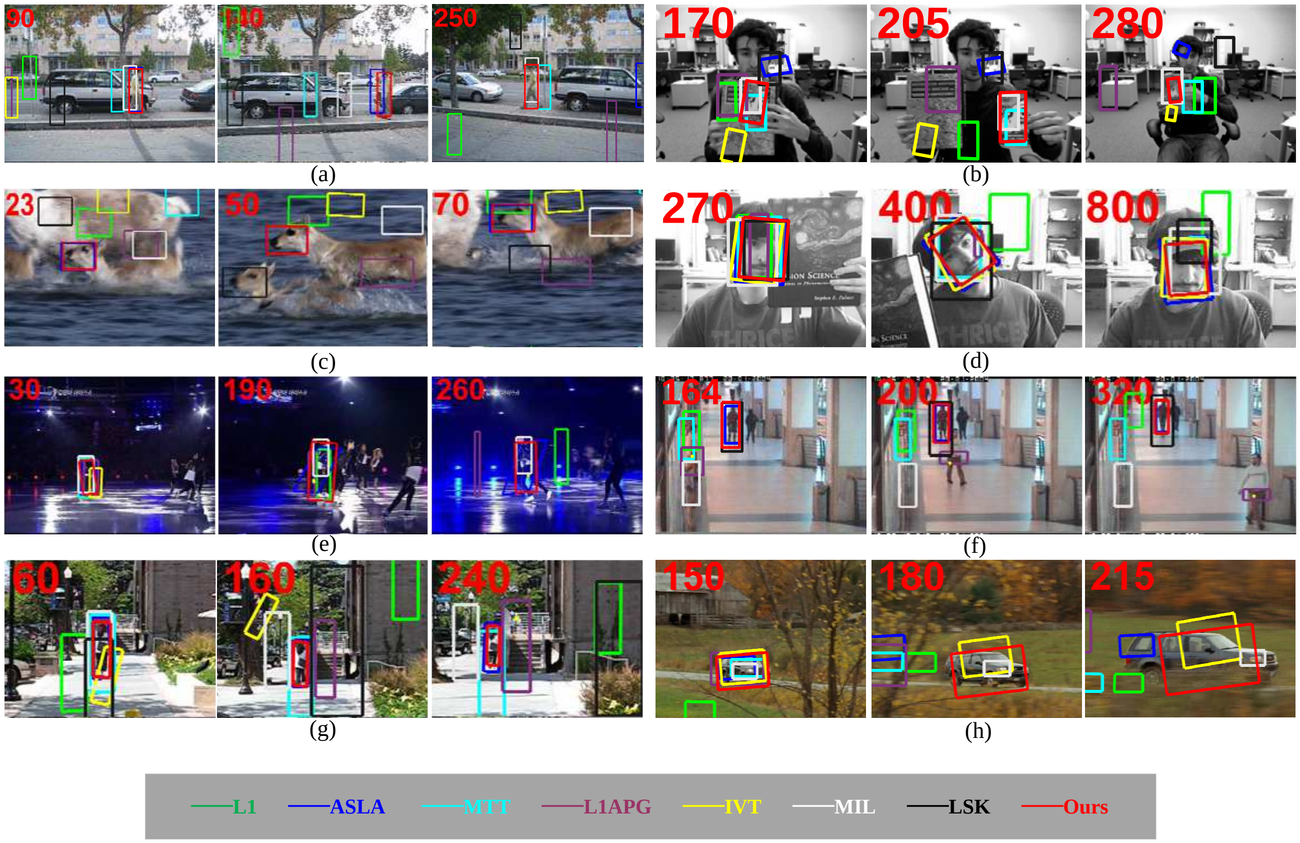

Figure 4. Representative frames of some sampled tracking results. And subfigures from top to bottom, left to right: (a) - (h), from video David3, ClifBar, Deer, Faceocc2, Skating1, Caviar1, Human7 and Carscale.

Table 1. Performance in terms of ”center location error” (CLE) in pixels. Red and blue colors indicate the best and 2nd best performance, respectively.

Sequence	LSK	L1	L1APG	MTT	IVT	MIL	ASLA	KCF	Ours
Car4	258.6	4.1	223.0	96.0	2.6	60.1	3.5	19.1	4.3
Carscale	34.5	66.8	81.2	83.7	11.7	27.3	48.8	43.0	10.9
David3	162.6	100.4	204.4	363.3	100.2	38.4	87.4	5.5	6.0
Deer	117.0	97.9	197.9	15.2	123.8	225.8	10.7	23.4	12.3
Faceocc2	19.1	11.1	16.2	9.9	8.4	14.1	3.5	10.5	7.5
Caviar3	23.7	65.9	26.6	64.8	66.2	57.8	2.2	23.4	17.7
ClifBar	71.1	75.9	72.1	35.7	62.6	7.8	61.1	41.6	5.0
Human7	51.4	103.7	27.8	17.0	54.1	21.9	2.9	57.6	3.4
Faceocc1	5.5	6.5	8.3	17.3	11.7	32.2	7.7	77.3	6.4
Skating1	90.5	32.6	145.4	256.3	31.9	86.2	47.2	22.7	10.7
Caviar1	8.0	34.6	95.0	55.2	98.5	88.2	1.6	4.9	4.9
Singer1	193.9	87.8	4.6	16.6	11.4	15.2	5.1	14.0	8.3
avg.	86.3	57.3	91.9	85.9	48.6	56.2	23.5	28.6	8.1

normalized to 32 × 32 pixels; different nearest neighbors ($k_1 = 5, k_2 = 8, k_3 = 10$) as the three ($m = 3$) local dictionaries were selected; and the iteration times T was fixed with 3; the ℓ_2 norm regularization parameters were set to $\lambda = 1, \beta = 0.1$; Especially, different λ were chosen to analyze the influence of tracking results.

To evaluate the proposed method against the state-of-the-art, 8 existing methods are selected, including KCF [5], ASLA [7], LSK [10], L1 trakcer [12], L1APG [3], MTT [25], IVT [16] and MIL [2].

Table 2. Performance in terms of ”overlap rate” e (in pixels). Red and blue colors indicate the best and 2nd best performance, respectively.

Sequence	LSK	L1	L1APG	MTT	IVT	MIL	ASLA	KCF	Ours
Car4	0.05	0.84	0.15	0.45	0.91	0.34	0.91	0.47	0.90
Carscale	0.51	0.36	0.55	0.49	0.62	0.42	0.45	0.41	0.70
David3	0.12	0.35	0.14	0.09	0.31	0.41	0.46	0.75	0.75
Deer	0.21	0.07	0.05	0.55	0.04	0.04	0.60	0.60	0.57
Faceocc2	0.56	0.67	0.34	0.70	0.74	0.61	0.80	0.72	0.76
Caviar3	0.36	0.20	0.19	0.14	0.13	0.11	0.84	0.14	0.48
ClifBar	0.12	0.20	0.27	0.29	0.11	0.53	0.21	0.25	0.65
Human7	0.17	0.05	0.27	0.48	0.11	0.29	0.81	0.28	0.77
Faceocc1	0.82	0.88	0.84	0.72	0.82	0.59	0.87	0.10	0.88
Skating1	0.28	0.39	0.10	0.10	0.06	0.31	0.40	0.45	0.50
Caviar1	0.55	0.28	0.28	0.28	0.27	0.25	0.89	0.69	0.80
Singer1	0.21	0.24	0.77	0.42	0.54	0.34	0.79	0.36	0.65
avg.	0.33	0.38	0.33	0.39	0.39	0.35	0.67	0.44	0.70
fps.	6.57	0.28	4.41	1.02	16.41	0.86	0.95	89.8	1.28

Results: Fig.4 shows screen shots of tracking results from different trackers. Tab.1 shows the performance of these methods based on the center location error (CLE), where a small CLE value indicates more accurate hence better tracking. Tab.2 shows the performance of different methods based on the overlap rate between the tracked bounding box and the ground truth box from these methods on several videos. The overlap rate is defined as $e = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where R_T and R_G are the area of tracked and ground truth box, respectively.

4.1. Qualitative Evaluation

Heavy Occlusion: Fig.4(a), (d), (f) and (h) demonstrate that the proposed method performs well in terms of position and rotation when the target undergoes severe occlusion. In the David3 sequence, IVT, L1APG and MTT completely fail to track at frames #34, #57 and #82. After David passes through the tree, ASLA and our method can effectively locate the target, whereas MIL suffers a slight drifts. In the Faceocc2 and Caviar1 Sequence, ASLA and our method performs better than the other methods in terms of tracking accuracy. When comes to the target occluded by branches in the Carscale sequences at frame #172, most methods suffers from severe drift while MIL and IVT are far from satisfaction. Only our proposed method shows a preferable tracking result. In sum, because of the non-negativity constraint, encoding coefficients are nonnegative from the selected positive templates. These selected positive templates cannot be regarded as negative templates as shown in Fig.1. Tracking results verify that the non-negativity constraint and model update scheme with the severe occlusion detection method are reasonable.

Shape Deformation and Rotation Variation: The trackers are easily confused if the object has changed in appearance of the target dramatically. Fig.4(b), (e) and (h) illustrate the tracking results in the ClifBar, Skating1 and Carscale sequences with scale and rotation variation of the card, the skater and the car. In the ClifBar sequence, our proposed method and MIL perform favorably better than other methods despite that the target undergoes severe scale and rotation variation. The skater in Fig.4(e) dramatically deforms her pose. Only our proposed method precisely tracks the skater to some extent under the condition of complicated background and low light. The first half of Carscale sequences, only our method could adaptive the change in size of the car. At frame #215, the car undergoes not only scale variation but also heavy occlusion and fast motion, it is difficult for our method to achieve a satisfying performance.

Abrupt Motion and Camera Shake: Fig.4(c) and (g) shows the tracking results on the Deer and Human7 sequences. It is difficult to predict the location of this deer and this woman accurately when they undergo an abrupt motion and camera shake. Most trackers lose tracking accuracy in Human7 sequence and even suffer from severe drifting in the Deer sequence. Our tracker and ASLA are able to distinguish the target from their surrounding background, and handles drift problem.

In summary, our test results on these videos with heavy occlusions or motion blur have shown that the proposed tracker and ASLA are effective and robust. In terms of sequences with scale and shape variation, only our proposed tracker performs favorably better than other methods.

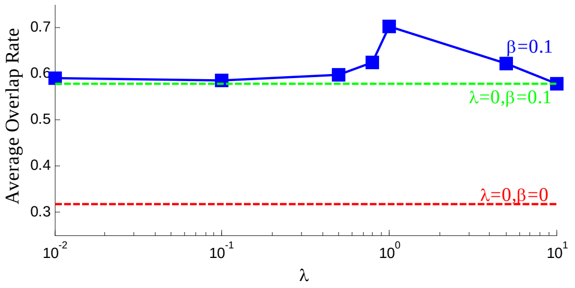

Figure 5. The average overlap rate of our proposed tracker versus parameter λ on these twelve sequences.

4.2. The Regularization Parameter Selection

Fig.5 shows the influence of regularization parameter λ with different values $(0, 0.01, 0.1, 0.5, 0.8, 1, 5, 10)$ on average overlap rate. Without regularization term $(\lambda = \beta = 0)$ shown in the red dashed line, drifts happen in most sequences with a very low average overlap rate $(e = 0.3175)$, forming a sharp contrast to these methods with regularization term. Under $\beta = 0.1$, when λ ranges from 0.01 to 10, the average overlap rate (in blue line) steadily increases and then falls down, where it reaches the summit at $\lambda = 1$. The green dashed line represents $\lambda = 0, \beta = 0.1$, just for a fair comparison with the blue fold line. If the value of λ is too small, or with image data in the same order of magnitude $(\lambda \in [0.01, 0.1])$, there is slight influence on the final tracking results. If the value of λ is too large, it is easily overfitting and loss function pales in importance. It is inevitable to seek for a tradeoff between the accuracy of appearance model and the regularization term.

5. Conclusion

This paper proposes an effective tracker with regularization term on encoding coefficients and different numbers of nearest neighbours in the multiple local dictionaries. The non-negativity constraint on encoding coefficients is substituted by the ℓ_2 regularization term. This replacement is rational for our computer vision applications. It not only leads to these elements nonnegative, but also has an average effect on them. These characteristics guarantee the tracking results more reliable and robust. Moreover, the optimal convex combination of multiple local dictionaries is learned from approximated LLC. And our occlusion detection method effectively prevents positive templates to update when the target undergoes severe occlusion. Experimental results demonstrate that our proposed algorithm is able to track the target accurately with challenging factors.

Acknowledgement

This research is partly supported by NSFC, China (No: 61273258) and 863 Plan, China (No. 2015AA042308). Jie Yang is the corresponding author.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 798–805, 2006.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust L1 tracker using accelerated proximal gradient approach. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1830–1837, 2012.

[4] C. Gong, K. Fu, A. Loza, Q. Wu, J. Liu, and J. Yang. PageRank tracker: From ranking to tracking. *IEEE Transactions on Cybernetics*, 44(6):882–893, 2014.

[5] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596, 2015.

[6] Z. Hong, X. Mei, D. Prokhorov, and D. Tao. Tracking via robust multi-task multi-view joint sparse representation. *Proceedings of the IEEE International Conference on Computer Vision*, pages 649–656, 2013.

[7] X. Jia, H. Lu, and M. H. Yang. Visual tracking via adaptive structural local sparse appearance model. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1822–1829, 2012.

[8] J. Kwon, K. M. Lee, and F. C. Park. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 991–998, 2009.

[9] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[10] B. Liu, J. Huang, C. Kulikowski, and L. Yang. Robust visual tracking using local sparse appearance model and k-selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2968–2981, 2013.

[11] H. Liu, M. Yuan, F. Sun, and J. Zhang. Spatial neighborhood-constrained linear coding for visual object tracking. *IEEE Transactions on Industrial Informatics*, 10(1):469–480, 2014.

[12] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2259–2272, 2011.

[13] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient $\ell 1$ tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264. IEEE, 2011.

[14] A. M. Ostrowski. Note on bounds for determinants with dominant principal diagonal. *Proceedings of the American Mathematical Society*, 3(1):26–30, 1952.

[15] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

[16] D. A. Ross, J. Lim, R.-s. Lin, and M.-h. Yang. Incremental Learning for Robust Visual Tracking, 2008.

[17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[18] N. Wang, J. Wang, and D. Y. Yeung. Online robust non-negative dictionary learning for visual tracking. *Proceedings of the IEEE International Conference on Computer Vision*, pages 657–664, 2013.

[19] Q. Wang, F. Chen, W. Xu, and M.-H. Yang. Online discriminative object tracking with local sparse representation. In *WACV*, pages 425–432, 2012.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[21] Y. Wu, J. Lim, and M.-H. Yang. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):1–1, 2015.

[22] Y. Wu, B. Shen, and H. Ling. Visual tracking via online non-negative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):374–383, 2014.

[23] S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46(7):1772–1788, 2013.

[24] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *Computer Vision–ECCV 2012*, pages 470–484. Springer, 2012.

[25] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, 101(2):367–383, 2013.

[26] T. Zhang, S. Liu, N. Ahuja, M. H. Yang, and B. Ghanem. Robust Visual Tracking Via Consistent Low-Rank Sparse Learning, 2014.

[27] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, volume 23, pages 1838–1845, 2012.

[28] T. Zhou, X. He, K. Xie, K. Fu, J. Zhang, and J. Yang. Robust visual tracking via efficient manifold ranking with low-dimensional compressive features. *Pattern Recognition*, 48(8):2459–2473, 2015.