Least Soft-thresold Squares Tracking

Dong Wang
Dalian University of Technology
wangdong.ice@gmail.com

Huchuan Lu
Dalian University of Technology
lhchuan@dlut.edu.cn

Ming-Hsuan Yang
University of California at Merced
mhyang@ucmerced.edu

Abstract

In this paper, we propose a generative tracking method based on a novel robust linear regression algorithm. In contrast to existing methods, the proposed Least Soft-thresold Squares (LSS) algorithm models the error term with the Gaussian-Laplacian distribution, which can be solved efficiently. Based on maximum joint likelihood of parameters, we derive a LSS distance to measure the difference between an observation sample and the dictionary. Compared with the distance derived from ordinary least squares methods, the proposed metric is more effective in dealing with outliers. In addition, we present an update scheme to capture the appearance change of the tracked target and ensure that the model is properly updated. Experimental results on several challenging image sequences demonstrate that the proposed tracker achieves more favorable performance than the state-of-the-art methods.

1. Introduction

Visual tracking plays a critical role in computer vision that finds many practical applications (e.g., motion analysis, video surveillance, vehicle navigation and human-computer interaction). Although significant progress has been made in the past decades, developing a robust tracking algorithm is still a challenging problem due to numerous factors such as partial occlusion, illumination variation, pose change, complex motion, and background clutter.

Tracking algorithms can be classified as either generative (e.g., [7, 1, 20, 19, 21, 15]) or discriminative (e.g., [2, 10, 4, 11, 27, 18, 30, 14]) methods. Generative methods focus on searching for the regions which are the most similar to the tracked targets, while discriminative methods cast tracking as a classification problem that distinguishes the tracked targets from the surrounding backgrounds. In this work, we propose a robust generative tracker which is able to handle partial occlusion and other challenging factors effectively.

Among the generative methods, the trackers based on linear representation maintain holistic appearance information and therefore provide a compact notion of the "thing" being tracked, which may facilitate some advanced vision tasks [7, 20]. These methods often adopt a dictionary (e.g., a set of basis vectors from a subspace or a series of templates) to describe the tracked target. A given candidate sample is linearly represented by the dictionary, and the representation coefficient and reconstruction error are computed, from which the corresponding likelihood (belonging to the object class) is determined. Ross *et al.* [20] propose an incremental visual tracking (IVT) method which represents the tracked target by a low dimensional PCA subspace (a set of PCA basis vectors) and assumes that the error is Gaussian distributed with small variances (i.e., small dense noise). Therefore, the representation coefficient can be obtained by a simple projection operator, which is equivalent to the ordinary least squares solution under the assumption that the dictionary atoms are orthogonal. The reconstruction error is computed by the objective function of the ordinary least squares methods. While the IVT method is effective to handle appearance change caused by illumination variation and pose variation, it is not robust to some challenging factors (e.g., partial occlusion and background clutter) due to the following two reasons. First, ordinary least squares methods have been shown to be sensitive to outliers due to the formulation based on reconstruction error with Gaussian noise assumption. Second, the IVT method uses new observations to update the observation model without detecting outliers and processing them accordingly. Other recent tracking algorithms [16, 12] based on the Gaussian noise assumption or the ordinary least squares methods have similar problems as the IVT method.

Motivated by the success of sparse representation-based face recognition [28], Mei *et al.* [19] develop a novel ℓ_1 tracker that uses a series of target templates and trivial templates to model the tracked target, where the target templates are used to describe the object class to be tracked and trivial templates are used to deal with outliers (e.g., partial occlusion) with the sparsity constraints. For tracking, a candidate sample can be sparsely represented by both target and trivial templates, and its corresponding likelihood is determined by the reconstruction error with respect to target templates. We note that this formulation is a linear regres-

sion problem with sparsity constraints on the representation coefficients. Recently, several methods have been proposed to improve the ℓ_1 tracker in terms of both speed and accuracy by using accelerated proximal gradient algorithm [5], replacing raw pixel templates with orthogonal basis vectors [24, 26, 25], modeling the similarity between different candidates [31], to name a few. Although these algorithms consider outliers by using additional trivial templates, this formulation can be generalized with better understanding. In this work, we show that the linear regression with the Gaussian-Laplacian noise assumption is more effective in dealing with outliers for object tracking. In addition, from the viewpoint of linear regression, it is not suitable to estimate the likelihood based on the reconstruction error with respect to target templates. We present a novel distance function to compute the distance between a candidate and the object class.

In this paper, we present a generative tracking algorithm based on linear regression. The contributions of this work are as follows. First, we introduce a novel linear regression method, Least Soft-thresold Squares (LSS), which assumes that the error vectors follow the i.i.d Gaussian-Laplacian distribution. Second, we present an efficient iteration method to solve the LSS problem and propose a LSS distance to measure the dissimilarity between the observation vector and dictionary. We note that the LSS method is related to the robust regression with the Huber loss function and is effective in detecting outliers. Compared with the least squares distance, the LSS distance is more effective in measuring the distance between the observation vector and dictionary when outliers occur. Third, we design a generative tracker by using the LSS method, where the dictionary consists of PCA basis vectors. The likelihood of each candidate is computed based on the LSS distance. Furthermore, we update the tracker by using an effective update scheme. Numerous experiments on challenging image sequences with comparisons to state-of-the-art tracking methods demonstrate the effectiveness of the proposed model and algorithm.

2. Least Soft-thresold Squares

2.1. Regression with Gaussian-Laplacian Noise

The objective of linear regression is to fit a linear model (i.e, estimate model parameters) to a series of noisy observations:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^{d\times 1}$ is a d-dimensional observation vector, $\mathbf{x} \in \mathbb{R}^{k\times 1}$ denotes the k-dimensional parameter (or coefficient) vector to be estimated and $\mathbf{A} = [\mathbf{r}_1; \mathbf{r}_2; ...; \mathbf{r}_d] \in \mathbb{R}^{d\times k}$ represents the input data matrix ($\mathbf{r}_i$ is the i-th row of $\mathbf{A}$). We denote $\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{x} = [e_1; e_2; ...; e_d]$ (i.e, $e_i = y_i - \mathbf{r}_i \cdot \mathbf{x},\ i = 1, ..., d$) as the error or residual term.

The coefficient $\mathbf{x}$ can be obtained by maximizing the posteriori probability $p(\mathbf{x}|\mathbf{y})$, which is also equivalent to maximizing the joint likelihood probability $p(\mathbf{x}, \mathbf{y})$. Assume there is a uniform prior, the coefficient $\mathbf{x}$ is estimated by $\widehat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{x}} p(\mathbf{e})$, which is the maximum likelihood estimation (MLE).

The errors $e_1, e_2, ..., e_d$ are usually assumed to be independently and identically distributed (i.i.d) according to some probability density function (PDF) $f_\theta(e_i)$, where θ is the parameter set that characterizes the probability distribution. Thus, the likelihood of the estimator (the joint probability of the error term $\mathbf{e}$) is $p(\mathbf{e}) = \prod_{i=1}^{d} f_\theta(e_i)$. To maximize the likelihood function is equivalent to minimizing the objective function $L_\theta(e_1, e_2, ..., e_d) = \sum_{i=1}^{d} \rho_\theta(e_i)$, where $\rho_\theta(e_i) = -\log f_\theta(e_i)$.

When the error $\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{x}$ follows the Gaussian distribution ($e_i \in \mathcal{N}\left(0, \sigma_N^2\right)$[1]), the MLE solution is equivalent to the ordinary least squares (OLS) solution

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, \tag{2}$$

and the closed form solution is $\widehat{\mathbf{x}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{y}$. Although the OLS method is easy to solve, it is sensitive to outliers due to the Gaussian noise assumption. If the error $\mathbf{e}$ follows the Laplacian distribution ($e_i \in \mathcal{L}(0, \sigma_L)$[2]), the MLE solution is equivalent to least absolute deviations (LAD) solution,

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_1. \tag{3}$$

Compared with the OLS method, the LAD method is robust to outliers. However, it is difficult to be solved by using either the simplex-based methods [6] or the iteratively reweighted least squares methods [22].

In this paper, we model error vector $\mathbf{e}$ as an additive combination of two independent components: an i.i.d Gaussian noise vector $\mathbf{n}$ ($n_i \in \mathcal{N}\left(0, \sigma_N^2\right)$) and an i.i.d Laplacian noise vector $\mathbf{s}$ ($s_i \in \mathcal{L}(0, \sigma_L)$),

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} + \mathbf{s}, \tag{4}$$

where the Gaussian component models small dense noise and the Laplacian one aims to handle outliers [3].

[1] e_i is a zero-mean Gaussian random variable with variance σ_N^2 and its PDF is $f_{\mathcal{N}}(e_i) = \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left(-\frac{e_i^2}{2\sigma_N^2}\right)$.

[2] e_i is a zero-mean Laplacian random variable with variance σ_L^2 and its PDF is $f_{\mathcal{L}}(e_i) = \frac{1}{\sqrt{2}\sigma_L} \exp\left(-\frac{\sqrt{2}|e_i|}{\sigma_L}\right)$.

[3] The similar idea of decomposing the noise into two components (e.g., decomposing the noise into sparse and non-sparse ones [3, 8] for achieving robust motion estimation) have appeared in computer vision community. While the goals are similar, the formulations and derivations are different. In this work, the proposed algorithm focuses on not only handling occlusion but also deriving a novel distance metric to compare the target candidate and the target template under the outlier condition, which will presented later.

The additive combination of i.i.d Gaussian and i.i.d Laplacian noise variables is also called Gaussian-Laplacian distribution [23]. Its joint PDF is $p(\mathbf{e}) = \prod_{i=1}^{d} f_{\mathcal{NL}}(e_i)$, where the PDF $f_{\mathcal{NL}}(e_i)$ is given by the convolution,

$$\begin{aligned} & f_{\mathcal{NL}}(e_i) \\ &= f_{\mathcal{N}}(n_i) * f_{\mathcal{L}}(s_i) \\ &= \int f_{\mathcal{L}}(s_i) f_{\mathcal{N}}(e_i - s_i)\, ds_i \\ &= \frac{1}{2\sqrt{2}\sigma_N} \exp\left(-\frac{e_i^2}{2\sigma_N^2}\right) \left[\begin{array}{l} erfcx\left(\frac{\sigma_N}{\sigma_L} - \frac{e_i}{\sqrt{2}\sigma_N}\right) \\ +erfcx\left(\frac{\sigma_N}{\sigma_L} + \frac{e_i}{\sqrt{2}\sigma_N}\right) \end{array} \right], \end{aligned} \tag{5}$$

where $erfcx(x) = \exp(x^2) erfc(x)$ and $erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$. Compared with the Gaussian and Laplacian distributions, the PDF of the Gaussian-Laplacian distribution is complex. Thus, it is difficult to obtain a simple objective function (such as Eqs. 2 and 3) directly.

Because of this, we treat the Laplacian noise term $\mathbf{s}$ as missing values with the same Laplacian prior, and therefore the joint likelihood $p(\mathbf{y}, \mathbf{x})$ is converted to $p(\mathbf{y}, \mathbf{x}, \mathbf{s})$.

$$\begin{aligned} & p(\mathbf{y}, \mathbf{x}, \mathbf{s}) \\ &= p(\mathbf{y}|\mathbf{x}, \mathbf{s})\, p(\mathbf{x}, \mathbf{s}) \\ &= p(\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{s})\, p(\mathbf{s}) \\ &= K \exp\left\{ -\frac{1}{\sigma_N^2} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1 \right) \right\}, \end{aligned} \tag{6}$$

where $K = \left(\frac{1}{\sqrt{2}\sigma_L}\right)^d \left(\frac{1}{\sqrt{2\pi}\sigma_N}\right)^d$ and $\lambda = \frac{\sqrt{2}\sigma_N^2}{\sigma_L}$. Thus, to maximize the joint likelihood $p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ is equivalent to minimizing the function $\frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1$ with respect to both $\mathbf{x}$ and $\mathbf{s}$.

2.2. Least Soft-thresold Squares Regression

To maximize the joint likelihood of Eq. 6, we consider the objective function:

$$L(\mathbf{x}, \mathbf{s}) = \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1, \tag{7}$$

and the optimal solution is $[\widehat{\mathbf{x}}, \widehat{\mathbf{s}}] = \arg\min_{\mathbf{x},\mathbf{s}} L(\mathbf{x}, \mathbf{s})$. The above objective function is based on the standard least squares criterion and an ℓ_1 regularization term on $\mathbf{s}$, and thus is convex but not differentiable everywhere. To the best of our knowledge, there is no closed-form solution for this optimization problem, so we present an iterative algorithm.

Proposition 1: Given $\widehat{\mathbf{s}}$, the optimal $\widehat{\mathbf{x}}$ can be computed by the ordinary least squares solution $\widehat{\mathbf{x}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top (\mathbf{y} - \widehat{\mathbf{s}})$.

Proposition 2: Given $\widehat{\mathbf{x}}$, the optimal $\widehat{\mathbf{s}}$ can be obtained by a soft-thresholding (or shrinkage) operation $\mathcal{S}_\lambda(\mathbf{y} - \mathbf{A}\widehat{\mathbf{x}})$, where $\mathcal{S}_\tau(\mathbf{x}) = \max(\mathbf{x} - \tau, 0)\mathrm{sgn}(\mathbf{x})$, where $\mathrm{sgn}(\cdot)$ is the sign function.

Let $\mathbf{P}$ denote $\left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top$, which can be precomputed before the iterative process. By Propositions 1 and 2, the optimization can be solved efficiently. The iterative operation is terminated when a stopping criterion is met (e.g., the difference of objective values between two iterations or the number of iterations). As our algorithm consists of two main components: ordinary least squares and soft-thresholding operation, we denote it as the *Least Soft-threshold Squares Regression* method.

Table 1. Least Soft-threshold Squares Regression

Input: An observation vector $\mathbf{y}$, matrix $\mathbf{A}$, percomputed matrix $\mathbf{P} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top$, and a small constant λ.
1: Initialize $\mathbf{s}_0 = \mathbf{0}$ and $i = 0$
2: Iterate
3: Obtain $\mathbf{x}_{i+1}$ via $\mathbf{x}_{i+1} = \mathbf{P}(\mathbf{y} - \mathbf{s}_i)$
4: Obtain $\mathbf{s}_{i+1}$ via $\mathbf{s}_{i+1} = \mathcal{S}_\lambda(\mathbf{y} - \mathbf{A}\mathbf{x}_{i+1})$
5: $i \leftarrow i + 1$
6: Until convergence or termination
Output: $\widehat{\mathbf{x}}, \widehat{\mathbf{s}}$

It is clear that $L(\mathbf{x}_{i+1}, \mathbf{s}_{i+1}) \leq L(\mathbf{x}_{i+1}, \mathbf{s}_i) \leq L(\mathbf{x}_i, \mathbf{s}_i)$ [4]. Hence, the proposed iterative algorithm converges to a local minimal value. As the objective function is convex, it also obtains the global minimal solution.

Remark 1: The proposed least soft-threshold squares regression is equivalent to robust regression with the Huber loss function:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \sum_{i=1}^{d} f(e_i), \quad e_i = y_i - \mathbf{r}_i \cdot \mathbf{x}, \tag{8}$$

where

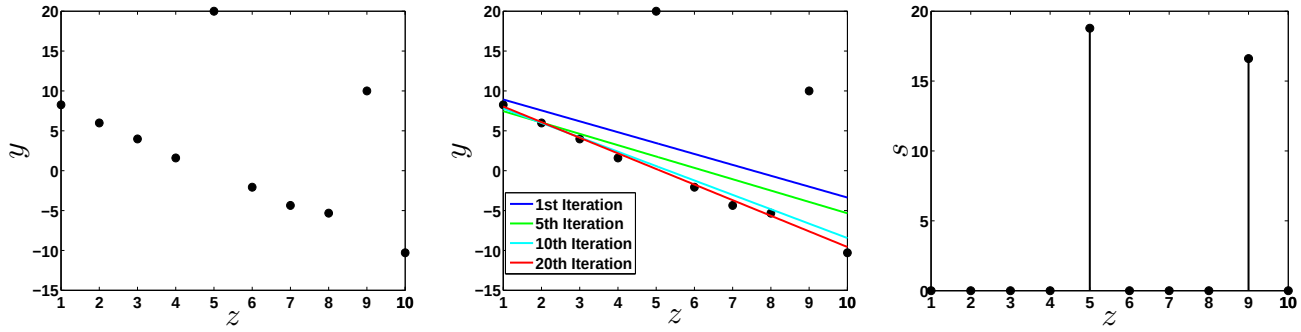
$$f(e) = \begin{cases} e^2/2, & |e| \leq \lambda \\ \lambda|e| - \lambda^2/2, & |e| > \lambda \end{cases} \tag{9}$$

is the Huber loss function and $\mathbf{r}_i$ denotes the i-th row of $\mathbf{A}$. A detailed explanation can be found in the supplementary material.

We note the approach to compute robust regression with the Huber loss function is less efficient than the proposed method as it is generally solved by using the iteratively reweighted least squares scheme which requires solving a weighted least squares problem (i.e., pseudo-inverse) at each iteration. As mentioned before, the pseudo-inverse matrix can be percalculated before the iterative process in the proposed LSS method.

Remark 2: The non-zero components in $\mathbf{s}$ can be used to identify outliers.

[4] $L(\mathbf{x}_{i+1}, \mathbf{s}_i) = \min_{\mathbf{x}} L(\mathbf{x}, \mathbf{s}_i) \leq L(\mathbf{x}_i, \mathbf{s}_i)$, and $L(\mathbf{x}_{i+1}, \mathbf{s}_{i+1}) = \min_{\mathbf{s}} L(\mathbf{x}_{i+1}, \mathbf{s}) \leq L(\mathbf{x}_{i+1}, \mathbf{s}_i)$



(a) Observation samples. (b) Iterative process in the LSS method. (c) The Laplacian noise term.

Figure 1. Robust line fitting by using the least soft-threshold squares (LSS) regression.

Figure 1 shows an example of fitting a straight line when small Gaussian noise and outliers occur simultaneously. It requires to estimate an accurate parameter for a straight line function $y = az + b$, where z is the input variable, y is the output variable, and a and b are parameters to be determined. Figure 1(a) shows ten observation sample pairs $\{z_i, y_i\}, i = 1, ..., 10$, where $z_i = i$ in this case (z_5 and z_9 are two outliers). Denote that $\mathbf{y} = [y_1; y_2; ...; y_{10}]$, $\mathbf{z} = [z_1; z_2; ...; z_{10}]$, $\mathbf{A} = [\mathbf{z}, \mathbf{1}]$ and $\mathbf{x} = [a; b]$, we use the proposed least soft-threshold squares (LSS) method to estimate the parameter $\mathbf{x}$. Figure 1(b) illustrates the iterative process of the LSS method (λ is set to 1), from which it is clear that the proposed LSS is not sensitive to outliers. We note that the result after the first iteration is the same as the result obtained using the OLS method, which also shows that the proposed LSS method is more robust than the OLS method. In addition, as shown in Figure 1(c), the non-zero Laplacian noise components correspond to the outliers.

2.3. Least Soft-thresold Squares Distance

We represent the matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_k]$, where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$. The vector, $\mathbf{Ax}$, can be viewed as a linear combination of the columns of $\mathbf{A}$ ($\mathbf{Ax} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + ... + x_k\mathbf{a}_k$). The matrix $\mathbf{A}$ is known as dictionary or basis matrix, and the vector $\mathbf{a}_i$ is called an atom or basis vector.

For some vision applications (such as tracking), it requires not only to estimate the coefficient accurately but also to define a distance between a noisy observation and the dictionary or the subspace. This generative perspective has commonly been exploited in subspace-based methods [7, 20]. The distance is usually defined to be inversely proportional to the maximum joint likelihood with respect to the coefficient $\mathbf{x}$,

$$\begin{aligned} & d(\mathbf{y}; \mathbf{A}) \\ & \propto -\log \max_{\mathbf{x}} p(\mathbf{y}, \mathbf{x}) \\ & = -\log \max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}). \end{aligned} \tag{10}$$

Take the ordinary least squares method for example (i.e., uniform prior), we have

$$\begin{aligned} & -\log \max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \\ & \propto -\log \max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) \\ & \propto -\log \max_{\mathbf{x}} \exp\left(-\tfrac{1}{2}\|\mathbf{y} - \mathbf{Ax}\|_2^2\right) \\ & \propto \min_{\mathbf{x}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{Ax}\|_2^2. \end{aligned} \tag{11}$$

Recall that the OLS method assumes the observation vector with i.i.d Gaussian noise, the distance $\mathbf{y}$ and $\mathbf{A}$ can therefore be defined as,

$$\begin{aligned} & d_{OLS}(\mathbf{y}; \mathbf{A}) \\ & = \min_{\mathbf{x}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{Ax}\|_2^2 \\ & = \tfrac{1}{2}\left\|\mathbf{y} - \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{y}\right\|_2^2. \end{aligned} \tag{12}$$

Similarly, under the i.i.d Laplacian noise assumption, the distance between $\mathbf{y}$ and $\mathbf{A}$ can be defined as,

$$d_{LAD}(\mathbf{y}; \mathbf{A}) = \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{Ax}\|_1, \tag{13}$$

which is difficult to be calculated (as mentioned in Section 2.1).

In this work, we adopt the Gaussian-Laplacian distribution to model observation noise. Therefore, we define the distance between $\mathbf{y}$ and $\mathbf{A}$ under the i.i.d Gaussian-Laplacian noise assumption as,

$$d_{LSS}(\mathbf{y}; \mathbf{A}) = \min_{\mathbf{x},\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \mathbf{Ax} - \mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_1. \tag{14}$$

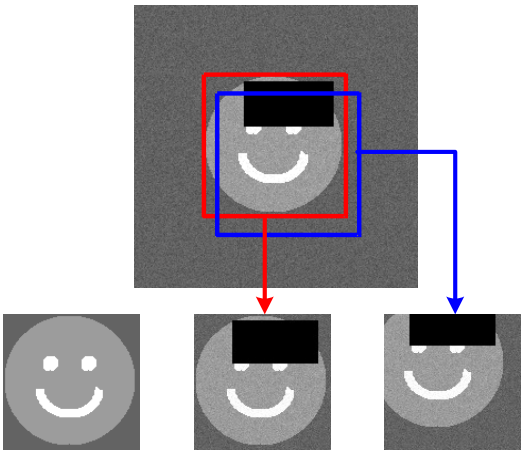
As it is related to the Least Soft-thresold Squares (LSS) regression, we denote it as the LSS distance.

Figure 2 illustrates a toy example of good and bad candidates for template matching with partial occlusion. For simplification, we merely use a single template shown in Figure 2(a). Figure 2(b) and (c) show good and bad candidates (shown in red and blue boxes respectively). In Table 2, we report the OLS distance and the LSS distance between the template and different candidates. We denote the

Table 2. The OLS and LSS distances between the template and different candidates of Figure 2.

	d_{OLS}	d_{LSS} ($\lambda = 0.05$)	d_{LSS} ($\lambda = 0.1$)
Good Candidate	27.78	5.83	10.88
Bad Candidate	25.51	7.21	12.51

template as $\mathbf{t}$, the good candidate as $\mathbf{y}_G$ and the bad candidate as $\mathbf{y}_B$ respectively. In this example, $d_{OLS}(\mathbf{y}_B; \mathbf{t})$ is smaller than $d_{OLS}(\mathbf{y}_G; \mathbf{t})$, which means the bad candidate is picked if the OLS distance is used. On the other hand, the good candidate is selected ($d_{LSS}(\mathbf{y}_G; \mathbf{t}) < d_{LSS}(\mathbf{y}_B; \mathbf{t})$) when the proposed LSS distance is used. Thus, we note that the proposed LSS distance is better than the OLS distance for handling outliers (e.g., partial occlusion).



(a) Template **(b) Good Candidate** **(c) Bad Candidate**

Figure 2. A toy example of good and bad candidates for template matching.

3. Least Soft-thresold Squares Tracking

In this paper, visual tracking is treated as a dynamic Bayesian inference task with a hidden Markov model. Given a set of observed image vectors $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t\}$ up to the t-th frame, the aim is to estimate the target state variable $\mathbf{x}_t$ by using the maximum a posteriori estimation,

$$\widehat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t^i} p\left(\mathbf{x}_t^i | \mathbf{y}_{1:t}\right), \tag{15}$$

where $\mathbf{x}_t^i$ indicates the i-th sample of the state $\mathbf{x}_t$. Based on the Bayes theorem, the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be estimated recursively by,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \mathbf{x}_{t-1}, \tag{16}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the motion model that describes the state transition between consecutive frames, and $p(\mathbf{y}_t|\mathbf{x}_t)$ is the observation model that estimates the likelihood of an observed image patch belonging to the object class. The affine motion model is used in this work and the state transition is formulated by random walk, i.e., $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a diagonal covariance matrix that indicates the variances of affine parameters.

Observation model: In this paper, we assume that the tracked target object is generated by a PCA subspace (spanned by $\mathbf{U}$ and centered at $\boldsymbol{\mu}$) with i.i.d Gaussian-Laplacian noise

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{U}\mathbf{z} + \mathbf{n} + \mathbf{s}, \tag{17}$$

where $\mathbf{y}$ denotes an observation vector, $\mathbf{U}$ represents a matrix of column basis vectors, $\mathbf{z}$ indicates the coefficients of basis vectors, $\mathbf{n}$ is the Gaussian noise component and $\mathbf{s}$ is the Laplacian noise component.

Based on the discussion in Section 2, under the i.i.d Gaussian-Laplacian noise assumption, the distance between the vector $\mathbf{y}$ and the subspace $(\mathbf{U}, \boldsymbol{\mu})$ is the least soft-threshold squares distance,

$$d(\mathbf{y}; \mathbf{U}, \boldsymbol{\mu}) = \min_{\mathbf{z},\mathbf{s}} \frac{1}{2} \|\overline{\mathbf{y}} - \mathbf{U}\mathbf{z} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1, \tag{18}$$

where $\overline{\mathbf{y}} = \mathbf{y} - \boldsymbol{\mu}$. Thus, for each observation $\mathbf{y}^i$ corresponding to a predicted state $\mathbf{x}^i$, we firstly solve the following optimization problem,

$$\left[\widehat{\mathbf{z}^i}, \widehat{\mathbf{s}^i}\right] = \arg\min_{\mathbf{z}^i,\mathbf{s}^i} \frac{1}{2} \left\|\overline{\mathbf{y}^i} - \mathbf{U}\mathbf{z}^i - \mathbf{s}^i\right\|_2^2 + \lambda \left\|\mathbf{s}^i\right\|_1, \tag{19}$$

where i denotes the i-th sample of the state $\mathbf{x}$ (without loss of generality, we drop the frame index t). As the PCA basis vectors $\mathbf{U}$ is orthogonal, the per-computed matrix $\mathbf{P}$ can be simply set to $\mathbf{U}^\top$. After the optimal $\widehat{\mathbf{z}^i}$ and $\widehat{\mathbf{s}^i}$ are obtained, the least soft-threshold squares distance can be calculated by $d\left(\mathbf{y}^i; \mathbf{U}, \boldsymbol{\mu}\right) = \frac{1}{2} \left\|\overline{\mathbf{y}^i} - \mathbf{U}\widehat{\mathbf{z}^i} - \widehat{\mathbf{s}^i}\right\|_2^2 + \lambda \left\|\widehat{\mathbf{s}^i}\right\|_1$. Then the observation likelihood can be measured by

$$p\left(\mathbf{y}^i|\mathbf{x}^i\right) = \exp\left(-\gamma d\left(\mathbf{y}^i; \mathbf{U}, \boldsymbol{\mu}\right)\right), \tag{20}$$

where γ is a constant controlling the shape of the Gaussian kernel.

Model Update: We note that the non-zero components in the Laplacian noise term can be used to identify outliers. Thus, we present a simple yet effective update scheme. After obtaining the best candidate state of each frame, we extract its corresponding observation vector $\mathbf{y}_o = \left[y_o^1; y_o^2; ...; y_o^d\right]$ and infer the Laplacian noise term $\mathbf{s}_o = \left[s_o^1; s_o^2 ...; s_o^d\right]$. Then we reconstruct the observation vector

by replacing the outliers with its corresponding parts of the mean vector $\boldsymbol{\mu}$,

$$y_r^i = \begin{cases} y_o^i, & s_o^i = 0 \\ \mu^i, & s_o^i \neq 0 \end{cases}, \tag{21}$$

where $\mathbf{y}_r = [y_r^1; y_r^2; ...; y_r^d]$ denotes the reconstructed vector and $\boldsymbol{\mu} = [\mu_r^1; \mu_r^2; ...; \mu_r^d]$ is the mean vector. The reconstructed sample is cumulated and then used to update the tracker (i.e., PCA basis vectors $\mathbf{U}$ and the mean vector $\boldsymbol{\mu}$) by using an incremental principal component analysis (PCA) method [20].

4. Experiments

The proposed tracker is implemented in MATLAB and runs at 3.5 frames per second on a PC with Intel i7-3770 CPU (3.4 GHz) with 32 GB memory. The regularization constant λ is set to 0.1 in all experiments. For each sequence, the location of the tracked target is manually labeled in the first frame. We resize each image observation to 32×32 pixels and use 16 eigenvectors for PCA representation. As a trade-off between effectiveness and speed, 600 particles are adopted and our tracker is incrementally updated every 5 frames. The MATLAB source codes, datasets and supplementary materials are available on our websites (http://ice.dlut.edu.cn/lu/publications.html, http://faculty.ucmerced.edu/mhyang/pubs.html).

In this work, we use fifteen challenging image sequences from prior work [20, 19, 4, 15, 29] and the CAVIAR data set (http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/). The challenging factors of these sequences include partial occlusion, illumination variation, pose change, background clutter and motion blur. We evaluate the proposed tracker against ten state-of-the-art algorithms, including the FragT [1], IVT [20], MIL [4], VTD [15], TLD [14], APGL1 [5], MTT [31], LSAT [17], SCM [32], ASLSA [13] and OSPT [26] trackers. For fair evaluation, we use the source codes provided by the authors and run them with adjusted parameters.

4.1. Quantitative Evaluation

We evaluate the above-mentioned algorithms using two criteria: the center location error and the overlap rate. Table 3 reports the average center location errors in pixels, where a smaller average error means a more accurate result. In addition, we use the segmentation criterion in the PASCAL VOC challenge [9] to evaluate the overlap rate. Given the tracking result (bounding box) of each frame R_T and the corresponding ground truth bounding box R_G, the overlap score is defined as $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. Table 4 reports the average overlap rates, where larger average scores mean more accurate results.

4.2. Qualitative Evaluation

Severe Occlusion: We test several sequences (*Occlusion1, Occlusion2, Caviar1, Caviar2, Caviar3, DavidOutdoor*) with heavy or long-time partial occlusion, scale change and rotation. Figure 3 (a-c) demonstrate that the proposed method performs well in terms of position, rotation and scale when the target undergoes severe occlusion. This can be attributes to two reasons: (1) the proposed LSS distance takes outliers (e.g., occlusion) into account explicitly; and (2) the update scheme is able to avoid degrading the observation model by removing the outliers from new observed samples. In addition, the SCM and ASLAS methods also achieve good performance in most cases as both of them include part-based representations with overlapping patches. The IVT method is sensitive to partial occlusion (*Occlusion2, Caviar1, Caviar3, DavidOutdoor*) since the OLS distance is not effective to handle outliers. The MIL and TLD methods do not perform well when the target object is occluded by a similar object (*Caviar1, Caviar2, Caviar3*). This can be explained by that the rectangle features they adopted (generalized Haar-liker features or binary patterns) are less effective when similar objects occlude each other.

Illumination Change: Figure 3 (d) shows the tracking results in the sequences (*DavidIndoor, Car4, Singer1*) with significant illumination variation and both scale and pose change. We can see that the MIL and TLD methods are less effective in these cases (e.g., *DavidIndoor* #0350 and *Singer1* #0090). Due to the use of incremental PCA algorithm, the proposed tracker achieves good performance in dealing with the appearance change caused by light change. For the same reason, the IVT and ASLSA methods also perform well.

Background Clutter: Figure 3 (e) demonstrates the tracking results in the *Car11, Deer* and *Football* sequences with background clutter. These videos also pose other challenging factors including illumination variation (*Car11*), fast motion (*Deer*) and partial occlusion (*Football*). As the proposed LSS distance encourages good matching results when outliers occur, our tracker performs better than other methods in these videos (e.g., *Deer* #0052 and *Football* #0315).

Fast Motion: Figure 3 (f) illustrates the tracking results on the *Jumping*, *Owl* and *Face* sequences. It is difficult to predict the locations of the tracked objects when they undergo abrupt motion. Furthermore, the appearance change caused by motion blur poses great challenges for capturing the tracked targets accurately and updating the observation models properly. We can see that the TLD and proposed methods perform better than other algorithms (e.g., *Owl* #0248 and *Face* #0254). We note that the TLD method is equipped with a re-initialization mechanism which facilitates object tracking.

Table 3. Average center location error (in pixels). The best three results are shown in red, blue, and green fonts.

Sequence	FragT	IVT	MIL	VTD	TLD	APGL1	MTT	LSAT	SCM	ASLAS	OSPT	LSST(Ours)
Occlusion1	5.6	9.2	32.3	11.1	17.6	6.8	14.1	5.3	3.2	10.8	4.7	5.3
Occlusion2	15.5	10.2	14.1	10.4	18.6	6.3	9.2	58.6	4.8	3.7	4.0	3.1
Caviar1	5.7	45.2	48.5	3.9	5.6	50.1	20.9	1.8	0.9	1.4	1.7	1.4
Caviar2	5.6	8.6	70.3	4.7	8.5	63.1	65.4	45.6	2.5	62.3	2.2	2.3
Caviar3	116.1	66.0	100.2	58.2	44.4	68.6	67.5	55.3	2.2	2.2	45.7	3.1
DavidOutdoor	90.5	53.0	38.4	61.9	173.0	233.4	65.5	101.7	64.1	87.5	5.8	6.4
DavidIndoor	148.7	3.1	34.3	49.4	13.4	10.8	13.4	6.3	3.4	3.5	3.2	4.3
Singer1	22.0	8.5	15.2	4.1	32.7	3.1	41.2	14.5	3.7	5.3	4.7	3.5
Car4	179.8	2.9	60.1	12.3	18.8	16.4	37.2	3.3	3.5	4.3	3.0	2.9
Car11	63.9	2.1	43.5	27.1	25.1	1.7	1.8	4.1	1.8	2.0	2.2	1.6
Deer	92.1	127.5	66.5	11.9	25.7	38.4	9.2	69.8	36.8	8.0	8.5	10.0
Football	16.7	18.2	16.0	4.1	11.8	12.4	6.5	14.1	10.4	18.0	33.7	7.6
Jumping	58.4	36.8	9.9	63.0	3.6	8.8	19.2	55.2	3.9	39.1	5.0	4.8
Owl	148.0	141.4	148.9	86.8	8.2	104.2	184.3	110.7	7.3	7.6	47.4	6.2
Face	48.8	69.7	134.7	141.4	22.3	148.9	127.2	16.5	125.1	95.1	24.1	12.3
Average	67.8	40.2	55.5	36.7	28.6	51.5	45.5	37.5	18.2	23.4	13.1	5.0

Table 4. Average overlap rate. The best three results are shown in red, blue, and green fonts.

Sequence	FragT	IVT	MIL	VTD	TLD	APGL1	MTT	LSAT	SCM	ASLAS	OSPT	LSST(Ours)
Occlusion1	0.90	0.85	0.59	0.77	0.65	0.87	0.79	0.90	0.93	0.83	0.91	0.89
Occlusion2	0.60	0.59	0.61	0.59	0.49	0.70	0.72	0.33	0.82	0.81	0.84	0.86
Caviar1	0.68	0.28	0.25	0.83	0.70	0.28	0.45	0.85	0.91	0.90	0.89	0.89
Caviar2	0.56	0.45	0.26	0.67	0.66	0.32	0.33	0.28	0.81	0.35	0.71	0.80
Caviar3	0.13	0.14	0.13	0.15	0.16	0.13	0.14	0.58	0.87	0.82	0.25	0.85
DavidOutdoor	0.39	0.52	0.41	0.42	0.16	0.05	0.42	0.36	0.46	0.45	0.77	0.76
DavidIndoor	0.09	0.69	0.23	0.23	0.50	0.63	0.53	0.72	0.75	0.77	0.76	0.75
Singer1	0.34	0.66	0.34	0.79	0.41	0.83	0.32	0.52	0.85	0.78	0.82	0.80
Car4	0.22	0.92	0.34	0.73	0.64	0.70	0.53	0.91	0.89	0.89	0.92	0.92
Car11	0.09	0.81	0.17	0.43	0.38	0.83	0.58	0.49	0.79	0.81	0.81	0.84
Deer	0.08	0.22	0.21	0.58	0.41	0.45	0.60	0.35	0.46	0.62	0.61	0.58
Football	0.57	0.55	0.55	0.81	0.56	0.68	0.71	0.63	0.69	0.57	0.62	0.69
Jumping	0.14	0.28	0.53	0.08	0.69	0.59	0.30	0.09	0.73	0.24	0.69	0.65
Owl	0.09	0.22	0.09	0.12	0.60	0.17	0.09	0.13	0.79	0.78	0.48	0.81
Face	0.39	0.44	0.15	0.24	0.62	0.14	0.26	0.69	0.36	0.21	0.68	0.76
Average	0.35	0.51	0.32	0.50	0.51	0.49	0.45	0.52	0.74	0.66	0.72	0.79

5. Conclusion

In this paper, we propose a Least Soft-thresold Squares (LSS) regression method that assumes the noise is Gaussian-Laplacian distributed, and apply it to object tracking. We present an efficient iteration algorithm to solve the LSS problem, which achieves the global minimal solution. We derive a LSS distance to measure the difference between an observation sample and the dictionary. The LSS distance is effective in handling outliers and therefore provides an accurate match, which facilitates object tracking (e.g., in dealing with partial occlusion). In addition, we develop a robust generative tracker based on the proposed LSS method and a simple update scheme. Both quantitative and qualitative evaluations on challenging image sequences show that the proposed tracker performs favorably against several state-of-the-art algorithms. In the future, we will extend the LSS method to solve other vision problems (e.g., face recognition).

Acknowledgements: D. Wang and H. Lu are supported by the National Natural Science Foundation of China #61071209 and #61272372. M.-H. Yang is supported by the US National Science Foundation CAREER Grant #1149783 and IIS Grant #1152576.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] S. Avidan. Ensemble tracking. *TPAMI*, 29(2):261, 2007.

[3] A. Ayvaci, M. Raptis, and S. Soatto. Occlusion detection and motion estimation with convex optimization. In *NIPS*, pages 100–108, 2010.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[5] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust ℓ_1 tracker using accelerated proximal gradient approach. In *CVPR*, pages 1830–1837, 2012.

[6] I. Barrodale and F. D. K. Roberts. An improved algorithm for discrete ℓ_1 linear approximation. *SIAM Journal on Numerical Analysis*, 10(5):839–848, 1973.

[7] M. J. Black. EigenTracking : Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *IJCV*, 26(1):63–84, 1998.

[8] Z. Chen, J. Wang, and Y. Wu. Decomposing and regularizing sparse/non-sparse components for motion field estimation. In *CVPR*, pages 1776–1783, 2012.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[10] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006.

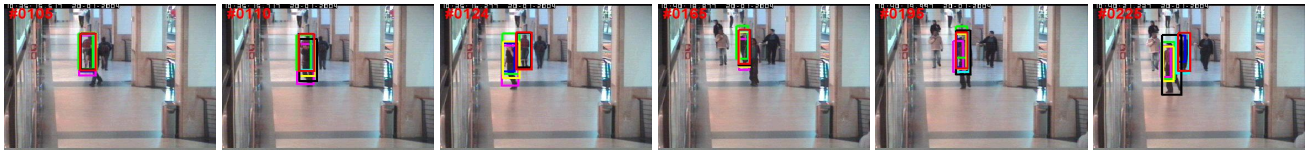
[11] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270, 2011.

[12] W. Hu, X. Li, X. Zhang, X. Shi, S. J. Maybank, and Z. Zhang. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *IJCV*, 91(3):303–327, 2011.

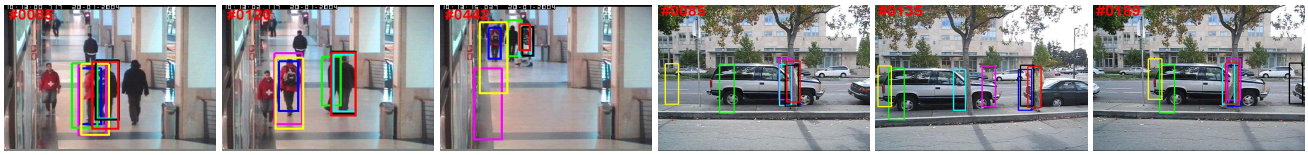
[13] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012.

[14] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2012.

(a) Tracking results on sequence *Occlusion1* and *Occlusion2* with heavy occlusion and in-plane rotation.

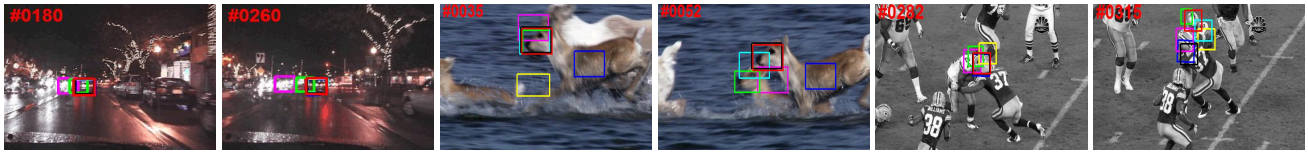


(b) Tracking results on sequence *Caviar1* and *Caviar2* with partial occlusion and scale change.



(c) Tracking results on sequence *Caviar3* and *DavidOutdoor* with severe occlusion, scale change and pose variation.

(d) Tracking results on sequence *DavidIndoor*, *Car4* and *Singer1* with illumination variation.



(e) Tracking results on sequence *Car11*, *Deer* and *Football* with background clutter.

(f) Tracking results on sequence *Jumping*, *Owl* and *Face* with abrupt motion.

Figure 3. Sample tracking results on fifteen challenging image sequences. This figure demonstrates the results of the IVT [20], MIL [4], TLD [14], APGL1 [5], SCM [32], ASLAS [13] and the proposed methods. More results can be found in the supplementary material.

[15] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[16] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental log-Euclidean Riemannian subspace learning. In *CVPR*, pages 1–8, 2008.

[17] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, pages 1313–1320, 2011.

[18] H. Lu, S. Lu, D. Wang, S. Wang, and H. Leung. Pixel-wise spatial pyramid-based hybrid tracking. *TCSVT*, 22(9):1365–1376, 2012.

[19] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. In *ICCV*, pages 1436–1443, 2009.

[20] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[21] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, pages 723–730, 2010.

[22] E. J. Schlossmacher. An iterative technique for absolute deviations curve fitting. *JOSA*, 68(344):857–859, 1973.

[23] I. W. Selesnick. The estimation of laplace random vectors in additive white gaussian noise. *TSP*, 56(8-1):3482–3496, 2008.

[24] D. Wang and H. Lu. Object tracking via 2DPCA and ℓ_1-regularization. *SPL*, 19(11):711–714, 2012.

[25] D. Wang and H. Lu. On-line learning parts-based representation via incremental orthogonal projective non-negative matrix factorization. *SP*, 93:1608–1623, 2013.

[26] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *TIP*, 22(1):314–325, 2013.

[27] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011.

[28] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2):210–227, 2009.

[29] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *ICCV*, pages 1100–1107, 2011.

[30] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV*, pages 864–877, 2012.

[31] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049, 2012.

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