Robust visual tracking using structurally random projection and weighted least squares

Zhang, S., Zhou, H., Jiang, F., & Li, X. (2015). Robust visual tracking using structurally random projection and weighted least squares. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(11), 1749-1760. https://doi.org/10.1109/TCSVT.2015.2406194

Published in:
IEEE Transactions on Circuits and Systems for Video Technology

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Robust visual tracking using structurally random projection and weighted least squares

Shengping Zhang, *Member, IEEE*, Huiyu Zhou, Feng Jiang, *Member, IEEE*, Xuelong Li, *Fellow, IEEE*

***Abstract*—Sparse representation based visual tracking approaches have attracted increasing interests in the community in recent years. The main idea is to linearly represent each target candidate using a set of target and trivial templates while imposing a sparsity constraint onto the representation coefficients. After we obtain the coefficients using L1-norm minimization methods, the candidate with the lowest error, when it is reconstructed using only the target templates and the associated coefficients, is considered as the tracking result. In spite of promising system performance widely reported, it is unclear if the performance of these trackers can be maximised. In addition, computational complexity caused by the dimensionality of the feature space limits these algorithms in real-time applications. In this paper, we propose a real-time visual tracking method based on structurally random projection and weighted least squares techniques. In particular, to enhance the discriminative capability of the tracker, we introduce background templates to the linear representation framework. To handle appearance variations over time, we relax the sparsity constraint using a weighed least squares (WLS) method to obtain the representation coefficients. To further reduce the computational complexity, structurally random projection is used to reduce the dimensionality of the feature space while preserving the pairwise distances between the data points in the feature space. Experimental results show that the proposed approach outperforms several state-of-the-art tracking methods.**

***Index Terms*—Visual tracking, sparse representation, structural random projection, weighted least squares.**

I. Introduction

VISUAL tracking provides a means to estimate the state of a specific target in an image sequence. There is an overwhelming need for its applications in multiple research fields, including intelligent video surveillance, human computer interaction and robot navigation, where visual tracking has demonstrated its value and importance in the past few decades [1]–[21].

Visual tracking is usually formulated as a search task where an appearance model is firstly used to represent the target to be tracked in a previous frame and then a search strategy is utilized to find the state of the target in the current frame. Therefore, how to effectively model the appearance of the target and how to accurately calculate its state are two key steps in a successful tracking system. Although a variety of tracking algorithms have been developed in the past few decades, the performance of these visual tracking methods barely meet the minimum requirements of real applications. The major challenge of visual tracking is that it is very difficult to design a powerful appearance model which should not only discriminate the target from its background but also be robust against appearance variations of the target over time. To improve the discriminative ability, some promising approaches have been proposed considering visual tracking as a two-class classification or detection problem. Many elegant features in pattern recognition can be used to effectively discriminate the target from its background. However, it is hard to obtain an approach immune to target appearance variations such as pose changes, shape deformation, illumination changes, and partial occlusion.

Traditional appearance representation methods rely on various features obtained either by hand-designing [2], [22], [23] or learning from data [24]–[27]. In spite of certain discriminative abilities, these appearance representation methods cannot maintain the desired tracking performance at all times. Recently, sparse representation based tracking methods [28]–[31] (refer to [32] for a comprehensive review) have attracted increasing interests due to its robustness against appearance variations. These methods are used to linearly represent each target candidate using a set of target templates and trivial templates (the column vectors of an identity matrix) with a sparsity constraint made to the representation coefficients. After obtaining the coefficients via a ℓ_1-norm minimization method, we can obtain the reconstruction error for each candidate, which is calculated using the target templates and the corresponding coefficients. The candidate with the lowest reconstruction error is considered as the tracking result. Although positive performance has been reported, it is unclear if the sparsity constraint can make the tracking performance better. Because the trivial templates are capable of representing any image, a large number of trivial templates will be activated in the linear representation, which violates the sparsity assumption of the representation coefficients. On the other hand, extensive computational costs caused by solving ℓ_1-norm minimization

This work was supported in part by the National Natural Science Foundation of China (No. 61300111, No. 61100096 and No.61125106), the China Postdoctoral Science Foundation (No. 2014M550192), the Research Fund for the Doctoral Program of Higher Education of China (No. 20132302120084), the Key Research Program of the Chinese Academy of Sciences (Grant No. KGZD-EW-T03). H. Zhou is in part supported by UK EPSRC under Grant EP/H049606/1.

S. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, Shandong, P. R. China. E-mail: s.zhang@hit.edu.cn

H. Zhou is with the Institute of Electronics, Communications and Information Technology, Queen's University of Belfast, Belfast, BT3 9DT, United Kingdom. E-mail: h.zhou@ecit.qub.ac.uk

F. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, Heilongjiang, P. R. China. E-mail: fjiang@hit.edu.cn

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China. Email: xuelong_li@opt.ac.cn

limit the use of these trackers in real-time applications.

To improve the performance of the existing sparse representation based tracking methods, in this paper, we propose a real-time visual tracking method based on the combination of structurally random projection and weighted least squares. To enhance the discriminative ability of the proposed tracker, we introduce a set of additional background templates to the linear representation framework. To make our method robust against appearance variations during tracking, we release the sparsity constraint using weighed least squares (WLS) to solve the linear representation problem. Another advantage of using WLS is that it has an analytic solution, which enables the proposed tracking method to work in real-time. To further reduce the computational complexity, structurally random projection is used to reduce the feature dimensionality while preserving the pairwise distances between the data points in the feature space.

The contribution of the proposed method is three-fold: **1**) The weighed least squares method releases the sparsity constraint imposed by the traditional sparse representation methods and achieves sufficient robustness against appearance variations. **2**) By introducing background templates to the linear representation framework, we are capable of discriminating the target to be tracked from its background. **3**) The dimensionality of the feature representation is significantly reduced using structurally random projection. In the meantime, the pairwise distances between the data points in the feature space are kept. All of these aspects make our tracker perform well in real-time.

The rest of the paper is organized as follows. In Section II, we review the related work reported in the literature. Section III gives a detailed description of the proposed method. Experimental results are reported and analyzed in Section IV. We conclude this paper in Section V.

II. Related work

Inspired by the success of sparse representation in face recognition [33], recently, sparse representation based visual tracking becomes overwhelming [28], [31], [32], [34]–[36]. The first sparse representation based tracking method was presented in [28], which is implemented under the widely used particle filter framework [37], [38] and represents each target candidate (corresponding to a particle) $\mathbf{y}$ using a set of target and trivial templates. Let $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_f}] \in \mathbb{R}^{d \times n_f}$ and $\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_d] \in \mathbb{R}^{d \times d}$ be the target and trivial template sets, respectively. The target templates are manually obtained at the first frame and then updated in an online style over time. The trivial templates have the same size as the target templates but only have one non-zero element in each template.[1] The linear representation can be written in a matrix form as

$$\mathbf{y} = \mathbf{F}\boldsymbol{\alpha}_F + \mathbf{I}\boldsymbol{\alpha}_I = [\mathbf{F}, \mathbf{I}] \begin{bmatrix} \boldsymbol{\alpha}_F \\ \boldsymbol{\alpha}_I \end{bmatrix} \doteq \mathbf{X}\boldsymbol{\alpha} \tag{1}$$

where $\boldsymbol{\alpha}_F \in \mathbb{R}^{n_f}$ and $\boldsymbol{\alpha}_I \in \mathbb{R}^d$ are coefficients associated with the target and the trivial templates, respectively. Mei *et al.* [28] assumed that if the candidate $\mathbf{y}$ is the tracking result, it should be in the subspace spanned by all the target templates. Therefore, the coefficient vector $\boldsymbol{\alpha}$ is sparse and can be obtained by solving the following ℓ_1-norm minimization problem

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{2}$$

where λ is the regularization parameter that controls the importance of the sparsity constraint. The weight of the i-th target candidate can be computed as

$$w_i = \exp(-\frac{\|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha}_F\|_2^2}{\delta}) \tag{3}$$

where δ is a parameter that controls the shape of the exponential function.

Due to the use of ℓ_1-norm minimization, Mei *et al.*'s method is called as a ℓ_1 tracker. Although good performance has been reported in [28], there are two areas in their approach that can be further improved. The first one is the unreasonable sparsity assumption related to the representation coefficients. Because the trivial templates are capable of representing any image, when the candidate $\mathbf{y}$ is background, a large number of trivial templates will be activated, which has been witnessed in our experiments. In this case, the sparsity assumption of the representation coefficients does not hold. In [31], to avoid the activation of the trivial templates, the ℓ_2- norm constraint on the coefficients corresponding to the trivial templates is introduced

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \gamma\|\boldsymbol{\alpha}_I\|_2 \tag{4}$$

where γ is set to be a small constant when the target is not occluded and zero otherwise. The introduction of $\|\boldsymbol{\alpha}_I\|_2$ in the objective function can make the target image well approximated by a sparse linear combination of the target templates with a small residual, which therefore causes a larger weight to be assigned to the target image. Although this modification can be used to eliminate the effect of the sparsity assumption to some extent and therefore improve the tracking performance, it still activates the trivial templates when the candidate is background, which will lead to non-sparse representation coefficients. To overcome this problem, Zhang *et al.* [35] proposed to use a learned basis to replace the trivial templates. This basis is learned in order to produce a sparse representation for the difference between the candidate and the target templates.

The other drawback of the ℓ_1 tracker is that solving the Eq. (2) is a time-consuming process. If the preconditioned conjugate gradients (PCG) [39] are adopted to solve the L1-norm minimization problem, the run time is determined by the product of the total number of all the PCG steps with all the iterations and the cost of each PCG step. The total number of the PCG iterations depends on the value of the regularization parameter λ. In the experiments with $\lambda = 0.15$, the total number of PCG is approximately a few hundred times. For a PCG step, the most expensive operator is a matrix-vector product which has $\mathcal{O}(d^2 + d \times n)$ computational complexity, where d is the feature dimensionality and n is the number of the templates. Motivated by the sparse signal recovery power of compressive sensing, Li *et al.* [30] accelerated the ℓ_1-norm

[1] See Figure 1 in [28] for an illustration example.

minimization by reducing the feature dimensionality using a hash table or random projection which meets the Restricted Isometry Property (RIP) [40]. Let $\mathbf{\Phi} \in \mathbb{R}^{\tilde{d} \times d}$ be the projection matrix, the coefficients $\boldsymbol{\alpha}$ can be computed by

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{\Phi}\mathbf{y} - \mathbf{\Phi}\mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{5}$$

When we set $\tilde{d} \ll d$, the dimensionality of the ℓ_1 minimization is significantly reduced while the original high dimensionality $\mathbf{y}$ can still be fully recovered from the reduced $\mathbf{\Phi}\mathbf{y}$.

To compute the coefficients shown in Eq. (2), we need computationally expensive ℓ_1-norm minimization. However, the particle weights defined in Eq. (3) generate a reconstruction error measured in ℓ_2-norm, which has a lower bound $\|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha}_F\|_2^2 \geq \|\mathbf{y} - \mathbf{F}\widehat{\boldsymbol{\alpha}}_F\|_2^2$, where

$$\widehat{\boldsymbol{\alpha}}_F = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha}_F\|_2^2 \tag{6}$$

Instead of reducing the computational complexity of the ℓ_1-norm minimization, Mei *et al.* [29] proposed to reduce the number of ℓ_1-norm minimization by excluding unimportant particles using the reconstruction error bound computed via fast ℓ_2-norm minimization shown in Eq. (6).

The aforementioned methods employ sparse representation to globally encode each target candidate through the target templates. In the literature, there are also different kinds of methods [34], [41] which used local sparse representation to model target appearance. These methods first construct a dictionary from the local patches sampled from the training images that contain the tracked target and then use the dictionary to encode local patches sampled from each target candidate or template. The coding coefficients are used as features to describe the appearance of the target candidate or template. However, due to the locality of the sampling, these appearance modeling methods have a poor discriminative ability. To overcome this disadvantage, Zhong et al. [42] integrated both local and global sparse appearance models. In [43], [44], structural sparse appearance modeling was proposed, which exploited the spatial layout of the locally sampled patches to increase the discriminative ability. In [45], a more sophisticated method was proposed, where discriminative sparse coding was directly used to enhance the discriminative power of the resulting coding coefficients.

III. Proposed Method

In this section, we present the proposed tracking method based on structurally random mapping and weighted least squares. In contrast to ℓ_1 trackers which only use target templates, our proposed framework uses both target and background templates to represent each candidate. When the total reconstruction error is minimized, the target and the background templates compete against each other in the linear representation. After reducing the feature dimensionality using structurally random mapping, we compute the representation coefficients by the weighted least squares technique. The reconstruction errors obtained by the target and the background templates are used to discriminate the target from its background. An overview of the proposed tracking method is shown in Fig. 1

A. The tracking framework

The proposed method is implemented using a sequential importance sampling (also known as particle filter) framework [37], [38], which is a popular computation method to recursively approximate the posterior distribution of state variables characterizing a dynamic system. It consists of two stages: prediction and updating. Let $\mathbf{z}_t$ and $\mathbf{I}_t$ be the state variables and the observation at time t, respectively. The posterior distribution of $\mathbf{z}_t$ given all the available observations $\mathbf{I}_{1:t-1} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{t-1}\}$ up to time $t-1$ can be predicated using the state transition model $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ as

$$p(\mathbf{z}_t|\mathbf{I}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{z}_{t-1}|\mathbf{I}_{1:t-1})d\mathbf{z}_{t-1} \tag{7}$$

At time t, the observation $\mathbf{I}_t$ is available, and the posterior distribution of $\mathbf{z}_t$ is updated using the Bayes rule as

$$p(\mathbf{z}_t|\mathbf{I}_{1:t}) = \frac{p(\mathbf{I}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{I}_{1:t-1})}{p(\mathbf{I}_t|\mathbf{I}_{1:t-1})} \tag{8}$$

Using the sequential importance sampling technique, the posterior distribution $p(\mathbf{z}_t|\mathbf{I}_{1:t})$ is approximated by a set of N weighted samples (also called particles) $\{\mathbf{z}_t^i, w_t^i\}_{i=1,\ldots,N}$, where w_t^i are the importance weights of particles $\mathbf{z}_t^i$. Let $q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})$ be the importance distribution from which the particles are drawn, the importance weights of $\mathbf{z}_t^i$ are updated as

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{I}_t|\mathbf{z}_t^i)p(\mathbf{z}_t^i|\mathbf{z}_{t-1}^i)}{q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})} \tag{9}$$

To avoid the degeneracy case where the weights of some particles may keep increasing for no reason, particles are re-sampled according to their importance weights so as to generate a set of equally weighted particles. In case a bootstrap filter is applied [37], where the state transition distribution is chosen as the importance distribution $q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{z}_{t-1})$, the weights are updated using the observation likelihood $w_t^i = p(\mathbf{I}_t|\mathbf{z}_t^i)$.

Particle filter is firstly used for contour tracking in [46]. Pérez *et al.* [38] used particle filtering for tracking targets parameterized within a rectangle region, *e.g.*, using color histogram to describe the states. The key step of the particle filter for visual tracking is to compute the weight for each particle using the observation likelihood. In practice, the observation likelihood $p(\mathbf{I}_t|\mathbf{z}_t^i)$ is computed as the similarity between the target template and the target candidate parameterized by the particle $\mathbf{z}_t^i$ using the appearance models. In the next subsection, we will present how to use our appearance model with a multi-scale pyramid matching to assign a proper weight to each particle.

State transition model: In this work, we adopt the six parameters of an affine transformation to model the target state $\mathbf{z} = (x, y, \theta, \zeta, \rho, \tau)$ which denote horizontal and vertical translations, rotation angles, scales, aspect ratios and skew directions. Using these affine parameters, we can crop a sub-image from the current image and then normalize it to the size $w \times h$. To sample particles, we adopt the second-order autoregressive dynamical model [47] $\mathbf{z}_t \sim \mathcal{N}(g(\mathbf{z}_{t-1}, \mathbf{z}_{t-2}), \mathbf{\Sigma})$, where $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is the norm distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$, $g(\mathbf{z}_{t-1}, \mathbf{z}_{t-2}) = c_1\mathbf{z}_{t-1} + c_2\mathbf{z}_{t-2}$,where two

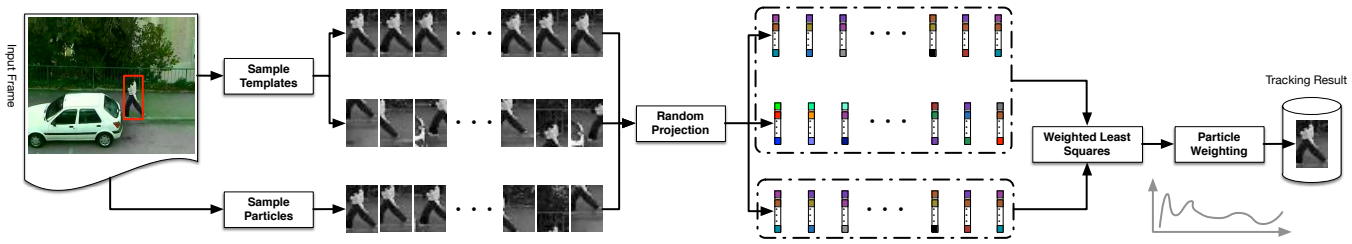


Fig. 1. Overview of the proposed tracking method.

constants c_1 and c_2 are used to define a constant acceleration model. After we sample the candidates, each candidate is assigned a weight based on the appearance modeling. In the following section, we will introduce how an observation likelihood is computed based on the sparse representation.

Discriminative observation model: Comparing to the existing L1 trackers, the first contribution of the proposed method is the use of the background templates in the linear representation form to replace the trivial templates. The motivation of this practice is two-folds: As we discussed before, the trivial templates in the L1 trackers will be activated when they are used to represent the background, leading to non-sparse representation. The use of the background templates in the linear representation form can help solve this problem because these background templates can be used to reconstruct the background candidate with minimal errors, compared to the trivial templates. It should be noted that the objective function is used to minimize the reconstruction error instead of the sparsity of the coefficients as the L1 trackers do. The other advantage of using background templates is that both the target and the background templates in the linear representation form can be used to discriminate the target candidates from the background candidates, when we design a reasonable discriminative function based on the reconstruction errors and the coefficients.

At the first frame $t = 1$, the state of the target is manually labeled. Let $(x_0, y_0, \theta_0, \zeta_0, \rho_0, \tau_0)$ be the initial state of the target. We assume that both the target templates and the background templates just have different center coordinates with individual corresponding initial states. Let $(x_f^{(i)}, y_f^{(i)})$ be the center coordinates of the i-th target template. In this work, we sample $(x_f^{(i)}, y_f^{(i)})$ using a simple Gaussian distribution with mean (x_0, y_0) and initial variance $(1, 1)$. For the i-th sampled target template, let $\mathbf{f}_i$ be the feature vector extracted from the cropped sub-image using $(x_f^{(i)}, y_f^{(i)}, \theta_0, \zeta_0, \rho_0, \tau_0)$ as affine parameters. In this work, we use the stacked pixel intensities as the features for efficiency, therefore, $\mathbf{f}_i \in \mathbb{R}^d$, where $d = w \times h$. Let $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_f}] \in \mathbb{R}^{d \times n_f}$ represent all the target templates.

To sample the background templates, we use a method similar to the above one to sample the target template but with a larger variance. In particular, we set the sampling variance as (w, h). Let $(x_B^{(i)}, y_B^{(i)})$ be the center coordinates of the i-th sampled background template. To avoid the sampled background templates of being close to the initial target, we set

$$x_B^{(i)} = \begin{cases} x - \frac{1}{8}w, & \text{if } x_B^{(i)} > x - \frac{1}{8}w \\ x + \frac{1}{8}w, & \text{if } x_B^{(i)} < x + \frac{1}{8}w \end{cases} \tag{10}$$

Similarly, we set

$$y_B^{(i)} = \begin{cases} y - \frac{1}{8}h, & \text{if } y_B^{(i)} > y - \frac{1}{8}h \\ y + \frac{1}{8}h, & \text{if } y_B^{(i)} < y + \frac{1}{8}h \end{cases} \tag{11}$$

Let $\mathbf{b}_i \in \mathbb{R}^d$ be the i-th sampled background template and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_b}] \in \mathbb{R}^{d \times n_b}$ be the background templates.

Let $\mathbf{y}$ be the feature vector of a target candidate, which can be linearly represented by both the target and the background templates

$$\mathbf{y} = \alpha_1 \mathbf{f}_1 + \ldots + \alpha_{n_f} \mathbf{f}_{n_f} + \beta_1 \mathbf{b}_1 + \ldots + \beta_{n_b} \mathbf{b}_{n_b} \tag{12}$$

Assume $\mathbf{X} = [\mathbf{F}, \mathbf{B}]$ and $\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}$. The linear system can be rewritten in a matrix form:

$$\mathbf{y} = [\mathbf{F}, \mathbf{B}] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \doteq \mathbf{X}\boldsymbol{\gamma} \tag{13}$$

Once the representation coefficients are obtained, then the weight of the i-th candidate can be computed as

$$w_i = \exp(-\frac{\varepsilon_f - \varepsilon_b}{\delta}) \tag{14}$$

where δ is a constant, $\varepsilon_f = \|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha}\|_2^2$ and $\varepsilon_b = \|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|_2^2$ are the reconstruction errors when we represent the candidate with the target and the background templates, respectively. The tracking result is the particle with the largest weight, whose index can be represented as

$$\hat{i} = \arg\max_i w_i \tag{15}$$

Since the appearance of the tracked target and the background around the target change gradually over time, both the target and the background templates should be updated during the tracking process. However, considering the inefficiency of updating the target templates in every frame, in this work, we update the background templates every 5 frames (this number is determined experimentally for the best performance).

B. Dimensionality reduction using structurally random projection

The dimensionality of the target or background template is extremely high, typically in the order of $10^3 \sim 10^5$, resulting in expensive computational costs in the tracking

stage. Therefore, it is necessary to reduce the dimensionality of the template space. Usually, the dimensionality reduction is conducted using a linear transformation, e.g., widely used PCA transformation. Let $\mathbf{\Phi} \in \mathbb{R}^{\hat{d} \times d}$ be the transformation matrix, which projects the high dimensional feature vector $\mathbf{y} \in \mathbb{R}^d$ onto a lower dimensional feature vector $\widetilde{\mathbf{y}} \in \mathbb{R}^{\hat{d}}$

$$\widetilde{\mathbf{y}} = \mathbf{\Phi}\mathbf{y} \tag{16}$$

A good transformation matrix needs to meet three requirements if not more: 1) The distance between a pair of high dimensional feature vectors can be preserved after they have been projected onto a lower dimensional space. 2) The computation should be efficient. 3) A small memory is requested. Although many dimensionality reduction methods have been proposed in the literature, they usually involve a complex training stage, which makes them less suitable for real-time tracking. Johnson and Lindenstrauss (JL) [48] stated that any set of n feature vectors in d-dimensional Euclidean space could be projected onto $\hat{d} = \mathcal{O}(\epsilon \log n)$-dimensional Euclidean space by a random matrix so that all the pairwise distances are preserved with an arbitrarily small factor ϵ. Baraniuk *et al.* [49] showed an interesting connection between the JL lemma and compressed sensing where a random matrix satisfying the JL lemma also holds true for the restricted isometry property [50]. In other words, a random matrix, $\mathbf{\Phi}$, can enable the JL lemma to project the vectors $\mathbf{y}$ in a high-dimensional space into the vector $\widetilde{\mathbf{y}}$ in a lower-dimensional space. In addition, all the pairwise distances in the high dimensional space are preserved and $\mathbf{y}$ can be recovered from $\widetilde{\mathbf{y}}$ with a minimum error.

Random projection matrices (RP) satisfying the JL lemma have been used for visual tracking in the literature. Li *et al.* [30] used RP to reduce the features' dimensionality in a ℓ_1 tracker. However, a RP matrix is usually dense, and the computational complexity and the memory requirement are formed as $\mathcal{O}(d\epsilon^{-2} \log N)$. One of the solutions to speed up the projection process is to use a sparse random matrix. Zhang *et al.* [51] used the sparse random projection (SRP) matrix proposed in [52] to compress the features' dimensionality. As the number of the nonzero entries of the SRP matrix is, on average, 3 times less than those of the dense RP matrix, the speed of the SRP is 3 times faster than that of the dense RP matrix. However, the SRP matrix cannot be further sparse without incurring a penalty to its dimensionality. To speed up the projection's computation process further, Ailon and Chazelle [53] proposed the fast JL-Transform (FJLT) matrix. The computational complexity of using the FJLT matrix is roughly $\mathcal{O}(d \log d + \epsilon^{-2} \log^3 n)$, which is much smaller than that of the RP and the SRP. When ϵ is relatively small, the FJLT matrix requires a high dimensional vector, e.g., $\|\mathbf{y}\|_\infty \leq \mathcal{O}((d/\hat{d})^{-1/2})$. In addition, although FJLT matrix is a very sparse matrix, its entries are still random and thus, a certain amount of memory size is required to store the matrix elements.

Recently, Do *et al.* proposed a fast and efficient compressive sampling method using Structurally Random Matrices (SRM) [54], [55]. A structurally random matrix $\mathbf{\Phi}$ is a product of three matrices:

$$\mathbf{\Phi} = \sqrt{\frac{d}{\hat{d}}}\mathbf{D}\mathbf{\Psi}\mathbf{R} \tag{17}$$

where $\mathbf{R}$ is a $d \times d$ random diagonal matrix whose diagonal entries R_{ii} are i.i.d Bernoulli random variables $p(R_{ii} = \pm 1) = \frac{1}{2}$, $\mathbf{\Psi}$ is a $d \times d$ orthonormal matrix where the absolute magnitudes of all the entries are on the order of $\mathcal{O}(\frac{1}{\sqrt{d}})$, and $\mathbf{D}$ is a $\hat{d} \times d$ matrix composed of nonzero rows of a random diagonal matrix with diagonal entries D_{ii} being i.i.d binary random variables and $p(D_{ii} = 1) = \frac{\hat{d}}{d}$. When we use the SRM, the projection $\mathbf{z}$ can be obtained efficiently as follows: 1) pre-randomize $\mathbf{y}$ by randomly flipping the sign of the entries of $\mathbf{y}$, 2) apply a fast transform to the randomized $\mathbf{y}$, and 3) randomly choose $\hat{d}$ as transform coefficients.

Using a structurally random matrix $\mathbf{\Phi}$, the target and background templates can be projected onto a lower dimensional space. Let $\widetilde{\mathbf{F}} = [\widetilde{\mathbf{f}}_1, \widetilde{\mathbf{f}}_2, \ldots, \widetilde{\mathbf{f}}_{n_f}] \in \mathbb{R}^{\hat{d} \times n_f}$ and $\widetilde{\mathbf{B}} = [\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2, \ldots, \widetilde{\mathbf{b}}_{n_b}] \in \mathbb{R}^{\hat{d} \times n_b}$ be the projected target and background templates, respectively, where

$$\widetilde{\mathbf{f}}_i = \mathbf{\Phi}\mathbf{f}_i, \quad \forall i \in [1, n_f] \tag{18}$$

$$\widetilde{\mathbf{b}}_i = \mathbf{\Phi}\mathbf{b}_i, \quad \forall i \in [1, n_b] \tag{19}$$

C. Obtaining coefficient using weighted least squares

The linear system (Eq. (13)) introduced above can only be used in an ideal case where the target does not undergo appearance variations that may be caused by occlusion, pose changes, illumination changes and noise. However, in practical applications, appearance variations are inevitable. Therefore, in this paper, we consider a more robust linear system described as follows

$$\mathbf{y} = \widetilde{\mathbf{X}}\boldsymbol{\gamma} + \mathbf{e} \tag{20}$$

where $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{F}}, \widetilde{\mathbf{B}}]$, $\mathbf{e} \in \mathbb{R}^{\hat{d}}$ is the representation error caused by appearance variations. Let $E[\mathbf{e}|\widetilde{\mathbf{X}}] = 0$ and $Var[\mathbf{e}|\widetilde{\mathbf{X}}] = \mathbf{\Omega}$ be the mean and variance of the error, respectively. Assume that $\mathbf{\Omega}$ is a diagonal matrix. The linear representation (Eq. (20)) is a Weighted Least Squares (WLS) problem where the representation coefficients $\boldsymbol{\gamma}$ can be derived by minimizing the squared Mahalanobis length of the residuals

$$\boldsymbol{\gamma} = \arg\min_{\boldsymbol{\gamma}} (\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\gamma})^T \mathbf{\Omega}^{-1/2} (\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\gamma}) \tag{21}$$

which leads to the explicit solution

$$\boldsymbol{\gamma} = (\widetilde{\mathbf{X}}^T \mathbf{\Omega}^{-1/2} \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{\Omega}^{-1/2} \mathbf{y} \tag{22}$$

Once $\mathbf{\Omega}$ is given, the representation coefficients can be efficiently computed with only matrix operators. However, for practical applications, $\mathbf{\Omega}$ is usually not known. It can be estimated by using the Feasible Generalized Least squares (FGLS) algorithm [56]. Firstly, assuming that there is not a representation error, the representation problem is degraded to the Original Least Squares (OLS) problem and the representation coefficients can be estimated as

$$\boldsymbol{\gamma} = (\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{y} \tag{23}$$

Accordingly, the representation residuals can be obtained as

$$\mathbf{e} = \mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\gamma} \tag{24}$$

Then the variance matrix $\boldsymbol{\Omega}$ can be derived as the diagonal matrix of the squared residuals [56]

$$\boldsymbol{\Omega} = diag(\mathbf{e})^2 \tag{25}$$

Now reconsidering the original weighted least squares (Eq. (20)) problem, we estimate its solution using Eq. (22). Repeat the above procedures (Eqs. (24), (25), and (22)) until the coefficient vector $\boldsymbol{\gamma}$ converges to a stable point. Algorithm 1 summarizes such an iteration process. Given the obtained coefficient vector, the weight of the i-th particle can be computed using Eq. (14). The tracking result in the current time refers to the particle with the largest weight. The outline of the proposed tracking algorithm is presented in Algorithm 2.

Algorithm 1: Sovling Eq. (22) using the Feasible Generalized Least squares.

Input: Given template set $\widetilde{\mathbf{X}}$, candidate feature $\mathbf{y}$ and maximal iteration number L
Output: Representation coefficients $\boldsymbol{\gamma}_{FGLS}$

Solve the Original Least squares (OLS) problem $\boldsymbol{\gamma} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$;
for $i = 1$ **to** L **do**
  Compute residuals $\mathbf{e} = \mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\gamma}$;
  Compute variance matrix $\boldsymbol{\Omega} = diag(\mathbf{e})^2$;
  Update coefficients $\boldsymbol{\gamma} = (\widetilde{\mathbf{X}}^T\boldsymbol{\Omega}^{-1/2}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\boldsymbol{\Omega}^{-1/2}\mathbf{y}$;
end

Algorithm 2: Proposed tracking algorithm

Input: Given initial state $\mathbf{z}_0$, observations $\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T\}$
Output: Target states $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$

Sampling target and background templates from $\mathbf{I}_0$ given $\mathbf{z}_0$;
for $t = 1$ **to** T **do**
  Sampling N target candidates $\{\mathbf{z}_t^1, \mathbf{z}_t^2, \ldots, \mathbf{z}_t^N\}$ from $\mathbf{z}_{t-1}$;
  for $i = 1$ **to** N **do**
    Computing coefficients vector using Algorithm 1;
    Computing weight ω_t^i of the i-th candidate using Eq. (14);
    Obtaining the current state $\mathbf{z}_t = \mathbf{z}_t^{\hat{i}}$ where $\hat{i} = \arg\max_i \omega_t^i$;
  end
  if *mod(t, 5)==1* **then**
    Sampling background templates from $\mathbf{I}_t$ given $\mathbf{z}_t$;
  end
end

IV. Experiments

In this section, we present the experimental results to validate the effectiveness of the proposed method and also compare it with other state-of-the-art methods. Firstly, we introduce the experiment protocols including the used dataset and the evaluation metrics. Then we present the quantitative and qualitative comparisons against the other methods.

A. Experiment Protocols

1) Datasets: Recently, a large scale benchmark library[2] for visual tracking was built by Wu *et al*. [57]. It contains a total of 50 test sequences collected from recent literatures. The authors have manually tagged the test sequences with 11 attributes, which represent the challenging aspects in visual tracking including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, low resolution. The length of these sequences vary from 71 to 3872. In addition to the sequences, this benchmark also contains the codes of publicly available visual trackers as well as their tracking results on the sequences.

2) Evaluated trackers: To demonstrate the effectiveness of the proposed tracking framework, we select several baseline trackers including the incremental visual tracker (IVT) [26], the multiple instance learning tracker (MIL) [58], the online AdaBoost tracker (OAB) [59], the L1 tracker using accelerated proximal gradient (L1APG) [31], the compressive sensing based tracker (CS) [30], and the scaled fast compressive tracker (SFCT) for comparison. Note that except the SFCT and our tracker, the results of other trackers are directly obtained from the benchmark. To test the SFCT tracker, we download the code from the authors's website[3] and run it on the benchmark sequences without manually tuning the parameters for the individual sequences.

3) Evaluation metrics: Two frame based metrics are widely used to assess the performance of a tracker: 1) *center location error*, which is defined as the Euclidean distance between the center location of the tracked target and the manually labeled ground-truth position; 2) *bounding box overlap* which is the ratio of the areas of the intersection and the union of the bounding boxes indicating the tracked object and the ground-truth. To measure the overall performance of a tracker on a test sequence, we adopt the *success rate* and *precision score* metrics. The former is computed as the percentage of the image frames which has a bounding box overlap larger than a given threshold. The latter is the percentage of image frames which have a center position error less than a given threshold. In each case, when multiple thresholds are used, a curve is provided to show how success rates or precision scores are affected by the threshold value. These curves are called **Success plot** and **Precious plot**, respectively. To ease the comparison, we average the Success and Precious curves of a tracker over all the sequences that represent a tracking challenge to obtain per challenge Success and Precious plots.

[2] http://visual-tracking.net
[3] http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm

	Ours	OAB	L1APG	IVT	MIL	SFCT	CT
Occlusion	0.505	0.361	0.340	0.327	0.300	0.310	0.260
Illumination variation	0.482	0.301	0.296	0.292	0.285	0.265	0.251
Scale variation	0.417	0.332	0.341	0.319	0.316	0.306	0.257
Background clutter	0.535	0.329	0.321	0.287	0.351	0.351	0.230
Deformation	0.523	0.361	0.313	0.303	0.328	0.344	0.249
Fast motion	0.454	0.350	0.283	0.210	0.315	0.339	0.217
Motion blur	0.470	0.326	0.266	0.219	0.279	0.289	0.197
In-plane rotation	0.488	0.350	0.352	0.331	0.334	0.295	0.255
Out-of-plane rotation	0.488	0.352	0.341	0.327	0.330	0.295	0.251
Out of view	0.589	0.394	0.309	0.302	0.301	0.342	0.262
Low resolution	0.312	0.314	0.297	0.195	0.208	0.321	0.114
Overall	0.509	0.375	0.362	0.342	0.338	0.324	0.258

TABLE I
AUC OF THE SUCCESS PLOTS OF THE STUDIED TRACKERS.

	Ours	OAB	L1APG	IVT	MIL	SFCT	CT
Occlusion	0.727	0.494	0.376	0.262	0.266	0.433	0.136
Illumination variation	0.701	0.385	0.380	0.412	0.340	0.327	0.296
Scale variation	0.648	0.494	0.474	0.477	0.450	0.457	0.374
Background clutter	0.753	0.433	0.404	0.414	0.446	0.434	0.286
Deformation	0.705	0.496	0.404	0.440	0.420	0.435	0.310
Fast motion	0.602	0.418	0.334	0.229	0.374	0.403	0.228
Motion blur	0.572	0.394	0.315	0.259	0.354	0.351	0.232
In-plane rotation	0.704	0.489	0.488	0.483	0.459	0.394	0.356
Out-of-plane rotation	0.712	0.497	0.475	0.483	0.454	0.401	0.348
Out of view	0.709	0.416	0.333	0.313	0.253	0.303	0.205
Low resolution	0.381	0.445	0.376	0.262	0.266	0.433	0.136
Overall	0.728	0.520	0.489	0.496	0.459	0.442	0.350

TABLE II
PRECIOUS SCORES FOR THE THRESHOLD = 20 PIXELS OF THE STUDIED TRACKERS.

The area under curve (**AUC**) of the success plot or the precision score for the threshold = 20 pixels is used to quantify the overall performance of a tracker for a challenge.

The conventional way to evaluate trackers is to run a tracker throughout a test sequence with an initialization from the ground-truthed position in the first frame. However, we found the initialization usually affects the performance of a tracker significantly. Therefore, it is necessary to test how robust a tracker is against different initialization states. In [57], Wu *et al*. proposed two ways to analyze a tracker's robustness against initialization: temporal robustness evaluation (**TRE**) that perturbs the initialization by starting a tracker at different frames and spatial robustness evaluation (SRE) that perturbs the initialization spatially by starting a tracker at different bounding boxes. In this work, we adopt the **SRE** for all the comparisons shown in this paper.

4) parameter Setup: The parameters related to the particle filter framework are set to be like those used in the benchmark [57]. The other parameters related to our method are set as $n_f = 50$, $n_b = 200$, $w = 32$, $h = 32$, $\hat{d} = 100$, $\delta = 0.4$ and $L = 5$ for all the sequences.

B. Quantitative Comparison

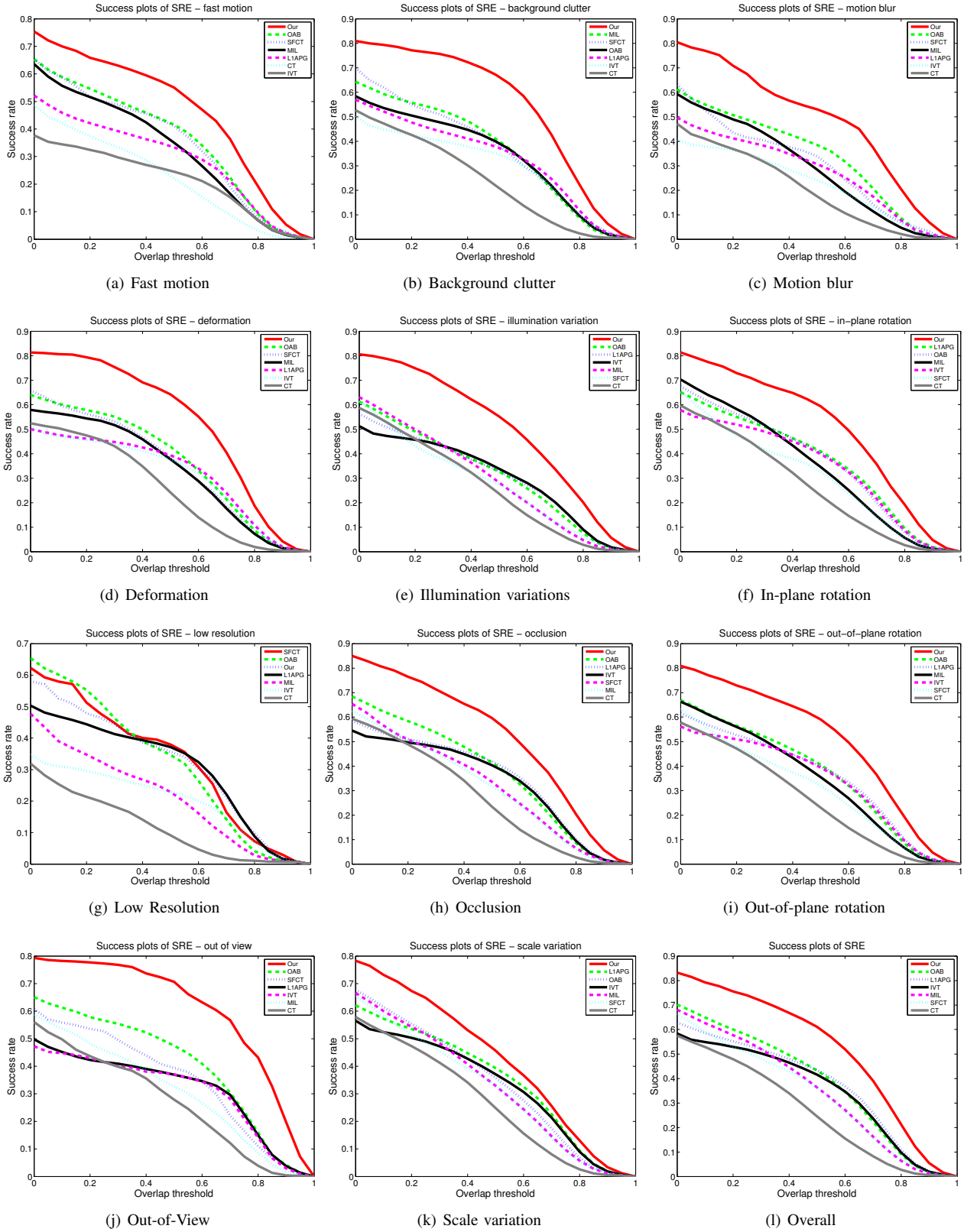
Fig. 2(a)–Fig. 2(k) and Fig. 3(a)–Fig. 3(k) show the success plots and precious plots for all the compared trackers averaging over the test sequences containing the same challenge, respectively. For example, Fig. 2(a) shows the success plots of all the compared trackers averaging over the results of all the test sequences containing fast motion. As we can see from these figures, our tracker achieves the best performance in all the challenges except the low resolution one where our tracker is slightly worse than the SFCT and OAB trackers. The success plots and precision plots of the compared trackers averaging over all the test sequences in the benchmark are shown in Fig. 2(l) and Fig. 3(l), respectively. To give quantitative comparison in numbers, Table I and Table II show the AUC values of the success plots in Fig. 2 and the precision scores for the threshold = 20 pixels in Fig. 3, respectively. The numbers in these tables quantitatively reflect the performance of the compared trackers on each individual challenge and also the entire benchmark. From the last rows of these tables, we can see that the overall performance of our tracker outperforms all the other methods.

C. Qualitative comparison

To qualitatively evaluate the tracking performance of the compared trackers, we show some tracking results on a subset of the benchmark in Fig. 4. We randomly select ten test sequences from the benchmark. For each selected sequence, we show the tracking results of all the compared trackers on six exemplar image frames. Note that we evenly select six frames over each sequence to make sure there is no bias when selecting the exemplar frames. As we can see, our tracker successfully tracks the targets in these frames on the *dog*, *doll*, *dude*, *fist*, *mhyang*, *suv*, *trellis* and *walking2* sequences, which mainly contain pose changes, partial occlusion and illumination changes. The results on these sequences indicate our tracker has strong abilities when it is used to handle these challenges. In contrast, as shown in the second row, the OAB, MIL, L1APG trackers fail to track the doll in the *doll* sequence. In the *suv* sequence, the IVT, CT and SFCT trackers also fail to track the car. Our tracker achieves superior performance on these sequences where the challenges are reasonably difficult, however, it loses the targets when the challenges are extremely difficult or the sequence simultaneously contains several challenges. For example, the *sylvester* sequence has both illumination and pose changes. In the 1345-th frame, our tracker loses the target since the target in this frame suffers from pose changes while the illumination is also changed. In the 597-th frame of the *woman* sequence, our tracker drifts away from the woman since the upper body of the woman was occluded by the tree and string. Generally speaking, these qualitative comparisons also validate the effectiveness of our proposed method.

D. Running speed

To investigate the computation efficiency of the proposed tracker, we compare the running speed of our tracker and three real-time trackers including the L1-APG tracker, the CS tracker and the SFCT tracker. To compare against ℓ_1 trackers, we also include the L1 [28] tracker and the BL1 [29] tracker. We test these methods on the woman sequence on a standard PC with an Intel Core 4 Duo 3.0 GHz processor and 4G RAM. As witnessed from Table III, our tracker achieves 29 frames per second, which is significantly faster than two ℓ_1 trackers and slightly faster than the CS tracker and the L1APG tracker but slightly slower than the SFCT tracker.



(a) Fast motion (b) Background clutter (c) Motion blur

(d) Deformation (e) Illumination variations (f) In-plane rotation

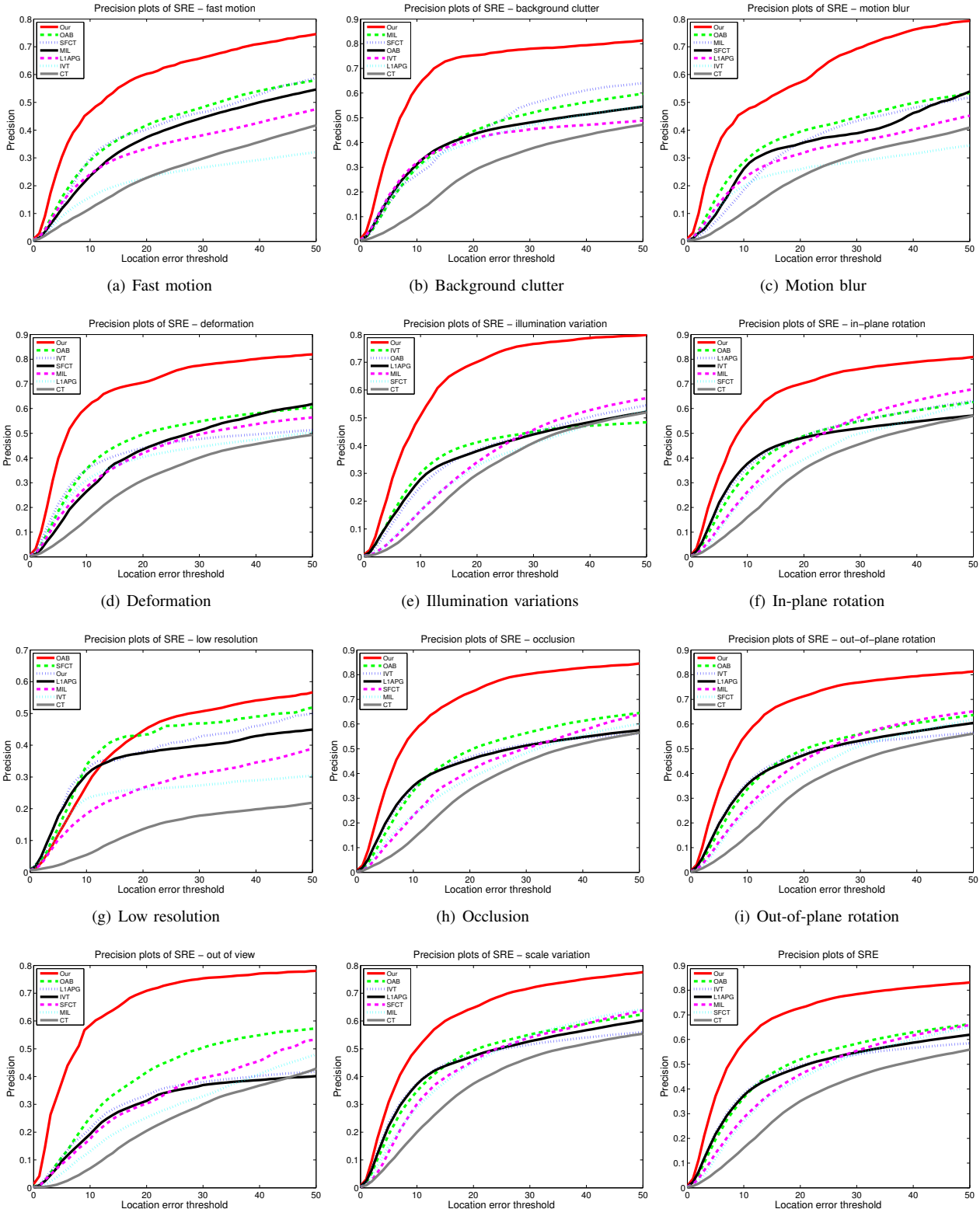
(g) Low Resolution (h) Occlusion (i) Out-of-plane rotation

(j) Out-of-View (k) Scale variation (l) Overall

Fig. 2. Success plots for the challenges considered in this work.

E. Discussion

In term of tracking effectiveness, our tracker achieves better performance than other state-of-the-art methods in all the tracking challenges except the low resolution one. The good performance of our tracker comes from two aspects: 1) both target and background templates are used in the



(a) Fast motion (b) Background clutter (c) Motion blur

(d) Deformation (e) Illumination variations (f) In-plane rotation

(g) Low resolution (h) Occlusion (i) Out-of-plane rotation

(j) Out-of-View (k) Scale variation (l) Overall

Fig. 3. Precision plots for the challenges considered in this work.

linear representation framework, which makes our tracker have strong abilities of distinguishing the tracked target from its background, and 2) a weighed least squares method is used to obtain the representation coefficients, which are robust to appearance variations over time. For the low resolution challenge, our tracker has slightly worse results than some of the other methods. The reason is that the random projection used in our method for feature dimensionality reduction loses partial

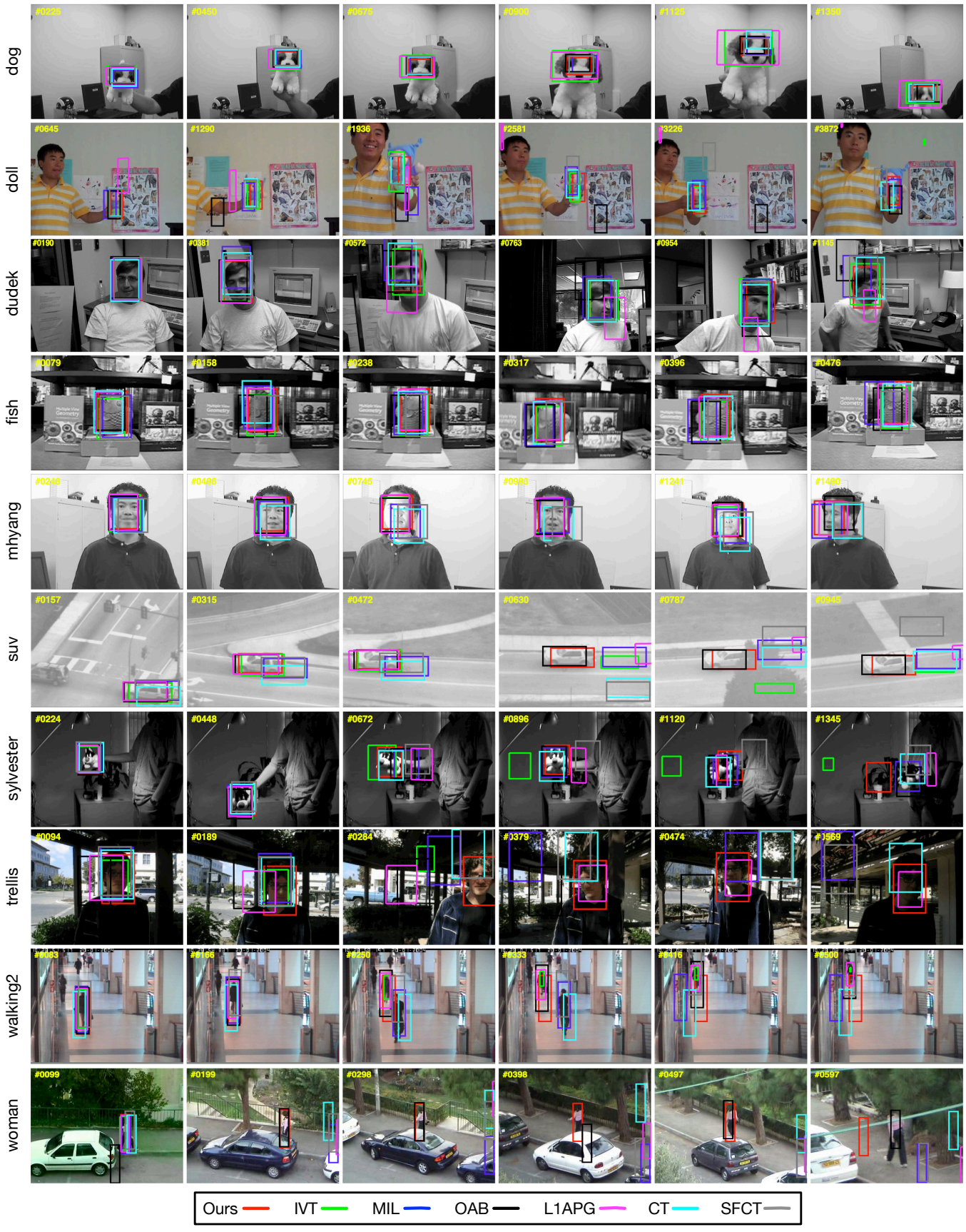


Fig. 4. Examples of tracking results of the compared methods on ten test sequences.

TABLE III
RUNNING SPEED COMPARISONS OF THE SELECTED TRACKERS ON THE WOMAN SEQUENCE

tracker	L1	BL1	CS	L1APG	SFCT	Ours
speed (frames/second)	0.18	0.51	19	23	35	29

useful information, especially when the sequence has very low resolution. Regarding the tracking efficiency, our tracker can achieve real-time operation since it avoids most of the computational costs required by those ℓ_1 trackers by adopting the fast weighed least squares method. However, our method is currently slower than the SFCT tracker as the randomly projection also causes additional computation costs especially when both targets and backgrounds are simultaneously used in the linear representation framework.

V. CONCLUSION

In this paper, to further improve the performance of the state-of-the-art sparse representation based visual tracking methods, we proposed a novel tracking method based on the weighted least squares and structural random projection. The weighed least squares technique releases the sparsity constraint imposed by the traditional sparse representation methods while achieving strong robustness against appearance variations. In addition, by introducing background templates into the linear representation framework, our method has strong capability of discriminating the tracked target from its background. On the other hand, the dimensionality of feature representation in our method is significantly reduced using structurally random projection while the pairwise distances between the data points in the feature space are preserved, reducing the computational complexity and making the proposed method feasible in real-time applications. Experimental results on a benchmark with 50 challenging sequences validate the effectiveness and efficiency of the proposed method.

REFERENCES

[1] H. Zhou, Y. Yuan, Y. Zhang, and C. Shi, "Non-rigid object tracking in complex scenes," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 98–102, 2009. 1

[2] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009. 1

[3] J. Wen, X. Gao, Y. Yuan, D. Tao, and J. Li, "Incremental tensor biased discriminant analysis: A new color-based visual tracking method," *Neurocomputing*, vol. 73, pp. 827–839, 2010. 1

[4] J. Wen, X. Gao, X. Li, D. Tao, and J. Li, "Incremental pairwise discriminant analysis based visual tracking," *Neurocomputing*, vol. 74, pp. 428–438, 2010. 1

[5] F. Monti and C. S. Regazzoni, "Ght based implementation of the expectation maximization for mixtures of multi-gaussians and its applications to video tracking," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 1278–1281, 2010. 1

[6] Z. Han, Q. Ye, and J. Jiao, "Combined feature evaluation for adaptive visual object tracking," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 69–80, 2011. 1

[7] Z. Han, J. Jiao, B. Zhang, Q. Ye, and J. Liu, "Visual object tracking via sample-based adaptive sparse representation (AdaSR)," *Pattern Recognition*, vol. 44, no. 9, pp. 2170–2183, 2011. 1

[8] T. A. Biresaw and C. S. Regazzoni, "A bayesian network for online evaluation of sparse features based multitarget tracking," *Proceedings of IEEE Conference on Image Processing*, pp. 429–432, 2012. 1

[9] X. Cao, J. Lan, P. Yan, and X. Li, "Vehicle detection and tracking in airborne videos by multi-motion layer analysis," *Machine Vision and Applications*, vol. 23, no. 5, pp. 921–935, 2012. 1

[10] X. Cao, Z. Shi, P. Yan, and X. Li, "Tracking vehicles as groups in airborne videos," *Neurocomputing*, vol. 99, pp. 38–45, 2013. 1

[11] Q. Huang, Z. Yang, W. Hu, L. Jin, G. Wei, and X. Li, "Linear tracking for 3-d medical ultrasound imaging," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1747–1754, 2013. 1

[12] X. Lu, Y. Yuan, and P. Yan, "Robust visual tracking with discriminative sparse learning," *Pattern Recognition*, vol. 46, no. 7, pp. 1762–1771, 2013. 1

[13] J. Fang, Q. Wang, and Y. Yuan, "Part-based online tracking with geometry constraint and attention selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 854–864, 2013. 1

[14] X. Lan, A. J. Ma, and P. C. Yuen, "Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1194–1201, 2014. 1

[15] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, vol. 131, pp. 227–236, 2014. 1

[16] Y. Yuan, J. Fang, and Q. Wang, "Robust superpixel tracking via depth fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 15–26, 2014. 1

[17] F. Yang, H. Lu, and M.-H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 2, pp. 242–254, 2014. 1

[18] Y. Wu, B. Ma, M. Yang, Y. Jia, and J. Zhang, "Metric learning based structural appearance model for robust visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 865–877, 2014. 1

[19] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2014. 1

[20] H. Zhou, M. Fei, A. Sadka, Y. Zhang, and X. Li, "Adaptive fusion of particle filtering and spatio-temporal motion energy for human tracking," *Pattern Recognition*, vol. 47, no. 11, pp. 3552–3567, 2014. 1

[21] S. Zhang, H. Zhou, H. Yao, Y. Zhang, K. Wang, and J. Zhang, "Adaptive normalhedge for robust visual tracking," *Signal Processing*, DOI: 10.1016/j.sigpro.2014.08.027. 1

[22] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *Proceedings of the 4th European Conference on Computer Vision*, pp. 343–356, 1996. 1

[23] A. Shahrokni, T. Drummond, and P. Fua, "Fast texture-based tracking and delineation using texture entropy," *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 1154–1160, 2005. 1

[24] A. Jepson, D. Fleet, and T. EI-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003. 1

[25] R. Collins, Y. Liu, and M. Leordeanu, "On-line selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005. 1

[26] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 8, pp. 125–141, 2007. 1, 6

[27] C. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010. 1

[28] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," *Proceedings of the 12th International Conference on Computer Vision*, pp. 1436–1443, 2009. 1, 2, 7

[29] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient L1 tracker with occlusion detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1257–1264, 2011. 1, 3, 7

[30] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1305 –1312, 2011. 1, 2, 5, 6

[31] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2012. 1, 2, 6

[32] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013. 1, 2

[33] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009. 2

[34] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1313–1320, 2011. 2, 3

[35] S. Zhang, H. Yao, H. Zhou, X. Sun, and S. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, vol. 100, no. 1, pp. 31–40, 2013. 2

[36] L. Wang, H. Yan, K. Lv, and C. Pan, "Visual tracking via kernel sparse representation with multikernel fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1132–1141, 2014. 2

[37] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001. 2, 3

[38] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Proceedings of European Conference on Computer Vision*, pp. 661–675, 2002. 2, 3

[39] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale L1-regularized least squares," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, no. 4, pp. 606–617, 2007. 2

[40] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information theory*, vol. 51, no. 12, pp. 4203–4215, 2005. 3

[41] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," *Proceedings of the 11th European Conference on Computer Vision*, pp. 624–637, 2010. 3

[42] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012. 3

[43] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2012. 3

[44] T. Bai and Y. Li, "Robust visual tracking with structured sparse representation appearance model," *Pattern Recognition*, vol. 45, no. 6, pp. 2390–2404, 2012. 3

[45] Y. Xie, W. Zhang, C. Li, S. Lin, Y. Qu, and Y. Zhang, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 539–553, 2014. 3

[46] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998. 3

[47] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2009. 3

[48] W. Johnson and J. Lindenstraussn, "Extensions of lipschitz mappings into a hilbert space," *Conference in Modern Analysis and Probability*, pp. 189–206, 1984. 5

[49] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008. 5

[50] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006. 5

[51] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligencen*, vol. 36, no. 10, pp. 2002–2015, 2014. 5

[52] D. Achlioptasi, "Database-friendly random projection: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003. 5

[53] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast johnson-lindenstrauss transform," *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 557–563, 2006. 5

[54] T. Do, T. Tran, and L. Gan, "Fast compressive sampling with structurally random matrices," *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 3369–3372, 2008. 5

[55] T. Do, L. Gan, N. Nguyen, and T. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, 2012. 5

[56] R. Little and D. Rubin, "Statistical analysis with missing data (2nd ed.)," *JohnWiley & Sons, Inc.*, 2002. 5, 6

[57] Y. Wu, J. Lim, and M. Yang, "Online object tracking: A benchmark," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013. 6, 7

[58] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligencen*, vol. 33, no. 8, pp. 1619–1632, 2011. 6

[59] H. Grabner and H. Bischof, "On-line boosting and vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 260–267, 2006. 6