Robust Multitask Multiview Tracking in Videos

Xue Mei*, *Senior Member, IEEE*, Zhibin Hong*, *Student Member, IEEE*,
Danil Prokhorov, *Senior Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Various sparse-representation-based methods have been proposed to solve tracking problems, and most of them employ least squares (LSs) criteria to learn the sparse representation. In many tracking scenarios, traditional LS-based methods may not perform well owing to the presence of heavy-tailed noise. In this paper, we present a tracking approach using an approximate least absolute deviation (LAD)-based multitask multiview sparse learning method to enjoy robustness of LAD and take advantage of multiple types of visual features, such as intensity, color, and texture. The proposed method is integrated in a particle filter framework, where learning the sparse representation for each view of the single particle is regarded as an individual task. The underlying relationship between tasks across different views and different particles is jointly exploited in a unified robust multitask formulation based on LAD. In addition, to capture the frequently emerging outlier tasks, we decompose the representation matrix to two collaborative components that enable a more robust and accurate approximation. We show that the proposed formulation can be effectively approximated by Nesterov's smoothing method and efficiently solved using the accelerated proximal gradient method. The presented tracker is implemented using four types of features and is tested on numerous synthetic sequences and real-world video sequences, including the CVPR2013 tracking benchmark and ALOV++ data set. Both the qualitative and quantitative results demonstrate the superior performance of the proposed approach compared with several state-of-the-art trackers.**

***Index Terms*— L1 minimization, least absolute deviation (LAD), multitask, multiview, sparse representation, tracking.**

I. Introduction

ONLINE object tracking is an important research topic in computer vision and is related to many practical applications, such as video surveillance and vehicle perception. Given an annotation of the object (bounding box in our paper) in the first frame, the task of a tracker is to estimate the target locations using the same annotation in subsequent video frames. Many model-free trackers [1], [2] have been designed to handle generic object tracking, where the prior knowledge about the target is absent. In general, designing a universally effective tracker is extremely difficult due to the presence of various challenges, such as appearance variations, occlusions (OCCs), and illumination changes.

In the community of visual tracking, the least squares (LSs) criterion, i.e., Euclidean distance, is usually used to approximate the sparse representation for tracking [3], [4]. The LS criterion performs well when the data distribution is Gaussian. It is popular because of its differentiability and smoothness properties, and it can be efficiently solved by gradient-based methods [5]. However, the noise in many real tracking scenarios is heavy-tailed, such as in the cases of background clutter (BC), Laplace noise, and salt-and-pepper noise, where the LS-based methods may degrade seriously, since these kinds of noise cannot be well estimated by the LS. According to [6] and [7], least absolute deviation (LAD) is much more robust than the LS, especially in the presence of heavy-tailed noise.

On the other hand, tracking problems can involve data that are represented by multiple views[1] of various types of visual features, including intensity [10], color [11], edge [12], wavelet [13], and texture. Relying on these multiple sources of information can significantly improve tracking performance as a result of their complementary characteristics [12], [14]–[16]. Given these cues from multiple views, an important problem is how to integrate them and build an appropriate model to explore their mutual dependence and independence.

Sparse representation has recently been introduced for tracking [3], in which a tracking candidate is sparsely represented as a linear combination of target templates and trivial templates. In particle filter (PF)-based tracking methods, particles around the current state of the target are randomly sampled according to a zero-mean Gaussian distribution. Each particle shares dependence with other particles. Original multitask learning in [17] aims to improve the performance of multiple related tasks by exploiting the intrinsic relationship among them. In [4], learning sparse representation of each particle is viewed as an individual task. However, Zhang *et al.* [4] assume that all tasks share a common set of features, which generally does not hold in visual tracking applications, since outlier tasks often exist. Outlier tasks are a set of minority tasks that do not share a common set of features with the majority of tasks. Furthermore, Mei and Ling [3] and Zhang *et al.* [4] only use the intensity feature to model the appearance change of the target. The intensity appearance model with L1 minimization is very robust to partial OCC and other tracking challenges [3]. However, it is very sensitive to shape deformation (DEF) of targets such as nonrigid objects.

Manuscript received January 29, 2014; revised January 23, 2015; accepted January 24, 2015. Date of publication February 26, 2015; date of current version October 16, 2015. This work was supported in part by the Toyota Research Institute North America, Ann Arbor, MI, USA, under Project 2013001793, and in part by the Australian Research Council under Project DP-120103730 and Project FT-130101457. Asterisk indicates equal contributions.

X. Mei and D. Prokhorov are with the Toyota Research Institute North America, Ann Arbor, MI 48108 USA (e-mail: nathanmei@gmail.com; dvprokhorov@gmail.com).

Z. Hong and D. Tao are with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology at Sydney, Sydney, NSW 2007, Australia (e-mail: zhibin.hong@student.uts.edu.au; dacheng.tao@uts.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2399233

[1]Regarding the term multiview learning [8], [9], we follow the machine learning convention, in which views refer to different feature subsets used to represent particular characteristics of an object.

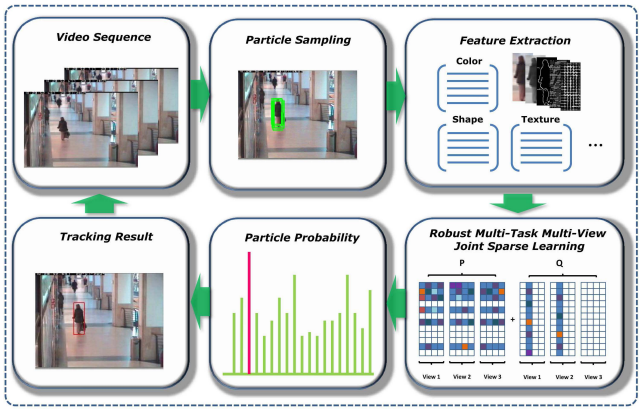


Fig. 1. Flowchart to illustrate the proposed tracking framework.

To overcome the above problems, we propose to take advantage of LAD and employ other visual features such as color, edge, and texture to complement intensity in the appearance representation, and to combine a multiview representation with a robust multitask learning [18] (Fig. 1). Within our proposed framework, the sparse representation for each view is learned as a linear combination of atoms from an adaptive feature dictionary, which enables the tracker to capture different statistics carried by different views. To exploit the interdependence shared between different views and particles, we impose the $\ell_{1,2}$-norm group-sparsity regularization on the representation matrix to learn the joint sparse representation for all views and over all particles, where learning the sparse representation for each view of a single particle is regarded as an individual task. The LAD instead of LS reconstruction error is used to learn the sparse representation and to improve the robustness of the learned representation. We decompose the sparse representation into two collaborative parts, thereby enabling them to learn representative coefficients and detect outlier tasks simultaneously. The proposed LAD formulation is effectively approximated by the Nesterov's smoothing method [19]. An efficient accelerated proximal gradient (APG) [5] scheme is employed to obtain the optimal solution via a sequence of closed-form updates. Although discriminative approaches can be sometimes more effective, generative methods often have better performance when the size of labeled data is small [20]. Many trackers are built on discriminative approaches [11], [21]–[24], but there are also many generative [3], [4], [25], [26] or even hybrid [27] methods that demonstrate superior performance in various scenarios. It should be noted that this paper aims to improve the previous generative approaches [3], [4] by considering the multiview setting in the sparse representation framework and exploring the relationship between different views among different particles. Although employing the discriminative setting in the current framework might further improve the performance, it is beyond the scope of this paper.

An earlier version of this paper has been published in [28]. The main differences between this paper and [28] are as follows. First, instead of using the popular LS criterion, we propose to take an advantage of LAD, which is more robust than the LS method, especially when the data are contaminated by outliers and noise, and use LAD reconstruction error during the sparse representation learning. Second, due to the nonsmoothness of Manhattan norm used in the LAD criterion, the APG method cannot be directly used, which is different from the case in [28]. Therefore, we approximate LAD using the Nesterov's smoothing method [19], and then efficiently solve the optimization using the APG scheme. Finally, we significantly increase the number of testing sequences for extensively evaluating the proposed tracker. The quantitative results demonstrate the superior performance of the new approach in comparison with the baseline trackers including the multitask multiview tracking LS (MTMVTLS) in [28] and many other trackers.

II. Related Work

An extensive review on tracking and multiview learning is beyond the scope of this paper. We refer readers to some recently published surveys [1], [29] for more details about existing trackers, and an extensive survey on multiview learning can be found in [30]. In this section, we review the works of relevance to our method including popular single-view-based trackers, multiview-based trackers and sparse representation-based trackers, multitask learning, and LAD.

Numerous existing trackers use single feature only and solve tracking in various ways. For instance, Comaniciu *et al.* [31] introduce a spatial kernel to regularize the color histogram-based feature representation of the target, which enables tracking to be reformulated as a gradient-based optimization problem solved by mean-shift. Babenko *et al.* [21] employ multiple instance learning (MIL) equipped with a Haar feature pool to overcome the label ambiguity problem. Ross *et al.* [32] present a tracking method that incrementally learns a low-dimensional subspace representation based on intensity features. Kalal *et al.* [33] propose a new tracking paradigm that combines the classical Lucas–Kanade-based tracker with an online learned random-forest-based detector using pixel-wise comparison of features. The learned detector is notable for enabling reacquisition following tracking failures.

The above trackers nevertheless tend to be vulnerable in particular scenarios due to the limitations of the adopted features. Various methods aim to overcome this problem by taking advantage of multiple types of features to enable a more robust tracker [12], [34]–[36]. In [37], two complementary features, color histogram and intensity gradient, are jointly considered to track a person's head. Moreno-Noguer *et al.* [34] propose a probabilistic framework allowing the integration of multiple features for tracking by considering cue dependencies. Kwon and Lee [12] propose a method termed visual tracking decomposition (VTD), which employs sparse principal component analysis to construct multiple basic observation models (basic trackers) based on multiple types of features. In [38], the visual tracker sampler (VTS) is further proposed by the authors to sample the basic trackers and probabilistically determine the acceptance of them.

Sparse representation was recently introduced for tracking in [3] which casts tracking as a sparse representation problem

in a PF framework [39] which was later used in [40]–[43]. In [4], a multitask learning [44] approach is applied to tracking by learning a joint sparse representation of all the particles in a PF framework. Compared with the original L1 tracker [3] that pursues the sparse representation independently, multitask tracking (MTT) achieves more robust performance by exploiting the interdependency between particles. In addition, [45] also tries to exploit the interdependency between particles and cast the tracking problem as a low-rank matrix learning problem. Multitask learning has also been successfully applied to face recognition [46] and image classification [47]. In [47], a multitask covariate selection model is used to classify a query image using multiple features from a set of training images, and a class-level joint sparsity regularization is imposed on class-level representation coefficients.

A well-known alternative of the popular LSs is the LAD [48]. Harter [7] comprehensively discusses the method of LS and its alternatives, and proposes that LAD is more robust than the LS method especially when the data are contaminated by outliers. In addition, a lot of works have been proposed to exploit the robustness of LAD in the linear approximation problems [6], [49]–[51]. Wang *et al.* [51] propose the LASSO regularized LAD regression that takes advantage of both the LAD and the LASSO. Guan *et al.* [6] also exploit the LAD and propose a Manhattan nonnegative matrix factorization for modeling the heavy-tailed Laplacian noise. The effectiveness of the proposed method was tested on both synthetic and real-world data sets, such as face images, natural scene images, surveillance videos, and multimodel data sets.

Motivated by the above advances, in this paper, we propose a multitask multiview tracking (MTMVT) method based on sparse representation to exploit the related information shared between particles and views in order to obtain improved performance. Moreover, we propose to employ the LAD and minimize the sum of absolute errors (SAEs) regularized by the group sparsity for the joint spare representation learning. In the rest of this paper, we denote the proposed MTMVTLSs and the MTMVT method using LAD as MTMVTLAD.

III. Particle Filter

The PF [52], also known as the sequential Monte Carlo method, is a Bayesian sequential importance sampling technique, which provides a convenient framework for estimating the posterior distribution of state variables and for simulating a dynamic system, such as the tracking process. The sophisticated PF has been extensively used in object tracking due to its nonlinear and non-Gaussian model assumption, regardless of the underlying distribution.

Let $\mathbf{y}_t$ denote the state variable describing the location and shape of a target at time frame t. Given all the available observations $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ until the current frame t, the predicting distribution of the target is inferred using the Bayes rule as

$$p(\mathbf{y}_t|\mathbf{x}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{y}_{t-1})p(\mathbf{y}_{t-1}|\mathbf{x}_{1:t-1})d\mathbf{y}_{t-1} \quad (1)$$

$$p(\mathbf{y}_t|\mathbf{x}_{1:t}) \propto p(\mathbf{x}_t|\mathbf{y}_t)p(\mathbf{y}_t|\mathbf{x}_{1:t-1}) \quad (2)$$

where $p(\mathbf{y}_t|\mathbf{y}_{t-1})$ is the state transition distribution and $p(\mathbf{x}_t|\mathbf{y}_t)$ is the observation likelihood estimated by the appearance model. In practice, the posterior probability is approximated by a finite set of n samples, i.e., particles, $\{\mathbf{y}_t^i\}_{i=1}^n$ with importance weight w_t^i. At each frame, the weight w_t^i is updated by the observation likelihood $p(\mathbf{x}_t|\mathbf{y}_t^i)$ following the strategy of the bootstrap filter [52]

$$w_t^i \propto w_{t-1}^i p\left(\mathbf{x}_t|\mathbf{y}_t^i\right). \quad (3)$$

Subsequently, a set of n equally weighted particles are resampled according to the importance weights using state transition distribution $p(\mathbf{y}_t|\mathbf{y}_{t-1})$.

In this paper, we let $\mathbf{y}_t = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_x, t_y)$ to describe the 2-D affine transformation of the target, where $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ are the affine transformation parameters and (t_x, t_y) are the translation parameters, which is analogous to the previous work [3]. The state transition distribution $p(\mathbf{y}_t|\mathbf{y}_{t-1})$ is simulated independently by the Gaussian distribution model, while the observation likelihood $p(\mathbf{x}_t|\mathbf{y}_t)$ reflecting the similarity between a target candidate and the target template is estimated by the reconstruction error described in Section IV. To model the observation likelihood $p(\mathbf{x}_t|\mathbf{y}_t)$, a region corresponding to state $\mathbf{y}_t$ is first cropped from the current frame. Multiple features are then extracted from the region and normalized to form a 1-D feature vector $\mathbf{x}_t$.

IV. Multitask Multiview Sparse Tracker

The L1 tracker [3] tackles tracking as finding a sparse representation in the template subspace. The representation is then used in a PF framework for visual tracking. However, appearance representation based only on intensity is prone to failure in difficult scenarios such as tracking nonrigid objects. Employing multiple types of features has proved to be beneficial for tracking because the ensemble of multiple views provides a comprehensive representation of the target appearance undergoing various changes such as illumination and DEF. However, combining multiple views by simply concatenating features into a high-dimensional feature vector is not a good option, since different features have different statistical properties [30]. Inspired by [4] and [47], the dependence of these views as well as the intrinsic relationship of sampled particles should be jointly considered. In this section, we propose to employ other visual features such as color, edge, and texture to complement intensity in the target appearance representation, and to combine a multiview representation with a robust multitask learning [18] to solve the visual tracking problem.

A. Sparse Representation-Based Tracker

In [3], the sparse representation of intensity feature x is formulated as the minimum error reconstruction through an L1-regularized minimization problem with nonnegativity constraints

$$\min_{\mathbf{w}} \| \mathbf{M}\mathbf{w} - \mathbf{x} \|_2^2 + \lambda \| \mathbf{w} \|_1, \quad \text{s.t. } \mathbf{w} \succcurlyeq 0 \quad (4)$$

where $\mathbf{M} = [\mathbf{D}, \mathbf{I}, -\mathbf{I}]$ is an over-complete dictionary that is composed of target template set $\mathbf{D}$ and positive and negative

trivial template sets $\mathbf{I}$ and $-\mathbf{I}$. Each column in $\mathbf{D}$ is a target template generated by reshaping pixels of a candidate region into a column vector; and the trivial templates $\mathbf{I}$ is modeled using an identity matrix. $\mathbf{w} = [\mathbf{a}^\top, \mathbf{e}^{+\top}, \mathbf{e}^{-\top}]^\top$ is composed of target coefficients $\mathbf{a}$ and positive and negative trivial coefficients $\mathbf{e}^+$, $\mathbf{e}^-$, respectively.

Finally, the observation likelihood is derived from the reconstruction error of $\mathbf{x}$ as

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{\Gamma} \exp\{-\alpha \parallel \mathbf{Da} - \mathbf{x} \parallel^2\} \tag{5}$$

where $\mathbf{a}$ is obtained by solving the L1 minimization (4), α is a constant controlling the shape of the Gaussian kernel, and Γ is a normalization factor.

B. Robust Multitask Multiview Sparse Learning With Least Absolute Deviation

We consider n particle samples, each of which has K different views (e.g., color, shape, and texture). Learning the sparse representation for each view of a single particle is regarded as an individual task, so there are a total of nK tasks to tackle for the joint sparse representations. For each view index $k = 1, \ldots, K$, denote $\mathbf{X}^k \in \mathbb{R}^{d_k \times n}$ as the feature matrix which is a stack of n columns of normalized particle image feature vectors of dimension d_k, where d_k is the dimension for the kth view. We denote $\mathbf{D}^k \in \mathbb{R}^{d_k \times N}$ as the target dictionary in which each column is a target template from the kth view, where N is the number of target templates. The target dictionary is combined with trivial templates $\mathbf{I}_{d_k} \in \mathbb{R}^{d_k \times d_k}$ to construct the complete dictionary $\mathbf{M}^k = [\mathbf{D}^k, \mathbf{I}_{d_k}] \in \mathbb{R}^{d_k \times h_k}$, where $h_k = N + d_k$.

Motivated by [18], we jointly evaluate K feature view matrices $\{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$ with n particles and learn the latent representations $\{\mathbf{W}^1, \ldots, \mathbf{W}^K\}$. The decomposed matrices $\mathbf{W}^k$s enable different views of particles to have different learned representations, and therefore exploit the independency of each view and capture the different statistical properties. Moreover, each representation matrix $\mathbf{W}^k$ is constructed by two collaborative components $\mathbf{P}^k$ and $\mathbf{Q}^k$, where $\mathbf{P}^k$ is regularized by row sparse constraint, which assumes that all particles share the same basis, while $\mathbf{Q}^k$ is regularized by column sparse constraint, which enables the capture of outlier tasks.

The same columns from each view in the dictionary should be activated to represent the particle in a joint sparse manner, since the corresponding columns represent the same sample of the object. Therefore, the corresponding decomposed weight matrices $\mathbf{P}^k$s and $\mathbf{Q}^k$s from all the views can be stacked horizontally to form two bigger matrices $\mathbf{P}$ and $\mathbf{Q}$, respectively. Each of them consists of the coefficients across all the views. Group LASSO penalty $\ell_{1,2}$ is applied to row groups of the first component $\mathbf{P}$ for capturing the shared features among all tasks over all views, where we define $\|\mathbf{P}\|_{1,2} = \sum_j (\sum_i \mathbf{P}_{j,i}^2)^{1/2}$, and $\mathbf{P}_{j,i}$ denotes the entry in the jth row and ith column in the matrix $\mathbf{P}$. The same group LASSO penalty is imposed on column groups of the second component $\mathbf{Q}$ to identify the outlier tasks simultaneously. The multiview sparse representations for all particles can be obtained by solving the following problem:

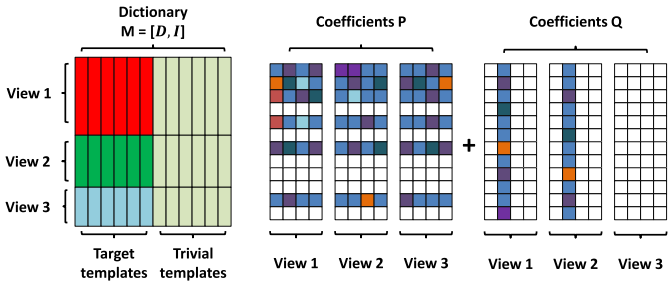


Fig. 2. Illustration for the structure of the learned coefficient matrices $\mathbf{P}$ and $\mathbf{Q}$, where entries of different color represent different learned values, and the white entries in $\mathbf{P}$ and $\mathbf{Q}$ indicate the zero rows and columns. Note that this figure demonstrates a case that includes four particles and three views, where the second particle is an outlier whose coefficients in $\mathbf{Q}$ comprise nonzero values.

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{k=1}^{K} f_L(\mathbf{M}^k\mathbf{W}^k - \mathbf{X}^k) + \lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^\top\|_{1,2} \tag{6}$$

where $f_L(\mathbf{X})$ is a cost function measuring the reconstruction errors during the representation learning, $\mathbf{W}^k = \mathbf{P}^k + \mathbf{Q}^k$, $\mathbf{P} = [\mathbf{P}^1, \ldots, \mathbf{P}^K]$, $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^K]$, and λ_1 and λ_2 are the parameters controlling the sparsity of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Fig. 2 shows the structure of the learned matrices $\mathbf{P}$ and $\mathbf{Q}$.

Note that the stacking of $\mathbf{P}^k$s and $\mathbf{Q}^k$s requires that $\mathbf{M}^k$s have the same number of columns. However, we can pad the matrices $\mathbf{M}^k$s with zero columns to make them the same number of columns in order to apply (6). The coefficients associated with the zero columns will be zeros based on the sparsity constraints from L1 regularization and do not impact the minimization function in terms of the solution. Without loss of generality, we assume $\mathbf{M}^k$s are sorted in descending order of the number of columns h_k, that is, $h_1 \geq h_2 \geq \cdots \geq h_K$. The new $\hat{\mathbf{M}}^k$ is defined as the zero padded matrix of $\mathbf{M}^k$, that is, $\hat{\mathbf{M}}^k = [\mathbf{M}^k, \mathbf{0}^k]$, where $\mathbf{0}^k \in \mathbb{R}^{d_k \times (h_1 - h_k)}$ and every element in 0^k is zero. We can replace $\mathbf{M}^k$ in (6) with $\hat{\mathbf{M}}^k$ and solve the same minimization problem.

For the cost function $f_L(\mathbf{MW} - \mathbf{X})$, a conventional selection is based on the Frobenius norm $f_L(\mathbf{MW} - \mathbf{X}) = 1/2\|\mathbf{MW} - \mathbf{X}\|_F^2$, which employs the Euclidean distance to measure the reconstruction error, i.e., minimizing the problem based on LS criterion. Then, the problem in (6) can be explicitly written as

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{k=1}^{K} \frac{1}{2}\|\mathbf{M}^k\mathbf{W}^k - \mathbf{X}^k\|_F^2 + \lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^\top\|_{1,2}. \tag{7}$$

The LS is popular due to its useful properties, namely, smoothness and differentiability, which enables application of efficient gradient-based methods, such as APG [44], as presented in our previous work [28]. However, as discussed in [6] and [7], LAD is much more robust than the ordinary LS in many applications, especially in the presence of heavy-tailed noise. The LAD estimate also arises as the maximum likelihood estimate if the errors have a Laplace distribution. To use LAD, we replace the Frobenius norm with the

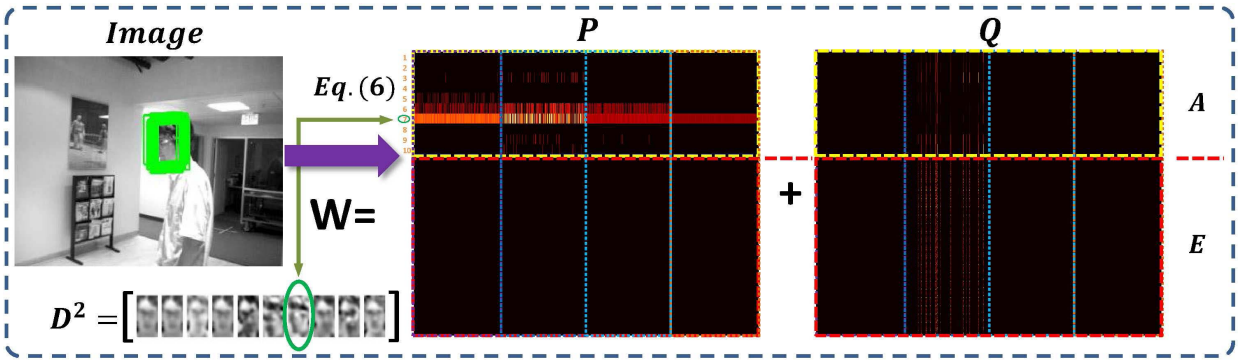


Fig. 3. Schematic example of the learned coefficients. We visualize the learned coefficient matrices **P** and **Q** for all particles across all views, which are color histograms, intensity, HOG, and LBP, respectively. Each matrix consists of four column parts corresponding to four different views, where the brighter color represents larger value in the corresponding matrix element. The seventh template in the dictionary is the most representative (which is circled in green in the shown intensity templates $\mathbf{D}^2$) and results in brighter values in the seventh row of **P** across all views (they are associated by the line with two arrows), while some columns in **Q** have brighter values which indicate the presence of outliers.

Manhattan norm, (6) becomes

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{k=1}^{K} \|\mathbf{M}^k\mathbf{W}^k - \mathbf{X}^k\|_{\mathbf{M}} + \lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^\top\|_{1,2}. \quad (8)$$

In particular, the problem in (8) minimizes the SAEs regularized by the group sparsity prior. Although (8) is still convex, each term in (8) is typically nonsmooth. The APG method can no longer be used. Fortunately, we will show the Nesterov's smoothing method [19] can be used to smooth the Manhattan norm in Section IV-D, and thus (8) can still be solved efficiently.

For a more intuitive view of the proposed formulation, we visualize a schematic example of the learned sparse coefficients in Fig. 3. The **W** can be decomposed in both horizontal and vertical directions. Vertically, $\mathbf{W} = [\mathbf{A}^\top, \mathbf{E}^\top]^\top$ consists of target coefficients **A** and trivial coefficients **E**, respectively, while horizontally, $\mathbf{W} = \mathbf{P} + \mathbf{Q}$ consists of information sharing matrix **P** and outlier identification matrix **Q**.

C. General Form and Special Cases

Before presenting the optimization method of (8), we would like to have a brief discussion about the proposed problem (6) in this section. The proposed optimization problem (6) can be generalized as

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{k=1}^{K} f_L(\mathbf{M}^k\mathbf{W}^k - \mathbf{X}^k) + \lambda_1||\mathbf{P}||_{p,q} + \lambda_2||\mathbf{Q}^\top||_{p,q} \quad (9)$$

where $\|\mathbf{P}\|_{p,q} = (\sum_j((\sum_i(\mathbf{P}_{j,i})^q)^{1/q})^p)^{1/p}$ is the $\ell_{p,q}$ norm of **P** and $\mathbf{P}_{j,i}$ represents the element in the jth row and ith column of **P**. To restrict a small number of dictionary templates to be selected by all particles across all views, let $p = 1$, then we get $\|\mathbf{P}\|_{1,q} = \sum_j(\sum_i(\mathbf{P}_{j,i})^q)^{1/q}$, which encourages **P** to be row sparse. For the options of q, we select three widely studied mixed norms $q \in \{1, 2, \infty\}$ as discussed in MTT [4]. Now, we discuss (9) with different combinations of λ_2, q, K, and f_L, which yields different trackers. If we restrict our tracker to the case of $\lambda_2 = +\infty$ and $K = 1$ for a single-view multitask problem, then we get $\mathbf{Q} = 0$. Therefore, (9) is degenerated to

$$\min_{\mathbf{P}} f_L(\mathbf{MP} - \mathbf{X}) + \lambda_1||\mathbf{P}||_{1,q} \quad (10)$$

where both the LS and LAD can be used. We note that if the LS is employed, where $f_L(\mathbf{MP} - \mathbf{X}) = 1/2\|\mathbf{MP} - \mathbf{X}\|_F^2$, (10) is exactly the same as the objective function used in MTT [4]. Furthermore, if we let $q = 1$, the obtained formulation is intrinsically the same as (4), which is used in the L1 tracker [3]. In this way, both the MTT tracker and the L1 tracker can be regarded as special cases of the proposed MTMVT in the single-view setting. Meanwhile, the LAD versions of MTT and L1 trackers can be naturally obtained within the proposed formulation in (10).

Here, we discuss another single-view version ($K = 1$) of MTMVT by appropriately setting $\lambda_2 > 0$, in which some nonzero columns of **Q** will be obtained if outliers exist. In particular, if we set $q = 2$, the MTT tracker with outlier handling can be obtained as follows:

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{W}} f_L(\mathbf{MW} - \mathbf{X}) + \lambda_1||\mathbf{P}||_{1,2} + \lambda_2||\mathbf{Q}^T||_{1,2} \quad (11)$$

where $\mathbf{W} = \mathbf{P} + \mathbf{Q}$, and the component **P** exploits the underlying relationships of majority particles, while the component **Q** is able to capture the outlier tasks simultaneously, which yields more robust representations.

D. Optimization With Approximated Least Absolute Deviation

In this section, we show how to solve (8) efficiently. First, the Manhattan norm is approximated by a smooth function using the method presented in [6] and [19], and then gradient-based method is applied to obtain the solution using a small number of closed-form updates.

Due to the separability property of Manhattan norm, we consider the following loss function of a single task:

$$g(\mathbf{M}, \mathbf{w}, \mathbf{x}) = \|\mathbf{Mw} - \mathbf{x}\|_1 \quad (12)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the single-view observation for a particle, $\mathbf{M} \in \mathbb{R}^{d\times h}$ is the dictionary, and $\mathbf{w} \in \mathbb{R}^h$ is the sparse representation. As shown in [6], the Nesterov's smoothing

method [19] can be used to approximate (12) and obtain a closed-form smoothed function as

$$g_\theta(\mathbf{M}, \mathbf{w}, \mathbf{x}) = \sum_{j=1}^{d} \|\mathbf{M}_{j,\cdot}\|_1 \psi_\theta\left(\frac{|\mathbf{M}_{j,\cdot}\mathbf{w} - \mathbf{x}_{(j)}|}{\|\mathbf{M}_{j,\cdot}\|_1}\right) \quad (13)$$

where $\mathbf{M}_{j,\cdot}$ is the jth row of $\mathbf{M}$, $\mathbf{x}_{(j)}$ is the jth entry of vector $\mathbf{x}$, and $\theta > 0$ is a parameter controls the smoothness. The larger the parameter θ the smoother the approximate function $g_\theta(\mathbf{M}, \mathbf{w}, \mathbf{x})$, but the worse the approximate accuracy. The function ψ_θ is a piecewise function defined as

$$\psi_\theta(\delta) = \begin{cases} \frac{\delta}{2\theta}, & 0 \le \delta \le \theta \\ \delta - \frac{\theta}{2}, & \delta > \theta. \end{cases} \quad (14)$$

According to [19], $g_\theta(\mathbf{M}, \mathbf{w}, \mathbf{x})$ is well defined and continually differentiable at any $\mathbf{w} \in \mathbb{R}^h$. Moreover, $g_\theta(\mathbf{M}, \mathbf{w}, \mathbf{x})$ is convex and its gradient with respect to $\mathbf{w}$ can be obtained as

$$\nabla_w g_\theta = \mathbf{M}^T \boldsymbol{\mu} \quad (15)$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the Lagrange multiplier vector and

$$\mu_{(j)} = \text{med}\left(-1, +1, \frac{\mathbf{M}_{j,\cdot}\mathbf{w} - \mathbf{x}_{(j)}}{\theta\|\mathbf{M}_{j,\cdot}\|_1}\right) \quad (16)$$

where $\text{med}(\cdot)$ is the median operator.

By applying (13), (8) can be approximated as

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{k=1}^{K} G_\theta(\mathbf{M}^k, \mathbf{W}^k, \mathbf{X}^k) + \lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^\top\|_{1,2} \quad (17)$$

where $G_\theta(\mathbf{M}^k, \mathbf{W}^k, \mathbf{X}^k) = \sum_{i=1}^{n} g_\theta(\mathbf{M}^k, \mathbf{w}_i^k, \mathbf{x}_i^k)$ is the cost function for the kth view of the n particles.

Let us denote by

$$\ell(\mathbf{P}, \mathbf{Q}) = \sum_{k=1}^{K} G_\theta(\mathbf{M}^k, \mathbf{W}^k, \mathbf{X}^k) \quad (18)$$

$$r(\mathbf{P}, \mathbf{Q}) = \lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^\top\|_{1,2}. \quad (19)$$

Note that now the objective function in (17) is a composite function of two parts, a differential empirical loss function $\ell(\mathbf{P}, \mathbf{Q})$ and a convex nonsmooth regularization $r(\mathbf{P}, \mathbf{Q})$, which has been extensively studied [5], [18], [44]. The APG method [44] is employed because of its well-known efficiency. In contrast to traditional subgradient-based methods that converge at sublinear rate, APG can obtain the globally optimal solution at quadratic convergence rate, which means APG achieves $O(1/m^2)$ residual from the optimal solution after m iterations.

We can apply the composite gradient mapping [5] to (17) and construct the following function:

$$\begin{aligned}\Phi(\mathbf{P}, \mathbf{Q}; \mathbf{R}, \mathbf{S}) = \ell(\mathbf{R}, \mathbf{S}) &+ \langle\nabla_{\mathbf{R}}\ell(\mathbf{R}, \mathbf{S}), \mathbf{P} - \mathbf{R}\rangle \\ &+ \langle\nabla_{\mathbf{S}}\ell(\mathbf{R}, \mathbf{S}), \mathbf{Q} - \mathbf{S}\rangle + \frac{\eta}{2}\|\mathbf{P} - \mathbf{R}\|_F^2 \\ &+ \frac{\eta}{2}\|\mathbf{Q} - \mathbf{S}\|_F^2 + r(\mathbf{P}, \mathbf{Q}). \end{aligned} \quad (20)$$

In $\Phi(\mathbf{P}, \mathbf{Q}; \mathbf{R}, \mathbf{S})$ comprises the regularization term $r(\mathbf{P}, \mathbf{Q})$ and the approximation of $\ell(\mathbf{P}, \mathbf{Q})$ by the first order Taylor expansion at point $(\mathbf{R}, \mathbf{S})$ regularized as the squared Euclidean distance between $(\mathbf{P}, \mathbf{Q})$ and $(\mathbf{R}, \mathbf{S})$, where η is a parameter controlling the step penalty and $\nabla_{\mathbf{R}}\ell(\mathbf{R}, \mathbf{S})$ and $\nabla_{\mathbf{S}}\ell(\mathbf{R}, \mathbf{S})$ denote the partial derivatives of $\ell(\mathbf{R}, \mathbf{S})$ with respect to $\mathbf{R}$ and $\mathbf{S}$. Recall that $\nabla_{\mathbf{R}}\ell(\mathbf{R}, \mathbf{S}) = \nabla_{\mathbf{S}}\ell(\mathbf{R}, \mathbf{S}) = \nabla_{\mathbf{W}}\ell(\mathbf{R}, \mathbf{S})$, so the partial derivatives $\nabla_{\mathbf{R}}\ell(\mathbf{R}, \mathbf{S})$ and $\nabla_{\mathbf{S}}\ell(\mathbf{R}, \mathbf{S})$ can be computed in closed-form by (15).

In the mth APG iteration, $(\mathbf{R}^{m+1}, \mathbf{S}^{m+1})$ is computed as a linear combination of $(\mathbf{P}^m, \mathbf{Q}^m)$ and $(\mathbf{P}^{m-1}, \mathbf{Q}^{m-1})$, so $(\mathbf{R}^{m+1}, \mathbf{S}^{m+1})$ stores the historical aggregation of $(\mathbf{P}, \mathbf{Q})$ in the previous iterations, which is conventionally called aggregation step. As suggested in [44], we set

$$\begin{aligned}\mathbf{R}^{m+1} &= \mathbf{P}^m + \alpha_m\left(\frac{1 - \alpha_{m-1}}{\alpha_{m-1}}\right)(\mathbf{P}^m - \mathbf{P}^{m-1}) \\ \mathbf{S}^{m+1} &= \mathbf{Q}^m + \alpha_m\left(\frac{1 - \alpha_{m-1}}{\alpha_{m-1}}\right)(\mathbf{Q}^m - \mathbf{Q}^{m-1}) \end{aligned} \quad (21)$$

where α_m can be set to $\alpha_0 = 1$ for $m = 0$ and $\alpha_m = 2/m + 3$ for $m \ge 1$, and $\mathbf{P}^0$, $\mathbf{Q}^0$, $\mathbf{R}^1$ and $\mathbf{S}^1$ are all set to zero matrix for the initialization. Once given the aggregation $(\mathbf{R}^m, \mathbf{S}^m)$, the solution for the mth iteration is obtained by computing the following proximal operator:

$$(\mathbf{P}^m, \mathbf{Q}^m) = \arg\min_{\mathbf{P},\mathbf{Q}} \Phi(\mathbf{P}, \mathbf{Q}; \mathbf{R}^m, \mathbf{S}^m). \quad (22)$$

With simple manipulations, the optimization problem (22) can be decomposed into two subproblems for $\mathbf{P}$ and Q, respectively, as

$$\mathbf{P}^m = \arg\min_{\mathbf{P}} \frac{1}{2}\|\mathbf{P} - \mathbf{U}^m\|_F^2 + \frac{\lambda_1}{\eta}\|\mathbf{P}\|_{1,2} \quad (23)$$

$$\mathbf{Q}^m = \arg\min_{\mathbf{Q}} \frac{1}{2}\|\mathbf{Q} - \mathbf{V}^m\|_F^2 + \frac{\lambda_2}{\eta}\|\mathbf{Q}^\top\|_{1,2} \quad (24)$$

where $\mathbf{U}^m = \mathbf{R}^m - 1/\eta\nabla_{\mathbf{R}}\ell(\mathbf{R}^m, \mathbf{S}^m)$ and $\mathbf{V}^m = \mathbf{S}^m - 1/\eta\nabla_{\mathbf{S}}\ell(\mathbf{R}^m, \mathbf{S}^m)$.

Following the decomposition, an efficient closed-form solution can be attained, respectively, for each row of $\mathbf{P}^m$ and each column of $\mathbf{Q}^m$ in the above subproblems (23) and (24) according to [53]

$$\begin{aligned}\mathbf{P}_{j,\cdot}^m &= \max\left(0, 1 - \frac{\lambda_1}{\eta\left\|\mathbf{U}_{j,\cdot}^m\right\|}\right)\mathbf{U}_{j,\cdot}^m \\ \mathbf{Q}_{\cdot,i}^m &= \max\left(0, 1 - \frac{\lambda_2}{\eta\left\|\mathbf{V}_{\cdot,i}^m\right\|}\right)\mathbf{V}_{\cdot,i}^m \end{aligned} \quad (25)$$

where $\mathbf{P}_{j,\cdot}^m$ denotes the jth row of $\mathbf{P}^m$ and $\mathbf{Q}_{\cdot,i}^m$ denotes the ith column of $\mathbf{Q}^m$. Finally, the solution of (17) can be obtained by iteratively computing (25) and updating $(\mathbf{U}^m, \mathbf{V}^m)$ until the convergence of $(\mathbf{P}, \mathbf{Q})$. The procedure of the presented algorithm is summarized in the Algorithm 1.

E. Outlier Rejection

Although a majority of particles will share the same dictionary basis, some outlier tasks may exist. The proposed MTMVT in (6) is capable of capturing the outliers by introducing the coefficient matrix $\mathbf{Q}$. In particular, if the sum of the L1 norm of the coefficients for the corresponding ith particle is larger than an adaptive threshold γ, as

$$\sum_{k=1}^{K} \left|\mathbf{Q}_i^k\right| > \gamma \quad (26)$$

Algorithm 1 Optimization Algorithm

Input: Features of K views for n candidate samples $\mathbf{X}$, dictionary $\mathbf{M}$, $\mathbf{P}^0$ and $\mathbf{Q}^0$, η, λ_1, λ_2

Output: $\mathbf{P}$, $\mathbf{Q}$

Initial $m = 1$, $\mathbf{P}^0 = 0$, $\mathbf{Q}^0 = 0$, $\mathbf{R}^1 = 0$, $\mathbf{S}^1 = 0$
while *not converged* **do**
  Compute $\nabla_{\mathbf{R}}\ell(\mathbf{R}^m, \mathbf{S}^m)$ and $\nabla_S\ell(\mathbf{R}^m, \mathbf{S}^m)$ using Equation (15)
  Compute $\mathbf{U}^m = \mathbf{R}^m - \frac{1}{\eta}\nabla_{\mathbf{R}}\ell(\mathbf{R}^m, \mathbf{S}^m)$
  Compute $\mathbf{V}^m = \mathbf{S}^m - \frac{1}{\eta}\nabla_{\mathbf{S}}\ell(\mathbf{R}^m, \mathbf{S}^m)$
  Compute $\mathbf{P}^m$ and $\mathbf{Q}^m$ using Equation (25)
  $\alpha_m = \frac{2}{m+3}$
  $\mathbf{R}^{m+1} = \mathbf{P}^m + \alpha_m(\frac{1-\alpha_{m-1}}{\alpha_{m-1}})(\mathbf{P}^m - \mathbf{P}^{m-1})$
  $\mathbf{S}^{m+1} = \mathbf{Q}^m + \alpha_m(\frac{1-\alpha_{m-1}}{\alpha_{m-1}})(\mathbf{Q}^m - \mathbf{Q}^{m-1})$
  $m = m + 1$
end while

Fig. 4. Examples of detected outliers. The green bounding boxes denote the outliers and the red bounding box denotes the tracked target. The outliers are detected out of 400 sampled particles. There are two outliers in the left frame and six outliers in the right frame.

where $\mathbf{Q}_i^k$ is the ith column of $\mathbf{Q}^k$, then it will be identified as an outlier and its observation likelihood will be set to zero. Therefore, the outliers will be ignored in the particle resampling process and the samples will be more efficiently used to focus on locating the target position. By denoting the number of detected outliers as n_o, the threshold γ is updated as follows:

$$\begin{cases} \gamma_{\text{new}} = \gamma_{\text{old}}\kappa, & n_o > N_o \\ \gamma_{\text{new}} = \gamma_{\text{old}}/\kappa, & n_o = 0 \\ \gamma_{\text{new}} = \gamma_{\text{old}}, & 0 < n_o \leq N_o \end{cases} \tag{27}$$

where κ is a scaling factor and N_o is a predefined threshold for the number of outliers. We select $\gamma = 1$, $\kappa = 1.2$ and $N_o = 20$ based on experiments. Fig. 4 shows examples with detected outliers.

F. Tracking Using Robust Multitask Multiview Sparse Representation

In reference to the tracking result, the observation likelihood of the tracking candidate i is defined as

$$p_i = \frac{1}{\Gamma}\exp\left\{-\alpha\sum_{k=1}^{K}\left\|\mathbf{D}^k\mathbf{A}_i^k - \mathbf{X}_i^k\right\|^2\right\} \tag{28}$$

where $\mathbf{A}_i^k \in \mathbb{R}^N$ is the coefficients of the ith candidate corresponding to the target templates of the kth view. The tracking result is the particle that has the maximum observation likelihood. It should be noted that both MTMVTLS and MTMVTLAD employ (28) to estimate the observation likelihood although different criterion are used to learn the sparse representation.

Algorithm 2 Tracking via Robust Multitask Multiview Sparse Representation

Input: Particle set $\mathcal{Y}_{t-1} = \{\mathbf{y}_{t-1}^i\}_{i=1}^n$, current complete dictionary $\mathbf{M}_{t-1} = \{\mathbf{M}_{t-1}^1, \cdots, \mathbf{M}_{t-1}^K\}$ for K views

Output: Estimated target $\mathbf{y}_t^*$, particle set $\mathcal{Y}_t = \{\mathbf{y}_t^i\}_{i=1}^n$, updated complete dictionary $\mathbf{M}_t = \{\mathbf{M}_t^1, \cdots, \mathbf{M}_t^K\}$ for K views

/*$\mathcal{Y}_t \leftarrow \mathcal{Y}_{t-1}$ */
for $k = 1$ to n **do**
  Draw particle $\mathbf{y}_t^i$ from $\mathbf{y}_{t-1}^i$ using the state transition distribution
end for
for $k = 1$ to K **do**
  Extract the features $\mathbf{X}^k$ according to $\mathcal{Y}_t$
end for
Estimate the robust joint sparse representation $\mathbf{W}, \mathbf{P}, \mathbf{Q}$ using (6)
Detect outliers using (26) and set $p_i = 0$ for all outliers
For the remaining particles, compute p_i using (28))
Find the best candidate $\mathbf{y}_t^*$ using $\arg\max_i p_i$
$\mathbf{M}_t \leftarrow$ Update templates
/*Resampling*/
$\mathcal{Y}_t \leftarrow$ Resample $\{\mathbf{y}_t^i\}_{i=1}^n$ with respect to $\{p_i\}_{i=1}^n$

G. Template Update

In the course of tracking, object appearance remains the same only episodically, but eventually the template is no longer an accurate model of the object appearance. To handle appearance variations, the target dictionary $\mathbf{D}$ is progressively updated using an approach similar to [3] and [4]. In particular, each target template in $\mathbf{D}$ is assigned a weight which represents its importance. At each frame, the norm of the learned coefficients $\mathbf{A}_i^k$s for the target particle is used to update the recorded weight of each template in $\mathbf{D}$. Once the angle between the tracked target and the most representative template (the template with the largest coefficient norm) is larger than a predefined threshold β, the template with the smallest recorded weight is replaced by the tracked target. We summarize the proposed tracking algorithm in Algorithm 2.

V. Experiments

In this section, we introduce the implementation details of the proposed trackers and report the experimental results by extensively evaluating the proposed tracking methods on numerous video sequences[2] including a comprehensive tracking benchmark [2], which is recently published

[2]Some of the sequences are publicly available in the following websites: http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml; http://www.cs.toronto.edu/~dross/ivt/; http://cv.snu.ac.kr/research/~vtd/; http://www4.comp.polyu.edu.hk/~cslzhang/CT/CT.htm [54]; http://www.eng.tau.ac.il/~oron/LOT/LOT.html [55]; http://lrs.icg.tugraz.at/research/houghtrack/ [56]

TABLE I
AVERAGE OVERLAP AND SUCCESS RATES (PERCENTAGES)

Sequence	Struck	L1T	MTT	IVT	VTD	MIL	MTMVTLS	MTMVTLAD
Shaking	0.31 (0.15)	0.58 (0.59)	0.30 (0.12)	0.02 (0.01)	0.72 (0.96)	0.57 (0.58)	**0.75 (0.99)**	**0.76 (0.99)**
Kitesurf	0.17 (0.23)	0.49 (0.61)	0.31 (0.32)	0.20 (0.26)	0.03 (0.05)	0.74 (0.89)	**0.70 (0.95)**	**0.71 (0.95)**
Girl	0.73 (0.96)	0.60 (0.76)	**0.71 (0.98)**	0.34 (0.28)	0.34 (0.30)	0.37 (0.24)	**0.74 (0.97)**	0.70 (0.95)
David1	0.64 (0.83)	0.36 (0.36)	0.51 (0.53)	0.57 (0.55)	0.27 (0.22)	0.29 (0.05)	**0.72 (0.97)**	**0.72 (0.97)**
Faceocc2	**0.79 (1.00)**	0.68 (0.74)	**0.80 (1.00)**	0.68 (0.76)	0.60 (0.65)	0.72 (0.99)	0.77 (0.98)	0.77 (0.98)
Jumping	0.64 (0.83)	0.10 (0.07)	0.23 (0.15)	0.36 (0.47)	0.10 (0.10)	0.52 (0.45)	**0.70 (0.95)**	**0.71 (0.93)**
Gym	0.62 (0.75)	0.03 (0.03)	0.20 (0.16)	0.20 (0.19)	**0.66 (0.90)**	0.43 (0.43)	0.58 (0.78)	**0.62 (0.85)**
Bolt	0.49 (0.46)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.14 (0.09)	0.62 (0.82)	**0.67 (0.84)**	**0.62 (0.86)**
Skating1	0.46 (0.57)	0.29 (0.33)	0.17 (0.15)	0.06 (0.06)	**0.67 (0.92)**	0.19 (0.19)	0.67 (0.89)	**0.64 (0.91)**
Singer1	0.31 (0.22)	0.65 (0.92)	**0.77 (1.00)**	0.44 (0.37)	**0.79 (1.00)**	0.31 (0.21)	0.55 (0.50)	0.68 (0.94)
Basketball	0.38 (0.45)	0.12 (0.02)	0.17 (0.20)	0.16 (0.08)	**0.72 (0.94)**	0.26 (0.24)	0.62 (0.78)	**0.68 (0.92)**
David2	0.68 (0.91)	0.39 (0.52)	0.42 (0.56)	0.26 (0.35)	0.37 (0.51)	0.37 (0.46)	**0.69 (0.94)**	**0.67 (0.92)**
DH	0.45 (0.54)	0.14 (0.09)	0.47 (0.62)	0.07 (0.08)	0.53 (0.49)	0.53 (0.46)	**0.59 (0.83)**	**0.56 (0.67)**
Shop	0.45 (0.36)	0.76 (0.99)	**0.79 (0.99)**	0.57 (0.41)	0.32 (0.36)	0.25 (0.34)	**0.76 (1.00)**	0.76 (0.99)
Animal	0.56 (0.73)	0.05 (0.06)	0.60 (0.80)	0.03 (0.04)	**0.70 (0.97)**	0.37 (0.27)	0.58 (0.87)	**0.61 (0.89)**
Bird2	0.59 (0.55)	0.53 (0.57)	0.09 (0.09)	0.39 (0.48)	0.10 (0.13)	0.35 (0.19)	**0.67 (0.91)**	**0.71 (0.95)**
Tiger1	0.38 (0.47)	0.18 (0.15)	0.32 (0.33)	0.11 (0.13)	0.13 (0.12)	0.59 (0.70)	**0.71 (0.94)**	**0.71 (0.93)**
Lemming	0.52 (0.62)	0.11 (0.15)	0.27 (0.35)	0.26 (0.36)	0.45 (0.56)	0.35 (0.37)	**0.62 (0.81)**	**0.67 (0.89)**
Sylv	**0.75 (0.99)**	0.66 (0.91)	**0.76 (1.00)**	0.55 (0.76)	0.66 (0.78)	0.76 (0.96)	0.72 (0.94)	0.75 (0.96)
Cliffbar	0.34 (0.41)	0.42 (0.48)	0.59 (0.62)	0.34 (0.44)	0.48 (0.67)	0.53 (0.52)	**0.70 (0.88)**	**0.63 (0.80)**
Car4	0.49 (0.38)	0.54 (0.46)	0.74 (1.00)	0.85 (1.00)	0.53 (0.56)	0.28 (0.27)	**0.86 (1.00)**	**0.86 (1.00)**
Average	0.51 (0.59)	0.37 (0.42)	0.44 (0.52)	0.31 (0.34)	0.44 (0.54)	0.45 (0.46)	**0.68 (0.89)**	**0.69 (0.92)**

The quantitative comparison on the 21 sequences. The figures outside the brackets and the figures inside the brackets are the average overlap and the success rates, respectively. The **RED** number indicates the best performance, while the **GREEN** indicates the second best. The ranking is primarily based on the success rates. If the success rates scores are equal, then we compare the average overlap.

in CVPR2013.[3] We also evaluated MTMVTLAD on the ALOV++ [57], which is recommended by an anonymous reviewer.

A. Implementation Details

To evaluate the effectiveness of the MTMVTLS and MTMVTLAD, they were implemented using four complementary features as four different views. We employed four popular features: 1) color histograms; 2) intensity; 3) histograms of oriented gradients (HOGs) [58]; and 4) local binary patterns (LBPs) [59]. The HOG is a gradient-based feature that captures edge distribution of an object. LBP is powerful for representing object texture. Moreover, to ensure the quality of extracted features, a simple but effective illumination normalization method used in [60] is applied before the intensity feature extraction. The unit-norm normalization is applied to the extracted feature vector of each particle view, respectively.

For all reported experiments, we set $\lambda_1 = \lambda_2 = 0.5$ for MTMVTLS, $\lambda_1 = 1.25$, $\lambda_2 = 1$, and $\theta = 0.1$ for MTMVTLAD, respectively. For both MTMVTLS and MTMVTLAD, we set the number of particles $n = 400$ [the same for L1 tracker (L1T) and MTT], the number of template samples $N = 10$. The template of intensity is set to one third of the size of the initial target (half size for those whose shorter side is <20). The color histogram, HOG, and LBP are extracted in a larger template that doubles the size of the intensity template. The threshold for template update β is set to 60.

Currently, the proposed tracker MTMVTLAD is implemented using MATLAB without special code optimization. The computational time of the proposed tracker mainly consists of two parts: 1) feature extraction and 2) the optimization solved by Algorithm 1. The feature extraction can be significantly accelerated using parallel programming based on GPU. However, we did not explore GPU programming, leaving this step for the future work. In Algorithm 1, the computational complexity of each iteration is dominated by the gradient computation in Step 3 of complexity $O(nKdh)$. Therefore, the runtime of tracker depends on the dimensionality of features, the number of particles, and the number of views. Practically, in the experiment on *EXTsequences* reported in Table IV, it runs at 1.8 s/frame on average on this multicore system: 2.9-GHz Intel Xeon E5-2690, 32-GB RAM.

B. Evaluation on Publicly Available Sequences

In this section, we validate the effectiveness of the proposed trackers by extensively performing the experiments on 21 publicly available sequences. All original sized images are used in contrast with resizing to the same size implemented in [28]. The titles of used sequences are listed in Table I. First, we qualitatively compare MTMVTLS and MTMVTLAD with six other popular trackers: 1) Struck [22]; 2) L1T [3]; 3) MTT [4]; 4) tracking with MIL [21]; 5) incremental learning for visual tracking (IVT) [32]; and 6) VTD [12]. It should be noted that VTD is a multiview tracker that employs hue, saturation, intensity, and edge templates for the features. We conducted the experiments by running source codes provided by the original authors. The recommended parameters are set for initialization.

The *Shaking*, *Kitesurf*, *Girl*, *Faceocc2*, *David1*, and *Jumping* sequences track human faces under different circumstances and challenges. The experimental results show that both

[3]26th IEEE Conference on Computer Vision and Pattern Recognition, 2013.

Fig. 5. Tracking results of different algorithms. Frame indexes are shown in the top left of each figure. (a) *Shaking*. (b) *Kitesurf*. (c) *Girl*. (d) *David1*. (e) *Faceocc2*. (f) *Jumping*. (g) *Gym*. (h) *Bolt*. (i) *Skating1*. (j) *Singer1*. (k) *Basketball*. (l) *David2*. (m) *DH*. (n) *Shop*.

MTMVTLS and MTMVTLAD are able to handle the scale changes, pose changes, fast motion (FM), OCC, appearance variation, illumination change, and angle variation problems encountered in face tracking tasks. For example, the *Shaking* sequence captures a person performing on stage. The task is to track his face under significant illumination changes and appearance variations. The IVT drifts from the target quickly due to the severe appearance variation. Struck and MTT are prone to drift during the illumination change. In contrast, our trackers are more robust to the illumination changes as a result of the employment of rich feature types. In the *David1* sequence, a moving face is tracked, which presents many challenges such as pose and scale changes. Compared with L1T and MTT, MTMVTLS and MTMVTLAD successfully track the target under different challenges due to the robustness of the additional features. From the experiments, we find that IVT is vulnerable to the appearance variations, while VTD is prone to drift in OCC scenarios. Some representative frames can be found in Fig. 5(a)–(f).

In Fig. 5(g)–(n), we show some example frames for a group of eight sequences, where the tasks are to track the human bodies in different settings. In particular, the *DH*, *Gym*, *Bolt*, *Skating1*, and *Basketball* sequences track fast moving human bodies in sport scenarios. In the *DH* sequence, Struck, L1T, MTT, and IVT lose the target because of the distracting background and FM. The VTD is prone to drift and only track part of the target. In the *Gym*, *Bolt*, *Skating1*, and *Basketball* sequences, the poses of targets changes rapidly and the appearance deforms frequently, which make them more challenging for existing trackers. Both L1T and IVT fail on all the sequences. Struck fails on the *Skating1* sequences due to the severe pose and illumination changes. MTT loses the targets soon on the *Bolt* and *Gym* sequences due to the DEF of the targets. The VTD succeeds in the *Skating1* and *Basketball* sequences because of the benefit of multiple types of features but drifts apart from the target in the *Bolt* sequence. In addition, VTD fails in the *David2* and *Shop* sequences with the presence of OCC. In contrast,

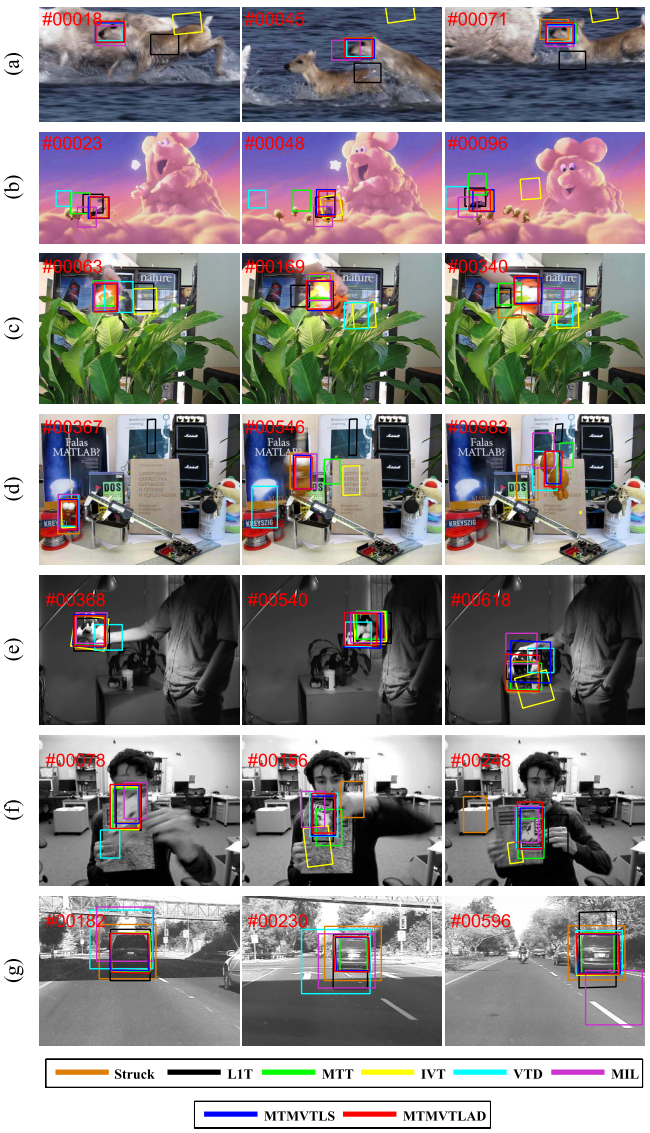


Fig. 6. Tracking results of different algorithms. Frame indexes are shown in the top left of each figure. (a) *Animal*. (b) *Bird2*. (c) *Tiger1*. (d) *Lemming*. (e) *Sylv*. (f) *Cliffbar*. (g) *Car4*.

MTMVTLS and MTMVTLAD successfully track all these targets in our experiments, which indicates the proposed tracker is not as sensitive to shape DEF as previous single-view trackers, due to the effective use of the complementary features and the capability of detecting outliers. Moreover, MTMVTLAD appears to be more robust than MTMVTLS in the *Singer1* and *Basketball* sequences, where MTMVTLS tends to include some background into the bounding box.

In the last group of seven sequences, the tasks are varying from tracking animal in wild or car in road, to tracking moving dolls or object indoor. Some representative frames of these sequences are shown in Fig. 6(a)–(g). The *Animal* sequence shown in Fig. 6(a) tracks the head of a fast running deer. The main challenges are the FM and BC. In the *Animal* sequence, MTT, VTD, MTMVTLS, and MTMVTLAD succeed in tracking the target over the whole sequence, while MIL and Struck are only able to track a part of the target though does not lose it. The IVT gradually drifts from the target after the third frame and totally loses the target in the sixth frame. L1T fails in the presence of FM and motion blur (MB). The multitask manner appears to make MTT, MTMVTLS, and MTMVTLAD more robust than L1T. In the *Tiger1*, *Lemming*, and *Sylv* sequences, the tasks are to track moving dolls in indoor scenes. Almost all the trackers compared can track the doll in the earlier part of the *Sylv* sequence. However, IVT loses the target when it undergoes pose changes. The *Tiger1* and *Lemming* sequences are much harder due to the significant appearance changes, OCC, in-plane rotations (IPRs), and distractive background, so all trackers continuously lock in the background except MTMVTLS and MTMVTLAD. Our trackers faithfully tracks the dolls, and obtain the best performances. Some example shots of these three sequences are shown in Fig. 6(c)–(e). In the *Car4* sequence, MTT, IVT, MTMVTLS, and MTMVTLAD perfectly track the moving car despite the dramatic illumination and scale changes, which are shown in Fig. 6(g). In contrast, VTD and MIL lose the target and L1T tends to include much of the background area into the bounding box when the car is moving under the bridge, which leads to significant illumination changes.

To quantitatively evaluate the performance of each tracker, we compute the bounding box overlap S_o of r_t and r_g in each frame, where r_t is the bounding box outputted by a tracker and r_g is the ground truth bounding box. The bounding box overlap is defined as $S_o = |r_t \cap r_g|/|r_t \cup r_g|$, where $\cap$ and $\cup$ denote the intersection and union of two regions, respectively. For more comprehensive comparison, we also compute the success rate R_o by counting the percentage of frames whose overlap S_o is bigger than a threshold $t_o = 0.5$. The average overlap as well as the success rate R_o of the eight comparative trackers on the 21 sequences are summarized in Table I. It can be clearly seen that the proposed MTMVTLS and MTMVTLAD achieve the best average performances over all the tested sequences compared with the other five popular trackers. Moreover, MTMVTLAD appears to be more robust than MTMVTLS and achieves a slightly better performance in this data set.

C. Evaluation on Noisy Sequences

In the previous section, we compare the proposed trackers with five other trackers on 21 challenging sequences. Most of these sequences are captured under restricted environment without the contamination of noise. However, in the real-world setting, the video images may be contaminated by various of noise, which makes the tracking task even harder. In this section, we tested the proposed trackers on the sequences contaminated by different types of synthetic noise and real-world noise, e.g., snow and rain. The composition of the tested sequences and the qualitative comparison are detailed below.

1) Evaluation on Noisy Video Sequences: In reality, the targets and the scene can be contaminated by many kinds of noise. To evaluate robustness to noise, we contaminated the above 21 sequences with different types of synthetic noise including Gaussian, Laplace, and salt and pepper noise to simulate the noise in real world and evaluate our proposed tracker on them. By synthesizing noisy images, we can choose different additive noise and control the noise level at the same time so we can better understand robustness of our method

TABLE II
PARAMETERS OF SYNTHETIC DATA SET

Dataset	Mean	Variance	Noise density
Gaussian 1	0.02	0.01	–
Gaussian 2	0.02	0.05	–
Gaussian 3	0.02	0.1	–
Gaussian 4	0.02	0.5	–
Laplace 1	0.02	0.01	–
Laplace 2	0.02	0.05	–
Laplace 3	0.02	0.1	–
Laplace 4	0.02	0.5	–
salt & pepper 1	–	–	0.01
salt & pepper 2	–	–	0.05
salt & pepper 3	–	–	0.1
salt & pepper 4	–	–	0.5

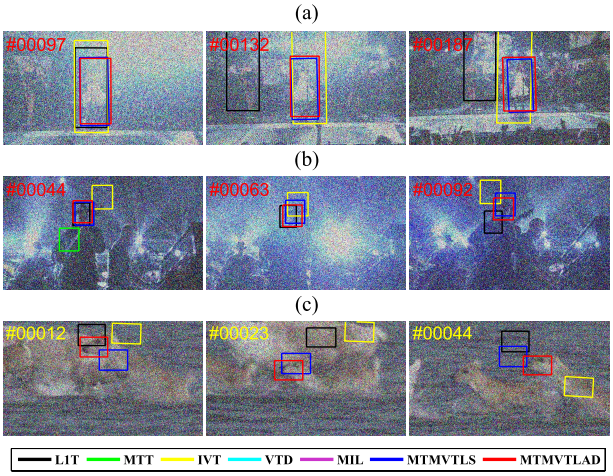


Fig. 7. Some examples of the contaminated sequences. (a) *Singer1 (Gaussian 4)*. (b) *Shaking (Laplace 4)*. (c) *Animal (salt and pepper 4)*.

TABLE III
AVERAGE SUCCESS RATES IN THE CONTAMINATED DATA SETS

Sequence	L1T	MTT	IVT	VTD	MIL	MTMVTLS	MTMVTLAD
Gaussian1	0.36	0.50	0.36	0.53	0.41	**0.73**	**0.83**
Gaussian2	0.42	0.35	0.38	0.49	0.37	**0.69**	**0.70**
Gaussian3	0.34	0.29	0.37	0.47	0.42	**0.65**	**0.66**
Gaussian4	0.15	0.03	0.29	0.29	0.41	**0.42**	**0.45**
Laplace1	0.45	0.45	0.34	0.54	0.40	**0.83**	**0.80**
Laplace2	0.36	0.37	0.35	0.53	0.42	**0.73**	**0.73**
Laplace3	0.34	0.31	0.37	0.49	0.41	**0.59**	**0.63**
Laplace4	0.29	0.11	0.29	0.39	0.43	**0.51**	**0.49**
Sault & Peppr1	0.38	0.58	0.42	0.54	0.44	**0.87**	**0.88**
Sault & Peppr2	0.36	0.43	0.39	0.45	0.44	**0.82**	**0.82**
Sault & Peppr3	0.33	0.36	0.39	0.51	0.42	**0.68**	**0.78**
Sault & Peppr4	0.21	0.04	0.25	0.26	**0.43**	**0.38**	0.37
Average	0.33	0.32	0.35	0.46	0.42	**0.66**	**0.68**

The **RED** number indicates the best performance, while the **Green** indicates the second best.

to noise. Similar approach that adds synthetic noise to the video sequences to test robustness of the tracking method has been adopted in [61]. Each type of noise is generated by four sets of different parameters indicating four light-to-heavy levels, and 12 additional groups of sequences are created, i.e., 252 (21×12) sequences in total. The parameters to generate the synthetic noise are summarized in Table II. Some examples of the contaminated sequences as well as the qualitative results are shown in Fig. 7. To quantitatively compare the tracking performance on the 12 data sets, we summarized the average success rates in Table III. From Table III, we can see all trackers to some extent degraded in terms of performance on these noisy data sets. In particular, L1T and MTT appear to be more sensitive to the noise due to the use of single type of feature and the heuristic strategy used for template update. Interestingly, IVT may have slightly better performance in some of the contaminated video data sets comparing with the performance on the original data set. This phenomenon, in which addition of some noise to the input data during training may sometimes improve the generalization and therefore boost the performance, has been noted in [62]. VTD, MTMVTLS, and MTMVTLAD achieve better performances on average compared with L1T, MTT, and IVT because of the adoption of multiple types of features. MIL is comparable with VTD in terms of average performance since MIL appears to be insensitive to the noise levels. This suggests that the Haar feature associated with MIL is just robust to our synthetic noise. On average, MTMVTLAD achieves better performance than MTMVTLS and obtain the best average performance over all comparative trackers and 12 tested data sets.

2) Evaluation on EXTsequences: To further evaluate the robustness of the proposed tracker to noise in real world, we collect nine more video sequences, which are taken in extreme weather. We call this set of sequences as *EXTsequences*. The first group of six sequences deals with tracking moving vehicles in bad weather (e.g., storm, snow) and associated challenges, e.g., OCC by the windshield wiper, illumination changes, and scale changes. Some example frames as well as the tracking results can be found in Fig. 8(a)–(f). The second group of three sequences deals with tracking human faces undergoing appearance variation due to IPR. For some example frames [Fig. 8(g) and (h)]. In these sequences, the visibility of faces is severely affected by snowstorm and spray. However, MTMVTLS and MTMVTLAD are able to track the targets faithfully. To quantitatively evaluate the performance, we again summarize the average overlap and the success rates in Table IV. The quantitative results demonstrate that MTMVTLAD is robust to noise and it obtains the best average performance comparing with other state-of-the-art trackers.

D. Evaluation on CVPR2013 Tracking Benchmark

To evaluate the overall performance of the proposed tracker under different scenarios and demonstrate the improvement with respect to previous methods, in this section, we conduct the experiments on the CVPR tracking benchmark [2] and compare the proposed tracker with numerous state-of-the-art trackers and its own baseline methods. The CVPR tracking benchmark is a comprehensive tracking benchmark, which is designed for tracking performance evaluation. It consists of 50 fully annotated sequences. Each sequence is tagged with the attributes indicating to the presence of different challenges: 1) illuminative variation; 2) scale variation; 3) OCC; 4) DEF; 5) MB; 6) FM; 7) IPR; 8) out-of-plane-rotation (OPR); 9) out-of-view; 10) BCs; and 11) low resolution. To evaluate the strength and weakness of different methods, the sequences are categorized according to the attributes, and 11 challenge subsets are created. In [2], the evaluation is based on two kinds of metrics, i.e., the precision plot and success plot. To obtain the precision plot,

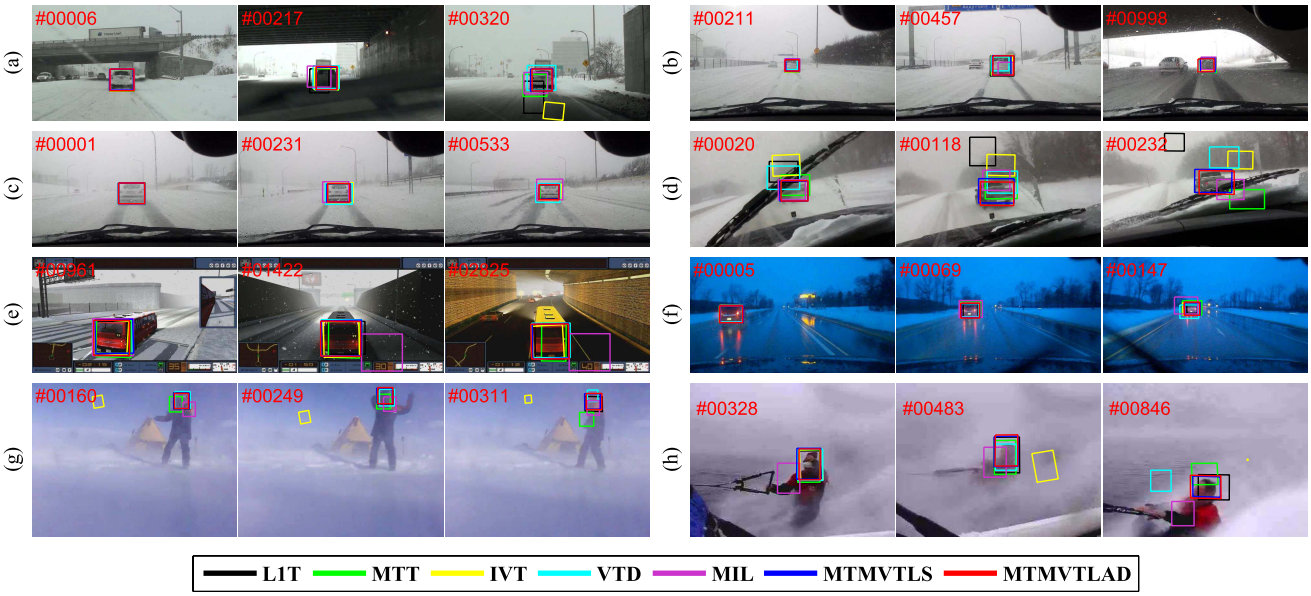


Fig. 8. Tracking results of different algorithms on *EXTsequences*. Frame indexes are shown in the top left of each figure. (a) *Storm*. (b) *Winter1*. (c) *Winter2*. (d) *Snow1*. (e) *Snow2*. (f) *DarkCar*. (g) *Antarctica*. (h) *Skiing1 and 2*.

TABLE IV
AVERAGE OVERLAP AND SUCCESS RATES (PERCENTAGES)

Sequence	Frames	L1T	MTT	IVT	VTD	MIL	MTMVTLS	MTMVTLAD
Storm	455	0.44 (0.33)	0.63 (0.54)	0.45 (0.48)	0.83 (0.98)	0.56 (0.46)	**0.86 (1.00)**	**0.85 (1.00)**
Winter1	998	0.81 (1.00)	**0.84 (1.00)**	0.81 (1.00)	0.67 (0.76)	0.60 (0.62)	0.81 (1.00)	**0.83 (1.00)**
Winter2	533	0.86 (1.00)	0.85 (1.00)	0.73 (1.00)	0.68 (1.00)	0.63 (0.88)	**0.88 (1.00)**	**0.88 (1.00)**
Snow1	290	0.05 (0.06)	0.35 (0.46)	0.20 (0.23)	0.24 (0.08)	0.47 (0.48)	**0.72 (0.98)**	**0.80 (0.96)**
Snow2	4435	0.86 (1.00)	0.72 (1.00)	0.74 (0.85)	0.86 (1.00)	0.24 (0.29)	**0.88 (1.00)**	**0.89 (1.00)**
DarkCar	147	**0.76 (1.00)**	0.53 (0.39)	0.55 (0.50)	0.62 (0.56)	0.45 (0.32)	**0.62 (0.89)**	0.61 (0.83)
Antarctica	580	0.50 (0.58)	0.24 (0.25)	0.01 (0.01)	0.61 (0.75)	0.17 (0.10)	**0.60 (0.92)**	**0.62 (0.93)**
Skiing1	486	0.74 (0.99)	0.77 (0.98)	0.46 (0.67)	0.73 (0.95)	0.31 (0.36)	**0.78 (0.99)**	**0.82 (1.00)**
Skiing2	846	0.51 (0.38)	0.72 (0.89)	0.02 (0.02)	0.17 (0.21)	0.04 (0.04)	**0.76 (0.89)**	**0.74 (0.90)**
Average	-	0.61 (0.70)	0.63 (0.72)	0.44 (0.53)	0.60 (0.70)	0.39 (0.39)	**0.77 (0.96)**	**0.78 (0.96)**

The quantitative comparison on EXTsequences. The figures outside the brackets and the figures inside the brackets are the average overlap and the success rates, respectively. The **RED** number indicates the best performance, while the **GREEN** indicates the second best. The ranking is primarily based on the success rates. If the success rates scores are equal, then we compare the average overlap.

we calculate the center location error (CLE), which is the distance between the centers of the tracking result and the manually labeled ground truth for each frame. The precision plot shows the percentage of frames whose CLE is within a given threshold and uses a representative precision score for ranking by choosing an appropriate threshold ($r = 20$). Another metric is to compute the bounding box overlap S_o which has been defined in Section V-B. The number of frames whose overlap S_o is larger than the given threshold t_o is counted. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. In success plot, the ranking is based on the area under curve (AUC) instead of a specific threshold. For the comparative trackers, it currently includes 29 popular tracking algorithms, including the Struck, MTT, IVT, VTD, and MIL, which have been tested in previous sections, and the L1APG [25] (a newer version of L1T). For more details about the benchmark, we refer readers to the original paper [2].

1) Comparison With Trackers on CVPR2013 Tracking Benchmark: We run the one-pass evaluation on the benchmark using the proposed trackers MTMVTLS and MTMVTLAD and compare them with the 29 popular tracking methods previously evaluated in [2]. We also compare the proposed trackers with the latest version of LRT [63], which has similar motivation as our methods. In [63], the consensus between particles is enforced through low-rank minimization.

It should be noted that we strictly follow the protocol proposed by the authors and use the same parameters for all the sequences. For comparison, we use the online available[4] tracking results and the unified tool provided by [2] to compute the evaluation plots. For the results of [63], we downloaded the code from the authors' website.[5] In the CVPR tracking benchmark, the proposed MTMVTLAD and MTMVTLS achieve overall the best and the second best performance using the precision plot as the metric, which is shown Fig. 9. MTMVTLS and MTMVTLAD also rank in the top ten from all 32 trackers

[4]http://visual-tracking.net/

[5]http://nlpr-web.ia.ac.cn/mmc/homepage/tzzhang/Project_Tianzhu/zhang_IJCV14/RobustVisualTrackingViaConsistentLow-RankSparse.html and ran it on all sequences using the default parameters.

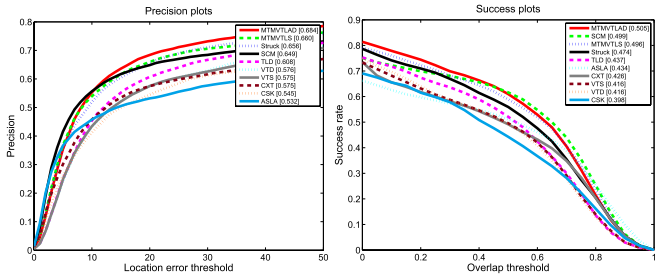


Fig. 9. Precision plots and success plots on the CVPR2013 tracking benchmark. The values appearing in the legend of the precision plot are the precision scores in the threshold of 20, while the ones in success plots are the AUC scores. Only the top 10 trackers are presented, while the other trackers can be found in [2]. The trackers appearing in the legend are Struck [22], SCM [27], TLD [33], LRT [63], VTD [12], VTS [64], CXT [65], CSK [66], and ASLA [26].

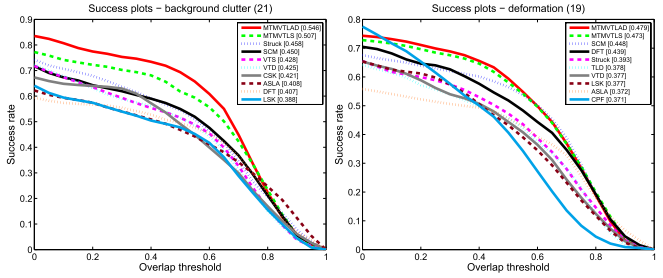
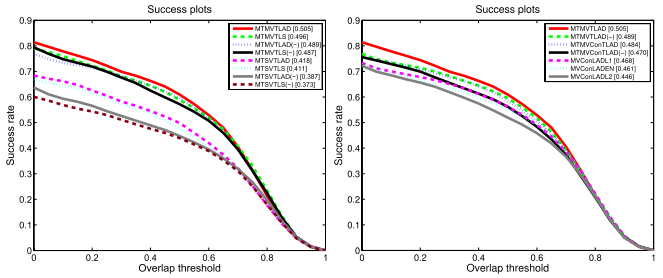


Fig. 10. Success plots for BC and DEF subsets of CVPR2013 tracking benchmark. The value appearing in the title is the number of sequences in the specific subset. The values appearing in the legend are the AUC scores. Only the top 10 trackers are presented, while the other trackers can be found in [2]. The trackers appearing in the legend are DFT [67], LSK [42], and CPF [68].

over all challenge subsets using either the measurement of precision plots or success plots. According to the results, MTMVTLS and MTMVTLAD are more robust to BC, DEF, IPR and OPR challenges comparing with other 30 trackers because the proposed methods can effectively take advantage of complementary features. Moreover, MTMVTLAD takes the first places in 6 out of 11 challenge subsets when using the success plot as the metric because LAD is advantageous to learn more appropriate representations. We show the success plots of the BC and DEF subsets in Fig. 10, but omit other figures due to the space limits.

2) Comparison With Baseline Methods: In previous sections, we have demonstrated the superior performance of MTMVTLAD comparing with MTMVTLS and other state-of-art trackers. However, it is also important to compare MTMVTLAD with its baseline variants to demonstrate the component-wise contributions to the performance of the proposed tracker.

First, we validate the improvement brought by the robust multitask multiview representation by testing it in both single-view and multiview settings and comparing it with their corresponding baseline variants. To this end, we implement two multitask single-view ($K = 1$) trackers based on (7) and denote them, respectively, as MTSVTLS and MTSVTLS($-$), where MTSVTLS($-$) is constructed by removing the functional component $\mathbf{Q}$ (no outlier handling). It should be noted that the formulation of MTSVTLS($-$) is the same



(a) (b)

Fig. 11. Success plots of MTMVTLAD and its baseline variants on the CVPR2013 tracking benchmark. The values appearing in the legend are the AUC scores. (a) Comparisons with the variants of MTMVTLAD. (b) Comparisons with trackers based on concatenated features and different regularization.

as MTT [4] since MTT is a special case of the proposed general form (9) discussed in Section IV-C. Both MTSVTLS and MTSVTLS($-$) are using intensity feature only, similar to MTT [4]. We also implement a multitask multiview tracker MTMVTLS($-$) similar to MTMVTLS but removing the functional component $\mathbf{Q}$. To test the improvement brought by the LAD formulation, we construct the corresponding LAD-based version and denote them by MTSVTLAD, MTSVTLAD($-$), and MTMVTLAD($-$), respectively. We ran these variants on the CVPR benchmark 2013 and compared MTMVTLAD with them using the success plot. As shown in Fig. 11(a), the multiview-based trackers significantly outperform the single-view-based trackers, which demonstrates the advantage of using complementary information. In addition, comparing with the trackers without the outlier handling, the trackers which explicitly consider outliers generally achieve better AUC scores, which suggests that outliers should be specifically considered during the multitask representation learning. Last but not least, the LAD-based trackers outperform the corresponding LS-based trackers, which validates the robustness of the learned representation based on LAD criterion.

As discussed previously, directly concatenating multiple features into a long feature vector is not a good way to handle multiple features. To validate this point, we concatenate multiple features and implement a baseline tracker based on the formulation (8), where we let $K = 1$. We call it as MTMVConTLAD. Using the concatenated features, we also implement several variants, including MTMVConTLAD($-$), which is the same as MTMVConTLAD but removes the functional component $\mathbf{Q}$ (no outlier handling), and three trackers MVConLADL1, MVConLADL2, MVConLADEN, which use L1, L2, and Elastic Net regularizers [69], respectively. It should be noted that all these variants can be easily implemented based on the method presented in Section IV-D, along with the soft thresholding [70] for L1 regularizer. We also ran these variants on the CVPR benchmark 2013 and compare MTMVTLAD with them using the success plot, which is shown in Fig. 11(b). It shows MTMVConTLAD and MTMVConTLAD($-$) outperform MVConLADL1, MVConLADL2, MVConLADEN, which validates expected improvements brought by considering all particles in a

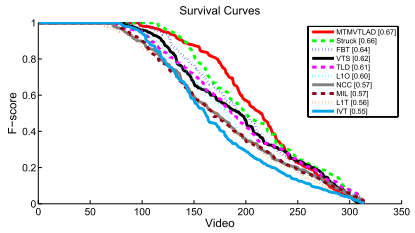


Fig. 12. Survival curves for top 10 trackers in ALOV++ data set. The average F-scores over all sequences are specified in the legend. The trackers appearing in the legend are Struck [22], FBT [71], VTS [64], TLD [33], L1O [43], NCC [72], MIL [21], L1T [3], and IVT [32].

multitask setting. In addition, MTMVConTLAD does not perform as good as the proposed MTMVTLAD, which suggests that the multiple features should not be concatenated directly.

E. Evaluation on ALOV++ Data Set

Recently, Smeulders *et al.* [57] have developed the Amsterdam Library of Ordinary Videos data set, named ALOV++, which consists of 14 challenge subsets, 315 sequences of which focuses on systematically and experimentally evaluating trackers' robustnesses in a large variety of situations including light changes, low contrast, OCC, and so on. In [57], survival curves based on F-score were proposed to evaluate trackers' robustnesses. To obtain the survival curve of a tracker, a F-score for each video is computed as $F = 2(\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall})$, where precision $= n_{\text{tp}}/(n_{\text{tp}} + n_{\text{fp}})$, recall $= n_{\text{tp}}/(n_{\text{tp}} + n_{\text{fn}})$, and n_{tp}, n_{fp}, n_{fn} denote the number of true positives (overlap $S_o >= 0.5$), false positives, and false negatives in a video. A survival curve shows the performance of a trackers on all videos in the data set. The videos are sorted according to the F-score. By sorting the videos, the graph gives a comparative view in cumulative rendition of the quality of the tracker on the whole data set. We refer the reader to the original paper [57] and the author's website[6] for details about the data set and the evaluation tools.

To evaluate the proposed MTMVTLAD tracker on ALOV++ data set, we downloaded the videos and ground truth data from the website,[6] and ran MTMVTLAD on all of the 315 sequences using the ground truth of the first frame as initialization. We compare our tracker with 19 popular trackers[7] evaluated in [57]. We show the survival curves of the top ten trackers and the average F-scores over all sequences in Fig. 12. As shown in the figure, the average F-score of MTMVTLAD in ALOV++ data set is 0.67, which is better than Struck [22] with 0.66 and is also much better than 0.62 of VTS [64], another multiview-based tracker. In ALOV++ data set, MTMVTLAD achieves the best overall performance over 20 compared trackers using the evaluation metric of average F-score. For better understanding of the overall performance

[6]http://imagelab.ing.unimore.it/dsm/

[7]Please refer to [57] and the references within for the details about the compared trackers. The evaluation results of these trackers were obtained from the authors of [57].

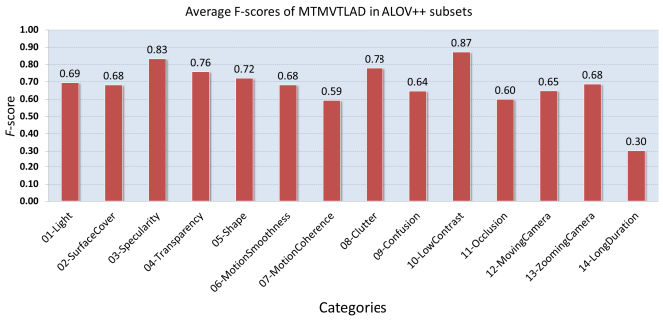


Fig. 13. Respective average F-scores of the proposed MTMVTLAD tracker in 14 ALOV++ challenge subsets.

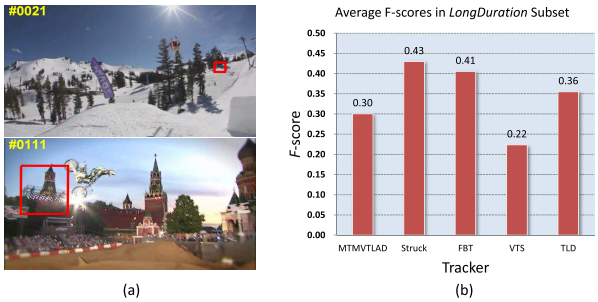


Fig. 14. Failure cases of MTMVTLAD. (a) Failure cases on *Skiing* and *MotorRolling* sequences of CVPR2013 benchmark. (b) Failure case in the *LongDuration* subset of ALOV++ data set. The numbers appear on the top of each bar is the tracker's average F-score over 10 sequences of the *LongDuration* subset.

of the proposed tracker, we also report the respective average F-scores of MTMVTLAD in 14 ALOV++ challenge subsets in Fig. 13.

F. Discussion

The experimental results demonstrate robust tracking performance of our approach. However, our tracker can indeed fail in some scenarios, which are shown in Fig. 14. Our tracker can fail when the objects undergo very large pose transformation caused by rotation or scale changes. For example, Fig. 14(a) shows two failure cases of MTMVTLAD on *Skiing* and *MotorRolling* sequences of CVPR2013 benchmark where MTMVTLAD loses the targets when the targets undergo rotations and/or change their appearance and scale. Another failure case of MTMVTLAD is on the *LongDuration* subset of ALOV++ data set. On this subset, the trackers run on 10 long sequences where some of targets may move completely out of the frame and then reappear. MTMVTLAD does not perform well and obtains a low F-score on this subset as shown in Fig. 14(b). It is possible that the tracker locks on an irrelevant patch when the target is fully occluded. We expect our future investigation to address this failure mode of the proposed tracker.

VI. Conclusion

In this paper, we have presented a LAD-based robust multitask multiview sparse learning method for PF-based tracking. By appropriately introducing the $l_{1,2}$ norm regularization, the

method not only exploits the underlying relationship shared by different views and different particles, but also captures the frequently emerging outlier tasks which have been previously ignored. The proposed regularized LAD problem is effectively approximated by the Nesterov's smoothing method and efficiently solved by the APG. We implemented our method using four types of complementary features, i.e., intensity, color histogram, HOG, and LBP, and extensively tested it on numerous challenging sequences including publicly available sequences, synthetic noisy sequences, real-world noisy sequences, and two comprehensive tracking data sets. The experimental results demonstrate that the proposed method is capable of taking advantage of multiview data and correctly handling the outlier tasks. Compared with several popular trackers, our tracker demonstrates superior performance. Moreover, the proposed method can potentially be extended to handle data obtained from sensors other than cameras.

ACKNOWLEDGMENT

The authors would like to thank the handling editor and anonymous reviewers for their constructive comments.

REFERENCES

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, Oct. 2013, Art. ID 58.

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.

[3] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[4] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2042–2049.

[5] Y. Nesterov, "Gradient methods for minimizing composite objective function," Center Oper. Res. Econometrics (CORE), Catholic Univ. Louvain, Louvain-la-Neuve, Belgium, CORE Discussion Paper 2007/76, 2007.

[6] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor. (2012). "MahNMF: Manhattan non-negative matrix factorization." [Online]. Available: http://arxiv.org/abs/1207.3438

[7] H. L. Harter, "The method of least squares and some alternatives: Part I," *Int. Statist. Rev.*, vol. 42, no. 2, pp. 147–174, Aug. 1974.

[8] N. Quadrianto and C. H. Lampert, "Learning multi-view neighborhood preserving projections," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, WA, USA, Jun. 2011, pp. 425–432.

[9] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.

[10] X. Mei, S. K. Zhou, and F. Porikli, "Probabilistic visual tracking via robust template matching and incremental subspace update," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 1818–1821.

[11] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[12] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2010, pp. 1269–1276.

[13] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[14] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel, "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[15] W. Du and J. Piater, "A probabilistic approach to integrating multiple cues in visual tracking," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France, Oct. 2008, pp. 225–238.

[16] W. Liu and D. Tao, "Multiview Hessian regularization for image annotation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2676–2687, Jul. 2013.

[17] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[18] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Brussels, Belgium, Dec. 2012, pp. 895–903.

[19] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, Dec. 2005.

[20] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifier: A comparison of logistic regression and naive Bayes," in *Proc. Ann. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2001, pp. 841–848.

[21] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[22] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE 13th Int. Conf. Comput. Vis.*, Colorado Springs, CO, USA, Nov. 2011, pp. 263–270.

[23] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time object tracking via online discriminative feature selection," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4664–4677, Dec. 2013.

[24] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Tracking using multilevel quantizations," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 155–171.

[25] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1830–1837.

[26] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1822–1829.

[27] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1838–1845.

[28] Z. Hong, X. Mei, D. Prokhorov, and D. Tao, "Tracking via robust multi-task multi-view joint sparse representation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 649–656.

[29] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[30] C. Xu, D. Tao, and C. Xu. (2013). "A survey on multi-view learning." [Online]. Available: http://arxiv.org/abs/1304.5634

[31] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[32] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.

[33] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 49–56.

[34] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras, "Dependent multiple cue integration for robust tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 670–685, Apr. 2008.

[35] B. Stenger, T. Woodley, and R. Cipolla, "Learning to track with multiple observers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 2647–2654.

[36] J. H. Yoon, D. Y. Kim, and K.-J. Yoon, "Visual tracking via adaptive tracker selection with multiple features," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 28–41.

[37] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Mumbai, India, Jun. 1998, pp. 232–237.

[38] J. Kwon and K. M. Lee, "Tracking by sampling and integrating multiple trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1428–1441, Jul. 2014.

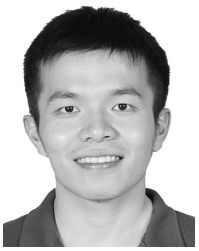
[39] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[40] Z. Hong, X. Mei, and D. Tao, "Dual-force metric learning for robust distracter-resistant tracker," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 513–527.
[41] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1305–1312.
[42] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1313–1320.
[43] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling L1 tracker with occlusion detection," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2661–2675, Jul. 2013.
[44] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell, "Accelerated gradient method for multi-task sparse learning problem," in *Proc. 9th IEEE Int. Conf. Data Mining*, Miami, FL, USA, Dec. 2009, pp. 746–751.
[45] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 470–484.
[46] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 980–993, Mar. 2015.
[47] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 3493–3500.
[48] O. J. Karst, "Linear curve fitting using least deviations," *J. Amer. Statist. Assoc.*, vol. 53, no. 281, pp. 118–132, Mar. 1958.
[49] I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete l_1 linear approximation," *SIAM J. Numer. Anal.*, vol. 10, no. 5, pp. 839–848, Oct. 1973.
[50] E. J. Schlossmacher, "An iterative technique for absolute deviations curve fitting," *J. Amer. Statist. Assoc.*, vol. 68, no. 344, pp. 857–859, Dec. 1973.
[51] L. Wang, M. D. Gordon, and J. Zhu, "Regularized least absolute deviations regression and an efficient algorithm for parameter tuning," in *Proc. 6th Int. Conf. Data Mining*, Hong Kong, Dec. 2006, pp. 690–700.
[52] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.
[53] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proc. Conf. Uncertainty Artif. Intell.*, Montreal, QC, Canada, Jun. 2009, pp. 339–348.
[54] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 864–877.
[55] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1940–1947.
[56] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 81–88.
[57] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[58] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, Jun. 2005, pp. 886–893.
[59] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[60] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.
[61] C. Wang, M. de La Gorce, and N. Paragios, "Segmentation, ordering and multi-object tracking using graphical models," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 747–754.
[62] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1995.
[63] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, Jan. 2015.
[64] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. IEEE 13th Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 1195–1202.
[65] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1177–1184.
[66] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 702–715.
[67] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1910–1917.
[68] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. 7th Eur. Conf. Comput. Vis.*, Copenhagen, Denmark, May 2002, pp. 661–675.
[69] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc., Ser. B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[70] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[71] H. T. Nguyen and A. W. M. Smeulders, "Robust tracking using foreground-background texture discrimination," *Int. J. Comput. Vis.*, vol. 69, no. 3, pp. 277–293, May 2006.
[72] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *Proc. SPIE*, vol. 4387, pp. 95–102, Mar. 2001.

Xue Mei (SM'14) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA.

He was with the Automation Path-Finding Group in Assembly and Test Technology Development and Visual Computing Group, Intel Corporation, Santa Clara, CA, USA. He is currently a Senior Research Scientist with the Department of Future Mobility Research, Toyota Research Institute North America, Ann Arbor, MI, USA, a Toyota Technical Center Division. He is an Adjunct Professor with Anhui University, Hefei. His current research interests include computer vision, machine learning, and robotics with a focus on intelligent vehicles research.

Dr. Mei is an Area Chair of the Winter Conference on Computer Vision in 2015, and a lead Organizer of the My Car Has Eyes: Intelligent Vehicle With Vision Technology Workshop at the Asian Conference on Computer Vision in 2015. He serves as a lead Guest Editor of the Special Issue on Visual Tracking for Computer Vision and Image Understanding.



Zhibin Hong (S'14) received the bachelor's degree in electronics engineering from the South China University of Technology, Guangzhou, China, in 2010. He is currently pursuing the Ph.D. degree with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology at Sydney, Sydney, NSW, Australia.

His current research interests include computer vision, machine learning, and data mining.

Danil Prokhorov (SM'02) began his career in Saint Petersburg, Russia, in 1992. He was a Research Engineer with the St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, Saint Petersburg. He was involved in automotive research in 1995. He was an intern with the Ford Scientific Research Laboratory, Dearborn, MI, USA, in 1995. In 1997, he became a Ford Research Staff Member, where he was involved in application-driven research on neural networks and other machine learning methods. Since 2005, he has been with the Toyota Technical Center (TTC), Ann Arbor, MI, USA. He is currently in charge of the Department of Future Mobility Research, Toyota Research Institute North America, Ann Arbor, a TTC Division. He has authored over 100 papers in various journals and conference proceedings and many patents in a variety of areas.

Dr. Prokhorov has served as the International Neural Network Society President from 2013 to 2014, a member of the IEEE Intelligent Transportation Systems Society Board of Governors, the U.S. National Science Foundation Expert, and the Associate Editor/Program Committee Member of many international journals and conferences.

Dacheng Tao (F'15) is currently a Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology at Sydney, Sydney, NSW, Australia. He mainly applies statistics and mathematics to data analytics. His current research interests include computer vision, data science, image processing, machine learning, neural networks, and video surveillance.

Dr. Tao's research results have expounded in one monograph and over 100 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the *Journal of Machine Learning Research*, the *International Journal of Computer Vision*, the Conference on Neural Information Processing Systems, the International Conference on Machine Learning, the Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, the International Conference on Artificial Intelligence and Statistics, and the International Conference on Data Mining (ICDM), and the ACM Special Interest Group on Knowledge Discovery and Data Mining, with several best paper awards, such as the Best Theory/Algorithm Paper Runner Up Award in the IEEE ICDM in 2007, the best student paper award in the IEEE ICDM in 2013, and the ICDM 10-Year Highest-Impact Paper Award in 2014.