Pattern Recognition 45 (2012) 2390–2404

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Robust visual tracking with structured sparse representation appearance model

Tianxiang Bai, Y.F. Li*

Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 4 May 2011
Received in revised form
12 September 2011
Accepted 9 December 2011
Available online 23 December 2011

Keywords:
Appearance model
Block-sparsity
Orthogonal matching pursuit
Sparse representation
Visual tracking

ABSTRACT

In this paper, we present a structured sparse representation appearance model for tracking an object in a video system. The mechanism behind our method is to model the appearance of an object as a sparse linear combination of structured union of subspaces in a basis library, which consists of a learned Eigen template set and a partitioned occlusion template set. We address this structured sparse representation framework that preferably matches the practical visual tracking problem by taking the contiguous spatial distribution of occlusion into account. To achieve a sparse solution and reduce the computational cost, Block Orthogonal Matching Pursuit (BOMP) is adopted to solve the structured sparse representation problem. Furthermore, aiming to update the Eigen templates over time, the incremental Principal Component Analysis (PCA) based learning scheme is applied to adapt the varying appearance of the target online. Then we build a probabilistic observation model based on the approximation error between the recovered image and the observed sample. Finally, this observation model is integrated with a stochastic affine motion model to form a particle filter framework for visual tracking. Experiments on some publicly available benchmark video sequences demonstrate the advantages of the proposed algorithm over other state-of-the-art approaches.

1. Introduction

Visual tracking has been one of the essential and fundamental tasks for intelligent video analysis as it can be widely applied in video surveillance, human motion understanding, human–computer interaction, etc. Not surprisingly, there is a fruitful literature in tracking algorithms development that reports promising results under various scenarios [1]. However, tracking the non-stationary appearance of objects undergoing significant pose, illumination variations and occlusions still remains a challenge for the community. Generally speaking, a visual tracking algorithm can be decomposed into three components: an appearance model which captures the visual characteristics of the target and evaluates the similarity between observed samples and the model; a motion model which locates the target between successive frames with certain motion hypothesis; and an optimization strategy which associates the appearance model with the motion model and finds the most likely location in the current frame. In this work, we concentrate mainly on designing a robust appearance model that confronts the aforementioned difficulties.

Supplementary material related to this article can be found online at doi:10.1016/j.patcog.2011.12.004.

Traditional appearance models such as template [2–4] or subspace representations [5–8] usually focus on approximating the target appearance itself, which is sensitive to gross errors caused by occlusion. Recently, a class of appearance modeling techniques named sparse representation has been shown to give state-of-the-art robustness against various disturbances, particularly, in the sense of occlusion [9–11]. Different from conventional approaches, the sparse representation based methods attempt to jointly estimate the target appearance as well as the occlusion by finding a sparse linear combination over a basis library containing target and trivial templates. The pioneer work was reported in [9], where the sparse representation of observed samples is achieved via ℓ_1-minimization. Although this method, referred to as the ℓ_1 tracker, appears to promise robustness for visual tracking, its extensive computational cost ensures that the tracker is not readily applicable to practical implementations. Another drawback is that the expressiveness of this model is limited as the target appearance can only be represented by the subspace spanned by the training templates directly cropped from the images, which makes it difficult to handle significant view or pose changes. Many remedies have been proposed to improve the performance, e.g., by reducing the data dimension via feature extraction, replacing the ℓ_1-minimization by Orthogonal Matching Pursuit (OMP), and enforcing the dynamic

* Corresponding author. Tel.: +852 3442 8410; fax: +852 2788 8423.
E-mail addresses: tx.bai@student.cityu.edu.hk, tianxiangbai@gmail.com (T. Bai), meyfli@cityu.edu.hk (Y.F. Li).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.12.004

group sparsity to the model [10,11]. However, the performance of these methods, which depend on feature extraction, may become degenerate associating with the reduction of the data dimension. In addition, these algorithms neither attempt to account for any prior information about the corruption or occlusion, nor the discriminative nature of sparse representation.

In this paper, we give a deeper insight into the sparse representation model for the visual tracking problem to develop an efficient algorithm. Our contention is that, in reality, the model goes beyond simple sparsity by considering a priori information on the predefined structure of the basis library and contiguous spatial distribution of occlusions. We present a structured sparse representation appearance model, which is robust against partial occlusion, pose variations and illumination changes. As illustrates in Fig. 1, in our proposed model, the nonzero entries in the sparse coefficient vector have a particular structure that arises from practical visual tracking. Our work is original in casting the appearance model as a sparse linear combination of a structured union of subspaces instead of individual templates. We show that with this structured model, a more robust and efficient implementation of the sparse representation for visual tracking is feasible. To further reduce the computational load, we introduce the Block Orthogonal Matching Pursuit (BOMP) [12] rather than ℓ_1-minimization or OMP to solve the structured sparse representation problem. We also exploit the inherent discriminativity of BOMP to eliminate the invalid observations during tracking, which results in more accurate and efficient tracking results. In addition, inspired by the adaptive appearance models [6,13,14], we replace the raw target templates with incrementally learned Eigen templates, which enrich the expressiveness of our appearance model. Finally, the tracking procedure leads to a particle filter framework with a stochastic affine motion model.

The remaining part of this paper is organized as follows. We begin by reviewing the relevant work in the next section. Section 3 gives the details of the structured sparse representation based appearance model, BOMP algorithm and model update scheme. The integration of our proposed appearance model and particle filter for visual tracking is described in Section 4. Section 5 presents experiments and performance evaluation of our tracker and we conclude this paper in Section 6.

2. Related works

There is a rich literature in appearance modeling and representation that aim at tackling non-stationary appearance tracking problems. In this section, we review the two most relevant topics that motivate this work: subspace representation and sparse representation methods.

Subspace representation aims at adapting the appearance variations with a low-dimensional subspace based on the core desire for dimensionality reduction. These methods have been justified that they are effective approaches to model the object appearance undergoing pose and illumination changes for visual tracking. Over the years, various subspace models have been proposed for visual tracking ranging from static subspace representations to adaptive learning subspace approaches. In early works, the geometry and illumination based parametric models [3] and eigenspace representations [5,8] are used for evaluating the similarity between the current observed images and the model. However, these methods mainly rely on training before tracking that may lead to failure if the target experiences variations which are outside the set of training samples. Moreover, in many real applications, it is neither practical to construct a rich sample set nor perform extensive training offline. What is needed, therefore, is an adaptive model that has the capabilities of online learning and updating for time-varying appearance representations. Ross et al. propose an incremental Principal Component Analysis (PCA) based learning subspace model that shows robustness to gradual changes in pose, scale and illumination [6]. Later on, more sophisticated subspace formulation such as graph-based learning subspace [15,16], Tensor Subspace [17], Riemannian subspace [18] and data-driven constrained subspace [7] are applied to find a more optimal subspace for performance improvement. These methods, benefiting from the adaptive learning or updating scheme, usually exhibit superiority compared to the static models. However, the above algorithms, static or adaptive,

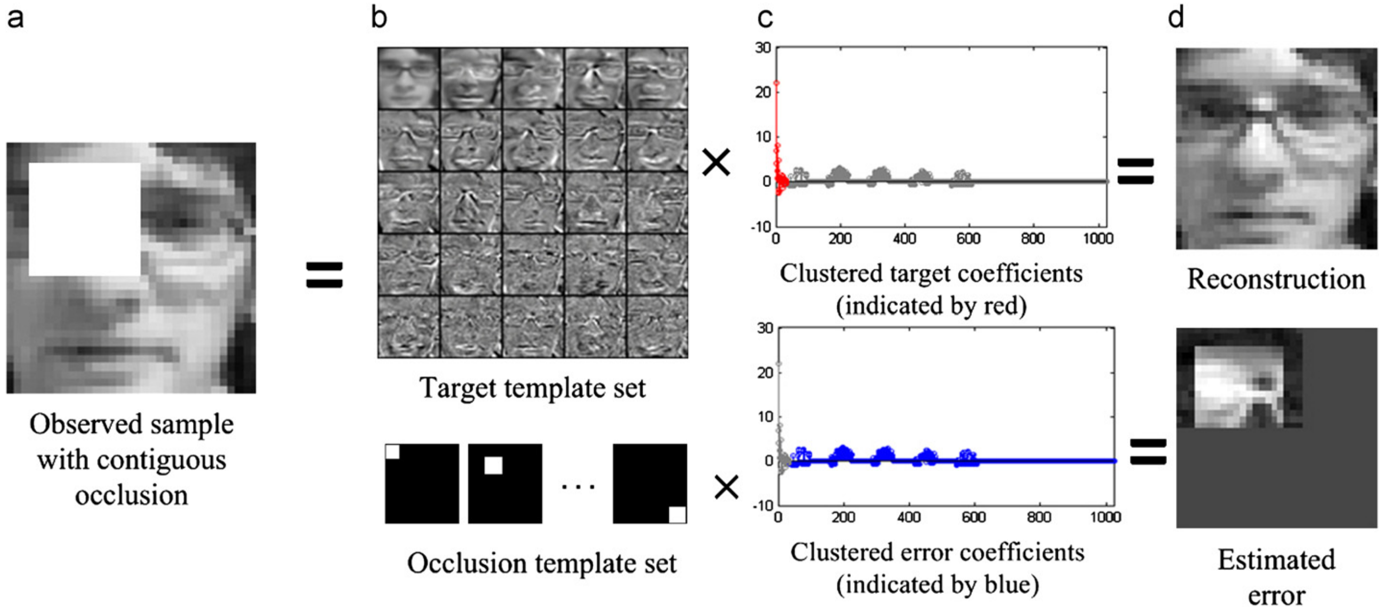
a
b
c
d
Observed sample with contiguous occlusion
=
Target template set
Occlusion template set
...
×
×
30
20
10
0
-10
0
200
400
600
800
1000
Clustered target coefficients (indicated by red)
Clustered error coefficients (indicated by blue)
=
=
Reconstruction
Estimated error

Fig. 1. Illustration of the structured sparse representation appearance model. An occluded target of interest (a) is represented as a sparse linear combination of the Eigen templates and occlusion templates in the basis library (b). The decomposed coefficients appear clustering distribution (c) corresponding to the predefined basis library structure and contiguous occlusions. The appearance of the target and the occlusion (d) can be jointly estimated.

usually do not have mechanisms to handle occlusions and they could suffer from failure caused by occlusion in a long duration.

Recent advancements in sparse representation indicate another path for us to model the appearance of an object by means of sparsity. The sparse representation based appearance models are initially inspired by the work on robust face recognition, where the discriminative nature of sparse representation is used as a classifier that achieves by far the best recognition rate in constrained experiments [19]. However, this method does not deal with pose variations and misalignment errors, which usually fails to recognize the true object in less controlled environments [20]. Subsequently, Mei and Ling propose a sparse representation based appearance model in a visual tracking scenario, in which the target appearance is expressed as a sparse linear combination with a basis library consisting of target templates, positive and negative trivial templates via ℓ_1-minimization [9]. The tracking result is then assigned to the observed sample that has the smallest reconstruction residual with the target templates and corresponding target coefficients. Qualitative experiments exhibit impressive robustness of such approach, but the large computational load prohibits its further application in reality. Furthermore, this model also has the same weakness as the sparse representation based classifier in [19], which is sensitive to pose variations. Various improvements are proposed such as collaborative tracking with two stage sparse optimization [10] and real-time compressive sensing tracking [11]. Both of these methods attempt to accelerate the algorithm by straightforward data dimension reduction with feature extraction and achieves the sparsity with a greedy algorithm rather than with ℓ_1-minimization. More recently, Han et al. [21] explore an alternative appearance model formulation with sparse representation, which casts the tracking as finding a sparse representation of sub-image feature sets sampled around the target. The tracking result is associated with the candidate holding the most similar coefficient distribution with the tracked object. This method is successful in the experiments of occluded cases. However, it is not clear if such a method is able to track objects effectively when they undergo significant illumination and pose variations.

This paper studies the problem of designing a novel robust appearance model for visual tracking, which is distinct from aforementioned methods in the following two perspectives. From the subspace representation point of view, our model jointly represents the target appearance and occlusion with a structured union of subspaces, which is adaptively constructed via a structured sparse representation, rather than a traditional subspace with fixed formation. This strategy provides a more flexible mechanism and richer expressiveness to harness the partial occlusion problems. From the sparse representation point of view, we analyze the intrinsic structure in the framework of sparse representation with the prior knowledge of contiguous occlusion in visual tracking. Such structured model can effectively alleviate the computational load without reducing the data dimension and yield more robust performance than the sparse representation appearance model in conventional sense, thereby ignoring the additional structure in the problem. Consequently, with an incremental learning subspace scheme, our model facilitates tracking objects undergoing significant illumination and pose change as well as occlusion.

3. Structured sparse representation based appearance model

3.1. Sparse representation based appearance model

In visual tracking, it is reasonable to assume that the target appearance can be represented by a linear subspace during a short time interval [6,7,22]. Mathematically, we assume an observed target sample $\mathbf{y} \in \mathbb{R}^L$ (stacked by columns to form a 1D vector) approximately lies in a subspace spanned by d given training templates $\mathbf{A}_{\boldsymbol{T}} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_d] \in \mathbb{R}^{L\times d}(L>d)$

$$\mathbf{y} \approx \mathbf{A}_T\mathbf{x} = \mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \cdots + \mathbf{a}_dx_d, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T \in \mathbb{R}^d$ is the corresponding target coefficient vector that denotes the contribution of each template in the gallery.

To harness the unpredictable partial occlusions, the query sample $\mathbf{y}$ is represented with the linear subspace $\mathbf{A}_T\mathbf{x}$, plus a sparse error $\mathbf{e} \in \mathbb{R}^L$ due to occlusion

$$\mathbf{y} = \mathbf{A}_T\mathbf{x} + \mathbf{e} = [\mathbf{A}_T\ \ \mathbf{A}_e]\begin{bmatrix}\mathbf{x}\\ \mathbf{e}\end{bmatrix} = \mathbf{A}\boldsymbol{\omega}. \tag{2}$$

Here, $\mathbf{A} = [\mathbf{A}_T\ \mathbf{A}_e] \in \mathbb{R}^{L\times(d+L)}$ is a basis library that consists of target template set $\mathbf{A}_T$ and occlusion template set $\mathbf{A}_e$, which is set to the identity matrix $\mathbf{I}$. It is sensible to assume $\boldsymbol{\omega} \in \mathbb{R}^{d+L}$ is a sparse coefficient vector with non-uniform density, where the $\mathbf{e}$ part of $\boldsymbol{\omega}$ is sparse, but the $\mathbf{x}$ part is usually dense. This is because, in most of the video systems that have moderate frame rate, the spatial coverage of occlusion usually changes gradually over time. Suppose the subspace model $\mathbf{A}_T\mathbf{x}$ can represent the target appearance together with occlusion well in the previous time step. The innovation $\mathbf{e}$ caused by the moving occlusion in the current time step is expected to occupy only a sparse spatial support. When the vector $\boldsymbol{\omega}$ is sparse enough, it is shown that the target coefficients $\mathbf{x}$ and sparse error $\mathbf{e}$ can be jointly recovered by solving the ℓ_0 norm minimization problem [19]

$$\min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_0 \text{ subject to } \|\mathbf{y} - \mathbf{A}\boldsymbol{\omega}\|_2 < \varepsilon, \tag{3}$$

where $\|\cdot\|_0$ is the ℓ_0 norm which counts the number of nonzero entries and $\|\cdot\|_2$ is the ℓ_2 norm. The parameter ε indicates the level of the reconstruction error. Solving the minimum ℓ_0 norm of the underdetermined problem, however, is both numerically unstable and NP-hard [23]. The ℓ_1-tracker suggests that the sparse solution can be obtained by converting the ℓ_0-minimization to an ℓ_1 norm convex optimization problem equivalently [9]. The tracking result is specified to the candidate that has the smallest residual against the recovered image with the target template set and corresponding target coefficients (i.e. argmin$\|\mathbf{y} - \mathbf{A}_T\mathbf{x}\|_2$). In addition, a heuristic template update scheme is proposed to adapt the appearance varying during tracking.

Although the ℓ_1-minimization is able to solve the sparse representation problem via convex optimization in polynomial time by standard scientific software [24,25], the costly computation and complex implementation make them unavailable for real-time visual tracking. Alternatively, the greedy approximation approaches such as Matching Pursuit (MP) [26] and Orthogonal Matching Pursuit (OMP) [27] show superiority in efficiency and implementation and comparable results for the ℓ_0-minimization approximation [28]. When the basis library $\mathbf{A}$ and sparse coefficient vector $\boldsymbol{\omega}$ are block structured, a more efficient implementation of MP and OMP is possible [12]. In the next session, we investigate the block sparse nature together with the Block Orthogonal Matching Pursuit (BOMP) algorithm in visual tracking applications and achieve more efficient and effective implementations.

3.2. From sparsity to structured sparsity

The prototype sparse representation framework is applicable to jointly model the target appearance and occlusion well without considering any priori distribution of the sparse coefficients. In fact, their sparse supports of $\boldsymbol{\omega}$ in (2) often have an underlying order. Fig. 1 illustrates an example of structured sparse supports. Considering a query sample occluded by a 20% contiguous patch, it is expected that the nonzero entries are clustered in target coefficient vector **x**. This is true when the test target sample **y** can be well approximated by the training template set $\mathbf{A}_T$. Another observation is that, in a practical visual tracking scenario, the partial occlusion often appears as a contiguous spatial distribution in the observed target sample. It was shown that analyzing the local image patches can promote the recognition performance against partial occlusion whose spatial support is unknown but contiguous [29–31]. In our approach, we employ such local region based method to resolve the partial occlusion problem, and argue that this approach preserves the block sparsity in the sparse representation framework. We apply this scheme in both the observed samples and the training templates since there is a one-to-one spatial mapping relation between them. As shown in Fig. 2, we first partition the observed sample and each of the training templates into R local parts (Fig. 2(b)) and follow by stacking the partitioned regions into 1D vectors, respectively. Then the 1D observed sample **y** and training templates $\mathbf{a}_i$, $i \in [1,d]$ are formed by concatenating the corresponding vectors in order. With previous local region analysis, the contiguous occlusion can be encoded as a block sparse vector having clustered nonzero entries (Fig. 2(c)).

Recent works report impressive improvements of the recovery threshold and efficiency by considering the block structured sparsity [12,32,33]. In this work, we explicitly take the aforementioned structured sparsity into account, treating the basis library $\mathbf{A}=[\mathbf{A}_T\,\mathbf{I}]$ and the sparse coefficient vector $\boldsymbol{\omega}=[\mathbf{x}\,\mathbf{e}]^{\mathrm{T}}$ in the view of concatenating by $R+1$ identical blocks (Fig. 3). The first block in **A** and $\boldsymbol{\omega}$ maps the target template set $\mathbf{A}_T$ and target coefficient vector **x**, respectively, while the remaining blocks correspond to the partitioned local regions and their coefficients. Mathematically, we assume that the dimension of the observed target sample **y** in (1) is L, the length of the block is d and $L=Rd$ with integer R. Thus the basis library $\mathbf{A} \in \mathbb{R}^{L\times(d+L)}$ in (2) can be viewed as a concatenation of $R+1$ identical blocks that have the same length of d

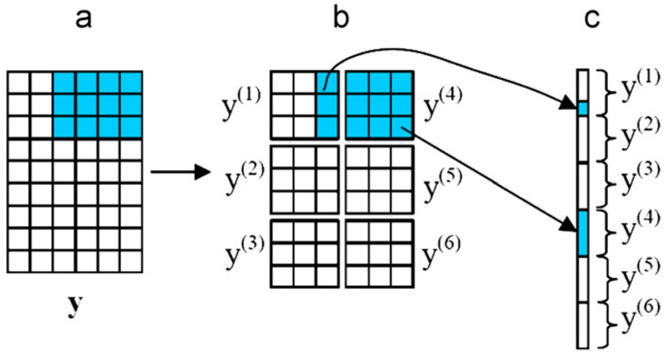

Fig. 2. Local region analysis that preserves block sparsity. The contiguous occlusion (highlighted with blue) can be stacked as a block sparse vector that has clustered nonzero entries. (a) Observed holistic sample or template image. (b) Partitioned local regions. (c) Stack the partitioned patches into 1D vector. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\mathbf{A}=[\underbrace{\mathbf{a}_1\cdots\mathbf{a}_d}_{\mathbf{A}[1]}|\underbrace{\mathbf{a}_{d+1}\cdots\mathbf{a}_{2d}}_{\mathbf{A}[2]}|\cdots|\underbrace{\mathbf{a}_{Rd+1}\cdots\mathbf{a}_{(R+1)d}}_{\mathbf{A}[R+1]}], \tag{4}$$

where $\mathbf{a}_i \in \mathbb{R}^L$ and $\mathbf{A}[l] \in \mathbb{R}^{L\times d}$ represent the ith column and lth block of the basis library **A**, respectively. Accordingly, we denote the block sparse coefficient vector $\boldsymbol{\omega}$ as

$$\boldsymbol{\omega}=[\underbrace{\omega_1\cdots\omega_d}_{\boldsymbol{\omega}^T[1]}|\underbrace{\omega_{d+1}\cdots\omega_{2d}}_{\boldsymbol{\omega}^T[2]}|\cdots|\underbrace{\omega_{Rd+1}\cdots\omega_{(R+1)d}}_{\boldsymbol{\omega}^T[R+1]}]^T \tag{5}$$

where ω_i and $\boldsymbol{\omega}[l] \in \mathbb{R}^d$ are the ith entry and the lth block of the vector $\boldsymbol{\omega}$, respectively. Conventionally, a vector $\boldsymbol{\omega} \in \mathbb{R}^{d+L}$ is called m-sparse if it has at most m nonzero entries. The definition of sparsity can be extended to block sparsity. A block k-sparse vector $\boldsymbol{\omega}$ is defined as the vector that has at most k $\boldsymbol{\omega}[l]$ with nonzero Euclidean norm, namely $\|\boldsymbol{\omega}\|_{2,0} \le k$ [12]. Denoting

$$\|\boldsymbol{\omega}\|_{2,0}=\sum_{l=1}^{R+1} I(\|\boldsymbol{\omega}[l]\|_2>0), \tag{6}$$

where $I(\cdot)$ is the indicator function.

In [12], the authors prove that orthogonalizing the columns within the blocks is able to achieve a higher recovery threshold. It is also shown that this orthogonalization preserves the block sparsity level, i.e., the distribution of the block with nonzero entries in $\boldsymbol{\omega}$ will not be changed if we orthogonalize each block within the basis library. The occlusion template set is the identity matrix that satisfies this condition as all the columns in the identity matrix are orthogonal mutually. The target template set, however, is usually not orthogonal, but has a high coherence because the templates are similar to each other. Thus, we exploit the Principal Components Analysis (PCA) approach to orthogonalize the target template set and form an Eigen template set instead

$$\mathbf{y}=[\mathbf{U}\,\mathbf{A}_e]\begin{bmatrix}\mathbf{c}\\ \mathbf{e}\end{bmatrix}=\mathbf{D}\boldsymbol{\alpha}, \tag{7}$$

where $\mathbf{D}=[\mathbf{U}\,\mathbf{A}_e] \in \mathbb{R}^{L\times(d+L)}$ is the new basis library consisting of the Eigen template set $\mathbf{U} \in \mathbb{R}^{L\times d}$ and the original occlusion template set $\mathbf{A}_e=\mathbf{I}$. And the vector $\boldsymbol{\alpha}=[\mathbf{c}\,\mathbf{e}]^T \in \mathbb{R}^{d+L}$ contains the decomposed coefficients that correspond to the Eigen template set and occlusion template set. The Eigen template set **U** is obtained by Singular Value Decomposition (SVD): $\mathbf{A}_T=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$.

Using the Eigen templates in place of the raw templates comes from the following two drawbacks in the original heuristic template

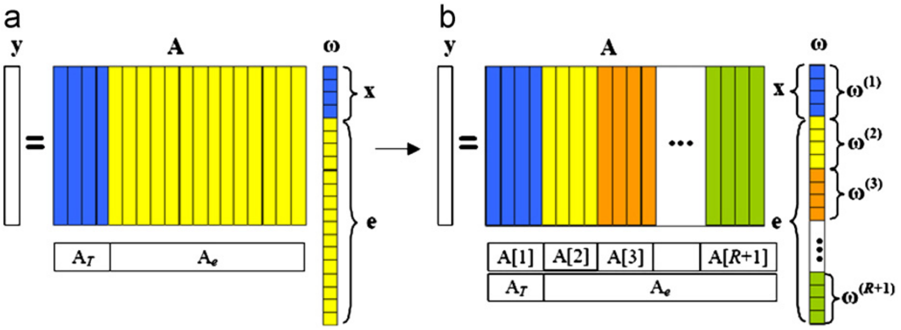

Fig. 3. Illustration of the block structured basis library **A** and coefficient vector $\boldsymbol{\omega}$.

update scheme proposed in [9]. First, the expressiveness of the subspace spanned by the raw templates is limited and hard to harness the significant view point and pose variations, because it deals with the templates that are only obtained from the previous couple of time instants. Second, the prototype ℓ_1-tracker is vulnerable to failure in the case of the basis library is updated with the background image patches or significant occluded tracking results. This is because the wrong templates are also possibly activated for approximating the observations and achieve high likelihood with the background or occlusion image patches. The Eigen template model can solve and avoid these problems effectively because it has the capability of learning the temporal correlation from the past appearance data by incremental SVD update procedure. This model has been successfully and prevalently used in visual tracking scenarios [5–6,8]. The learned Eigen templates provide a richer description than the raw templates to deal with the severe pose changes, since they span an optimal subspace that has the smallest reconstruction error with not only the current but also the past tracked appearance. Furthermore, it can also remedy the drift problems result from the incorrect updating by enforcing the previous appearance information into the target template set. It is notable that, we discard the nonnegativity constraints in [9], as PCA allows the Eigen templates and coefficients to be of arbitrary sign, and involves complex cancellations between positive and negative numbers. Imposing the nonnegativity constraints may lead to unexpected reconstruction failures.

3.3. *Block orthogonal matching pursuit (BOMP)*

Considering the block structured sparse representation, intuitively, the BOMP [12] is applied for seeking the sparsest linear combination for target representation efficiently. The BOMP algorithm consists of three major stages in each iteration: matching stage, estimation stage and updating stage. The major difference between BOMP and standard OMP is the matching stage. BOMP selects the block having the highest correlation whereas only one best matched template is chosen by OMP. Once the block is found, the corresponding coefficients are estimated via least-squares minimization at the estimation stage. Then the residual is updated at the third stage.

An inherent benefit of the BOMP algorithm comes from its discriminative ability of inferring whether the observed samples are invalid. Since we know a priori that a valid observation should be better represented by the target templates rather than the occlusion templates and it is able to achieve the highest correlation with the target template set. The matching stage thus can act as a classifier that eliminates the outliers by judging whether the target template set is picked in the first iteration. It is also possible to shorten the running time of the algorithm if we terminate the loops in an early stage once the observed sample is determined as outliers. Fig. 4 depicts the inferred outliers in our experiments. We show that this discriminative nature promotes both efficiency and accuracy of our tracker in the experiment session. The BOMP with outlier elimination procedure is summarized in Algorithm 1.

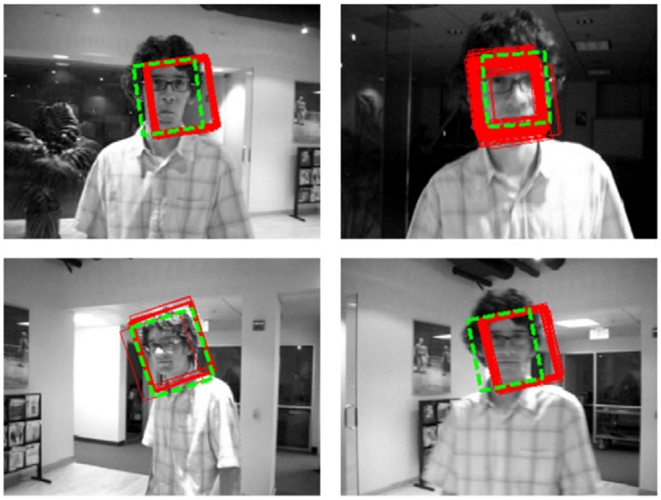

Fig. 4. Illustration of inferred outliers (enclosed in the red solid box) and estimated tracking target (indicated with green dashed box). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1. BOMP with outlier elimination

Input: Given the observation sample $\mathbf{y}$ and basis library $\mathbf{D}$.

1. Initialization: Initialize the residual as $\mathbf{r}_0 = \mathbf{y}$, $l=1$ and $\boldsymbol{\alpha} = [\,]$.
2. Matching stage: Choose one block that is best matched to $\mathbf{r}_{l-1}$ according to

   $$i_l = \arg\max_i \|\mathbf{D}^T[i]\mathbf{r}_{l-1}\|_2, \tag{8}$$

   where $\mathbf{D}[i]$ is the ith block of basis library $\mathbf{D}$.
3. Outlier elimination: Break and return $\mathbf{y}$ as an outlier if $l=1$ and $i_l \neq 1$.
4. Estimation stage: Solve the least-squares problem

   $$\min \left\| \mathbf{y} - \sum_{i \in \mathbf{I}} \mathbf{D}[i]\boldsymbol{\alpha}_l[i] \right\|_2, \tag{9}$$

   where $\boldsymbol{\alpha}_l[i]$ is the estimated block coefficients of vector $\boldsymbol{\alpha}$ over the set of chosen indices: $\mathbf{I} = \{i_1, \ldots, i_l\}$.
5. Updating stage: Update the residual

   $$\mathbf{r}_l = \mathbf{y} - \sum_{i \in \mathbf{I}} \mathbf{D}[i]\boldsymbol{\alpha}_l[i]. \tag{10}$$

6. Increment l and go back to step 2, until the ℓ_2-norm of residual $||\mathbf{r}_l||_2$ is below the destination threshold or the maximum number of iterations is reached.

Output: The sparse coefficient vector $\boldsymbol{\alpha}$.

3.4. *Computational complexity*

The computational cost of OMP is much better compared with the ℓ_1-norm minimization with standard convex programming for sparse representation problems [27,28]. We now address the further computational complexity reduction of BOMP based on standard OMP algorithm by exploiting the block structure. The computational time of OMP is dominated by the matching stage (step 2) and the least squares estimation stage (step 4). Suppose there are m nonzero entries in total in the block k-sparse coefficient vector $\boldsymbol{\alpha}$ and we assume all m nonzero entries are clustered into k d-length blocks identically ($k = m/d$). The total cost of running the matching and estimation stage in OMP are $O(mLN)$ and $O((m+1)mL/2)$, respectively [27]. The computational load of BOMP at the matching stage in each iteration is the same as OMP because all the templates are selected inevitably for inner product calculation. However, BOMP picks d templates (length of the block) at a time. Thus, the number of iterations is reduced to k and the time complexity drops to O(kLN) if the pervious assumption holds. Similarly, the computational complexity of the estimation stage decreases to $O((k+1)kdL/2)$. The margin total time complexity ratio between

BOMP and OMP is $O(kLN/mLN)=O(1/d)$ for the matching stage and $O(((k+1)kdL/2)/((m+1)mL/2))=O(k+1)/(kd+1)\approx O(1/d)$ for the estimation stage. It means that the time complexity of BOMP is $1/d$ of OMP if all the whole nonzero entries are clustered by d-length blocks identically.

In practice, the previous assumption may not be strictly satisfied. The coefficient vector $\boldsymbol{\alpha}$ is likely not block k-sparse if the sampled observations are not well aligned to the target. The algorithm has to take more loops to approximate the observations and generate denser distributed coefficients. However, such computational load for invalid samples is undesired and time consuming. On the other hand, as mentioned earlier, it is sensible to assume that the spatial coverage of innovation caused by occlusion in each time constant is sparsely distributed. It is also shown that the tolerance of occlusion is 33% for reliable face recognition with ℓ_1-minimization [19]. Although the theoretical bound of the greedy algorithm is somewhat lower than ℓ_1-minimization, it is worth to note that, the role of sparse representation in visual tracking is to jointly approximate the target appearance and occlusion rather than perfect reconstruction for face recognition. We thus argue that the tolerance of occlusion in our model is consistent with [19], which means the sparse error vector $\mathbf{e}$ has at most one third of the blocks that have nonzero entries. Since it is proven that the number of iterations for recovering a block k-sparse vector is at most k [12], we set the maximum number of iterations to $k=\lfloor R/3+1\rfloor$, where $\lfloor\cdot\rfloor$ returns the nearest integer less than or equal to the value inside. The value of k counts the blocks for coding the target appearance and occlusion. And the stopping threshold of the residual is set to 0.1 empirically in all our experiments.

3.5. Incremental learning scheme

Since the appearance of a target changes over time, the fixed Eigen templates are impossible to capture the appearance variations for a long duration. It is important to update the model online to enhance the adaptivity of the tracker. In our approach, we employ the incremental PCA algorithm to perform an incremental learning procedure when a new target sample is observed [34,35]. To make this paper self-contained we briefly introduce the incremental PCA algorithm here. Suppose we have an existing data matrix $\mathbf{B}\in\mathbb{R}^{L\times d}$ with its SVD $\mathbf{B}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ and a matrix $\mathbf{C}\in\mathbb{R}^{L\times n}$ whose columns contain new observation samples. All the observed samples are normalized by a zero-mean-unit-norm to partially compensate for photometric contrast variations and zero mean is assumed by removing the sample mean. Considering the following partitioned form of $[\mathbf{B}\,\mathbf{C}]$

$$[\mathbf{B}\,\mathbf{C}]=[\mathbf{U}\,\mathbf{K}]\begin{bmatrix}\boldsymbol{\Sigma} & \mathbf{U}^T\mathbf{C}\\ 0 & \mathbf{K}^T\mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{V}^T & 0\\ 0 & \mathbf{I}\end{bmatrix}=[\mathbf{U}\,\mathbf{K}]\mathbf{R}\begin{bmatrix}\mathbf{V}^T & 0\\ 0 & \mathbf{I}\end{bmatrix}, \tag{11}$$

where $\mathbf{K}$ is the orthonormal basis of $\mathbf{C}$ orthogonal to $\mathbf{U}$ and

$$\mathbf{R}=\begin{bmatrix}\boldsymbol{\Sigma} & \mathbf{U}^T\mathbf{C}\\ 0 & \mathbf{K}^T\mathbf{C}\end{bmatrix}$$

is an upper triangular matrix having size $d+n$. We can obtain $\mathbf{K}$ and $\mathbf{R}$ via QR decomposition of $[\mathbf{U}\boldsymbol{\Sigma}\,\mathbf{C}]$: $[\mathbf{U}\,\mathbf{K}]\mathbf{R}=[\mathbf{U}\boldsymbol{\Sigma}\,\mathbf{C}]$. In order to update the SVD of $[\mathbf{B}\,\mathbf{C}]$, it is necessary to diagonalize $\mathbf{R}$ with $\mathbf{R}\overset{\mathrm{SVD}}{=}\mathbf{U}'\boldsymbol{\Sigma}'\mathbf{V}'^T$. We can then rewrite (11) as

$$[\mathbf{B}\,\mathbf{C}]=([\mathbf{U}\,\mathbf{K}]\mathbf{U}')\boldsymbol{\Sigma}'\left(\mathbf{V}'^T\begin{bmatrix}\mathbf{V}^T & 0\\ 0 & \mathbf{I}\end{bmatrix}\right). \tag{12}$$

With this incremental updating scheme, the new Eigen template set $\tilde{\mathbf{U}}$ can be calculated with $\tilde{\mathbf{U}}=[\mathbf{U}\,\mathbf{K}]\mathbf{U}'$. In each update procedure, the first d Eigen templates remain to form the Eigen template set.

4. Proposed tracking algorithm with particle filter

We embed the structured sparse representation appearance model into a Bayesian inference framework to form a robust tracking algorithm. The model recursively updates the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ over the target state $\mathbf{x}_t$ given all the observation $\mathbf{y}_{1:t}=\{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_t\}$ up to and include time t. Applying Bayes' theorem, we have the Bayes filter

$$p(\mathbf{x}_t|\mathbf{y}_{1:t})\propto p(\mathbf{y}_t|\mathbf{x}_t)\int_{\mathbf{x}_{t-1}}p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}, \tag{13}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the observation model and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the motion model. In the particle filter framework [36], the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is recursively approximated by a set of weighted samples $\{\mathbf{x}_{t-1}^{(i)},\pi_{t-1}^{(i)}\}_{i=1}^{N}$, where $\pi_{t-1}^{(i)}$ is the weight for particles $\mathbf{x}_{t-1}^{(i)}$ and N is the total number of particles. We substitute the integration of the Bayes filter (13) with Monte Carlo approximation

$$p(\mathbf{x}_t|\mathbf{y}_{1:t})\approx kp(\mathbf{y}_t|\mathbf{x}_t)\sum_i\pi_{t-1}^{(i)}p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}), \tag{14}$$

where k is a normalization constant. The candidate samples $\{\mathbf{x}_{t-1}^{(i)}\}_{i=1}^{N}$ are drawn from an importance distribution $q(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{y}_{1:t})$ and then the weights of the samples are updated as

$$\pi_t^i=\pi_{t-1}^i\frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{y}_{1:t})}. \tag{15}$$

It is often reasonable to choose the motion model prior $q(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{y}_{1:t})=p(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$ as the importance density function. The motion model predicts the current state given the previous state. In this paper, an affine image warping is applied to model the target motion of two consecutive frames. We formulate the state vector $\mathbf{x}_t=(x_t,y_t,\eta_t,s_t,\beta_t,\phi_t)$ at time t with six parameters of affine transformation where x_t,y_t denote the x, y translation and $\eta_t,s_t,\beta_t,\phi_t$ represent the rotation angle, scale, aspect ratio, and skew direction at time t respectively. Each parameter in $\mathbf{x}_t$ is governed by a Gaussian distribution around their previous state $\mathbf{x}_{t-1}$ and assumed they are mutually independent

$$p(\mathbf{x}_t|\mathbf{x}_{t-1})=\mathbb{N}(\mathbf{x}_t;\mathbf{x}_{t-1},\boldsymbol{\psi}), \tag{16}$$

where $\boldsymbol{\psi}$ is the covariance matrix. Then, applying the Grenander's factored sampling algorithm, (16) can be rewritten as [40]

$$\pi_t^i=p(\mathbf{y}_k|\mathbf{x}_t^i), \tag{17}$$

where $p(\mathbf{y}_k|\mathbf{x}_t^i)$ is the observation model. The observation model indicates the similarity between an observed target candidate and the recovered image. It is determined by

$$p(\mathbf{y}_t|\mathbf{x}_t^i)=\begin{cases}0 & \text{if } \mathbf{x}_t^i \text{ is outlier}\\ \exp^{-\lambda\mathbf{r}_t} & \text{else}\end{cases}, \tag{18}$$

where λ denotes the weighting parameter,[1] and $\mathbf{r}_t=\|\mathbf{y}_t-\hat{\mathbf{y}}_t\|_2$ is the residual between the observed target sample $\mathbf{y}_t$ and the recovered image $\hat{\mathbf{y}}_t=\mathbf{D}_t\boldsymbol{\alpha}_t$. Notice that, unlike previous mentioned work [9–11], where the observation likelihood barely relies on calculating the reconstruction with the target template set and the target coefficients (i.e. $\mathbf{A}_T\mathbf{x}$), we consider both the reconstruction of the appearance and the estimated occlusion in our tracking algorithm. The posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be approximated in the following weighted form:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t})\approx\sum_{i=1}^{N}\pi_t^{*i}\delta(\mathbf{x}_t-\mathbf{x}_t^i), \tag{19}$$

[1] We set λ to 5 in all of the experiments.

Table 1
The details of video sequences.

Video clip	Frames	Initial position	Challenges
Occluded Face2	1–814	158,107	Significant and long duration occlusion
David Indoor	1–462	160,106	Illumination and pose variations
Trellis	1–502	200,100	Drastic illumination and pose variations
Car4	1–658	109,92	Sudden illumination changes
Car11	1–392	88,138	Difficult illumination conditions
Sylvester	1–1344	145,77	Significant illumination and pose variations
PETS2001	1–210	301,185	Non-rigid shape deformation, occlusion and background cluttering
OneLeaveShopReenter2cor	1–260	121,153	Non-rigid shape deformation and significant occlusion

where π_k^{*i} is the normalized weight obtained by

$$\pi_k^{*i} = \frac{\pi_k^i}{\sum_{i=1}^N \pi_k^i}. \tag{20}$$

The current state is then estimated by maximum a posterior (MAP) that associates with the highest likelihood

$$\mathbf{x}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}}\, p(\mathbf{x}_t | \mathbf{y}_{1:t}). \tag{21}$$

The proposed tracking algorithm is summarized in Algorithm 2.

Algorithm 2. Structured sparse representation appearance model based tracker

Input: The initial state of the target $\mathbf{x}_0 = (x_0, y_0, \eta_0, s_0, \beta_0, \phi_0)$.

1. Initialization: Construct the target template set $\mathbf{A}_T$ and normalize using the zero-mean-unit-norm and generate the Eigen template set $\mathbf{U}$ via SVD.
2. Sampling: Sample the target area according to the affine motion model (16).
3. Sparse approximation: Calculate the block sparse coefficient vector $\boldsymbol{\alpha}$ through Algorithm 1 of each sample and obtain the likelihood under the observation model (18).
4. Maximum a posterior: Estimate the current state $\mathbf{x}_t$ by MAP according to the particle filter framework and store the result image $\mathbf{y}_t$.
5. Incremental learning: Update the Eigen template set with $\mathbf{y}_t$ using incremental PCA.
6. Advance to the next frame and go to step 2, until the last frame of the video is reached.

Output: The current state $\mathbf{x}_t$.

5. Experiments

In this section, we present experiments to demonstrate the efficiency and effectiveness of our proposed algorithm. Eight publicly available benchmark video sequences[2] are used for evaluating the performance of our tracker under challenges of significant occlusion, pose and illumination changes. The details of the selected video sequences are listed in Table 1, where the start and the end frames, the initial position for tracking (indicated by the image coordinates), and the main challenges are included. All the experiments are conducted using a MATLAB implementation on a 3 GHz machine with 2 GB RAM. Additionally, we also tested the prototype sparse representation based ℓ_1 tracker [9], IVT tracker [6] and Mean Shift (MS) [37] for comparison.[3] As ℓ_1 tracker and IVT tracker are also particle filter based approaches, we set the parameters of both the motion model and observation model as consistent as possible with our tracker for fair comparison. For the motion model, the settings of the variance for translation are maintained the same in all three particle filter based trackers. However, the definition of the remaining four parameters of the ℓ_1 tracker is different from others. We use the default value for the ℓ_1 tracker and keep the other two identical. Each observed target sample is resized to a 12 × 15 patch for the observation model of the first three trackers. Note that the default size of a sampled image is 32 × 32 for the IVT tracker; down sampling the image may lead to degenerate experimental results compared with [6]. We draw 600 particles per frame for all three particle filtering methods to approximate the observation likelihood. The models are set to update every frame in order to adapt the appearance changes in the full extent. The block size d is assigned to 30 to balance the computational efficiency and tracking accuracy. The effectiveness of d will be discussed in the session of quantitative analysis. Accordingly, we partition the observed samples and Eigen templates into six local areas in our experiments. We also adopt 30 Eigen templates for the IVT tracker. The MS tracker is implemented in C++ with OpenCV functions. All the trackers start with the same initial position of the videos. For the trackers that involve particle filter, the quantitative results are obtained over 25 runs. We perform the MS tracker only 1 time, since it is a deterministic tracker. The representative visual results are reported in Fig. 5–7 for qualitative analysis. In the experiments, our algorithm runs at around 1.5–1.8 s per frame whereas the original sparse representation based tracker with ℓ_1-norm optimization solver [9] spends about 27 seconds per frame. Our algorithm is over 15 times faster compared with the ℓ_1 tracker in the experiments. It is also possible that the efficiency of our tracker could be further increased using feature extraction such as random projection [11] and Gabor filtering [38] for dimension reduction, since the sparsity pattern is not affected by choosing a different basis for each subspace [32]. This point, however, is beyond the scope of the present paper.

5.1. Qualitative analysis

The experimental results from the first video (*Occluded Face2*) validates our argument that using the structured sparse representation scheme is able to improve the robustness against occlusion. The parameters of our tracker are set as $\boldsymbol{\psi} = [2\ \ 2\ \ 0.01\ \ 0.01\ \ 0.001\ \ 0.001]$. As can be seen in Fig. 5(a), our method can track the face during the whole sequence even when

[2] These video sequences are available at http://www.cs.toronto.edu/~dross/ivt/, http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml, http://www.cvg.cs.rdg.ac.uk/PETS2001/ and http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[3] The code of ℓ_1 tracker and IVT algorithm can be obtained from http://www.ist.temple.edu/ hbling/code data.htm and http://www.cs.toronto.edu/~dross/ivt/ respectively.

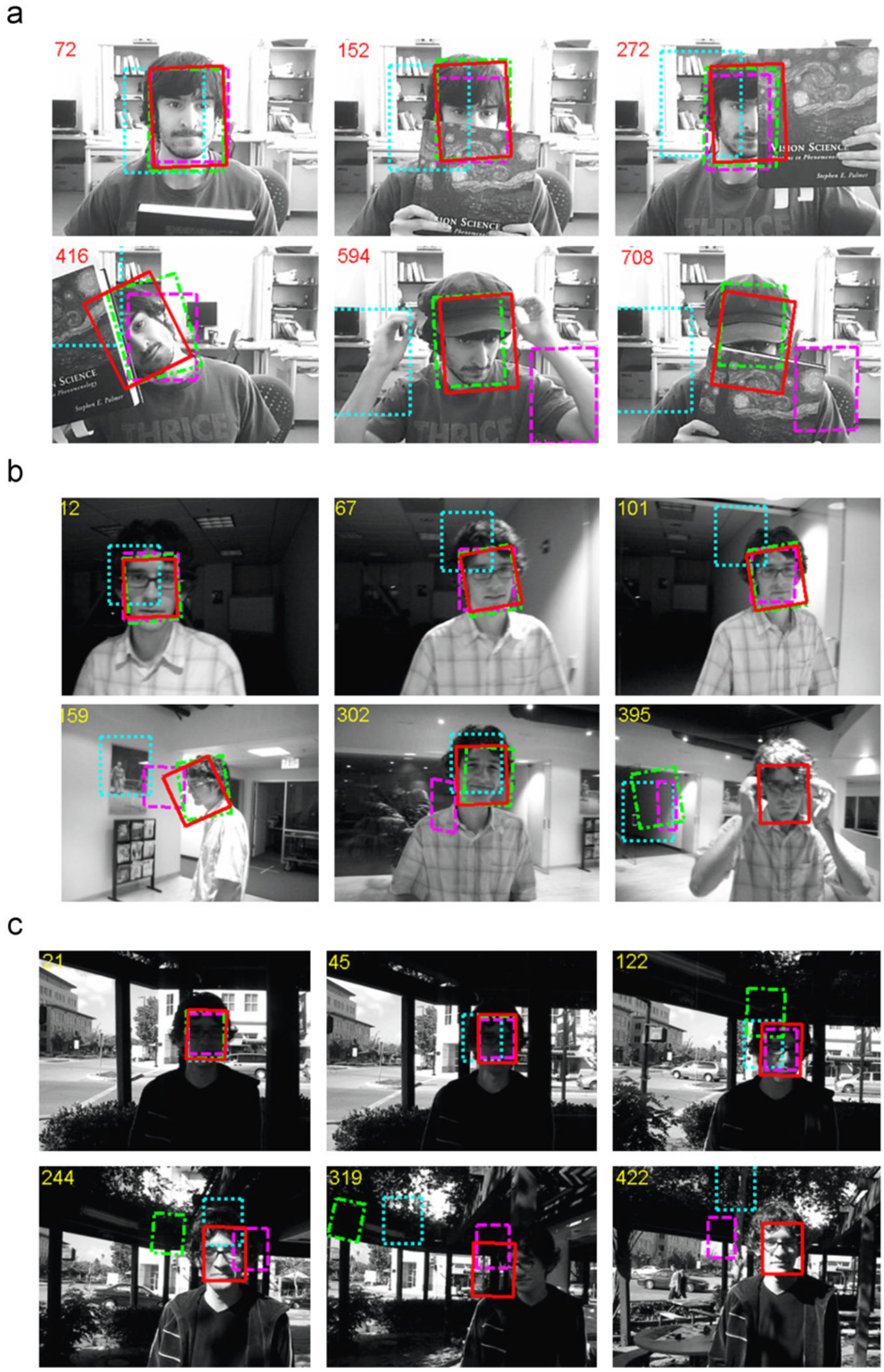

Fig. 5. Screen shots of a comparison tracking results of the proposed tracker (red solid box) with the ℓ_1 tracker (magenta dashed box), the IVT tracker (green dot dashed box) and the mean shift tracker (cyan dotted box). More results are shown in the accompanying video. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

it is significantly occluded by the book or cap and their combinations. However, the MS tracker is sensitive to the occlusions and drifts away when statistical features are similar between the background and the target. The ℓ_1 tracker shows competitive performance in the first 250 frames, but fails to locate the target at frame 594 due to the large pose varying. The IVT tracker drifts away from the face (frame 272, 416, 708), when long duration occlusion occurs.

We utilize *David Indoor* and *Trellis* sequences to evaluate the tracker performance under illumination changes, pose variations and a moving camera. The *David Indoor* data set (see Fig. 5(b)) is captured by a moving camera that experiences uncertain camera motions. We set the parameters as $\boldsymbol{\psi} = [5 \;\; 5 \;\; 0.01 \;\; 0.02 \;\; 0.002 \;\; 0.001]$. Our tracker is able to track the face of the person during the whole sequence. However, the IVT tracker fails in the last few frames and the ℓ_1 tracker drifts from the face in frame 159. The superior results produced by our model and IVT tracker confirm that the incremental learning subspace spanned by Eigen templates has richer expressiveness against pose and illumination changes than the ℓ_1 tracker that relies on the raw templates directly sampled from the images. The MS tracker totally loses the target in frame 111 where it encounters ambiguous grayscale statistics. The *Trellis* data set provides an

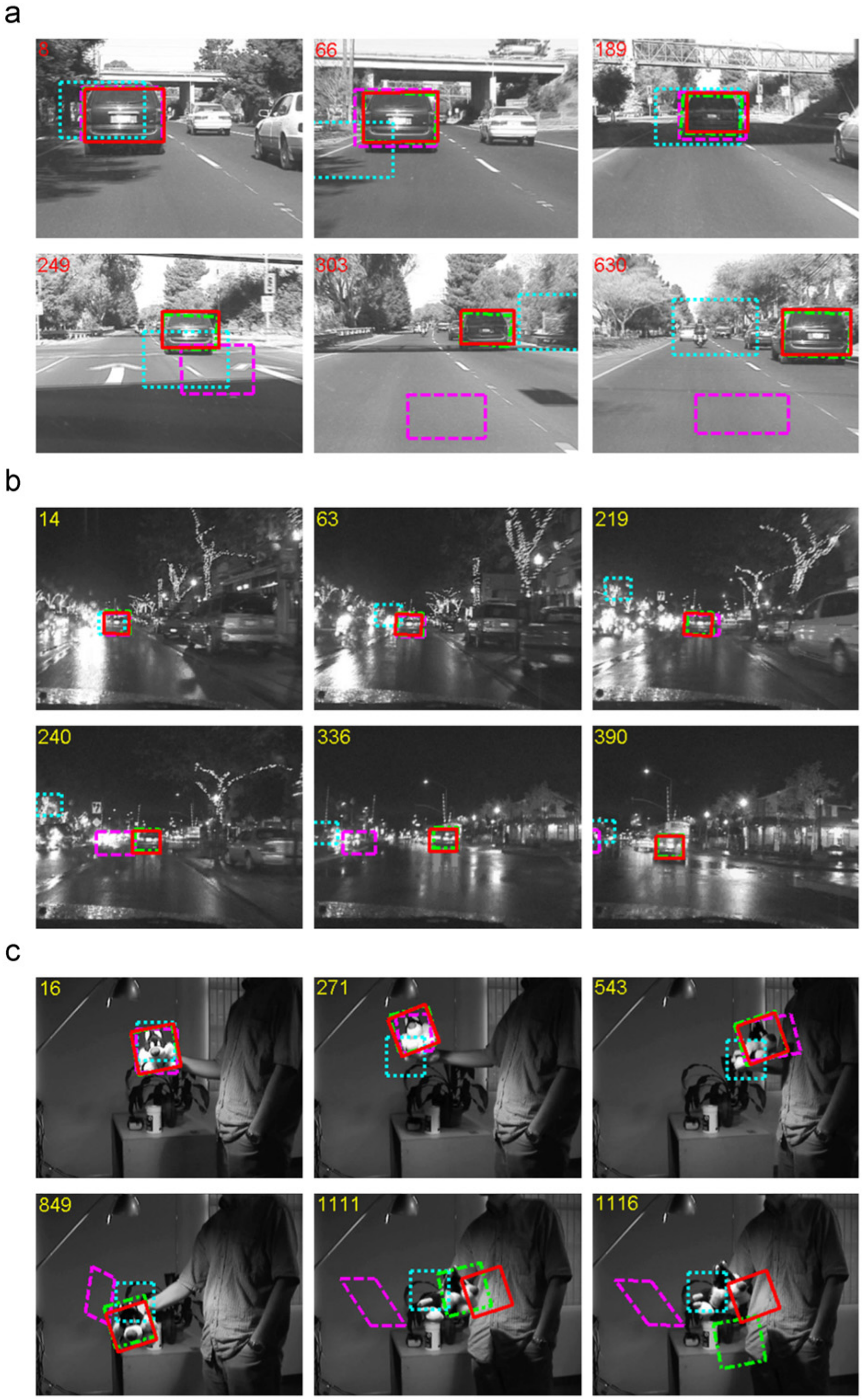

Fig. 6. Screen shots of a comparison tracking results of the proposed tracker (red solid box) with the ℓ_1 tracker (magenta dashed box), the IVT tracker (green dot dashed box) and the mean shift tracker (cyan dotted box). More results are shown in the accompanying video. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

even more challenging scenario where the person undergoes a combination of significant illumination and pose variations. The parameters for this video sequence are set as $\psi = [3\ 3\ 0.01\ 0.01\ 0.001\ 0.001]$. As illustrated in Fig. 5(c), no competitors can track the person in the whole sequence. Our tracker survives in the longest duration while the IVT tracker, ℓ_1 tracker and MS miss the target in early stage. Interestingly, our tracker exhibits the capability to resume tracking the target after the person turning round his head to the front (frame 422). The result implies that even though our tracker and IVT tracker share the same subspace learning scheme and motion parameter setting in the experiment, the integration with structured sparse representation is superior compared to the individual implementation of subspace representation based IVT tracker.

The *car4* and *car11* videos, shown in Fig. 6(a) and (b), represent some potential real-world applications of our tracker. The parameters are set as $\psi = [5\ \ 5\ \ 0.01\ \ 0.01\ \ 0.001\ \ 0.001]$ for these two

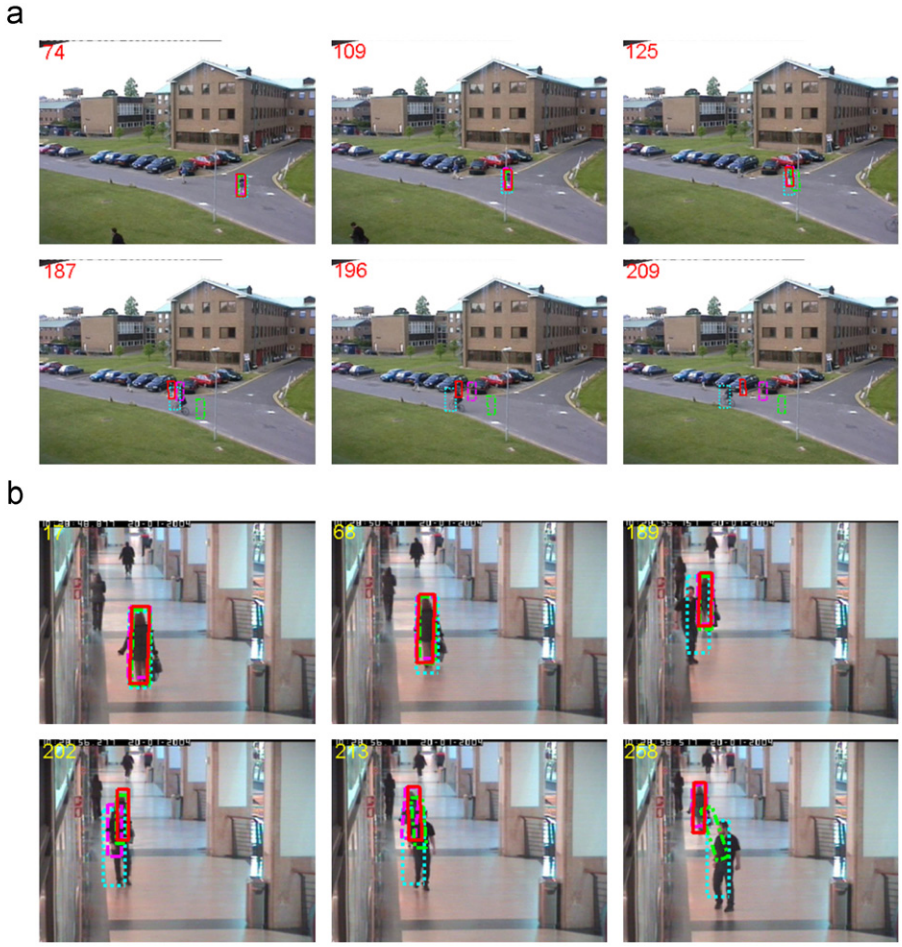

Fig. 7. Screen shots of a comparison tracking results of the proposed tracker (red solid box) with the ℓ_1 tracker (magenta dashed box), the IVT tracker (green dot dashed box) and the mean shift tracker (cyan dotted box). More results are shown in the accompanying video. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sequences. For the *car4* sequence, it tests the robustness of the trackers in sudden illumination changes. Although it is claimed that the ℓ_1 tracker can handle such situation via imposing the nonnegative constraint in the basis library [9], it still drifts from the target in our experiments. Similar results can be also found in [10]. It might be induced by the setting of the initial position or other parameters. Our tracker can track the car successfully, and the IVT tracker is also able to follow the car in most of the experiments. The MS tracker is unable to handle the illumination changes and has difficulties tracking the car in our experiment. The *car11* data set emphasizes that both MS and ℓ_1 tracker show unsatisfied performance in difficult illumination conditions. The MS tracker drifts from the car in the very beginning of the video, and the ℓ_1 tracker loses its way from frame 233.

The *Sylvester* sequence, shown in Fig. 6(c), exhibits challenges on dramatic illumination and pose variations. The parameters for this dataset are set as $\boldsymbol{\psi} = [3\ \ 3\ \ 0.01\ \ 0.01\ \ 0.001\ \ 0.001]$. The ℓ_1 tracker fails after frame 849, due to the large lighting and pose changes. Both our algorithm and IVT tracker are able to get through this frame, however, failures are observed in frame 1111 and 1116 for our tracker and the IVT tracker due to combination of extreme pose and illumination variation. The slight better performance of the IVT tracker may come from the benefits of using the forgotten factor, which allows the tracker to focus on the short-term observations. The MS tracker is able to follow the target for the whole clip but drifts away from the target frequently.

The *PETS2001* and *OneLeaveShopReenter2cor* sequences are used to test the performance of our algorithm in the case of tracking the non-rigid objects undergoing occlusions. The parameters for these two sequences are set as $\boldsymbol{\psi} = [1\ \ 1\ \ 0.005\ \ 0.005\ \ 0.001\ \ 0.001]$ and $[3\ \ 3\ \ 0.01\ \ 0.01\ \ 0.001\ \ 0.001]$, respectively. In the *PETS2001* sequence (Fig. 7(a)), our algorithm is able to track the person walking through the lamp pole and with occlusion by the bicyclist in most of the experiments. The IVT tracker loses the target from frame 125 due to the occlusion caused by the pole. Both the ℓ_1 tracker and MS tracker can track the target after this frame. But the ℓ_1 tracker gets stuck on the background that is similar with the target in frame 187, and the MS tracker is distracted by the bicyclist in frame 196. The experimental results of *OneLeaveShopReenter2cor* sequence, shown in Fig. 7(b), demonstrate that both the proposed tracker and ℓ_1 tracker can track the woman walking in the corridor when she is partially occluded by the man. However, the ℓ_1 tracker is not as stable as our tracker. Drift can be found in the 202th frame. On the other hand, the IVT tracker and MS tracker drift away from the target significantly when the occlusion occurs.

5.2. Quantitative analysis

The ground truth of *Occluded Face2*, *David Indoor* and *Sylvester* video clips are provided by the authors of [39]. For the other five sequences, we manually labeled the ground truth for the quantitative analysis. We use two metrics to quantify the performance of the proposed tracker and the reference trackers. The first one is the location error that measures the Euclidean distance between the tracking window center and the ground truth. The maximum, mean and standard deviation of the location error and the averaged location error with respect to frame number are summarized and plotted in Table 2 and Fig. 8, respectively. The second one is the failure rate that indicates the number of the tracking failure frame, divided by the total number of frames in a testing video clip. The tracking failure frame is indicated when the deviation of the tracking window center from the ground truth is larger than half of the diagonal length of the rectangle enclosing the target. The averaged failure rates in our experiments are reported in Table 3. We show that our tracker, on the mean location error metric, outperforms the other three competitors. And our method achieves the lowest standard deviation in the *Occluded Face2*, *David Indoor*, *Trellis*, *Car4*, *PETS2001* and *OneLeaveShopReenter2cor* sequences that implies the proposed tracker can attach the target closely against various disturbances. Overall, the lowest maximum, mean and standard deviation of the location errors are associated with the proposed tracker. We also observe that our tracker has the lowest failure rate compared with the other three trackers except for the *Sylvester* dataset. Considering the overall performance, our tracker only has 8.08% of failure rate which is far lower than the others in all eight video sequences that contain thousands of frames.

However, the above simplistic comparison does not necessarily imply that the proposed tracker is better than the reference trackers. We conduct a standard statistical one-sided hypothesis testing [41] on above two metrics to further evaluate whether the tracking performance is significantly improved. The test is made of the null hypothesis H_0 that the proposed tracker is not superior to the reference trackers against the alternative hypothesis H_1 that the proposed tracker is significantly better than the others. At the jth repetition, we calculate the sample performance differences

$$\Delta^j = C^j_{REF} - C^j_{SSR}, \tag{22}$$

where C^j_{REF} and C^j_{SSR} denote the quantified performance of the reference trackers and the proposed structured sparse representation appearance model based tracker, respectively. In our case, C^j represents the mean location error or the failure rate in run j. The hypothesis test is based on the sample mean of above differences

$$\overline{\Delta} = \frac{1}{J}\sum_{j=1}^{J} \Delta^j, \tag{23}$$

and its standard error

$$\delta_{\overline{\Delta}} = \sqrt{\frac{1}{J^2}\sum_{j=1}^{J}(\Delta^j - \overline{\Delta})^2}. \tag{24}$$

The alternative hypothesis H_1 is accepted (H_0 is rejected) if the test statistic $\overline{\Delta}/\delta_{\overline{\Delta}}$ exceeds a threshold μ_α, that represents a point on the standard Gaussian distribution corresponding to the upper-tail probability of α. In this work, we set the threshold $\mu_\alpha = 1.65$, where the corresponding significant level α is 0.05. It is worth noticing that, the test is not applicable to the cases that the standard error $\delta_{\overline{\Delta}}$ becomes zero. These cases usually come from the comparisons that both the competitors produce 0% failure rates in all experiments (both Δ^j and $\overline{\Delta}$ are 0), or the comparisons involve MS tracker that yields consistent results against the proposed tracker have 0% failure rates in all repetitions ($\Delta^j = \overline{\Delta}$). The results of the hypothesis testing on location error and failure rate with respect to different video sequences and all experiments are shown in Table 4. We observe that the test statistic in Table 4 is greater than $\mu_{0.05} = 1.65$ in the majority of experiments. Moreover, if we consider the overall performance, the alternative hypothesis H_1 is accepted. Therefore, it is reasonable to infer that the performance of our tracker is significantly superior to the other three state-of-the-art algorithms.

Although the results of the experiments show that the structured sparse representation strategy works well for visual tracking, there are still two unanswered questions regarding the contribution of the block structure and the discriminativity of BOMP to improve tracking performance. To demonstrate the effectiveness against partial occlusion of the block structure, we conduct tracking experiments on the *Faceocc2* sequence with varying block length d ($d = 1$, 5, 10, 15, 30, 60) and report the mean location errors in Fig. 9. Since the number of Eigen template can also affect the tracker's performance, we keep 30 Eigen templates in all the experiments to avoid the influence of the varying Eigen space representation. For the case of $d = 60$, we set the length of the first block as 30, while the length of others as 60. This setting allows us to eliminate the influence of the Eigen templates model to make sure that the varying block length is the only cause of the tracking performance difference. It is worth to note that, the BOMP is the traditional OMP when $d = 1$. As shown in Fig. 9, when the block length is moderate ($d = 10$, 15, 30), it is shown that the tracking performance with BOMP produces slightly better accuracy (11.26–11.70 pixels) compared with traditional OMP (11.98 pixels). However, we also observed that the location error rises significantly if the block length is too large ($d = 60$). This is because in our algorithm, the maximal sparsity level kd is set as a constant. As the block length d increases, the block sparsity level k reduces accordingly. A higher block sparsity level k can represent the appearance and occlusions more flexibly by selecting more blocks within the basis library and relevant local regions to construct the union of subspaces. A bigger block length d can increase the implementation speed of the algorithm. This suggests that a compromise between the running efficiency

Table 2
Location errors of the proposed tracker, IVT tracker, ℓ_1 tracker and mean shift. Bold font with underline indicates the best performance.

Video clip	IVT tracker			ℓ_1 Tracker			Mean shift			Proposed tracker		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
Occluded Face2	42.09	13.27	9.54	152.35	33.75	36.27	141.83	77.96	36.21	**38.21**	**11.32**	**8.44**
David Indoor	270.42	18.62	44.81	227.22	61.77	58.85	178.40	80.48	52.99	**19.87**	**4.66**	**2.79**
Trellis	344.99	124.09	85.16	220.34	65.34	65.34	198.20	70.88	55.00	**90.41**	**11.50**	**15.48**
Car4	177.03	9.77	28.81	209.76	71.82	60.61	138.36	53.69	30.33	**12.74**	**3.59**	**1.90**
Car11	8.53	2.33	**1.61**	114.23	26.88	34.97	155.28	85.23	47.81	**8.07**	**2.28**	1.64
Sylvester	189.11	28.91	45.32	163.84	36.46	43.63	**82.04**	22.23	**13.90**	146.91	**21.47**	30.30
PETS2001	114.65	28.55	30.94	43.28	5.78	8.50	**24.71**	8.33	4.45	27.68	**2.56**	**3.26**
OneLeaveShopReenter2cor	31.89	4.30	5.35	28.78	3.43	3.56	58.41	15.25	14.92	**11.22**	**2.23**	**1.55**
Overall	344.99	29.12	54.21	227.22	42.65	51.23	198.20	51.80	44.55	**146.91**	**10.87**	**19.09**

Std: standard deviation.

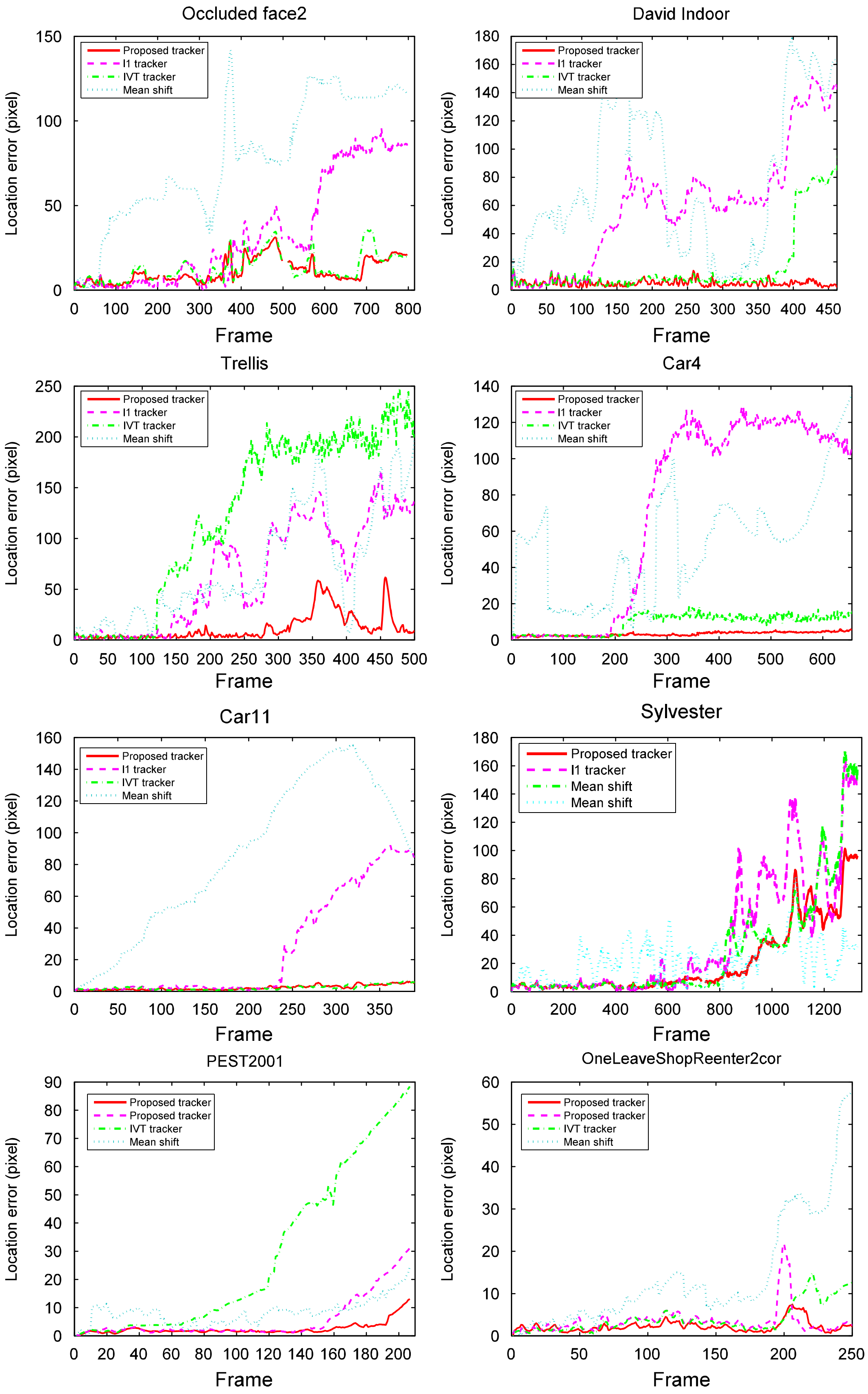

Fig. 8. Location error plots for the proposed tracker, ℓ_1 tracker, IVT tracker and mean shift for eight video sequences.

and tracking accuracy is needed for determining d. In this work, we show that setting $d=30$ is a rational trade off in our experiments.

We also assess the contribution of the outlier elimination procedure of BOMP to show how this inherent discriminative ability enhances the overall tracking performance. The *David Indoor* sequence is used for experiments, evaluating the tracking accuracy with and without outlier elimination. We re-run the tracker without removing the outliers elimination in the matching stage of BOMP. This caused a larger increase in mean location error, to 5.12 pixels, comparing with 4.66 pixels of our proposed approach. The location error curves are plotted in Fig. 10. Although the two curves are similar with each other in the first 350 frames, the curve of the approach without outlier elimination becomes apparently higher than that with such rejection criteria in the last 100 frames. In particular, a significant drift from the target is observed in the 445th frame without the outlier elimination scheme. Correspondingly, the running time of the algorithm without outlier elimination increases to 1.65 s per frame while the original tracker runs at 1.55 s per frame. We show that such discriminative capability of BOMP provides around 10% and 6% measurable boost to tracking accuracy and efficiency in our experiment, respectively. The reason for yielding more accurate tracking performance is that, the outlier elimination procedure essentially enlarges the effective sampling size so that the effects of degeneracy in the particle filter is reduced. According to [36], the effective sample size of the particle filter can be estimated by

$$\widehat{N_t^{\mathrm{eff}}} = \frac{1}{\sum_{i=1}^{N} (\pi_t^{*i})^2}, \qquad (25)$$

where N is the number of particles and π_t^{*i} is the normalized weight, which is calculated with the observation model (18) and normalized with (20). It has been shown that a larger effective

Table 3
Failure rates of the proposed tracker, IVT tracker, ℓ_1 tracker and mean shift. Bold font with underline indicates the best performance.

Video clip	IVT tracker (%)	ℓ_1 Tracker (%)	Mean shift (%)	Proposed tracker (%)
Occluded Face2	**0**	15.89	46.12	**0**
David Indoor	6.90	53.56	64.43	**0**
Trellis	75.63	51.90	65.87	**8.98**
Car4	6.77	55.06	44.51	**0**
Car11	**0**	36.14	87.05	**0**
Sylvester	25.73	37.06	**14.32**	24.13
PETS2001	46.86	14.14	8.57	**2.33**
OneLeaveShopReenter2cor	3.05	1.13	7.03	**0**
Overall	19.61	36.05	40.18	**8.08**

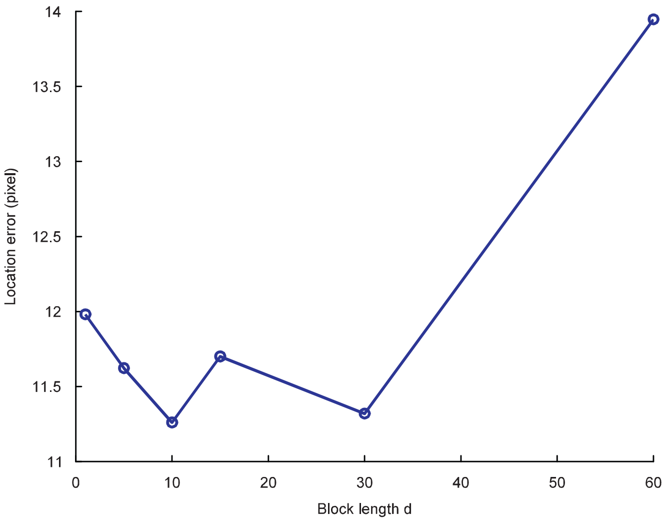

Fig. 9. Tracking location error with varying block length.

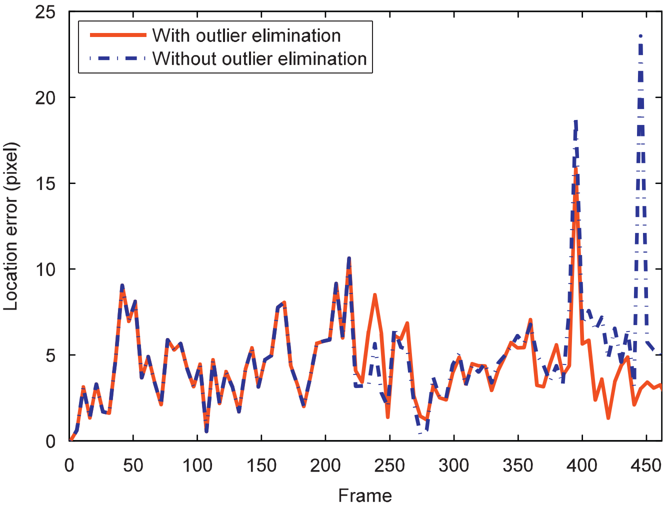

Fig. 10. Location error plots for comparison of with and without outlier elimination.

Table 4
Statistical test results for the proposed tracker against the IVT tracker, ℓ_1 tracker and mean shift in different video clips.

Test statistic $\overline{\Delta}/\sigma_{\overline{\Delta}}$	IVT tracker		ℓ_1 Tracker		Mean shift	
	Location	FR	Location	FR	Location	FR
Occluded Face2	3.23	N/A	11.55	5.47	185.58	N/A
David Indoor	3.29	2.54	9.09	10.16	72.54	N/A
Trellis	89.10	50.06	7.46	9.21	93.44	42.59
Car4	0.86	1.05	10.71	15.50	147.71	N/A
Car11	0.56	N/A	14.04	25.93	104.40	N/A
Sylvester	2.67	0.34	12.15	5.15	0.61	−3.91
PETS2001	9.13	10.94	5.08	5.36	20.52	5.63
OneLeaveShopReenter2cor	4.36	2.50	8.88	6.34	183.06	N/A
Overall	5.16	5.60	10.00	11.67	12.83	10.75

FR: failure rate.
N/A: not applicable.

sampling size $\widehat{N_t^{\text{eff}}}$ produces better tracking performance [40]. In our proposed observation model, the observation likelihood is assigned to be 0 once the candidate is identified as an outlier. More identified outliers can result in a smaller sum of squared normalized weights $\sum_{i=1}^{N}(\pi_t^{*i})^2$ in (25) that can contribute to a larger effective sample size for the particle filter.

6. Conclusions

In this paper, we have presented a robust visual tracking algorithm with a structured sparse representation appearance model. The structured sparse representation framework allows us to jointly model the appearance of the target and contiguous distributed occlusions with a sparse linear combination of structured union of subspaces. We state that this framework is applicable for visual tracking since it can capture the intrinsic structured distribution of sparse coefficients effectively in practice. For computational load reduction, we introduce the BOMP algorithm to solve the structured sparse representation problem. This leads to a significant improvement of the efficiency and towards to the practical implementation. In addition, an incremental PCA based Eigen template update scheme is proposed to improve the capturing of the changes of target appearance online. With this learning strategy, the appearance model has a richer expressiveness to tackle significant view and pose variation. We implement our method together with the state-of-the-art ℓ_1 tracker, IVT tracker and MS tracker on many publicly available benchmark video sequences. The empirical results show that our tracker achieves the most accuracy and robust performance in the tests with respect to partial occlusions, illumination and pose variations.

Although our proposed tracker performs well in our experiments, drifts are observed when there is extreme pose and illumination variation. This is because the description of subspace model is limited for approximating the complex and nonlinear manifold of appearance. A possible remedy is considering the background information in order to enforce additional discriminative power to the model. This issue should be investigated in the future work.

Acknowledgment

This work was supported by the Research Grants Council of Hong Kong (Project no. CityU 118311) and the University Grants Committee (Grant no. SEG_CityU 01). The authors would like to thank the editor and the anonymous reviewers for their constructive comments, which helped us to improve our paper.

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (4) (2006).
[2] C.F. Olson, Maximum-likelihood template matching, in: Proceedings of the IEEE Conference on CVPR, 2000.
[3] G.D. Hager, P.N. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1025–1039.
[4] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2004) 810–815.
[5] M.J. Black, A.D. Jepson, Eigen tracking: robust matching and tracking of articulated objects using a view-based representation, International Journal of Computer Vision 26 (1) (1998) 63–84.
[6] D.A. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (1–3) (2008) 125–141.
[7] M. Yang, Z. Fan, J. Fan, Y. Wu, Tracking nonstationary visual appearances by data-driven adaptation, IEEE Transactions on Image Processing 18 (7) (2009) 1633–1644.
[8] Z. Khan, T. Batch, F. Dellaert, A Rao–Blackwellized particle filter for eigentracking, in: Proceedings of the IEEE Conference on CVPR, 2004.
[9] X. Mei, H. Ling, Robust Visual Tracking using ℓ_1 minimization, in: Proceedings of the ICCV, 2009.
[10] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, C. Kulikowski, Robust and fast collaborative tracking with two stage sparse optimization, in: Proceedings of the ECCV, 2010.
[11] H. Li, C. Shen, Robust real-time visual tracking with compressive sensing, in: Proceedings of the IEEE Conference on Image Processing, 2010.
[12] Y.C. Eldar, P. Kuppinger, H. Bolcskei, Block-sparse signals: uncertainty relations and efficient recovery, IEEE Transactions on Signal Processing 58 (6) (2010) 3042–3054.
[13] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1296–1311.
[14] S.K. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Transactions on Image Processing 13 (11) (2004) 1491–1506.
[15] X. Zhang, W. Hu, S. Maybank, X. Li, Graph based discriminative learning for robust and efficient object tracking, in: Proceedings of the ICCV, 2007.
[16] Y. Zha, Y. Yang, D. Bi, Graph-based transductive learning for robust visual tracking, Pattern Recognition 43 (1) (2010) 187–196.
[17] X. Li, W. Hu, Z. Zhang, X. Zhang, G. Luo, Robust visual tracking based on incremental tensor subspace learning, in: Proceedings of the ICCV, 2007.
[18] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, J. Cheng, Visual tracking via incremental Log–Euclidean Riemannian subspace learning, in: Proceedings of the IEEE Conference on CVPR, 2008.
[19] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, M. Yi, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.
[20] A. Wagner, J. Wright, A. Ganesh, Z. Zihan, M. Yi, Towards a practical face recognition system: robust registration and illumination by sparse representation, in: Proceedings of the IEEE Conference on CVPR, 2009.
[21] Z. Han, J. Jiao, B. Zhang, Q. Ye, J. Liu, Visual object tracking via sample-based Adaptive Sparse Representation (AdaSR), Pattern Recognition 44 (1) (2011) 2170–2183.
[22] K.C. Lee, J. Ho, M.H. Yang, D. Kriegman, Visual tracking and recognition using probabilistic appearance manifolds, Computer Vision and Image Understanding 99 (3) (2005) 303–331.
[23] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Review 43 (1) (2001) 129–159.
[24] D.L. Donoho, Y. Tsaig, Fast solution of ℓ_1-norm minimization problems when the solution may be sparse, IEEE Transactions on Information Theory 54 (11) (2008) 4789–4812.
[25] E. Candes, J. Romberg, ℓ_1-magic: recovery of sparse signals via convex programming, 〈http://www.acm.caltech.edu/l1magic/〉, 2005.
[26] S.G. Mallat, Z. Zhifeng, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.
[27] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Transactions on Information Theory 53 (12) (2007) 4655–4666.
[28] S. Kunis, H. Rauhut, Random sampling of sparse trigonometric polynomials, II. Orthogonal matching pursuit versus basis pursuit, Foundations of Computational Mathematics 8 (6) (2008) 737–763.
[29] A.M. Martinez, Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (6) (2002) 748–763.
[30] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 2037–2041.
[31] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proceedings of the IEEE Conference on CVPR, 1994.
[32] Y.C. Eldar, M. Mishali, Robust recovery of signals from a structured union of subspaces, IEEE Transactions on Information Theory 55 (11) (2009) 5302–5316.
[33] M. Stojnic, F. Parvaresh, B. Hassibi, On the reconstruction of block-sparse signals with an optimal number of measurements, IEEE Transactions on Signal Processing 57 (8) (2009) 3075–3085.
[34] A. Levy, M. Lindenbaum, Sequential Karhunen-Loeve basis extraction and its application to images, IEEE Transactions on Image Processing 9 (8) (2000) 1371–1374.
[35] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: Proceedings of the ECCV, 2002.
[36] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188.
[37] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
[38] M. Yang, L. Zhang, D. Zhang, Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary, in: Proceedings of the ECCV, 2010.
[39] B. Babenko, S. Belongie, M.H. Yang, Visual tracking with online multiple instance learning, in: Proceedings of the IEEE Conference on CVPR, 2009.

[40] H. Chen, Y.F. Li, Dynamic View planning by effective particles for three-dimensional tracking, IEEE Transactions on System, Man, and Cybernetics—Part B: cybernetics 39 (1) (2009) 242–253.

[41] M. Kristan, S. Kovacic, A. Leonardis, J. Pers, A two-stage dynamic model for visual tracking, Transactions on System, Man, and Cybernetics—Part B: Cybernetics 40 (6) (2010) 1505–1520.

Tianxiang Bai received the B.S. and M.S. degrees in mechanical engineering from Guangzhou University, China, in 2006 and Guangdong University of Technology, China, in 2009, respectively. From 2008 to 2009, he was a visiting researcher in the Department of Mechanical Engineering at Korea Advanced Institute of Science and Technology, Republic of Korea. Currently, he is a Ph.D. student in the Department of Mechanical and Biomedical Engineering at City University of Hong Kong, Hong Kong. His research interests are robot vision and machine learning, especially in visual object detection and tracking.

Y.F. Li received the B.S. and M.S. degrees in electrical engineering from Harbin Institute of Technology China. He obtained the Ph.D. degree in robotics from the Department of Engineering Science, University of Oxford in 1993. From 1993 to 1995 he was a postdoctoral research staff in the Department of Computer Science at the University of Wales, Aberystwyth, UK. He joined City University of Hong Kong in 1995. His research interests include robot sensing, sensor guided manipulation, robot vision, 3D vision, visual tracking. He has served as an Associate Editor of IEEE Transactions on Automation Science and Engineering (T-ASE) and is currently serving as Associate Editor of IEEE Robotics and Automation Magazine (RAM).