Efficient Discriminative Nonorthogonal Binary Subspace with its Application to Visual Tracking

Ang Li, Feng Tang, Yanwen Guo, and Hai Tao

***Abstract*—One of the crucial problems in visual tracking is how the object is represented. Conventional appearance-based trackers are using increasingly more complex features in order to be robust. However, complex representations typically not only require more computation for feature extraction, but also make the state inference complicated. We show that with a careful feature selection scheme, extremely simple yet discriminative features can be used for robust object tracking. The central component of the proposed method is a succinct and discriminative representation of the object using discriminative non-orthogonal binary subspace (DNBS) which is spanned by Haar-like features. The DNBS representation inherits the merits of the original NBS in that it efficiently describes the object. It also incorporates the discriminative information to distinguish foreground from background. However, the problem of finding the DNBS bases from an over-complete dictionary is NP-hard. We propose a greedy algorithm called discriminative optimized orthogonal matching pursuit (D-OOMP) to solve this problem. An iterative formulation named iterative D-OOMP is further developed to drastically reduce the redundant computation between iterations and a hierarchical selection strategy is integrated for reducing the search space of features. The proposed DNBS representation is applied to object tracking through SSD-based template matching. We validate the effectiveness of our method through extensive experiments on challenging videos with comparisons against several state-of-the-art trackers and demonstrate its capability to track objects in clutter and moving background.**

***Index Terms*—Non-orthogonal binary subspace, object tracking, matching pursuit, efficient representation.**

I. Introduction

VISUAL object tracking in video sequences is an active research topic in computer vision, due to its wide applications in video surveillance, intelligent user interface, content-based video retrieval and object-based video compression. Over the past two decades, a great variety of tracking methods have been brought forward. Some of them include template/appearance based methods [1], [2], [3], [4], [5], layer based methods [6], [7], image statistics based methods [8], [9], [10], feature based methods [11], [12], contour based methods [13], and discriminative feature based methods [14], [15]. One of the most popular categories of methods is appearance based approaches which represent the object to be tracked using an appearance model and match the model to each new frame to determine the object state. In order to handle appearance variations, an appearance update scheme is usually employed to adapt the object representation over time. Appearance based trackers have shown to be very successful in many scenarios. However they may not be robust to background clutter where the object is very similar to background. In order to solve this problem, more and more complicated object representations which take into account colors, gradients and textures are used. However, extraction of the complicated features usually incurs more computation which slows down the tracker. Moreover, complex representation will make the inference much more complicated. One natural question to ask is how complicated features are really needed to track an object. In this paper, we show that with a careful feature selection scheme, extremely simple object representations can be used to robustly track objects.

Essentially, object tracking boils down to the image representation problem – what type of feature should be used to represent an object? An effective and efficient image representation not only makes the feature extraction process fast but also reduces computational load for object state inference. Traditional object representations such as raw pixels and color histograms are generative in natural, which are usually designed to describe the appearance of object being tracked while completely ignoring the background. Trackers using this representation may fail when the object appearance is very similar to the background. It is worth noting that some appearance based trackers model both foreground and background, for example in the layer tracker [7] the per-pixel layer ownership is inferred by competing the foreground and background likelihoods using a mixture of Gaussians. However the Gaussian model assumption degrades the representation power of the model. Subspaces are popular in modeling the object appearance. IVT [16] incrementally learns principal components of the object to adapt its appearance changes during tracking. Compressive Tracking [17] was proposed to project the object features into a subspace spanned by sparse binary basis. Most of these methods consider only the object appearance while not aware of the background context information.

Discriminative approaches have opened a promising new direction in the tracking literature by posing tracking as a classification problem. Instead of trying to build an appearance model to describe the object, discriminative trackers seek a decision boundary that can best separate the object and background (such as [18], [19], [20], [21]). The support vector tracker [18] (denoted as SVT afterwards) uses an offline-

A. Li is with the Department of Computer Science and the Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 20742. E-mail: angli@umiacs.umd.edu.

F. Tang is with Apple Inc. Email: sharp.tang@gmail.com.

Y. Guo is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China 210023. Email: ywguo@nju.edu.cn.

H. Tao is currently on leave with the Department of Computer Engineering, University of California, Santa Cruz, CA 95064. Email: taohai2006@gmail.com.

Manuscript received Month DD, YYYY; revised Month DD, YYYY.

learned support vector machine as the classifier and embeds it into an optical flow based tracker. Recently, Struck [20] employed the structural support vector machines to learn the object classifier and achieved the state-of-the-art performance. TLD [21] uses ferns as a rough classifier which generates object candidates and verifies these object patches using a nearest neighbor classifier. Collins et al. [14] are perhaps the first to treat tracking as a binary classification problem. A feature selection scheme based on variance ratio is used to select the most discriminative features for tracking in the next frame. Avidan's ensemble tracker [15] combines an ensemble of online learned weak classifiers using AdaBoost to classify pixels in the new frame. In discriminative spatial attention tracking [22], attention regions (AR) which are locally different from their neighborhoods are selected as discriminative tracking features. In [23], Gabor features are used to represent an object and the background. A differential version of Linear Discriminant Analysis classifier is built and maintained for tracking. In these trackers, the tracking result in the current frame is usually used to select training samples to update the classifier. This bootstrap process is sensitive to tracking errors – slight inaccuracies can lead to incorrectly labeled training examples, hence degrading the classifier and finally causing further drift. To solve this problem, researchers have proposed to use more robust learning algorithms such as semi-supervised learning and multiple instance learning to learn from uncertain data. In co-tracking [24], two semi-supervised support vector machines are built using color and gradient features to jointly track the object using co-training. In the online multiple instance tracking [25], the classifier is learned using multiple instance learning which only requires bag-level labels so that makes the learner more robust to localization errors. Leistner et al. [26] use online random forest for multiple instance which achieves faster and more robust tracker. In [27], the authors combine multiple instance learning and semi-supervised learning in a boosting framework to minimize the propagation of tracking errors. The algorithm proposed in [28] models the confusing background as virtual classes and solves the tracking problem in a multi-class boosting framework.

We propose an extremely simple object representation using Haar-like features for efficient object tracking. The representation is generative in nature in that it finds the features that can best reconstruct the foreground object. It is also discriminative because only those features that make the foreground representation different from background are selected. Our representation is based on the nonorthogonal binary subspace (NBS) method in [29]. The original NBS tries to select from an over-complete set of Haar-like features that can best represent the image. We propose a novel discriminative representation called discriminative nonorthogonal binary subspace(D-NBS) that extends the NBS method to incorporate discriminative information. The new representation inherits the merits of original NBS in that it can be used to efficiently describe the object. It also incorporates the discriminative information to distinguish foreground from background. The problem of finding a D-NBS subspace for a given template is NP-hard and even achieving an approximate solution is time consuming. We also propose a hierarchical search method that can efficiently find the subspace representation for a given image or a set of images. To make the tracker more robust, an update scheme is devised in order to accommodate object appearance variations and background change. We validate the effectiveness of our approach through extensive experiments on challenging videos and demonstrate its capability to track objects in clutter and moving background.

It is worth noting that there are also methods in machine learning that combines generative and discriminative models, for example [30], [31], [32], [33], [34], [35], [28]. Grabner et al. proposed to use boosting to select Haar-like features and these features are used to approximate a generative model [34]. Tu et al. proposed an approach to progressively learn a target generative distribution by using negative samples as auxiliary variables to facilitate learning via discriminative models [35]. This idea has been widely applied in later computer vision literatures. In [36], a generative subspace appearance model and a discriminative online support vector machine are used in the co-training framework for tracking objects. However, two different representations are used for generative and discriminative models which incurs extra computation for feature extraction. In this paper, we propose a principled method to extract a set of highly efficient features that are both generative and discriminative.

Recently, deep feature based trackers are emerging among the state-of-the-art performers [37], [38], [39]. The performance of these trackers gets boosted thanks to the strong feature extraction capability of deep neural networks. While deep neural net is able to extract high level visual semantics, the forward pass of neural net could be much slower as the network goes deeper. Although the tracking accuracy is of high priority, the tracking speed is still an important factor to consider because many complex computer vision tasks require a fast light-weight tracker with certain robustness.

A preliminary version of this work appeared as a conference paper [40]. This paper extends the previous version in the following perspectives:

- We devise a novel iterative D-OOMP method for fast computation of the D-NBS representation. This iterative method exploits the redundancy between the iterations of feature selection with a recursive formulation, hence significantly reducing the computational load.
- We propose a hierarchical D-OOMP algorithm that can speed up the search using a hierarchical dictionary obtained by feature clustering. This process dramatically reduces the computation cost and makes our approach applicable to large templates.
- We provide more detailed performance analysis and extensive experiments to show the superiority of the new method in this paper over the preliminary version of this work. We compare our tracker against 8 state-of-the-art trackers in 21 video sequences using comprehensive evaluation criteria.

The rest of this paper is organized as follows. In Section II, we briefly review Haar-like features and the non-orthogonal binary subspace approach. The Discriminative Nonorthogonal Binary Subspace (DNBS) formulation and its optimization algorithm (D-OOMP) are proposed in Section III. Section

IV introduces an equivalent DNBS formulation which speeds up the feature selection without loss of accuracy. Besides, a hierarchical strategy is further incorporated to boost the performance. In Section V, the application of DNBS to tracking is described. Both qualitative and quantitative experimental results are given in Section VI. Finally, we conclude the paper and discuss the future work in Section VII.

II. Background: Nonorthogonal Binary Subspace

Haar-like features and the variants have been widely employed in object detection and tracking [41], [42], [43], [44], [25] due to its computational efficiency. The original Haar-like features measure the intensity difference between black and white box regions in an image. This definition was modified in [29] as the sum of all the pixels in a white box region for the purpose of image reconstruction.

Definition 1 (Haar-like function): The Haar-like box function $\mathcal{H}$ for Nonorthogonal Binary Subspace is defined as,

$$\mathcal{H}_{u_0,v_0,w,h}(u,v) = \begin{cases} 1, & u_0 \le u \le u_0 + w - 1 \\ & v_0 \le v \le v_0 + h - 1 \\ 0, & \text{otherwise} , \end{cases} \quad (1)$$

where w and h represent the width and height of the box in the template. (u_0, v_0) represents the top-left location of the Haar-like box. The advantage of such box functions is that the inner product of the Haar-like base with any same-sized image template can be computed with only 4 additions, by pre-computing the integral image of the template.

The original NBS [29] approach tries to find a subset of Haar-like features from an overcomplete dictionary to span a subspace that can be used to reconstruct the original image. It is worth noting that in [29] the Haar-like functions have two types, i.e. one-box and symmetrical two-box functions. The symmetrical two-box functions are mainly designed for images with symmetric structure (e.g. frontal faces). We select only one-box functions to make it suitable for tracking arbitrary object that may not have symmetric structures.

Suppose that for any given image template $\mathbf{x} \in \mathbb{R}^{W \times H}$ of size $W \times H$ and the selected binary box features are $\{c_i, \phi_i\}(1 \le i \le K)$. c_i is the coefficient of box function ϕ_i. The NBS approximation is formulated as $\mathbf{x} = \sum_{i=1}^{K} c_i \phi_i + \varepsilon$, where ε denotes the reconstruction error. We define $\mathbf{\Phi}_K = [\phi_1, \phi_2, \ldots, \phi_K]$ as the basis matrix, each column of which is a binary base vector. Note that, this base set is non-orthogonal in general, therefore the reconstruction vector of template $\mathbf{x}$ should be calculated by the Moore-Pense pseudo-inverse such that

$$R_{\mathbf{\Phi}_K}(\mathbf{x}) = \mathbf{\Phi}_K(\mathbf{\Phi}_K^{\mathsf{T}}\mathbf{\Phi}_K)^{-1}\mathbf{\Phi}_K^{\mathsf{T}}\mathbf{x} . \quad (2)$$

Definition 2: For a given image template with width W and height H, a nonorthogonal binary feature dictionary $\mathbf{D}_{W,H}$ is specified such that

$$\mathbf{D}_{W,H} = \{\mathcal{H}_{u_0,v_0,w,h} \mid u_0, v_0, w, h \ge 1 \wedge u_0 + w - 1 \le W \wedge v_0 + h - 1 \le H\} . \quad (3)$$

The dictionary is composed of all possible Haar-like box functions which vary by the location and size of the white box. In our formulation introduced later in this paper, we represents the dictionary using a matrix $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots \psi_{N_\psi}]$ where each column vector is a vectorized Haar-like feature in dictionary $\mathbf{D}_{W,H}$. The total number of Haar-like box functions N_ψ in dictionary $\mathbf{D}_{W,H}$ is $W(W+1)H(H+1)/4$, thus the dictionary of base vectors is over-complete and highly redundant. The objective function for the optimal subspace selection with respect to a given image template is to minimize the reconstruction error using selected base vectors, which is formulated as

$$\arg\min_{\mathbf{\Phi}_K} \| \mathbf{x} - R_{\mathbf{\Phi}_K}(\mathbf{x}) \| . \quad (4)$$

In general, the problem of optimizing Eq. 4 is NP-hard. Greedy approximate solutions for example optimized orthogonal matching pursuit (OOMP) [29], [45] have been proposed to find a sub-optimal set of base vectors by iteratively selecting a base vector that minimizes the reconstruction error.

III. Discriminative Nonorthogonal Binary Subspace

The NBS method has been successfully used in computer vision applications such as fast template matching [29]. However, we find it less robust for applications such as object tracking. This is because tracking is essentially a binary classification problem to distinguish between foreground and background. NBS only considers the information embodied in the object image itself without any information about the background. To solve this problem, we propose the Discriminative Non-orthogonal Binary Subspace (D-NBS) image representation that extracts features using both positive samples and negative samples, i.e. foreground objects and background. The discriminative NBS method inherits the merits of the original NBS in that it can well describe the object appearance, and at the same time, it captures the discriminant information that can better separate the object from background.

A. Formulation

The objective of Discriminative NBS is to construct an object representation that can better distinguish between foreground object and background. The main idea behind Discriminative NBS is that we want to select features so that the reconstruction error for foreground is small while it is large for background. Different from the original NBS formulation Eq. 4 in which only the foreground reconstruction error is considered, in Discriminative NBS formulation, the objective function has both foreground and background reconstruction terms.

Let $\mathbf{\Phi}_K$ be the Discriminative NBS basis vectors with K bases and $R_{\mathbf{\Phi}_K}(\mathbf{X})$ be the reconstruction of $\mathbf{X}$ via $\mathbf{\Phi}_K$ using Eq. 2. Note that $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}]$ is a matrix of N_f recent foreground samples and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{N_b}]$ is a matrix of N_b sampled background vectors. The objective function for $\mathbf{\Phi}_K$ is to optimize

$$\arg\min_{\mathbf{\Phi}_K} \frac{1}{N_f} \| \mathbf{F} - R_{\mathbf{\Phi}_K}(\mathbf{F}) \|_F^2 - \frac{\lambda}{N_b} \| \mathbf{B} - R_{\mathbf{\Phi}_K}(\mathbf{B}) \|_F^2 \quad (5)$$

where $\| \cdot \|_F$ represents the Frobenius norm. The first term in the equation is the reconstruction error for the foreground and

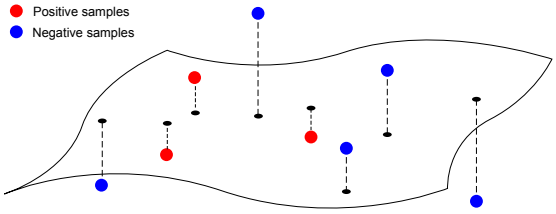


Fig. 1. An illustration of subspaces trained using multiple positive and negative samples: positive examples have smaller reconstruction errors while negative ones have larger reconstruction errors

the second term is the reconstruction error for the background. The objective is to find the set of base vectors to minimize the foreground reconstruct error while maximizing the background errors. This formulation can be interpreted as a hybrid form in which the generative and discriminative items are balanced by λ. As the feature dictionary is highly redundant and over-complete, the original NBS is under constrained. The second discriminative term can also be viewed as a regularization term to constrain the solution. An equivalent formulation of Eq. 5 is

$$\arg\max_{\mathbf{\Phi}_K} \frac{1}{N_f}\sum_{i=1}^{N_f}\langle \mathbf{f}_i, R_{\mathbf{\Phi}_K}(\mathbf{f}_i)\rangle - \frac{\lambda}{N_b}\sum_{i=1}^{N_b}\langle \mathbf{b}_i, R_{\mathbf{\Phi}_K}(\mathbf{b}_i)\rangle \ . \quad (6)$$

Conventional discriminative tracking approaches only model the difference between foreground and background, so they hardly memorize information about what the object looks like. Once losing track, they have weaker ability to recover, compared to generative trackers. The proposed approach has a generative component of object appearance, i.e., the model constrains the tracked result to be similar to the object in appearance. Such an enhanced model reduces the chance of losing track. In addition, such a combined generative-discriminative approach also helps recover the object from tracking failure.

B. Solution: Discriminative OOMP

It can be proved that solving Eq. 5 is NP hard, even verification of a solution is difficult. To optimize it, we propose an extension of OOMP (Optimized Orthogonal Matching Pursuit) [29] called discriminative OOMP. Similar to OOMP, discriminative OOMP is a greedy algorithm which computes adaptive signal representation by iteratively selecting base vectors from a dictionary.

We assume that totally K base vectors are to be chosen from the dictionary $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots, \psi_{N_\psi}]$ where N_ψ is the total number of base vectors in the dictionary. Supposing $k-1$ bases $\mathbf{\Phi}_{k-1} = [\phi_1, \phi_2, \ldots, \phi_{k-1}]$ have been selected, the k-th base is chosen to best reduce the construction errors for foreground and least for the background. Note that the candidate feature ϕ_i may not be orthogonal to the subspace $\mathbf{\Phi}_{k-1}$, the real contribution of ψ_i to increase Eq. 6 has to be offset by the component that lies in $\mathbf{\Phi}_{k-1}$. So the objective is to find the ψ_i which maximizes the following function:

$$\frac{1}{N_f}\sum_{j=1}^{N_f}\frac{|\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{f}_j)\rangle|^2}{\| \gamma_i^{(k)} \|^2} - \frac{\lambda}{N_b}\sum_{j=1}^{N_b}\frac{|\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{b}_j)\rangle|^2}{\| \gamma_i^{(k)} \|^2} \quad (7)$$

where $\gamma_i^{(k)} = \psi_i - R_{\mathbf{\Phi}_{k-1}}(\psi_i)$ is the component of base vector ψ_i that is orthogonal to the subspace spanned by $\mathbf{\Phi}_{k-1}$. $\varepsilon_{k-1}(\mathbf{x}) = \mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x})$ denotes the reconstruction error of $\mathbf{x}$ using $\mathbf{\Phi}_{k-1}$.

In each iteration of the base selection, the algorithm needs to search all the dictionary ψ_i to compute $\gamma_i^{(k)}$. Since the number of bases in dictionary is quadratic to the number of pixels in image, this process may be slow for large templates. To solve this problem, we further analyze the components of the above equation for for recursive formulation for fast computation.

Property 1 (Inner product): Let $\mathbf{\Phi}$ be a subspace in $\mathbb{R}^n$. For any point $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\langle \mathbf{x} - R_{\mathbf{\Phi}}(\mathbf{x}), \mathbf{y} - R_{\mathbf{\Phi}}(\mathbf{y})\rangle = \langle \mathbf{x}, \mathbf{y} - R_{\mathbf{\Phi}}(\mathbf{y})\rangle \quad (8)$$

where $R_{\mathbf{\Phi}}(\cdot)$ is the reconstruction of point with respect to subspace $\mathbf{\Phi}$.

Prop.1: Since $\mathbf{y} - R_{\mathbf{\Phi}}(\mathbf{y})$ is orthogonal to subspace $\mathbf{\Phi}$ and $R_{\mathbf{\Phi}}(\mathbf{x})$ lies in subspace $\mathbf{\Phi}$, hence $\langle R_{\mathbf{\Phi}}(\mathbf{x}), \mathbf{y} - R_{\mathbf{\Phi}}(\mathbf{y})\rangle = 0$ which is equivalent to Eq. 8. ■

Lemma 1:

$$\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{x})\rangle = \langle \psi_i, \mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x})\rangle \ . \quad (9)$$

Lemma 2: The norm of reconstruction residue of basis ψ_i with respect to subspace $R_{\mathbf{\Phi}_{k-1}}$ can be calculated recursively according to

$$\| \gamma_i^{(k)} \|^2 = \| \gamma_i^{(k-1)} \|^2 - \frac{|\langle \varphi_{k-1}, \psi_i\rangle|^2}{\| \varphi_{k-1} \|^2} \ . \quad (10)$$

Proof: See Appendix. ■

The denominator for each base vector $\| \gamma_i^{(k)} \|^2$ can be easily updated in each iteration, because the inner product $\langle \varphi_k, \psi_i\rangle$ can be quickly computed.

It is worth noting that reconstruction for any $\mathbf{x}$ (i.e. $R_{\mathbf{\Phi}_k}(\mathbf{x})$) can be efficiently computed by pre-computing $\mathbf{\Phi}_k(\mathbf{\Phi}_k^\mathsf{T}\mathbf{\Phi}_k)^{-1}$. The calculation of $\mathbf{\Phi}_k^\mathsf{T}\mathbf{x}$ is the inner products between $\mathbf{x}$ and the base vectors, which can be accomplished in $O(k)$ time using integral image. Thus, computing the reconstruction $R_{\mathbf{\Phi}_k}(\mathbf{x})$ simply costs $O(kWH)$ time, where W, H are respectively the width and height of the image template. As $\langle \varphi_k, \mathbf{x}\rangle$ and $\| \mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x}) \|^2$ can be pre-computed, the total computational complexity is $O(N_\psi K(N_f + N_b))$ with N_ψ the number of features in dictionary.

Below is the pseudo-code for D-OOMP where $\Sigma(\mathbf{x})$ represents the integral image of $\mathbf{x}$ and $\text{PROD}(\psi_i, \Sigma(\mathbf{x}))$ represents the inner product between Haar-like feature ψ_i and an arbitrary vector $\mathbf{x}$ which is calculated using its integral image.

Algorithm 1. D-OOMP for Haar-like features

Initialize dictionary $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots, \psi_{N_\psi}]$.
$denom(0, i) \leftarrow \| \psi_i \|^2, \forall i \in [1, N_\psi]$
for $k = 1$ **to** K **do**
  for $i = 1$ **to** N_ψ **do**
    $t \leftarrow \text{PROD}(\psi_i, \Sigma(\overline{\varphi_{k-1}}))$
    $denom(k, i) \leftarrow denom(k-1, i) - t^2$
    $num \leftarrow 0$
    for $j = 1$ **to** N_f **do**
      $num \leftarrow num + \frac{1}{N_f}\text{PROD}(\psi_i, \Sigma(\epsilon_{k-1}(\mathbf{f}_j)))$
    end for

for $j = 1$ **to** N_b **do**
  $num \leftarrow num - \frac{\lambda}{N_b}\text{PROD}(\psi_i, \Sigma(\epsilon_{k-1}(\mathbf{b}_j)))$
end for
score$_i \leftarrow num/denom(k, i)$
end for
The k-th basis is ψ_{opt} s.t. $opt = \arg\min_i \text{score}_i$.
end for

IV. Faster Computation of D-OOMP

Although the recursive computation of $\|\gamma_i^{(k)}\|^2$ improves the efficiency of D-OOMP, the optimization process is still slow due to the huge number of features in the dictionary. Another reason is that the computation of scores is proportional to the number of samples $N_f + N_b$. In this section we develop two algorithms to significantly reduce the amount of computation. The first one is an exact algorithm called Iterative D-OOMP that reduces the redundant computation in each iteration of feature selection with a recursive formulation. The second is an approximate method named hierarchical D-OOMP that uses hierarchical search to reduce the search space. We show that combining the two methods can achieve significant computational savings.

A. Iterative D-OOMP

In the above implementation, the time complexity of maximizing Eq. 7 for each feature is $O(N_f + N_b)$. Therefore, with the total number of foreground and background samples increasing, the computational load increases. This computation bottleneck will limit the applications of DNBS. Thus, we design to compute the feature scores iteratively with an equivalent formulation in which the foreground/background terms in Eq. 5 can be combined together and the time complexity will not be sensitive to the increasing of the example number.

To begin with, we denote $L_k(\psi_i)$ the item in Eq. 7 such that

$$L_k(\psi_i) = \frac{1}{N_f}\sum_{j=1}^{N_f}\frac{|\langle\gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{f}_i)\rangle|^2}{\| \gamma_i^{(k)} \|^2} - \frac{\lambda}{N_b}\sum_{j=1}^{N_b}\frac{|\langle\gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{b}_i)\rangle|^2}{\| \gamma_i^{(k)} \|^2} \tag{11}$$

The efficient computation of $L_k(\psi_i)$ plays a decisive role in speeding up the whole feature selection algorithm since it is exhaustively and repetitively calculated in each of the iterations. Through a series of equivalent transformations, we have the following proposition.

Proposition 1: During the selection procedure of the k-th basis, $L_k(\psi_i)$ can be calculated iteratively with known $L_{k-1}(\psi_i)$ such that

$$L_k(\psi_i) = \frac{1}{d_i^{(k)}}\left[d_i^{(k-1)}L_{k-1}(\psi_i) - 2\frac{\beta_i^{(k)}}{u_{k-1}}\langle\psi_i, \mathbf{I}_k\rangle + \left(\frac{\beta_i^{(k)}}{u_{k-1}}\right)^2 S_k\right] \tag{12}$$

where $d_i^{(k)} = \| \gamma_i^{(k)} \|^2$, $u_k = \| \varphi_k \|^2$ and $\beta_i^{(k)} = \langle\psi_i, \varphi_{k-1}\rangle$.

$$\mathbf{I}_k = \frac{1}{N_f}\sum_{j=1}^{N_f}\eta_k(\mathbf{f}_j) - \frac{\lambda}{N_b}\sum_{j=1}^{N_b}\eta_k(\mathbf{b}_j) \tag{13}$$

where $\eta_k(\mathbf{x}) = \langle\varphi_{k-1}, \mathbf{x}\rangle\varepsilon_{k-2}(\mathbf{x})$

$$S_k = \frac{1}{N_f}\sum_{j=1}^{N_f}\alpha_k^2(\mathbf{f}_j) - \frac{\lambda}{N_b}\sum_{j=1}^{N_b}\alpha_k^2(\mathbf{b}_j) \tag{14}$$

with $\alpha_k(\mathbf{x}) = \langle\varphi_{k-1}, \mathbf{x}\rangle$.

Proof: See Appendix. ■

It can be found from the above proposition that neither $\mathbf{I}_k$ or S_k are related to ψ_i, which indicates that they are same to each ψ_i and can be pre-computed before the main iteration of feature scoring. Then the computation of each feature score can be accomplished by with only two inner product calculations based on integral images and several multiplications. However, Eq. 12 only applies for situations when $k > 1$. Thus, the first binary base still has to be selected by the brute-force search, which theoretically costs $N_\psi(N_f + N_b)$ operations where N_ψ is the dictionary size. Let K be the expected number of features, (W, H) the size of template and N_f, N_b the numbers of foreground/background samples. Therefore, the time complexity of our approach achieves

$$\begin{aligned} &O(N_\psi(N_f + N_b) + KN_\psi + KWH(N_f + N_b) + K^2WH) \\ &= O((N_\psi + KWH)(K + N_f + N_b)) \\ &= O((N_\psi + KN_{\text{pix}})(N_s + K)) \end{aligned}$$

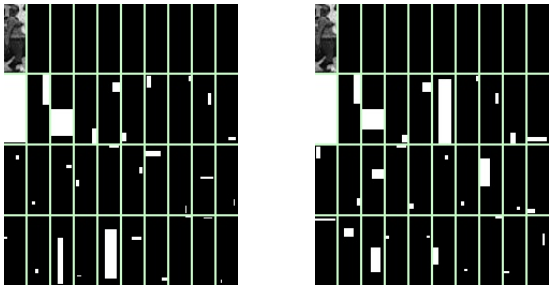
where $N_{\text{pix}} = WH$ is the number of pixels in template and $N_s = N_f + N_b$ is the total number of samples.

Algorithm 2. Iterative D-OOMP for Haar-like features

Initialize dictionary $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots, \psi_{N_\psi}]$.
Select the first feature φ_1 and initialize data.
for $k = 2$ **to** K **do**
  Pre-compute $\mathbf{I}_k$ according to Eq. 13.
  Pre-compute S_k according to Eq. 14.
  for $i = 1$ **to** N_ψ **do**
    Calculate $\beta_i^{(k)} \leftarrow \text{PROD}(\psi_i, \Sigma(\overline{\varphi_{k-1}}))$.
    Calculate $d_i^{(k)}$:$\|\gamma_i^{(k)}\|^2 \leftarrow \|\gamma_i^{(k-1)}\|^2 - (\beta_i^{(k)})^2$
    Calculate $L_k(\psi_i)$ according to Eq. 12.
  end for
  The k-th basis is ψ_{opt} s.t. $opt = \arg\min_i L_k(\psi_i)$.
end for

The iterative approach is theoretically an equivalent formulation of the original D-NBS, and thus it will not incur any additional error to the results. The most repetitive items are pre-computed to avoid redundant computation. Furthermore, the efficiency of the iterative approach stays much more stable as the number of samples increases, which is more suitable for those applications having a large number of training images.

An example image and the selected Haar-like features using Discriminative NBS are shown on the left of Figure 2. It is compared with the results selected using the original NBS shown on the right.



(a) DNBS (b) NBS

Fig. 2. Top 30 features selected using Discriminative NBS (left) and the original NBS (right) for an image. The two feature sets are in general similar to each other while the differences between the two feature sets are due to the negative samples integrated into DNBS.

B. Hierarchical D-OOMP

As can be observed, the major computation cost for D-OOMP lies in searching all the features in the dictionary. Therefore, one natural way to speed up is to reduce the dictionary size. In this section, we propose a hierarchical searching approach to reduce the search space. The features in the dictionary is first grouped into clusters using a fast iterative clustering method. This forms a two level hierarchy with the first one as the cluster centers and the second as all the rest in the same cluster. During the search procedure, a Haar-like feature is first compared with each of the cluster centers so that the ones that are far away from the candidate feature can be easily rejected.

1) Dictionary Clustering:

Definition 3 (μ-near basis set for a single feature): For Haar-like basis φ_i, we define its μ-near basis set to be $\mathcal{N}(\varphi_i, \mu) = \{\varphi_j | \langle \varphi_j, \varphi_i \rangle \geq \mu\}$.

Definition 4 (μ-near basis set for a set of features): For a set of Haar-like basis $\Phi = \{\varphi_i\}$, we define its μ-near basis set to be $\mathcal{N}(\Phi, \mu) = \cup_i \mathcal{N}(\varphi_i, \mu)$.

All features in the dictionary are grouped into clusters such that any feature has inner product larger than or equal to μ with the cluster center (i.e. μ-near basis set). The following three steps are iterated until all features have been assigned.

1) Randomly select cluster center c_i in the remaining feature set $\mathcal{F}$;
2) Bundle features in $\mathcal{N}(c_i, \mu)$ to cluster C_i.
3) Remove new cluster features: $\mathcal{F} = \mathcal{F} \setminus C_i$.

Afterwards, the dictionary $\mathbf{\Psi}$ is divided into groups of features $\mathbf{C} = \{C_1, C_2, \ldots\}$.

2) Efficient Dictionary Clustering: Observing that in the feature clustering process, the computation of $\mathcal{N}(c_i, \mu)$ is the most expensive operation. We describe a fast μ-near basis set retrieval method that leverages the special structural property of Haar-like box features.

For a cluster center ϕ and any feature ψ, the inner product is

$$\langle \phi, \psi \rangle = \frac{\text{CommonArea}(\phi, \psi)}{\sqrt{\text{Area}(\phi)} \cdot \sqrt{\text{Area}(\psi)}} \tag{15}$$

where $\text{CommonArea}(*, *)$ is the common area between the two rectangle features and $\text{Area}(*)$ denotes the area of a rectangle feature. In each iteration of the clustering algorithm, the center feature ϕ is selected and the remaining feature set is searched to select ψ's that satisfy the inequality

$$\langle \phi, \psi \rangle \geq \mu \ . \tag{16}$$

Integrating Eq. 15 and Eq. 16, we get

$$\frac{\text{CommonArea}(\phi, \psi)}{\sqrt{\text{Area}(\phi)} \cdot \sqrt{\text{Area}(\psi)}} \geq \mu \ . \tag{17}$$

Let w_* and h_* be the width and height of the non-zero rectangle of Haar-like feature $*$, then $\text{Area}(*) = w_* h_*$. The common area must be included in each of the two rectangle regions. A direct way is to search the bounding coordinates of feature ψ and to calculate their intersections. However, this method would be too much expensive. We instead search the bounding coordinates of the common area and infer feature ψ from the position of this common area. We suppose the common rectangle is of size $(w_\cap, h_\cap)$ and the extension from common rectangle to feature ψ is (l, r, t, b) indicating the left, right, top, and bottom margins respectively. Considering the fact that the common area between two rectangles is always a rectangle, we know that feature ψ is of size $(w_\psi, h_\psi) = (w_\cap + l + r, h_\cap + t + b)$. Therefore, Eq. 17 can be re-written as

$$\frac{w_\cap h_\cap}{\sqrt{w_\phi h_\phi}\sqrt{(w_\cap + l + r)(h_\cap + t + b)}} \geq \mu \tag{18}$$

and further simplified to

$$(w_\cap + l + r)(h_\cap + t + b) \leq \frac{w_\cap^2 h_\cap^2}{\mu^2 w_\phi h_\phi} = A_{\text{sup}} \ . \tag{19}$$

Intuitively, in the case that the common rectangle is completely included in the rectangle ϕ (no edge overlapping), it is certain that ψ is the same as the common rectangle. Thus, there should be limitations on the range of (l, r, t, b). Here, (x_*, y_*) is the coordinate of the top-left pixel of rectangle $*$.

$$0 \leq l \leq \begin{cases} \frac{A_{\text{sup}}}{h_\cap} - w_\cap, & x_\cap = x_\phi \\ 0, & x_\cap \neq x_\phi \end{cases}$$

$$0 \leq r \leq \begin{cases} \frac{A_{\text{sup}}}{h_\cap} - w_\cap - l, & x_\cap + w_\cap = x_\phi + w_\phi \\ 0, & x_\cap + w_\cap \neq x_\phi + w_\phi \end{cases}$$

$$0 \leq t \leq \begin{cases} \frac{A_{\text{sup}}}{(w_\cap - l - r)} - h_\cap, & y_\cap = y_\phi \\ 0, & y_\cap \neq y_\phi \end{cases}$$

$$0 \leq b \leq \begin{cases} \frac{A_{\text{sup}}}{(w_\cap - l - r)} - h_\cap - t, & y_\cap + h_\cap = y_\phi + h_\phi \\ 0, & y_\cap + h_\cap \neq y_\phi + h_\phi \end{cases}$$

Moreover, the size of common rectangle is limited to

$$w_\cap h_\cap \geq \mu^2 w_\phi h_\phi \ . \tag{20}$$

Finally, constrained search of $(x_\cap, y_\cap, w_\cap, h_\cap, l, r, t, b)$ leads to a fast implementation of dictionary clustering.

With dictionary pre-clustered, each iteration of the Discriminative OOMP can be performed hierarchically. The cluster centers are examined first, only those clusters that are close enough are further searched. This approximate solution can significantly reduce the computation load required with

minimal accuracy decreasing using carefully tuned parameter settings. The hierarchical D-OOMP algorithm is given as follows.

Algorithm 3. Hierarchical D-OOMP

Initialize dictionary $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots, \psi_{N_\psi}]$.
Cluster features with center index $\mathbf{C} = \{c_1, c_2, \ldots\}$ in accord with the given μ.
Select the first feature φ_1 and initialize data.
for $k = 2$ **to** K **do**
  Pre-compute $\mathbf{I}_k$ according to Eq. 13.
  Pre-compute S_k according to Eq. 14.
  for $i = 1$ **to** $|\mathbf{C}|$ **do**
    Calculate $\beta_{c_i}^{(k)} \leftarrow \text{PROD}(\psi_{c_i}, \Sigma(\overline{\varphi_{k-1}}))$.
    Calculate $d_{c_i}^{(k)}$:$\|\gamma_{c_i}^{(k)}\|^2 \leftarrow \|\gamma_{c_i}^{(k-1)}\|^2 - (\beta_{c_i}^{(k)})^2$
    Calculate $L_k(\psi_{c_i})$ according to Eq. 12.
  end for
  Get the optimal index: $opt = \arg\max_{c_i} L_k(\psi_{c_i})$
  for $i = 1$ **to** $|\mathbf{C}|$ **do**
    if $L_k(\psi_{c_i}) > L_k(\psi_{opt}) - ratio|L_k(\psi_{opt})|$ **then**
      for each feature ψ_j s.t. $\langle \psi_j, \psi_{c_i} \rangle \geq \mu$ **do**
        Calculate $L_k(\psi_j)$ according to Eq. 12.
        Update $opt = j$ if $L_k(\psi_j) > L_k(\psi_{opt})$.
      end for
    end if
  end for
  The k-th basis is $\phi_k = \psi_{opt}$.
end for

In the k-th iteration of feature selection, all the cluster centers are scored. Supposing the maximum is $L_k^{(\max)} = \max_{c_i}\{L_k(c_i)\}$, those groups whose central scores are bigger than $L_k^{(\max)} - \text{RATIO} \times |L_k^{(\max)}|$ are further examined. It is obvious that this pruning operation will lose some precision when this threshold is limited. We aim to seek a balance between efficiency and accuracy of Hierarchical D-OOMP here. The error score in Fig. 3 is defined using the function in Eq. 5. According to Fig. 3, when RATIO is between 0.3 and 0.6, the time consumption of the algorithm is relatively low (less than 1 seconds) while its accuracy is close to the original D-OOMP (when RATIO is infinitely large). We empirically set it to 0.5 in experiments.

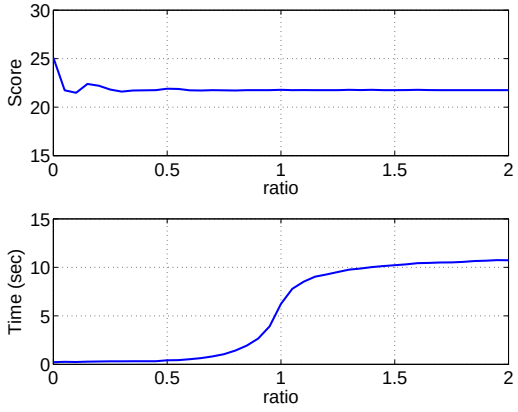


Fig. 3. Statistics on score and computational cost against the value of ratio in Hierarchical D-OOMP

C. Comparison of the Three Optimization Methods

We compare the performance of the original D-OOMP, Iterative D-OOMP and Hierarchical D-OOMP. As can be observed, there are several parameters that control the efficiency of D-OOMP, for example the number of bases selected and the number of samples used for training. The more features we need to select, the more time it takes. Also, in general, the more samples for training, the more computation it requires. In all of the following experiments, all the image templates are all of size 50×50. All time statistics are calculated excluding pre-processing.

In Fig. 4, we show the relation between the number of bases and optimization score by varying the number of bases from 1 to 100. As can be observed, the more bases, the better the solution is. The original D-OOMP and iterative D-OOMP have no difference in performance because the iterative D-OOMP is an equivalent transformation of the original D-OOMP. The hierarchical D-OOMP has slightly higher error because it yields an approximate solution.

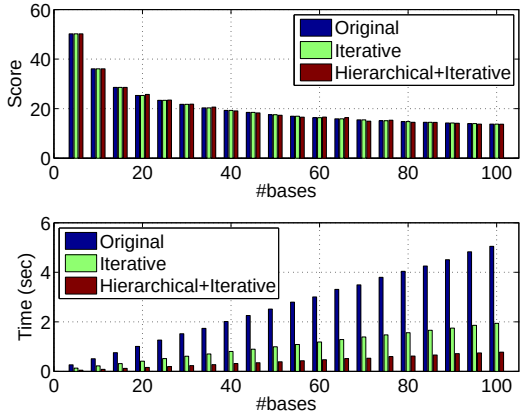


Fig. 4. Score and time against the number of bases, using 5 positive and 5 negative samples.

One of the major advantages of the two new efficient D-OOMP algorithms is that the computation is not sensitive to the total number of foreground and background samples. We here simply change the number of background samples from 5 to 100 and see how the reconstruction error and computation time change. The result is shown in Figure 5. As can be observed, as the number of training samples increases, the computation cost for the original D-OOMP goes linearly while the computation for iterative D-OOMP and hierarchical D-OOMP remains stable.

V. Tracking Using Discriminative NBS

We apply the DNBS representation to visual object tracking. With the DNBS object representation, we locate object position in the current frame through sum of squared difference (SSD)-based matching. Using discriminative NBS, the object is first compared with the possible locations in an region in the current frame around the predicted object position. The one with the minimum SSD value is chosen as the target object location. In order to accommodate object appearance changes,

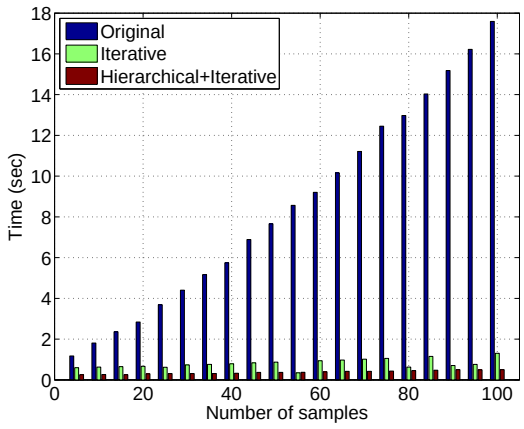


Fig. 5. Time consumption with respect to the number of background samples varying from 5 to 100.

the foreground and discriminative NBS are automatically updated every few frames.

A. Object Localization

We use SSD to match the template, due to its high efficiency of matching under the discriminative NBS representation. In each frame t, we specify a rectangular region centered at the predicted object location as the search window, in which the templates are sequentially compared with the referenced foreground $\mathbf{x} = R_{\mathbf{\Phi}_K^{(t)}}(\mathbf{f}_{\text{ref}}^{(t)})$.

Suppose that $\mathbf{x}$ is the object and $\mathbf{y}$ is a candidate object in the search window. The SSD between them is,

$$\text{SSD}(\mathbf{x},\mathbf{y}) = \| \mathbf{x}-\mathbf{y} \|^2 = \| \mathbf{x} \|^2 + \| \mathbf{y} \|^2 - 2\langle \mathbf{x},\mathbf{y} \rangle \ , \quad (21)$$

where $\| \cdot \|$ represents the L^2-norm and $\langle \cdot,\cdot \rangle$ denotes the inner product. $\mathbf{x}$ is approximated by DNBS $\mathbf{\Phi}_K$ (i.e. $R_{\mathbf{\Phi}_K^{(t)}}(\mathbf{f}_{\text{ref}}^{(t)}) = \sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)}$) , built using the approach in Section IV. Eq. 21 is then transformed to

$$\begin{aligned} &\text{SSD}(\sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)}, \mathbf{y}) \\ &= \| \sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)} \|^2 + \| \mathbf{y} \|^2 - 2 \sum_{i=1}^{K} c_i^{(t)} \langle \phi_i^{(t)}, \mathbf{y} \rangle \ . \end{aligned} \quad (22)$$

The first term is the same for all the candidate locations in the current frame, while the second and third ones can be computed rapidly using integral image. The online computational complexity of Eq. 22 is only $O(K)$, where K is the number of selected bases.

B. Subspace Update

Due to appearance changes of the object, the DNBS built in the previous frame might be unsuitable for the current frame. A strategy to dynamically update the subspace is necessary. Here we update the subspace every 5 frames. Once a new subspace needs to be computed, we first use the updated template and background samples from the current frame to compute the DNBS again as Eq. 5.

1) Template Update: The object template is also updated constantly to incorporate appearance changes and the updated template serves as the new positive sample. According to Eq. 5, DNBS is then constructed to better represent the object using an updated set of samples. Intuitively, these sampled foregrounds should recently appear, in order to more precisely describe the current status of the object. Many previous efforts have been devoted to template update (see [46]). One natural way is to choose the recent N_f referenced foregrounds. Another solution is to update the reference template in each frame, but this may incur considerable error accumulation. Simply keeping it unchanged is also problematic due to object appearance changes. A feasible way is to update the foreground by combining the frames using time-decayed coefficients. Here, we propose to update the foreground reference for every N_u frames,

$$\mathbf{f}_{\text{ref}}^{(t)} = \begin{cases} \mathbf{f}_0 & t = 0 \\ \gamma \mathbf{f}_{\text{ref}}^{(\lfloor (t-1)/N_u \rfloor N_u)} + (1-\gamma)\mathbf{f}_t & \text{otherwise} \ , \end{cases}$$

where $\mathbf{f}_0$ is the foreground specified in the first frame and $\mathbf{f}_t$ is the matched template at frame t. γ is the tradeoff, which is empirically set to 0.5 in our experiments. $\lfloor (t-1)/N_u \rfloor N_u$ is the frame at which the current subspace is updated. $\mathbf{f}_{\text{ref}}^{(\lfloor (t-1)/N_u \rfloor N_u)}$ is the object template at that frame. This means we are updating the template periodically instead of at each frame, which is more robust to tracking errors. This template updating scheme is compared with other methods and the results are shown in the experimental section.

2) Background Sampling: The background samples which closely resemble the reference foreground often interfere with the stability and accuracy of tracker. We sample the background templates which are similar to the current reference object and take them as the negative data in solving the DNBS. We compute a distance map in a region around the object and those locations that are very similar to the object are selected as the negative samples. This process can be done efficiently because the SSD distance map can be computed efficiently using Haar-like features and integral images. Once the distance map is computed, the local minima locations are used to select negative training examples by means of non-minimal suppression.

VI. Experiments

The proposed approach is evaluated on a set of sequences extracted from public video datasets. These sequences are challenging because of their background clutter and camera motion. Some key parameters, such as λ used in the DNBS formulation and μ used in hierarchical D-OOMP, are firstly discussed in this section. The qualitative tracking results are shown afterwards. To demonstrate the advantages of our approach and the benefit of the discriminative term in DNBS, we qualitatively compare our tracker with an NBS tracker which applies the original NBS object representation. We show in this comparison that the discriminative terms in DNBS help increase the tracking accuracy. Quantitative evaluations are conducted by comparing the success rates of our tracker against several state-of-the-art trackers. In addition, we also

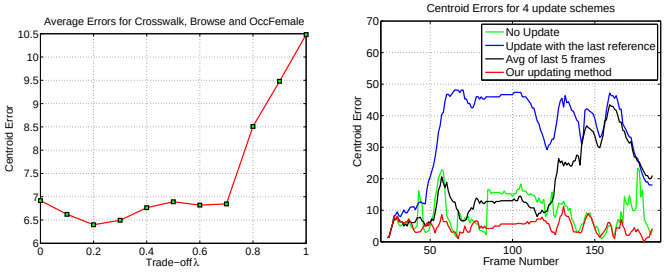
provide a comprehensive comparison by employing the evaluation protocols proposed by [47]. While achieving a relatively stable performance, our tracker is able to be processed in real-time.

A. Parameter Selection

Several parameters are used in the DNBS such as the trade-off λ between foreground and background reconstruction errors. Intuitively, those parameters can influence the accuracy of object reconstruction and the tracking performance. So we perform experiments on these paramters and discuss the justification of the selections.

The formulation of the DNBS balances the influence of the foreground and background reconstruction terms with a coefficient λ. Intuitively, it should be set to a small value to ensure the accuracy of foreground representation. To find the best value, we use several image sequences (*"Browse"*, *"Crosswalk"* and *"OccFemale"*) with ground-truth to quantitatively evaluate how this parameter affects the tracking accuracy. To generate more data for evaluation, we split each of the sequences into multiple subsequences initialized at different frames.

The tracking performance is evaluated using the mean distance error between the tracked object location and the groundtruth object center. Specifically, we initialize our tracker in each of the frame using the groundtruth as the bounding box and record average tracking errors for the subsequent 20 frames with different choices of the parameter λ. For each sequence, errors of all the subsequences under the same λ are averaged and plotted in Fig. 6(a). As is observed, the centroid error is relatively more stable and smaller when λ is set to 0.25.



(a) (b)

Fig. 6. (a) The influence of λ on the averaged tracking errors on multiple sequences. (b) Performance (centroid tracking errors) comparison among the four template updating schemes.

Another parameter for DNBS is the number of bases K used. The selection of this parameter depends on image content. In general, the more features we use, the more accuracy DNBS is able to reconstruct the object. However, more features bring more computational costs. As a tradeoff, we set $K = 30$. We empirically set the number of foreground templates N_f to 3 and that of background ones N_b to 3. These parameters are fixed for all the experiments.

We also conduct experiments to show the effectiveness of our template updating scheme. Here, we review several template updating methods mentioned above by comparing their tracking errors of video sequence *Browse*. These updating schemes include: 1) updating the current template with the previous one; 2) updating the current template with an average of previous 5 frames and our updating method. All of the schemes are initialized with the same bounding box at the first frame and the error of object center is computed with respect to the groundtruth. Fig. 6(b) shows that the time-decaying approach is more robust and stable.

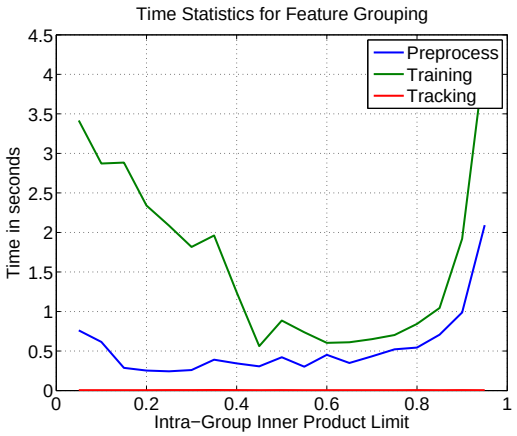


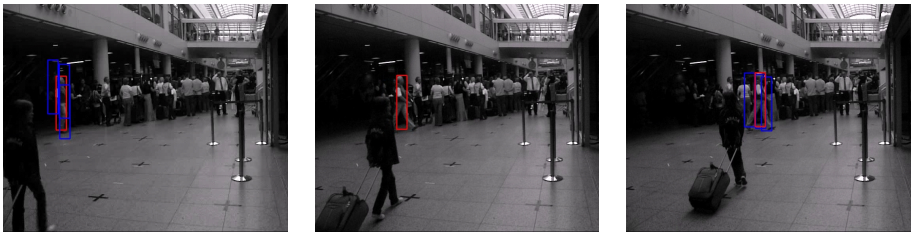
Fig. 7. Time statistics on building dictionary hierarchy (preprocessing), subspace feature selection (training) and object localization (tracking per frame) for Sequence *Crosswalk* with respect to μ.

Fig. 7 shows the detailed computational cost with respect to the selection of μ for tracking using hierarchical D-OOMP. The time statistics has three components: (a) preprocessing the Haar-like dictionary and setting up relevant parameters for tracking tasks (shown as the blue curve); (b) training, i.e., optimizing the DNBS formulation to obtain the up-to-date DNBS subspace representation (shown as the green curve); (c) tracking and localizing the foreground object in each frame (shown as red curve). The merit of using Haar-like features is revealed in Fig. 7 that the tracking procedure has an extremely low computation cost. Also, as the inner product upper limit μ changes we could find the optimal spot between 0.6 and 0.8 where the time consumption for all of the three procedures is the minimum.

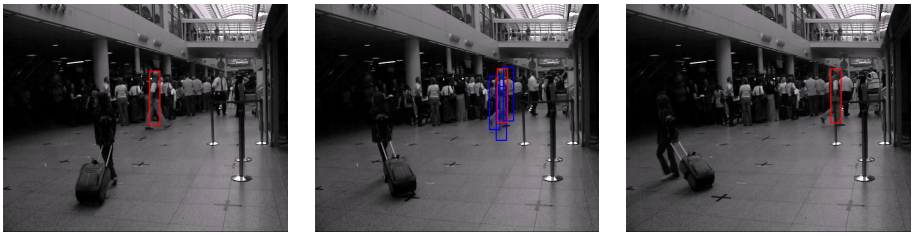
B. Qualitative Results

We apply our tracker to several challenging sequences to show its effectiveness. Qualitative results are demonstrated on pedestrian videos to show that our tracker can handle background clutter, heavy camera motion, and object appearance variations. In the following figures, red boxes indicate tracked object while blue ones are the negative samples selected when the object DNBS is update at that frame. The subspace is updated every 5 frames and if there is no update of subspace. No blue boxes (background samples) will be shown while no subspace update is performed.

We qualitatively compare the tracking results of the proposed DNBS approach with the NBS tracker to show the power of the additional discriminative term in Eq. 5. To make

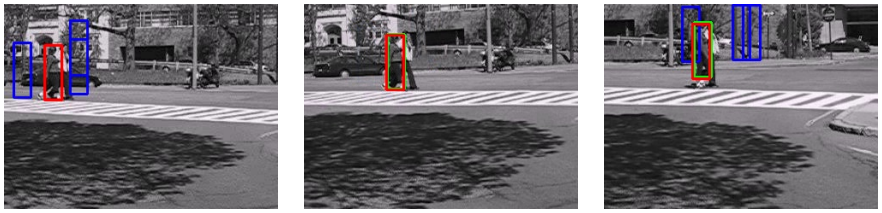


(a) Frame 1 (b) Frame 18 (c) Frame 46

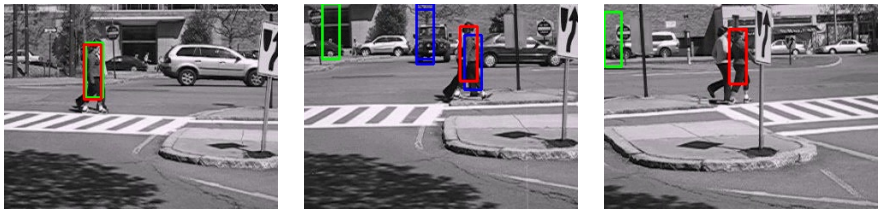


(d) Frame 57 (e) Frame 76 (f) Frame 89

Fig. 8. Sequence *Crowd*: The frames 1, 18, 46, 57, 76 and 89 are shown. The red boxes are tracked objects using DNBS, the green boxes are tracking results using NBS and the blue boxes in some of the frames are sampled backgrounds for subspace update in DNBS.



(a) Frame 1 (b) Frame 17 (c) Frame 51



(d) Frame 75 (e) Frame 106 (f) Frame 140

Fig. 9. Sequence *Crosswalk*: The frames 1, 17, 51, 75, 106and 140 are shown. The red boxes are tracked objects using DNBS, the green boxes are tracking results using NBS and the blue boxes in some of the frames are sampled backgrounds for subspace update in DNBS.

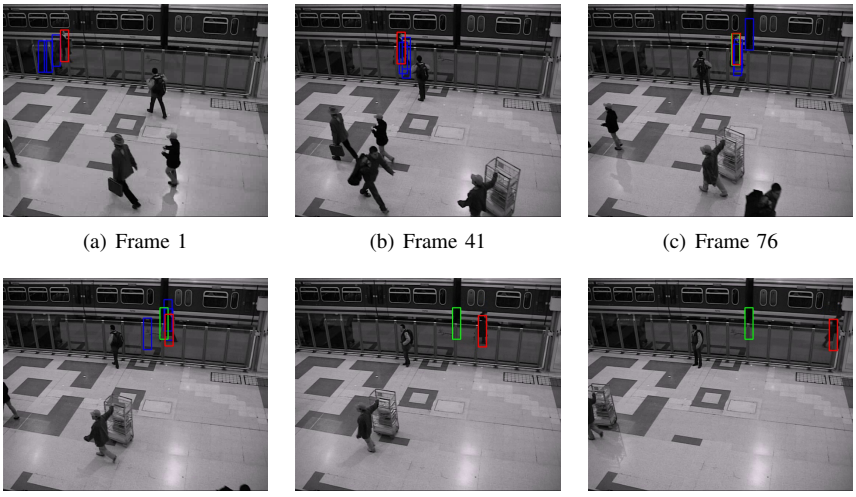
the comparison fair, we fix the number of selected bases for both NBS and DNBS to be 30.

Sequence *Crowd* (Fig. 8) is a video clip selected from PETS 2007 data set. In this sequence the background is cluttered with many distracters. As can be observed the object can still be well tracked. The frame is of size 720×576 and the object is initialized with a 26×136 bounding box.

Sequence *Crosswalk* (Fig. 9) has totally 140 frames, with two pedestrians walking together along a crowded street with an extremely cluttered background. The tracking result demonstrates the discriminative power of our algorithm. In this sequence the hand-held camera is extremely unstable. The shaky nature of the sequence makes it difficult to accurately track the pedestrians. Despite this, our algorithm is able to track the pedestrians throughout the entire 140 frames of the sequence.

Sequence *OccFemale* (Fig. 10) is a video clip selected from the PETS 2006 data set. Each frame is of size 720×576 and the object is initialized with a 22×85 bounding box at the beginning. It can be observed that the person being tracked is of low texture with very similar background and the person is also occluded by the fences periodically. In particular, the person's cloth is almost all black which makes it very similar to the black connector of the two compartments. When the person walks by the black connector at frame 90, the NBS tracker loses track (shown as a green box) while the DNBS tracker (shown as red box) can still keep track. This is because, at frame 76 this connector was selected as the background negative samples (the blue box) for model updating which makes the tracker aware of the distracting surroundings. The object can thus be tracked stably.

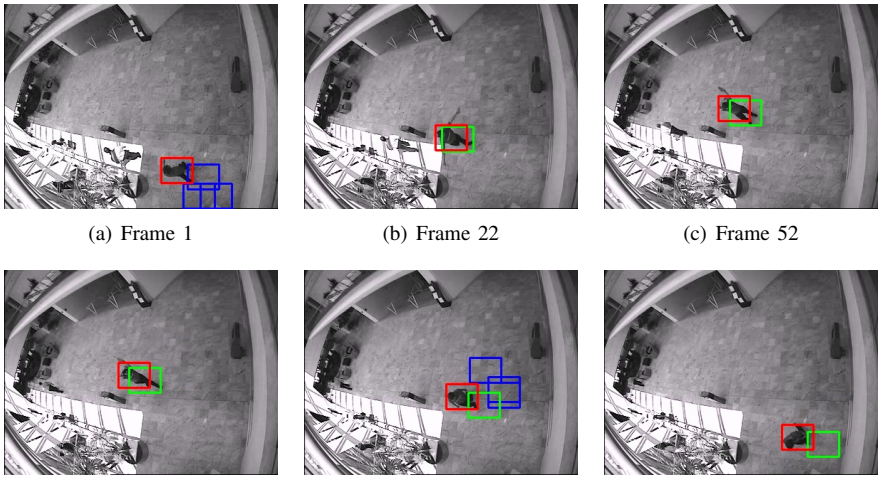
Sequence *Browse* (Fig. 11) is a video clip of frames 24-



(a) Frame 1 (b) Frame 41 (c) Frame 76

(d) Frame 91 (e) Frame 107 (f) Frame 153

Fig. 10. Sequence *OccFemale*: The frames 1, 41, 76, 91, 107, and 153 are shown. The red boxes are tracked objects using DNBS, the green boxes are tracking results using NBS and the blue boxes in some of the frames are sampled backgrounds for subspace update in DNBS.



(a) Frame 1 (b) Frame 22 (c) Frame 52

(d) Frame 92 (e) Frame 116 (f) Frame 142

Fig. 11. Sequence *Browse* : The frames 1, 22, 52, 92, 116, 142 are shown. The red boxes are tracked objects using DNBS, the green boxes are tracking results using NBS and the blue boxes in some of the frames are sampled backgrounds for subspace update in DNBS.

201 extracted from *Browse1.avi* in CAVIAR people (ECCV-PETS 2004) dataset[1]. This sequence is recorded by a distorted camera. Each frame is 384×288 pixels and the object is bounded by a 44×35 box. With significant distortion, the object can still be tracked.

In addition, we validated our tracker on other sequences from public video datasets such as Sequence *Courtyard*, Sequence *Ferry* which is extracted from PETS 2005 Zodiac Dataset[2], Sequence *CrowdFemale* extracted from PETS 2007[3], and sequences *boy*, *car4*, *couple*, *crossing*, *david*, *david2*, *fish*, *girl*, *matrix*, *mhyang*, *soccer*, *suv*, *trellis* which are used in previous literatures [47]. Qualitative video results for all of these 21 sequences are available at our demo webpage[4].

C. Quantitative Evaluation

In the quantitative evaluation, we compare our tracker with 8 state-of-the-art trackers, which are CSK [19], CT [17], DFT [48], IVT [16], L1APG [49], ORIA [50], Struck [20] and TLD [21], using 21 public video sequences.

In the first place, we give a detailed comparison in Table I where each of the trackers is evaluated on the same set of 21 video sequences. Each cell in the the table shows the percentage of successfully tracked frames with respect to the corresponding tracker-sequence pair. A frame is successfully tracked if and only if the overlap ratio (intersection area over union area) between tracked object and ground truth is higher

[1]CAVIAR Dataset, EC Funded CAVIAR project: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[2]PETS 2005 Zodiac: http://www.vast.uccs.edu/~tboult/PETS05/

[3]PETS 2007: http://www.cvg.reading.ac.uk/PETS2007/data.html

[4]DNBS webpage: http://www.cs.umd.edu/users/angli/dnbs

TABLE I

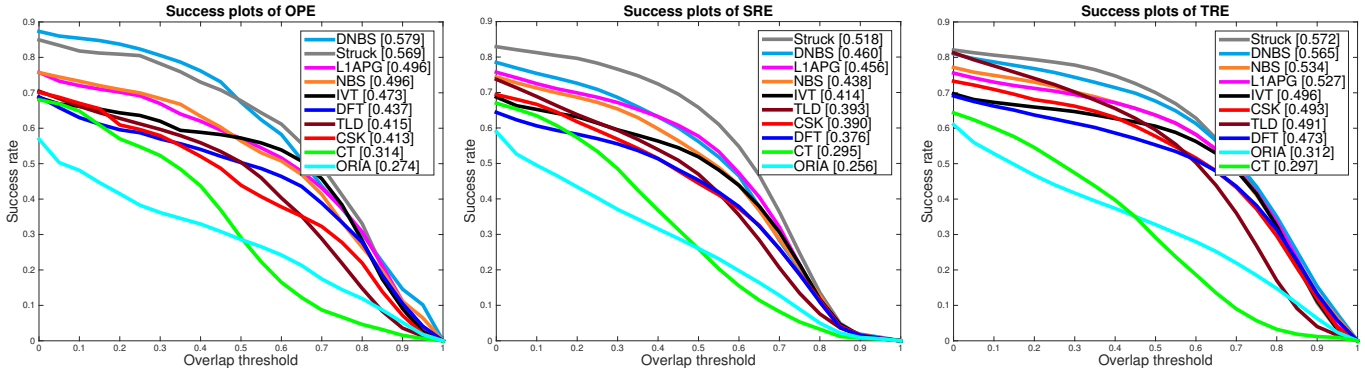
PERFORMANCE EVALUATION USING THE FRACTION OF SUCCESSFULLY TRACKED FRAMES. A FRAME IS SUCCESSFULLY TRACKED IF AND ONLY IF THE OVERLAP RATIO, I.E. INTERSECTION OVER UNION (IOU), IS HIGHER THAN 0.35 (ABOUT HALF OF THE BOUNDING BOX OVERLAPED), WITH RESPECT TO THE GROUND TRUTH BOUNDING BOX. THE TIMING OF EACH METHOD (FRAMES PER SECOND) IS COMPUTED WITH RESPECT TO A 44×35 OBJECT TEMPLATE. FOR EACH SEQUENCE, THE BEST SUCCESS FRACTION IS HIGHLIGHTED IN RED COLOR. OUR TRACKER RANKS 1ST BOTH IN THE AVERAGED SUCCESS RATE AND IN THE TOTAL NUMBER OF WINNING SEQUENCES WITH A MODERATE REAL TIME SPEED COMPARED TO ALL OTHER TRACKERS.

#	Sequence	CSK	CT	DFT	IVT	L1APG	ORIA	Struck	TLD	Proposed	
										NBS	DNBS
1	blackman	0.55	0.01	**1.00**	**1.00**	**1.00**	0.13	0.99	0.32	**1.00**	**1.00**
2	boy	0.84	0.64	0.48	0.33	0.93	0.18	**1.00**	**1.00**	0.43	0.44
3	browse	0.85	0.39	0.88	0.15	**0.90**	0.07	0.70	0.09	0.87	0.87
4	car4	0.67	0.35	0.26	**1.00**	0.32	0.24	0.73	0.27	0.28	0.30
5	couple	0.09	0.32	0.09	0.09	0.61	0.05	0.71	0.25	0.54	**0.99**
6	courtyard	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.34	**1.00**	**1.00**	0.96	**1.00**
7	crossing	0.88	0.99	0.68	0.40	0.25	0.21	**1.00**	0.52	0.37	0.71
8	crosswalk	0.06	0.09	0.09	0.09	0.10	0.01	0.66	0.12	0.55	**0.99**
9	crowd	**1.00**	0.01	**1.00**	**1.00**	**1.00**	0.47	**1.00**	0.57	**1.00**	**1.00**
10	crowdfemale	0.16	0.95	**1.00**	0.99	**1.00**	0.42	0.97	0.74	0.49	**1.00**
11	david	0.48	0.90	0.35	**0.96**	0.81	0.47	0.31	0.63	0.84	0.88
12	david2	**1.00**	0.00	0.60	**1.00**	**1.00**	0.70	**1.00**	**1.00**	**1.00**	**1.00**
13	ferry	0.28	0.27	0.29	0.27	0.27	0.05	0.30	0.81	**0.99**	**0.99**
14	fish	0.04	0.97	0.86	**1.00**	0.26	0.70	**1.00**	0.68	**1.00**	**1.00**
15	girl	0.48	0.35	0.29	0.21	0.99	0.55	**1.00**	0.87	0.78	0.70
16	matrix	0.01	0.10	0.06	0.02	0.11	**0.23**	0.19	0.12	0.02	0.02
17	mhyang	**1.00**	0.86	0.94	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
18	occfemale	0.56	**1.00**	0.99	0.99	0.83	0.06	0.85	0.61	0.57	0.99
19	soccer	0.16	**0.34**	0.23	0.16	0.21	0.17	0.16	0.15	0.24	0.24
20	suv	0.59	0.24	0.06	0.46	0.55	0.59	0.73	**0.98**	0.53	0.57
21	trellis	**0.85**	0.40	0.53	0.37	0.31	0.61	0.63	0.43	0.56	**0.85**
averaged success rate		0.55	0.48	0.56	0.59	0.64	0.35	0.76	0.58	0.67	**0.79**
# winning sequences		5	3	4	8	7	2	8	5	6	**11**
frames per second		**342**	76	10	26	1	14	7	24	22	17

than 0.35, i.e. more than half portion of the object is overlapped with the groundtruth bounding box. For each sequence, the highest success fractions are highlighted in red color. The last three rows show respectively the averaged success rate, the number of winning sequences, and the frames per second. The average success rate is computed by averaging the success rate for each of the sequences. According to the result, the proposed DNBS tracker performs the best (0.79 in average success rate) and Struck ranks 2nd with a very close success rate 0.76. The number of winning sequences is computed by counting the total number of sequences where each of the methods wins. The proposed DNBS approach wins in total 11 sequences (Blackman, Couple, Courtyard, Crosswalk, Crowd, Crowdfemale, Ferry, Fish, Mhyang, Occfemale and Trellis) which outperforms all the other trackers. Struck and IVT tied for the 2nd place with 8 winings. The number of frames tracked per second (FPS) is shown in the last row which is calculated using an object template of size 44×35. Our DNBS tracker runs in real-time at 17 frames per second which is faster than DFT, L1APG, ORIA and Struck.

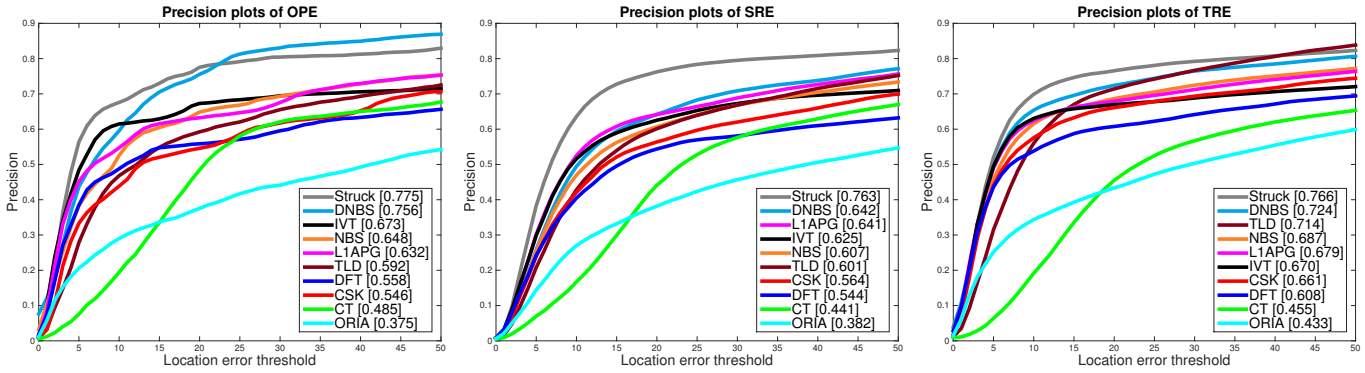
For a more comprehensive evaluation, we employ the quantitative evaluation protocols proposed by [47]. There are three criteria used in their evaluation benchmark: (a) one-pass evaluation (OPE) tests each tracker from the beginning of the sequence to the end; (b) spatial robustness evaluation (SRE) initializes the tracker 12 times on each sequence by spatially pertubing the initial bounding box and averages the performance of different initializations over all trials; and (c) temporal robustness evaluation (TRE) segments each sequence into 20 segments and tests the tracker on each segment independently and averages their performances over all trials. Besides, two error functions are employed: the centeroid distance from the tracked object location to the ground-truth location and the bounding box overlap (Intersection-Over-Union) ratio. Such comprehensive criteria provide a better evaluation of the robustness of trackers. The resulting precision curves using overlap rate are shown in Fig. 12 and curves using centroid distance error are shown in Fig. 13. According to the curves, the performance of our tracker is the best in OPE evaluation with overlap ratio, and ranks 2nd with any of the other 5 evaluation criteria.

According to quantitative results, it is hard to say that there is a tracker performs best in both computation cost and accuracy. In Table I, our DNBS tracker outperforms other trackers in 11 of these sequences and its speed is faster than half of the trackers. The second most accurate tracker is Struck. However, in the comprehensive evaluation (Fig. 13), Struck performs the best for 5 of the criteria for which our DNBS tracker is the second. We have to admit that Struck provides more robustness in the application of object tracking. However, its speed is twice slower than our DNBS tracker. Since our method is based on template matching, we observed that it is hard for our tracker to compete with Struck in every scenario, which is based on structural learning. When compared to methods using similar techniques, our DNBS tracker outperforms all those subspace representation based approaches used in our evaluation such as IVT, CT, and L1APG. Since our approach does not rely on partile filtering, it provides robustness and efficiency in videos with heavy camera motion. The drawback of our approach is that it does



(a) (b) (c)

Fig. 12. Quantitative results on the success rate with respect to overlap ratio over 21 sequences: (a) One-pass evaluation (b) Spatial-robustness evaluation (c) Temporal-robustness evaluation



(a) (b) (c)

Fig. 13. Quantitative results on the object center distance error evaluations over 21 sequences: (a) One-pass evaluation (b) Spatial-robustness evaluation (c) Temporal-robustness evaluation

not very well handle scale changes and non-rigid intra-object motions which is due to the nature of template matching. After all, our tracker is based on much simpler principles and algorithms which produces a relatively balanced performance in both accuracy and computation.

VII. Conclusion

We have proposed the Discriminative Nonorthogonal Binary Subspace, a simple yet informative object representation that can be solved using a variant of OOMP. The proposed DNBS representation incorporates the discriminate image information to distinguish the foreground and background, making it suitable for object tracking. We used SSD matching built upon the DNBS to efficiently locate object in videos. The optimization of DNBS is efficient as we proposed a suite of algorithms to accelerate the training process. Our experiments on challenging video sequences show that the DNBS-based tracker can stably track the dynamic objects. In the future, we intend to explore the applications of DNBS on other computer vision and multimedia tasks such as image copy detection and face verification.

Acknowledgment

The authors would like to thank Dr. Terry Boult for sharing the Zodiac video dataset.

References

[1] G. Hager, M. Dewan, and C. Stewart, "Multiple kernel tracking with ssd," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. I: 790–797.

[2] B. Han and L. Davis, "On-line density-based appearance modeling for object tracking," in *Proc. Int'l Conf. Computer Vision*, 2005, pp. II: 1492–1499.

[3] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, June 2004.

[4] M. J. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. European Conf. Computer Vision*, 1996, pp. 329–342.

[5] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, June 2001.

[6] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, October 2003.

[7] H. Tao, H. Sawhney, and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75–89, January 2002.

[8] D. Comaniciu, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.

[9] Z. Fan and Y. Wu, "Multiple collaborative kernel tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. II: 502–509.

[10] S. Birchfield and R. Sriram, "Spatiograms versus histograms for region-based tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. II: 1158–1163.

[11] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
[12] F. Tang and H. Tao, "Object tracking with dynamic feature graphs," in *Workshop on VS-PETS*, 2005, pp. 25–32.
[13] Y. Chen, Y. Rui, and T. Huang, "Jpdaf based hmm for real-time contour tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, pp. I:543–550.
[14] R. Collins, Y. Liu, and M. Leordeanu, "On-line selection of discriminative tracking features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, October 2005.
[15] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, February 2007.
[16] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *Int'l Journal Computer Vision*, vol. 77, no. 1-3, pp. 125–141, May 2008.
[17] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. European Conf. on Computer Vision*, Berlin, Heidelberg, 2012, pp. 864–877.
[18] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 184–191, 2001.
[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. European Conf. on Computer Vision*, 2012.
[20] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels." in *Int'l Conf. Computer Vision*, 2011, pp. 263–270.
[21] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
[22] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in *Proc. European Conf. on Computer Vision*, 2010, pp. I: 480–493.
[23] H. Nguyen and A. Smeulders, "Robust tracking using foreground-background texture discrimination," *Int'l Journal of Computer Vision*, vol. 69, no. 3, pp. 277–293, September 2006.
[24] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. Int'l Conf. Computer Vision*, 2007, pp. 1–8.
[25] B. Babenko, M.-H. Yang, and S. Belongie, "Visual Tracking with Online Multiple Instance Learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
[26] C. Leistner, A. Saffari, and H. Bischof, "Miforests: multiple-instance learning with randomized trees," in *Proc. European Conf. Computer Vision*, 2010, pp. 29–42.
[27] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, "On-line semi-supervised multiple-instance boosting," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, p. 1879.
[28] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-class LPBoost," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2010, pp. 3570–3577.
[29] F. Tang, R. Crabb, and H. Tao, "Representing images using nonorthogonal haar-like bases," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2120–2134, 2007.
[30] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems*, 1998, pp. 487–493.
[31] R.-S. Lin, D. A. Ross, J. Lim, and M.-H. Yang, "Adaptive discriminative generative model and its applications," in *Advances in Neural Information Processing Systems*, 2004.
[32] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 87–94.
[33] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum, "Classification with hybrid generative/discriminative models," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
[34] H. Grabner, P. M. Roth, and H. Bischof, "Eigenboosting: Combining discriminative and generative information," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, 2007.
[35] Z. Tu, "Learning generative models via discriminative approaches," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.
[36] Q. Yu, T. B. Dinh, and G. G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. European Conf. Computer Vision*, 2008, pp. 678–691.
[37] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision)*, 2015.
[38] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
[39] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
[40] A. Li, F. Tang, Y. Guo, and H. Tao, "Discriminative nonorthogonal binary subspace tracking," in *Proc. European Conf. on Computer Vision*, vol. 6313, 2010, pp. 258–271.
[41] P. Viola and M. Jones, "Robust real-time face detection," *Int'l Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
[42] T. Mita, T. Kaneko, and O. Hori, "Joint haar-like features for face detection," in *Proc. Int'l Conf. Computer Vision*, 2005, pp. 1619–1626.
[43] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.
[44] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 260–267.
[45] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 137–140, 2002.
[46] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.
[47] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2013.
[48] L. Sevilla-Lara, "Distribution fields for tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1910–1917.
[49] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Washington, DC, USA, 2012, pp. 1830–1837.
[50] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 1808–1814.

Ang Li received the B.S. degree in computer science and technology from Nanjing University, Nanjing, China, in 2011. He is currently working towards the Ph.D. degree in computer science at the University of Maryland, College Park, MD. His research interest is computer vision and machine learning.

Feng Tang received the BS and MEng degrees from the Computer Science Department of Zhejiang University in 2001 and 2004, and PhD degree in the Computer Engineering Department, University of Califor- nia, Santa Cruz in 2008. He joined Hewlett-Packard Laboratories Palo Alto in February, 2009 where he is currently a Senior Researcher. His research interests are in image representation, tracking, and machine learning. He received the best paper award for ACM Multimedia Modeling in 2011.

Yanwen Guo received the Ph.D. degree in applied mathematics from State Key Lab of CAD&CG, Zhejiang University in 2006. He is currently an Associate Professor at the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University. His research interests mainly include image and video processing, image representation, and geometry processing. He was a visiting researcher in the Department of Computer Science and Engineering, the Chinese University of Hong Kong in 2006 and a visiting professor in Department of Computer Science, the University of Hong Kong in 2008 and 2012.

Hai Tao eceived the BS and MS degrees in automation from Tsinghua University in 1991 and 1993, respectively, the MS degree in electrical engineering from Mississippi State University in 1995, and the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1999. From 1999 to 2001, he was a member of the technical staff in the Vision Technology Laboratory at Sarnoff Corp., New Jersey. Since July 2001, he has been with the Department of Computer Engineering at the University of California, Santa Cruz, where he is now an associate professor. His research interests include image and video processing, computer vision, vision-based computer graphics, and human-computer interaction. He has published more than 50 technical papers and holds 12 US patents. In 2004, he received the US National Science Foundation Faculty Early Career Development (CAREER) Award. He is a senior member of the IEEE and currently serves as an associate editor for the journals Machine Vision and Applications and Pattern Recognition.