Discriminative Nonorthogonal Binary Subspace Tracking

Ang Li[1], Feng Tang[2], Yanwen Guo[1,3], and Hai Tao[4]

[1] National Key Lab for Novel Software Technology, Nanjing University, China
[2] Multimedia Interaction and Understanding Lab, HP Labs Palo Alto, USA
[3] Jiangyin Institute of Information Technology of Nanjing University, China
[4] University of California, Santa Cruz, USA

Abstract. Visual tracking is one of the central problems in computer vision. A crucial problem of tracking is how to represent the object. Traditional appearance-based trackers are using increasingly more complex features in order to be robust. However, complex representations typically will not only require more computation for feature extraction, but also make the state inference complicated. In this paper, 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 image template using discriminative non-orthogonal binary subspace spanned by Haar-like features. These Haar-like bases are selected from the over-complete dictionary using a variation of the OOMP (optimized orthogonal matching pursuit). Such a 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 the foreground and background. We apply the discriminative NBS to object tracking through SSD-based template matching. An update scheme of the discriminative NBS is devised in order to accommodate object appearance changes. We validate the effectiveness of our method through extensive experiments on challenging videos and demonstrate its capability to track objects in clutter and moving background.

1 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 category of method is appearance based approaches, these trackers represent the object to be tracked using an appearance model and it is matched to each new frame to determine the object state. In order to handle appearance variations, an appearance update scheme

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 258–271, 2010.

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 the background. In order to handle this problem, more and more complicated object representations that take into account color, gradients, texture 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 feature is 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 the object. Effective and efficient image representation not only makes the feature extraction process fast but also reduces the computation for object state inference. Traditional object representations for example raw pixels, color histograms are generative in natural, they are usually designed to describe the appearance of the 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.

Recently, discriminative methods 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. The support vector tracker [16] (denoted as SVT afterwards) uses an offline-learned support vector machine as the classifier and embeds it into an optical flow based tracker. Collins et al. [14] were perhaps the first to treat tracking as a binary classification problem. A classifier is learnt in each frame to be used to locate object in the next frame. A feature selection scheme using variance ratio to select the most discriminative features is used to measure feature discriminability and select the best feature for tracking. Avidan's ensemble tracker [15] combines an ensemble of online learned weak classifiers using AdaBoost to label pixels in the next frame. After the data is labeled, the peak of the classification score map is detected to be the object. To handle the object appearance changes and maintain temporal coherence, in each frame some classifiers that do not perform well or have existed longer than a fixed number of frames get removed or pruned from the tracker, and new classifiers are trained to replace them. In co-tracking [17], two semi-supervised support vector machines are built for color and gradient features. A co-training framework is used to update the classifiers.

Previous discriminative trackers generally have two major problems. First, the tracker only relies on the classifier which can well separate the foreground and background and does not have any information what the object is like. This makes it hard to recover once the tracker makes a mistake. Second, discriminative trackers generally have a fixed image representation for all objects to be tracked and this representation is not updated any more. However, adaptive objective representation is more desirable in most cases because it can capture the characteristics of particular object being tracked.

In this paper, we propose an extremely simple object representation using Haar-like features that combines the advantage of generative trackers and discriminative trackers. 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 [18]. The original NBS tries to select from an over-complete dictionary a set of Haar-like features that can best represent the image. We extend the NBS method to incorporate discriminative information by adding a discriminative background term. The new representation is called discriminative non-orthogonal binary subspace. The discriminative nonorthogonal binary subspace is a compact representation of an image which is spanned by Haar-like rectangle base vectors. By approximating image patches with discriminative NBS, the inner product between templates could be obtained very fast using integral image trick. We show in this paper that such extremely simple features can be used for effective object tracking even when the object is similar to background.

The rest of this paper is organized as follows. In section 2, we briefly review Haar-like features and the non-orthogonal binary subspace approach. The discriminative nonorthogonal binary subspace is proposed in section 3. In section 4, the application of discriminative NBS to tracking is described. Afterwards, we provide both qualitative and quantitative experimental results in section 5. The paper is concluded in section 6.

2 Background: Nonorthogonal Binary Subspace

The original NBS [18] tries to find a subset of Haar-like features from an over-complete dictionary to span a subspace that can be used to reconstruct the original image.

The Haar-like box function ϕ for NBS is defined as,

$$\phi(u,v) = \begin{cases} 1, & u_0 \le u \le u_0 + w' - 1 \\ & v_0 \le v \le v_0 + h' - 1 \\ 0, & \text{otherwise} \ , \end{cases} \tag{1}$$

where w' and h' represent the width and height of the box in the template. (u_0, v_0) is its top-left pixel. 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.

Suppose that for a given image template $\mathbf{x} \in R^{WH}$ 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 expressed 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 a base matrix, each column of which is a chosen binary base vector. Note that, this base set is non-orthogonal in general, hence the reconstruction vector of template $\mathbf{x}$ is calculated as

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

The number of Haar-like box functions is $W(W+1)H(H+1)/4$, thus the dictionary of base vectors is highly redundant. In previous work, the NBS is used to approximate the image template. Thus, a specific small number of features are chosen from the over-complete dictionary to optimize the function

$$\arg\min_{\mathbf{\Phi}_K} \parallel \mathbf{x} - R_{\mathbf{\Phi_K}}(\mathbf{x}) \parallel \ . \tag{3}$$

Since the dictionary is highly redundant, the optimal solution to Eq.(3) is NP-hard. It is shown in [18,19] that a sub-optimal solution can be produced by a greedy algorithm named the optimized orthogonal matching pursuit (OOMP).

3 Discriminative Nonorthogonal Binary Subspace

The NBS method has been successfully used for fast template matching and face recognition [18]. However, it only considers the information embodied in the object image itself without any information about the rest of the image. In the applications such as video object tracking, which is essentially a classification problem, the background content should be taken into account in addition to the object template. To account for this, we propose a discriminative NBS (D-NBS) image representation that considers both foreground object 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 best separate the object from background.

3.1 Formulation

The objective of discriminative NBS is to construct an object representation that can separate object from background. This will facilitate vision tasks such as object tracking. In contrast to the original NBS, we formulate the discriminative NBS by finding the bases such that the foreground can be well separated with background for SSD based template matching.

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.(3) in which only the foreground reconstruction is considered, in discriminative NBS formulation, the objective function has foreground and background reconstruction terms.

Let $\mathbf{\Phi}_K$ be the discriminative NBS based 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 $\mathbf{F} = \left[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}\right]$

is a matrix of N_f recent foreground samples. $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{N_b}]$ is a matrix of N_b sampled background vectors. The objective function for $\mathbf{\Phi}_K$ is

$$\arg\min_{\mathbf{\Phi}_K} \left\{ \frac{1}{N_f} \parallel \mathbf{F} - R_{\mathbf{\Phi}_K}(\mathbf{F}) \parallel_F^2 - \frac{\lambda}{N_b} \parallel \mathbf{B} - R_{\mathbf{\Phi}_K}(\mathbf{B}) \parallel_F^2 \right\} , \tag{4}$$

where $\| \cdot \|_F$ represents the Frobenius norm. The first term in the equation is to make the foreground better approximated while the second one is to make the representation far away from background. This formulation is a hybrid approach in which the generative and discriminative items are balanced by λ.

To make it more clear, Eq.(4) can be transformed to

$$\arg\min_{\mathbf{\Phi}_K} \left\{ \frac{1}{N_f} \sum_{i=1}^{N_f} \parallel \mathbf{f}_i - R_{\mathbf{\Phi}_K}(\mathbf{f}_i) \parallel^2 - \frac{\lambda}{N_b} \sum_{i=1}^{N_b} \parallel \mathbf{b}_i - R_{\mathbf{\Phi}_K}(\mathbf{b}_i) \parallel^2 \right\} . \tag{5}$$

It can be further simplified to

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

3.2 Solution

It can be proved that Eq.(4) is a NP hard problem, even verification of a solution is difficult. To optimize the objective function, we propose an extension of OOMP(Optimized Orthogonal Matching Pursuit) [18] called discriminative OOMP. Similar to OOMP, discriminative OOMP is a greedy algorithm to compute adaptive signal representation by iterative selection of base vectors from a dictionary.

We assume that totally K base vectors are to be chosen from the base set $\mathbf{\Psi} = \{\psi_1, \psi_2, \ldots, \psi_{N_\psi}\}$. N_ψ is the total number of base vectors in the dictionary. Suppose $k-1$ bases $\mathbf{\Phi}_{k-1} = \{\phi_1, \phi_2, \ldots, \phi_{k-1}\}$ have been selected, the k-th base is chosen according to

$$\arg\max_{\psi_i} \left\{ \frac{1}{N_f} \sum_{j=1}^{N_f} \frac{|\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{f}_j) \rangle|^2}{\parallel \gamma_i^{(k)} \parallel^2} - \frac{\lambda}{N_b} \sum_{j=1}^{N_b} \frac{|\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{b}_j) \rangle|^2}{\parallel \gamma_i^{(k)} \parallel^2} \right\} , \tag{7}$$

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 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. We further analyze the above equation for simplification,

$$\langle \gamma_i^{(k)}, \varepsilon_{k-1}(\mathbf{x}) \rangle = \langle \psi_i - R_{\mathbf{\Phi}_{k-1}}(\psi_i), \mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x}) \rangle = \langle \psi_i, \mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x}) \rangle . \tag{8}$$

Since ψ_i is a box base, the inner product can be computed in $O(1)$ time with pre-computation of $\mathbf{x} - R_{\mathbf{\Phi}_{k-1}}(\mathbf{x})$ using integral image. Because

$$R_{\mathbf{\Phi}_k}(\mathbf{x}) = R_{\mathbf{\Phi}_{k-1}}(\mathbf{x}) + \frac{\varphi_k \langle \varphi_k, \mathbf{x} \rangle}{\| \varphi_k \|^2} \ , \tag{9}$$

where $\varphi_k = \phi_k - R_{\mathbf{\Phi}_{k-1}}(\phi_k)$ denotes the component of ϕ_k that is orthogonal to the subspace spanned by $\mathbf{\Phi}_{k-1}$, we therefore have

$$\| \gamma_i^{(k)} \|^2 = \| \psi_i - R_{\mathbf{\Phi}_{k-2}}(\psi_i) - \frac{\varphi_{k-1}\langle \varphi_{k-1}, \psi_i \rangle}{\| \varphi_{k-1} \|^2} \|^2 = \| \gamma_i^{(k-1)} \|^2 - \frac{|\langle \varphi_{k-1}, \psi_i \rangle|^2}{\| \varphi_{k-1} \|^2} \ . \tag{10}$$

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.

Note that the reconstruction for any $\mathbf{x}$ (i.e. $R_{\mathbf{\Phi}_k}(\mathbf{x})$) can be efficiently computed by pre-storing $\mathbf{\Phi}_k(\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1}$. The calculation of $\mathbf{\Phi}_k^T \mathbf{x}$ is the inner products between $\mathbf{x}$ and the base vectors, which can be accomplished in $O(k)$ time. Thus, computing the reconstruction simply costs $O(kWH)$ time, where W, H are respectively the width and height of the base 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.

3.3 Fast Search Using Coherence

As aforementioned, computation of the above algorithm is mainly spent on repetitive searching in the dictionary. Since, in the NBS framework, the size of base dictionary is proportional to $W^2 \cdot H^2$, the computational cost may increase dramatically as the template size increases. A natural way to accelerate it is to reduce the number of bases to be searched in each iteration. We propose to achieve this through basis filtering using coherence.

A μ-coherent dictionary $\mathbf{\Psi}$ has coherence μ for $0 \leq \mu \leq 1$, if $| \langle \psi_i, \psi_j \rangle | \leq \mu$ for all distinct $\psi_i, \psi_j \in \mathbf{\Psi}$. A 0-coherent base set is orthogonal. In general, bases with high coherence are likely to be redundant in representing the vector space. Coherence is used to reduce dictionary redundancy hence reducing the computation. Using coherence our algorithm can be accelerated by pruning all the base vectors with μ-coherent (μ is a given parameter) after each iteration of base selection.

An example image and the selected Haar-like features using discriminative NBS are shown in the left image of Figure 1. It is compared with the results selected using original NBS in the right image. Figure 2 shows the number of remaining bases for each coherence μ after selection of the largest Haar base. The template size is 50 × 50.

4 Tracking Using Discriminative NBS

With the discriminative NBS object representation, we locate object position in the current frame through sum of squared difference (SSD)-based matching.

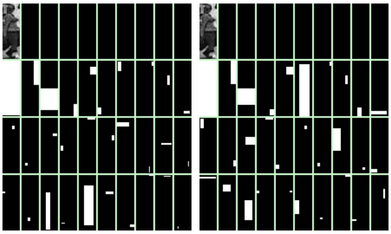

Fig. 1. Top 30 features selected using discriminative NBS (left) and the original NBS (right) for an image

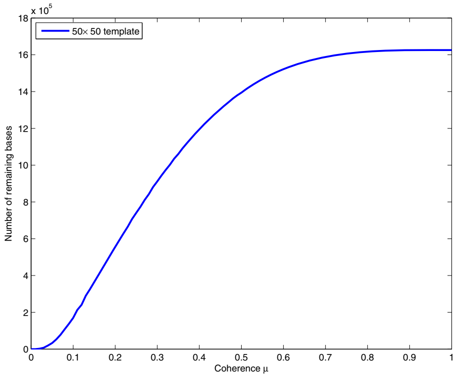

Fig. 2. The number of remaining bases after selecting the largest base

Using discriminative NBS, the object is first compared with the possible locations in an region around the object position detected in the previous frame. The one with the minimum SSD value is the target object location. In order to account for object appearance changes, the foreground and discriminative NBS are automatically updated every few frames.

4.1 Object Localization

The tracker starts from the predicted object position in the previous frame and searches the best matched template in an extended area around it. 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 surrounding the object position with a margin as the search window, in which the templates are sequentially compared with the referenced foreground $\mathbf{x} = R_{\mathbf{\Phi}_K^{(t)}}(\mathbf{f}_{\mathrm{ref}}^{(t)})$.

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

$$\mathrm{SSD}(\mathbf{x},\mathbf{y}) = \parallel \mathbf{x} - \mathbf{y} \parallel^2 = \parallel \mathbf{x} \parallel^2 + \parallel \mathbf{y} \parallel^2 - 2\langle \mathbf{x},\mathbf{y} \rangle \ , \tag{11}$$

where $\parallel \cdot \parallel$ represents the L^2-norm and $\langle \cdot,\cdot \rangle$ denotes the inner product. $\mathbf{x}$ is approximated by the discriminative NBS $\mathbf{\Phi}_K$ (i.e. $R_{\mathbf{\Phi}_K^{(t)}}(\mathbf{f}_{\mathrm{ref}}^{(t)}) = \sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)}$) , built using the approach in Section 3. Eq.(11) is then transformed to

$$\mathrm{SSD}(\sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)}, \mathbf{y}) = \parallel \sum_{i=1}^{K} c_i^{(t)} \phi_i^{(t)} \parallel^2 + \parallel \mathbf{y} \parallel^2 - 2 \sum_{i=1}^{K} c_i^{(t)} \langle \phi_i^{(t)}, \mathbf{y} \rangle \ . \tag{12}$$

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 with using integral image. The online computational complexity of Eq.(12) is only $O(K)$, where K is the number of selected bases.

4.2 Subspace Update

Due to appearance changes of the object, the discriminative NBS 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 discriminative NBS again as Eq.(4).

Template Update. The object template is also updated constantly to incorporate appearance changes, which serves as the new positive samples. As Eq.(4), NBS is constructed to better represent for a set of foreground templates. 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 [20]). 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,

$$\mathbf{f}_{\text{ref}}^{(t)} = \begin{cases} \mathbf{f}_0 & t = 0 \\ \gamma \mathbf{f}_{\text{ref}}^{(\lfloor (t-1)/N_u \rfloor N_u)} + (1-\gamma)\mathbf{f}_t & \text{otherwise} \ , \end{cases} \tag{13}$$

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 was 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 results are shown in the experiments section.

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 discriminative NBS. 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. Note this process can be done very efficiently because the SSD distance map can be computed very efficiently using Haar-like features and the integral image. Once the distance map is computed, locations which are local minima together with a non-minimal suppression are used to select negative samples.

5 Experiments

We first discuss in this section several key parameters used in constructing the discriminative NBS. Then we show qualitative tracking results of our approach on challenging sequences with significant background clutter and camera motion. To demonstrate the advantages of our approach, our tracking results are compared with three kinds of trackers: (1) a standard SSD tracker which uses direct patch matching, (2) an NBS tracker which applies the original NBS for object representation, and (3) a discriminative feature tracker proposed by Collins et al. in [14].

5.1 Parameter Selection

Several parameters are used in the discriminative NBS. Parameters with different settings will influence the accuracy of foreground reconstruction and tracking. We discuss here the justification of selecting them.

The formulation of the discriminative NBS 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 (mostly from PETS 2001 data set) with ground-truths to quantitatively evaluate how the parameter changes the tracking result. The tracking performance is evaluated as the mean distance error between the tracked location and the groundtruth object center. The discriminative NBS-based tracker with varying λ from 0 to 1 is applied to this sequence. The curve plotted in Fig. 3 shows the correlation of λ and centroid tracking error averaged over the whole sequence. Obviously, the centroid error is relatively more stable and smaller when λ is set to 0.25.

Another parameter for discriminative NBS is the number of bases K used. The selection of this parameter depends on image content. In general, the more features, the more accurate tracking, but it will also incur more computation. As a tradeoff, we set $K = 30$. Some other parameters we set empirically include: the number of foreground template N_f to 1 and background ones N_b to 3. These parameters are fixed for all the experiments in this paper.

We also conducted experiments to show the effectiveness of our template updating scheme. Here, we review several template updating methods mentioned above by comparing their tracking error of video sequence *browse*. These updating methods 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 methods are initialized with the same bounding box at the first frame and the error of object center is computed according to the ground truth. Figure 4 shows that the time-decaying approach is more robust and stable.

5.2 Tracking Results

Qualitative Results. We apply our tracker to several challenging sequences to show its effectiveness. We show some qualitative results on pedestrian videos

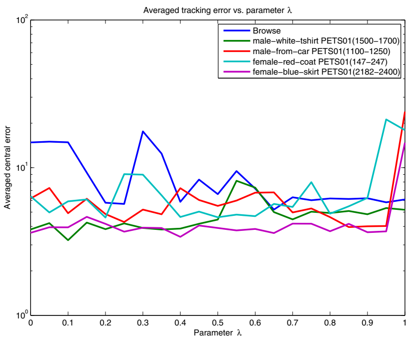

Fig. 3. The influence of λ on tracking errors. The y-axis is logarithmically scaled.

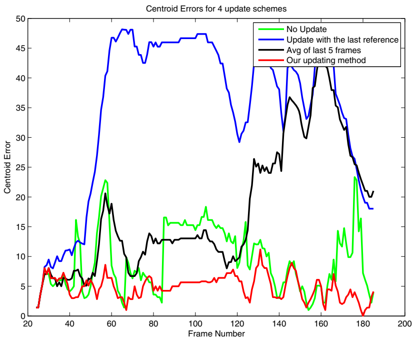

Fig. 4. Comparison for 4 template updating approaches

here to show that our tracker can handle background clutter, camera motion, and object appearance variations. In the following figures, red boxes indicate tracked object while blue boxes indicate the negative samples selected if there is a subspace update in that frame. The subspace is updated every 5 frames and if there is no update of subspace, no blue boxes (background samples) will be showed.

Sequence *Crosswalk* (Figure 5) 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 all the more difficult to accurately track the pedestrians. Despite this, our algorithm is able to track the pedestrians throughout the entire 140 frames of the sequence. Shai Avidan mentions in [15] that the Ensemble Tracker is able to track for the first 80 frames of the sequence but does not mention the performance for the remaining 60 frames.

Sequence *Browse* (Figure 6) is a video clip of frames 24-185 in *Browse1.avi* derived from CAVIAR people (ECCV-PETS 2004)Dataset [21]. This sequence is obtained 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.

Sequence *Courtyard* (Figure 7) is a video clip from 134th to 267th frame which records a person walking in the yard. The frame size is 720×480 and the object is manually bounded at frame 134 with a 41×101 red box. With moving background and variation of the object, our tracker can stably track the person.

Sequence *Crowd* (Figure 8) is a video clip (250th to 338th frames) selected from PETS 2007 Data set. In this sequence the background is very cluttered with many distracters. As can be observed the object can still be well tracked. The frame size is 720×576 and the object is initialized with a 26×136 bounding box.

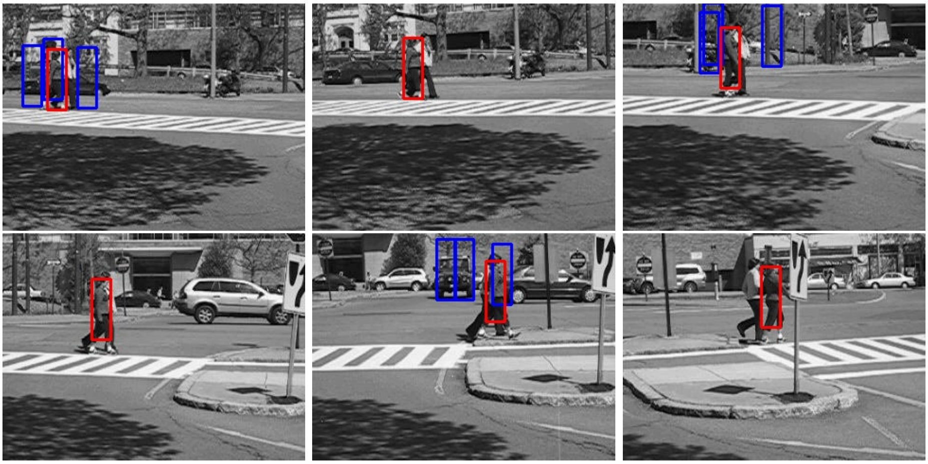

Fig. 5. *Crosswalk* sequence: The frames 0, 16, 50, 74, 105 and 139 are shown. The red boxes are the tracked objects and the blue boxes at $5k$ frame are the sampled backgrounds.

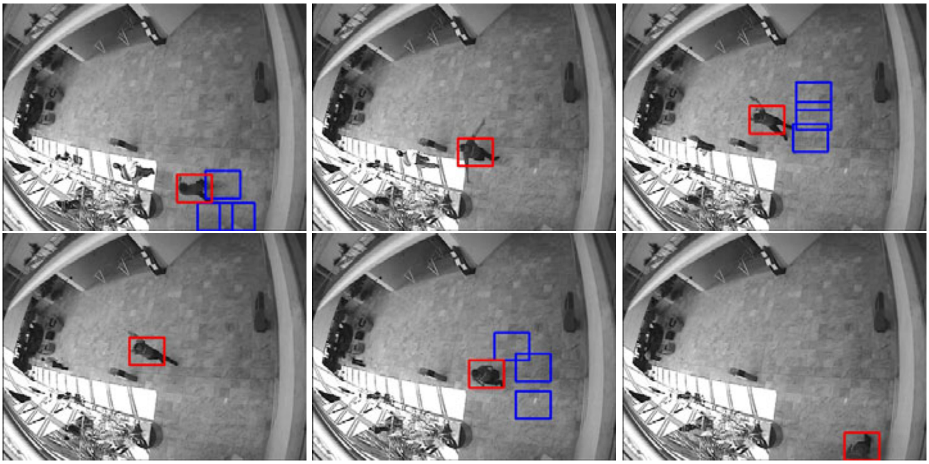

Fig. 6. *Browse* sequence: The frames 24, 45, 74, 115, 139, 185 are shown. The red boxes are the tracked objects and the blue boxes at $5k + 4$ frame are the sampled backgrounds.

Comparative result between our DNBS tracker and another discriminative tracker [14] is showed in Fig. 9. Sequence *Female* is a video clip in PETS 2007 data set. It starts from frame 826 to 870, each of which has 720×576 pixels. The object is initialized at the 826th frame of size 26×106. Collins' tracker drifts away at frame 841, while our method still keeps track all along.

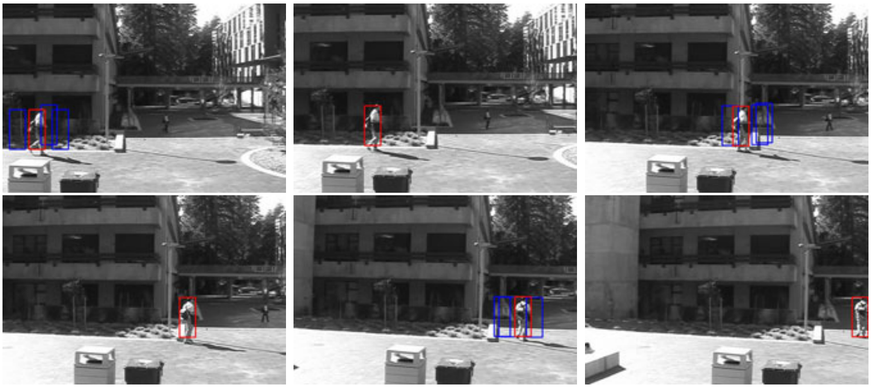

Fig. 7. *Courtyard* sequence: The frames 134, 153, 189, 205, 234, and 267 are shown. The red boxes are the tracked objects and the blue boxes at $5k + 4$ frame are the sampled backgrounds.

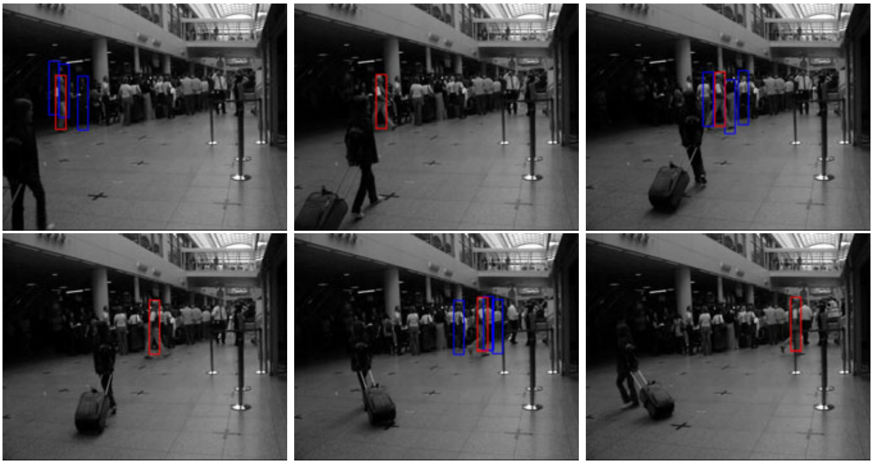

Fig. 8. *Crowd* sequence: The frames 250, 267, 295, 306, 325, and 338 are shown. The red boxes are the tracked objects and the blue boxes at $5k$ frame are the sampled backgrounds.

Quantitative Evaluation. In order to quantitatively evaluate the performance of our approach, we compare our results with the ground truth of the above two sequences (*Crosswalk* and *Browse*). The error is measured as the distance between the tracked object center location and the groundtruth object location in pixels. Figure 10 shows the results for three methods: (blue) SSD method, (green) NBS method, (red) Discriminative NBS method, and (light blue) a discriminative feature tracker proposed by Collins et al. [14]. The objects are initialized at the same position at the first frames and the reference templates are updated in

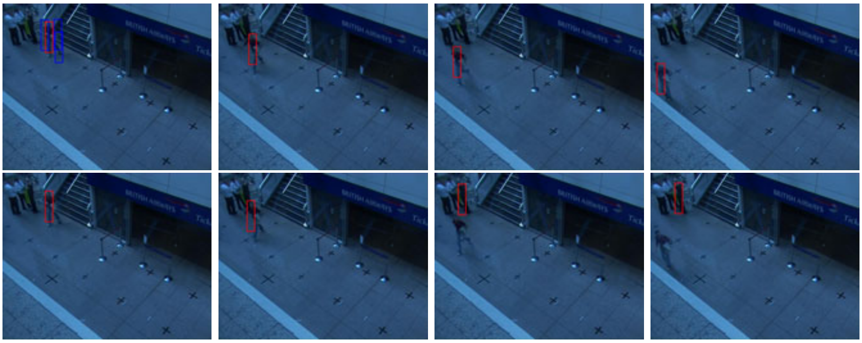

Fig. 9. *Female* sequence: The frames 826, 840, 854 and 870 are shown. The upper row shows results for DNBS tracker and the second row shows results for Collins' tracker.

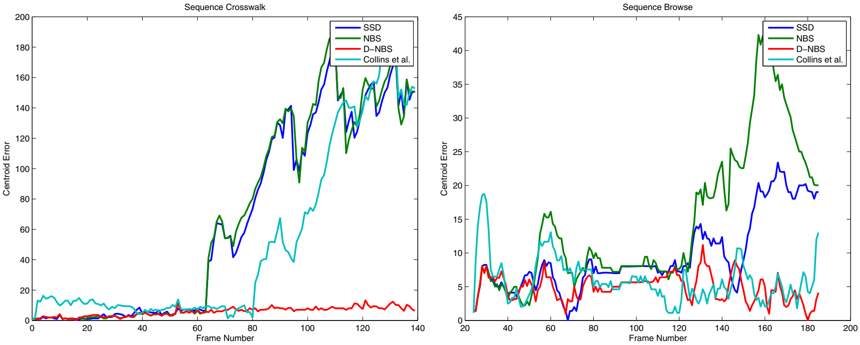

Fig. 10. Quantitative results for the *Crosswalk* and *Browse* sequence. The horizontal axis is the frame number and the vertical axis is the tracking error between the tracked object location and groundtruth.

the same way (with $N_u = 5$ and $\gamma = 0.5$) as mentioned in this paper. As can be observed, our approach is consistently better than these two methods.

6 Conclusions

We have proposed the discriminative NBS, a simple yet informative object representation that can be solved using a variant of OOMP. Such a representation incorporates the discriminate image information to distinguish the foreground and background, making it suitable to be used in object tracking. We use SSD matching built upon the discriminative NBS to efficiently locate object in video frames. Our experiments on challenging video sequences show that the discriminative NBS-based tracker can stably track the dynamic object. We intend to explore the application of discriminative NBS on other vision and multimedia tasks such as image copy detection in future.

Acknowledgments. This work was supported in part by the National Science Foundation of China under Grants 60703084, 60723003, 60721002, the National Fundamental Research Program of China (2010CB327903), and the Jiangsu Science Foundation (BK2009081).

References

1. Hager, G., Dewan, M., Stewart, C.: Multiple kernel tracking with ssd. In: Proc. CVPR, pp. I: 790–797 (2004)
2. Han, B., Davis, L.: On-line density-based appearance modeling for object tracking. In: Proc. ICCV, vol. II, pp. 1492–1499 (2005)
3. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. on PAMI 26, 810–815 (2004)
4. Black, M.J., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 329–342. Springer, Heidelberg (1996)
5. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. on PAMI 23, 681–685 (2001)
6. Jepson, A., Fleet, D., El-Maraghi, T.: Robust online appearance models for visual tracking. IEEE Trans. on PAMI 25, 1296–1311 (2003)
7. Tao, H., Sawhney, H., Kumar, R.: Object tracking with bayesian estimation of dynamic layer representations. IEEE Trans. on PAMI 24, 75–89 (2002)
8. Comaniciu, D.: Kernel-based object tracking. IEEE Trans. on PAMI 25, 564–577 (2003)
9. Fan, Z., Wu, Y.: Multiple collaborative kernel tracking. In: Proc. CVPR, vol. II, pp. 502–509 (2005)
10. Birchfield, S., Sriram, R.: Spatiograms versus histograms for region-based tracking. In: Proc. CVPR, vol. II, pp. 1158–1163 (2005)
11. Shi, J., Tomasi, C.: Good features to track. In: Proc. CVPR, pp. 593–600 (1994)
12. Tang, F., Tao, H.: Object tracking with dynamic feature graphs. In: Workshop on VS-PETS, pp. 25–32 (2005)
13. Chen, Y., Rui, Y., Huang, T.: Jpdaf based hmm for real-time contour tracking. In: Proc. CVPR, vol. I, pp. 543–550 (2001)
14. Collins, R., Liu, Y., Leordeanu, M.: On-line selection of discriminative tracking features. IEEE Trans. on PAMI 27, 1631–1643 (2005)
15. Avidan, S.: Ensemble tracking. IEEE Trans. on PAMI 29, 261–271 (2007)
16. Avidan, S.: Support vector tracking. IEEE Trans. on PAMI, 184–191 (2001)
17. Tang, F., Brennan, S., Zhao, Q., Tao, H.: Co-tracking using semi-supervised support vector machines. In: Proc. ICCV, pp. 1–8 (2007)
18. Tang, F., Crabb, R., Tao, H.: Representing images using nonorthogonal haar-like bases. IEEE Trans. on PAMI 29, 2120–2134 (2007)
19. Rebollo-Neira, L., Lowe, D.: Optimized orthogonal matching pursuit approach. IEEE Signal Processing Letters 9, 137–140 (2002)
20. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. on PAMI 25, 1296–1311 (2003)
21. EC Funded CAVIAR project, `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`