UC Santa Cruz
UC Santa Cruz Previously Published Works

Title

Representing images using nonorthogonal Haar-like bases

Permalink

https://escholarship.org/uc/item/7x2585wk

Journal

IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(12)

ISSN

0162-8828

Authors

Tang, Feng
Crabb, Ryan
Tao, Hai

Publication Date

2007-12-01

Peer reviewed

Representing Images Using Nonorthogonal Haar-Like Bases

Feng Tang, *Student Member*, *IEEE*, Ryan Crabb, *Student Member*, *IEEE*, and
Hai Tao, *Senior Member*, *IEEE*

Abstract—The efficient and compact representation of images is a fundamental problem in computer vision. In this paper, we propose methods that use Haar-like binary box functions to represent a single image or a set of images. A desirable property of these box functions is that their inner product operation with an image can be computed very efficiently. We propose two closely related novel subspace methods to model images: The nonorthogonal binary subspace (NBS) method and the binary principal component analysis (B-PCA) algorithm. NBS is spanned directly by binary box functions and can be used for image representation, fast template matching, and many other vision applications. B-PCA is a structure subspace that inherits the merits of both NBS (fast computation) and PCA (modeling data structure information). B-PCA base vectors are obtained by a novel PCA-guided NBS method. We also show that B-PCA base vectors are nearly orthogonal to each other. As a result, in the nonorthogonal vector decomposition process, the computationally intensive pseudoinverse projection operator can be approximated by the direct dot product without causing significant distance distortion. Experiments on real image data sets show a promising performance in image matching, reconstruction, and recognition tasks with significant speed improvement.

Index Terms—Nonorthogonal subspace, image representations, principal component analysis, image reconstruction.

✦

1 INTRODUCTION

LINEAR data representations are widely used in signal processing and data analysis. Representations such as the Fourier transform [1] and the wavelet transform [2] are designed for general data sets. Data specific subspace representations, on the other hand, can capture the statistics of particular data sets and are learned directly from the training data by optimizing some error measure. Several such methods include principal component analysis (PCA) [3], independent component analysis (ICA) [4], sparse coding [5], nonnegative matrix factorization (NMF) [6], and nonorthogonal binary subspace [7]. High-dimensional visual data often belongs to a low-dimensional manifold with the dimension roughly corresponding to the number of physical factors that affect the objects appearance. These factors may include the viewing angles, the light source directions, and the rigid or nonrigid motion of the object. The linear subspace method is extensively used in computer vision algorithms in order to obtain compact representations. Some of the many examples using this technique include the Eigenface method [8], the EigenTracking algorithm [9], illumination subspaces [10], [11], and the nonnegative matrix factorization method [6].

In this paper, we will present two novel efficient and compact subspace representations for a single image or a set of images. The two key ingredients in the new approach are the Haar-like box function approximation of the visual data and the associated theory of nonorthogonal subspace.

1.1 Haar-Like Box Functions and Their Properties

In recent years, Haar-like box functions have become a popular choice as image features due to the seminal work of Viola and Jones [12], [13]. Examples of such box functions are shown in Fig. 1. Formally, the binary function is defined as $f(u,v) \in \{0,1\}$, $1 \le u \le w$, $1 \le v \le h$, where w and h are the dimensions of the binary function. The single Haar-like box function is defined as

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

where w' and h' are the size of the white box in Fig. 1, and u_0 and v_0 are the upper left corner of the white box. For some symmetric objects like human faces, we can similarly define the vertically symmetric two-box binary function as

$$f(u,v) = \begin{cases} 1 & \begin{matrix} u_0 \le u \le u_0 + w' - 1 \\ v_0 \le v \le v_0 + h' - 1 \end{matrix} \\ 1 & \begin{matrix} w - u_0 - w' + 1 \le u \le w - u_0 \\ h - v_0 - h' + 1 \le v \le h - v_0 \end{matrix} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

One obvious property of binary box bases is that they are not necessarily orthogonal to each other. If the white boxes in two box functions have overlapping areas, their dot product is not zero. The main advantage of using these base functions is that the inner product of a data vector with each of them can be performed by several integer additions, instead of N floating point multiplications, where N is the dimension of the base

- *The authors are with the Department of Computer Engineering, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064. E-mail: {tang, rcrabb, tao}@soe.ucsc.edu.*
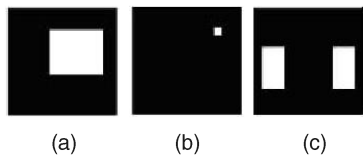
Fig. 1. Three typical one and two-box functions. (a) and (b) are one-box functions and (c) is a symmetric two-box function.

vectors. This is achieved by computing the integral image $f_i(i, j)$ of the original image $f(i, j)$, which is defined as

$$f_i(i, j) = \sum_{m=1}^{i} \sum_{n=1}^{j} f(m, n).$$

The dot product of an image with a one-box base function is the summation of a rectangular area of the image, which can be computed efficiently as

$$\sum_{i=top}^{bottom} \sum_{j=left}^{right} f(i, j) = f_i(bottom, right) - f_i(bottom, left-1)$$
$$- f_i(top-1, right) + f_i(top-1, left-1),$$

where $f(\cdot, \cdot)$ is the image function, and $f_i(\cdot, \cdot)$ is the integral image of f. top, $bottom$, $left$, and $right$ are the coordinates that define the rectangular area. This technique has been used in many applications [12], [13], [14], [7], [15], [16], [17]. These binary box functions are generally nonorthogonal and span an *nonorthogonal binary subspace* (NBS). In the applications that will be investigated in this paper, we will show that a significant speedup can be achieved by using NBS for image representation.

1.2 Related Work in Subspace Representation

Some subspace data representation methods have their roots in neural computing and have been widely employed in signal analysis. In sparse coding [5], the objective is to find a decomposition in which the base vectors are representative and sparse. Sparse coding yields base vectors closely resembling simple-cell receptive fields in the mammalian primary visual cortex. The closely related model of independent component anaylsis (ICA) [4] was shown to give similar results [18]. In standard sparse coding, the data is described as a combination of base vectors involving both additive and subtractive interactions. Lee and Seung proposed the non-negative matrix factorization (NMF) in [6]. By minimizing the reconstruction error while imposing the constraints that the elements of base vectors and reconstruction coefficients are nonnegative, the NMF produces a nonnegative representation of the data. Such a representation encodes the data using a few "active" components, which makes the subspace easy to interpret. However, because the sparseness given by NMF is somewhat of a side-effect rather than a goal, it is hard to control the degree of sparseness of the bases. Hoyer [19] proposed to combine these two methods into the so-called nonnegative sparse coding by posing both nonnegative and sparse constraints in the objective function. In sparse NMF, the uniqueness of the resulting basis and coefficients holds if the sparseness values are known. In [20], the authors use sparse component analysis for blind source separation by constraining the signals to be as sparse as possible (as many zeros as possible). In [21], Heiler and Schnorr proposed a method to

efficiently compute sparsity-controlled invariant image codes by a well-defined sequence of convex conic programs. In [22], instead of treating the image as a 1D vector in NMF, Hazan et al. proposed a method to factorize the 3D tensor of the image array with a nonnegative constraint to obtain a unique sparse set of factors that correspond to the parts of the object. In [23] and [24], the authors propose a basis pursuit (BP) approach to find an overcomplete basis set (the number of base vectors is larger than the data dimension) for general signals. In [25], Elad and Aharon proposed a method to obtain both sparse and overcomplete image representations using K-SVD for image denoising. The projection pursuit method [26] aims to find the optimal projection to separate the data clusters. However, the subspaces obtained by these methods are usually spanned by floating point base vectors. This makes the dot product with the data computationally expensive. In our work, we aim at finding a compact and efficient image representation using simple box functions. Generally, the number of base vectors is much smaller than the data dimension. Another work related to ours is [27], which approximates a matrix by a weighted sum of outer products of vectors whose elements are ±1 or 0, with the application of data compression. Our work is to represent data as a linear combination of a set of simple Haar-like base vectors.

1.3 Nonorthogonal Subspace and Matching Pursuit

A subspace can be represented using either orthogonal or nonorthogonal base vectors. Orthogonal subspaces such as DCT [28], Walsh-Hadamard transform [29], wavelet transform [30], [31], and PCA [3] are most often used in vision algorithms. Any pair of base vectors in an orthogonal subspace is orthogonal. For a nonorthogonal subspace, there exist one or more pairs of base vectors that are not orthogonal to each other.

Base vectors for most orthogonal subspaces can be obtained in a principled way with mathematical decompositions or factorizations. Unfortunately, the problem of searching for the best subspace representation in a set of predefined nonorthogonal base vector dictionary is known to be NP-hard [32]. Two of the popular greedy solutions to this problem include the MP [33] approach and the optimized orthogonal matching pursuit (OOMP) method [34], [35].

Let $D = [\phi_1, \phi_2, \ldots, \phi_N]$ denote an arbitrary redundant nonorthogonal base dictionary in a Hilbert space. The set of base vectors selected to represent the data vector or a set of data vectors $\mathbf{x}$ up to iteration k is denoted as $\Phi_k = [\phi_1, \phi_2, \ldots, \phi_k]$. The representation of $\mathbf{x}$ using Φ_k is denoted as $R_{\Phi_k}(\mathbf{x}) = \sum_{i=1}^{k} c_i \phi_i$.

- The Matching pursuit (MP) method [33] sequentially selects the base vector ϕ_k from D such that $|c_i| = |\langle \mathbf{x} - R_{\Phi_{k-1}}(\mathbf{x}), \phi_k \rangle|$ is maximized. However, in each iteration, the residual $\mathbf{x} - R_{\Phi_k}(\mathbf{x})$ may not be orthogonal to the subspace spanned by Φ_k unless the ϕ_is are orthogonal to each other. As a consequence, the $R_{\Phi_k}(\mathbf{x})$ is not the best approximation of $\mathbf{x}$ using k base vectors.

- Optimized orthogonal matching pursuit (OOMP) [34], [35] uses a similar criterion as MP to select the base vectors, but it maintains full backward orthogonality of the residual in each iteration. Thereby, the reconstruction of $\mathbf{x}$ using OOMP-selected base vectors Φ_k is orthogonal to the residual, hence, the

name *orthogonal* MP. In addition, OOMP selects the base vector ϕ_i that minimizes the corresponding residual error. Details of OOMP will be discussed in Section 2.2.

In the rest of the paper, $\mathbf{C} = [c_1, c_2, \ldots, c_k]^T = P_{\mathbf{\Phi}_k}(\mathbf{x}) = (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T \mathbf{x}$ are the projection coefficients. $\mathbf{Q} = (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T$ is called the projection matrix for the subspace. As a special case, when ϕ_is form an orthonormal system, the projection process can be simplified as $P_{\mathbf{\Phi}_k}(\mathbf{x}) = \mathbf{\Phi}_k^T \mathbf{x}$. When they are nonorthogonal, the projection process takes the general form $P_{\mathbf{\Phi}_k}(\mathbf{x}) = (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T \mathbf{x}$.

1.4 Contributions

In this paper, we propose NBS and the binary principal component analysis (B-PCA) to represent an image or a set of images. NBS is directly spanned by box functions and applied to accelerate several commonly used computer vision algorithms. NBS is then extended to B-PCA to capture the structure information of the images. B-PCA spans a subspace that approximates a PCA subspace. Each B-PCA base vector is a linear combination of a small number of Haar-like box functions. As the result, the dot product with each B-PCA base vector can be computed very efficiently.

The main contributions of this paper are:

- A nonorthogonal binary subspace that can represent images efficiently with box functions.
- A novel nonorthogonal image subspace representation called Binary Principal Component Analysis (B-PCA) with each base vector as a linear combination of Haar-like box functions.
- A fast PCA-guided optimized orthogonal matching pursuit (OOMP) algorithm for finding the B-PCA base vectors, including a theoretical analysis of the properties of B-PCA bases and the associated subspace projection.
- Applications of the proposed representations in image matching, reconstruction, and object recognition tasks.

The rest of the paper is organized as follows: In Section 2, we investigate the problem of representing an image or a set of images using Haar-like box functions and propose the nonorthogonal binary subspace (NBS) method. The applications of NBS in accelerating several widely used vision algorithms are addressed in Section 3. In Section 4, we extend the NBS to binary component anaylsis (B-PCA). The implementation and experiments of NBS are given in Section 5. In Section 6, we demonstrate the effectiveness of the B-PCA method in image reconstruction and recognition. Conclusion and future work are highlighted in Section 7.

2 NONORTHOGONAL BINARY SUBSPACE

2.1 The Problem Statement

We will denote the dictionary of all one-box and symmetric two-box functions as $D = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$, where each $\mathbf{b}_i$ is a column vector formed by reshaping the rectangular box function formally defined in Section 1.1. For an image of $W \times H$ pixels, there are $H(H+1)W(W+1)/4$ and $H(H+1)W(W-1)/16$ one-box and symmetric two-box functions, respectively. D itself is a highly redundant nonorthogonal

base vector set. For a given image $\mathbf{x} \in R^N$ and the dictionary $D = \{\mathbf{b}_i\}_{i \in I}$ that consists of a set of binary box base vectors $\mathbf{b}_i$ and an index set as $I = \{1, \ldots, N\}$, $\mathbf{x}$ can be approximated using base vectors as $\hat{\mathbf{x}} = \sum_{i \in \Lambda} c_i \mathbf{b}_i$, where c_i is the coefficient of base vector $\mathbf{b}_i$, Λ is a set of indices of the vectors in D, and $\Lambda \subseteq I$. Typically, $|\Lambda| \ll N$, so the representation is compact. The subspace spanned by $D_\Lambda = \{\mathbf{b}_i\}_{i \in \Lambda}$ can be represented by a matrix formed by the base vectors $\mathbf{B}_\Lambda = [\mathbf{b}_{l_1}, \ldots, \mathbf{b}_{l_{|\Lambda|}}]$. The subspace reconstruction operator is denoted as $R_{\mathbf{B}_\Lambda}(\cdot)$; therefore, $\hat{\mathbf{x}} = R_{\mathbf{B}_\Lambda}(\mathbf{x})$. For a nonorthogonal $\mathbf{B}_\Lambda$, the projection coefficients are computed as

$$P_{\mathbf{B}_\Lambda}(\mathbf{x}) = [c_1, \ldots, c_{|\Lambda|}]^T = (\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1} \mathbf{B}_\Lambda^T \mathbf{x}.$$

The reconstruction becomes

$$R_{\mathbf{B}_\Lambda}(\mathbf{x}) = \mathbf{B}_\Lambda (\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1} \mathbf{B}_\Lambda^T \mathbf{x}.$$

The goal of an optimal approximation algorithm is to find the set of base vectors indexed by Λ so that the following objective function is minimized:

$$\arg \min_\Lambda \|\mathbf{x} - \mathbf{B}_\Lambda (\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1} \mathbf{B}_\Lambda^T \mathbf{x}\| + \beta \sum_{i \in \Lambda} cost(\mathbf{b}_i), \quad (3)$$

where $\mathbf{B}_\Lambda = [\mathbf{b}_{l1}, \ldots, \mathbf{b}_{l|\Lambda|}]$ is the set of base vectors for NBS. $\sum_{i \in \Lambda} cost(\mathbf{b}_i)$ is the computational cost for the set of base vectors Λ. To make the problem simpler, we assume that the computation of all the boxes are the same, $cost(\mathbf{b}_i) = 1$, in other words, the computation cost term grows linearly with the number of base vectors. This is reasonable in our case because the computational cost for the bases is either three (one-box function) or seven operations (symmetric two-box function). Obviously, selecting more base vectors will reduce the residual, but on the other hand, more base vectors increase the representation complexity. The optimal solution is a trade-off of the two terms and is determined by β, which is the weight to balance these two terms.

When a set of images $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_S]$ are approximated using nonorthogonal binary bases, the formulation is identical except that the notations represent matrices instead of vectors

$$\arg \min_\Lambda \|\mathbf{X} - R_{\mathbf{B}_\Lambda}(\mathbf{X})\|_F^2 + \beta \sum_{i \in \Lambda} cost(\mathbf{b}_i), \quad (4)$$

where $\|\cdot\|_F^2$ is the Frobenius norm. Solving (1) and (2) is the fundamental problem in the newly emerging research area of highly nonlinear approximation. Solving the general form of this problem has been proved to be NP-hard [32]. It has been shown that even the verification of the optimal solution is difficult. However, attempts have been made recently to derive the performance bound under the assumption of low coherence [36]. Suboptimal greedy algorithms that converge to a local optimal solution exist and have been used extensively. A family of these algorithms is based on the celebrated matching pursuit method [33].

2.2 The Solution: OOMP

Finding the global optimal of (3) and (4) is NP-hard, and the solution may not be unique because of the overcompleteness of the base set. We employ a greedy algorithm (OOMP) to find a suboptimal solution that guarantees that given a parameter β, there is a unique solution to (3) and (4).
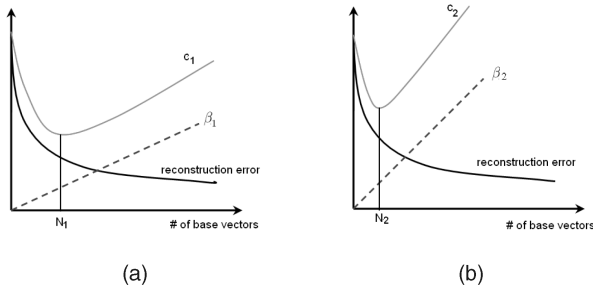
Fig. 2. Relation of the residual term and computation cost term in the objective function. The black solid curves are the reconstruction error versus the number of base vectors. The dotted curves are the computational cost term with different β, the upper curves are the sum of reconstruction residual and computational cost.

In practice, the choice of β is not intuitive. However, we find that if the computational cost is a convex function, there is an equivalent and more intuitive parameter: N, the number of base vectors to represent the signal; Fig. 2 shows the relation between the reconstruction residual and computation cost. The black solid curve is the reconstruction error decreasing with the number of base vectors; we can easily see it is a convex decreasing function (because the OOMP always selects the base vector that reduces the residual most). The dotted line is the computational cost term that grows linearly with the number of base vectors. The slope of this line is determined by β. The upper curve is the objective function, which is the combination of the two curves below. As can be observed, the objective function first decreases with the number of base vectors because with more base vectors, reconstruction error decreases, but at some point, the objective function tends to increase because more base vectors will increase the computation term. The optimal solution is the minimal point of the red curve. Given a set of images to approximate, the OOMP feature selection will decrease the reconstruction error when the number of base vectors increase; this curve is not affected by β. β only determines the slope of the computation cost line; different β correspond to different slopes (in Figs. 2a and 2b), and each β corresponds to one unique N, the number of base vectors used to approximate the input signal. Therefore, we can pick the number of base vectors N instead of β to control the greedy optimization. Another parameter that is adaptive to the data is the error ratio, which sets a threshold on the ratio of the residual and the original image. This method is discussed in detail in Section 4.3. One thing to note is that in practice, the binary box base vectors are not norm 1, but their norms can be precomputed and stored for efficient computation.

Optimized orthogonal matching pursuit (OOMP) [35] is a technique to compute adaptive signal representation by iterative selection of base vectors from a dictionary. Given the number of base vectors to be selected, the OOMP algorithm iteratively finds base vectors $\mathbf{B}_\Lambda = [\mathbf{b}_{l_1}, \ldots, \mathbf{b}_{l_{|\Lambda|}}]$ according to the following procedure. Suppose that after iteration $k-1$, the already selected $k-1$ base vectors are defined by the index set $\Lambda_{k-1} = (l_i)_{i=1}^{k-1}$. To find the base vector in iteration k, the OOMP prescribes to select the index $l_k = i$ that minimizes the new residual, which is equivalent to maximizing
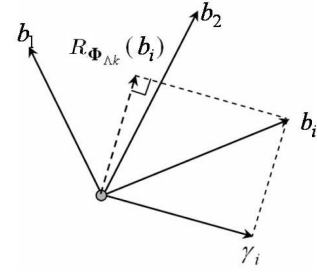

Fig. 3. Projection of a base vector $\mathbf{b}_i$ into the subspace formed by selected base vectors B_{Λ_k}. The $\gamma_i = \mathbf{b}_i - R_{\mathbf{B}_{\Lambda_k}}(\mathbf{b}_i)$ is the component of $\mathbf{b}_i$ that is orthogonal to the subspace spanned by $\mathbf{B}_{\Lambda_k}$.

$$\frac{|\langle \gamma_i, \varepsilon_{k-1} \rangle|}{\|\gamma_i\|}, \ \|\gamma_i\| \neq 0, \ i \in \bar{\Lambda}_k, \quad (5)$$

where $\varepsilon_{k-1} = \mathbf{x} - R_{\mathbf{B}_{\Lambda_{k-1}}}(\mathbf{x})$ is the reconstruction residual using $\mathbf{B}_{\Lambda_{k-1}}$, and $\gamma_i = \mathbf{b}_i - R_{\mathbf{B}_{\Lambda_{k-1}}}(\mathbf{b}_i)$ is the component of $\mathbf{b}_i$ that is orthogonal to the subspace spanned by $\mathbf{B}_\Lambda$. $R_{\mathbf{B}_\Lambda}(\mathbf{x}) = \mathbf{B}_\Lambda(\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1} \mathbf{B}_\Lambda^T \mathbf{x}$ is the reconstruction of the signal $\mathbf{x}$ using the nonorthogonal bases indexed by Λ_{k-1}. $\bar{\Lambda}_{k-1}$ is the subset of indices that are not selected in the previous $k-1$ iterations, that is, $\bar{\Lambda}_{k-1} = I - \Lambda_{k-1}$. The geometric interpretation of γ_i is illustrated in Fig. 3. An effective implementation of this optimization can be achieved by the forward adaptive biorthogonalization technique [37]. In essence, OOMP is a greedy algorithm that finds a suboptimal decomposition of a data vector using a number of base vectors from a nonorthogonal dictionary D.

3 APPLICATIONS OF NONORTHOGONAL BINARY SUBSPACE

The power of NBS for representing visual data will be shown in Section 5.1. In this section, we will show how NBS can be applied to accelerate a wide variety of computer vision algorithms such as normalized cross correlation (NCC), and sum of squared distance (SSD)-based template matching and recognition. Considerable work has been done on fast template matching, but to our knowledge, few of them use the binary box functions as image representation. In [38], a fast pruning method is proposed to reject false matches for NCC. In [39], Kiltahau et al. presented a method to accelerate the SSD using fast Fourier transform (FFT). In [40], the authors present a method to accelerate SSD using Walsh-Hadamard projection kernels and later extend to the gray-code kernels [41] for efficient filtering.

3.1 Fast Normalized Cross Correlation

NCC is a popular method for matching 2D patterns in images. When a $(2h+1) \times (2w+1)$ template $\mathbf{y}$ is correlated with an image $\mathbf{x}$ at the image location (u, v), the NCC is computed as

$$\mathrm{ncc}(u, v) = \frac{\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{X}(i,j) \mathbf{Y}(i,j)}{\sqrt{\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{X}(i,j)^2} \sqrt{\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{Y}(i,j)^2}} \quad (6)$$

with

$$\mathbf{X}(i,j) = \mathbf{x}(u+i, v+j) - \bar{x},$$
$$\mathbf{Y}(i,j) = \mathbf{y}(h+i, w+j) - \bar{y},$$
$$\bar{x} = \frac{1}{(2h+1)(2w+1)} \sum_{i=u-h}^{u+h} \sum_{j=v-w}^{v+w} \mathbf{x}(i,j), \qquad (7)$$
$$\bar{y} = \frac{1}{(2h+1)(2w+1)} \sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{y}(h+i, w+j).$$

In [42], a fast algorithm was developed to compute the denominator term

$$\sum_{i=-h}^{h} \sum_{j=-w}^{w} (\mathbf{x}(u+i, v+j) - \bar{x})^2. \qquad (8)$$

This is achieved by observing that this term can be simplified as $\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{x}(u+i, v+j)^2 - (2h+1)(2w+1)\bar{x}^2$. The first term and the mean $\bar{x}$ can be computed efficiently using integral images of the original image and the squared image. We can represent the template image using the proposed NBS method to precompute $\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{Y}(i,j)^2$ and speed up the numerator computation

$$\mathbf{y} \approx \sum_{i \in \Lambda} c_i \mathbf{b}_i. \qquad (9)$$

The numerator can be expanded as

$$\sum_{i=-h}^{h} \sum_{j=-w}^{w} (\mathbf{x}(u+i, v+j) - \bar{x})(\mathbf{y}(h+i, w+j) - \bar{y})$$
$$= \sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{x}(u+i, v+j)\mathbf{y}(h+i, w+j) \qquad (10)$$
$$- \sum_{i=-h}^{h} \sum_{j=-w}^{w} \bar{x}\mathbf{y}(h+i, w+j).$$

The first term can be computed using (9) as $\sum_{i \in \Lambda} c_i \langle \mathbf{x}_{u,v}, \mathbf{b}_i \rangle$, where $\mathbf{x}_{u,v}$ is the image patch centered at (u, v). The dot product with each base vector can be computed using three or seven integer additions with integral image, depending on one or two boxes in the base vector $\mathbf{b}_i$. Thus, the first term needs $|\Lambda|$ multiplications, between $3|\Lambda|$ and $7|\Lambda|$ additions. The second term needs a single multiplication and a single division because $\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{y}(h+i, w+j)$ is constant and can be precomputed; the computation of the $\bar{x}$ needs a single division. Therefore, the total computation for (8) is $|\Lambda| + 1$ floating point multiplications, one floating point division and between $3|\Lambda|$ and $7|\Lambda|$ integer additions. Since $|\Lambda| \ll N = (2h+1)(2w+1)$, the computation saving is significant compared to the cost without using NBS, which needs N multiplications and $2N$ additions. We will show in Section 5 how the number of base vectors used in approximating the template affects the matching accuracy.

It should be noted here that the cost of building an integral image is not significant because it is done only once for the whole image and only needs $2W \cdot H$ additions, where W and H are the image width and height.

3.2 Fast Template Matching Using SSD

When the SSD is used as the error measure for template matching, it is computed at each image location as

$$ssd(u, v) = \sum_{i=-h}^{h} \sum_{j=-w}^{w} (\mathbf{x}(u+i, v+j) - \mathbf{y}(h+i, w+j))^2$$
$$= \sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{x}(u+i, v+j)^2 + \sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{y}(h+i, w+j)^2$$
$$- 2 \sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{x}(u+i, v+j)\mathbf{y}(h+i, w+j). \qquad (11)$$

The first term can be computed efficiently using the squared integral image, and the second term is constant, which is precomputed. The last term can be computed using the NBS method with $|\Lambda|$ floating point multiplications and between $3|\Lambda|$ and $7|\Lambda|$ integer additions, as discussed in the previous section.

3.3 Subspace Expansion and Residual Computation

As mentioned in Section 2.1, the reconstruction of a data vector $\mathbf{x}$ in the subspace $\mathbf{B}_\Lambda$ is $R_{\mathbf{B}_\Lambda}(\mathbf{x}) = \mathbf{B}_\Lambda(\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1}\mathbf{B}_\Lambda^T \mathbf{x}$. To compute $\mathbf{B}_\Lambda^T \mathbf{x}$, between $3|\Lambda|$ and $7|\Lambda|$ additions are needed. $(\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1}$ can be precomputed, and the product between $(\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1}$ and $\mathbf{B}_\Lambda^T \mathbf{x}$ requires $|\Lambda|^2$ multiplications. The product of $\mathbf{B}_\Lambda$ and $(\mathbf{B}_\Lambda^T \mathbf{B}_\Lambda)^{-1}\mathbf{B}_\Lambda^T \mathbf{x}$ can be efficiently computed using at most $N|\Lambda|$ floating point additions, where N is the dimension of the data vector $\mathbf{x}$. Therefore, a total of $|\Lambda|^2$ floating point multiplications, at most $N|\Lambda|$ floating point additions, and $7|\Lambda|$ integer additions are needed. Without using NBS, the total computation is $2N|\Lambda|$ multiplications and $2N|\Lambda| - N - |\Lambda|$ additions. Similarly, to find the subspace reconstruction residual, additional N additions are needed. Note that we use the original box functions here rather than the normalized base vector; the reconstruction is the same as when the base vector norm is 1. This reduces the floating point multiplication operations. The proof can be found in the supplementary material.

3.4 Recognition in the Nonorthogonal Space

It is possible to perform object recognition directly using the projection $\mathbf{B}_\Lambda^T \mathbf{x}$ because the difference between nonorthogonal feature values and the feature values in the same subspace but spanned by orthogonal coordinate axes is a linear transformation. Two separable clusters remain separable under a linear transformation. Therefore, instead of using $N|\Lambda|$ multiplications and $(N-1)|\Lambda|$ additions to compute the projection, only $|\Lambda|$ multiplications (for the normalization coefficients) and between $3|\Lambda|$ and $7|\Lambda|$ additions are needed. The computation saving is significant. An application of using NBS for face recognition is shown in Section 5.

4 BINARY-PCA (B-PCA)

Our experiments show that the greedy search solution for NBS can represent images very well in the sense of least squares reconstruction error. However, for object recognition tasks, NBS has the following drawbacks:

- The NBS base vectors do not have an explicit physical interpretation. Although some of the binary box functions may have corresponding parts in the

images, generally, each base vector lacks a meaningful interpretation due to its oversimplified structure.

- The NBS base vectors are usually highly nonorthogonal to each other, which makes the L-2 norm in NBS significantly deviate from the Euclidean distance. Therefore, it leads to numerical problems in recognition. This phenomenon can be observed in our experiments in Section 5.3.
- Compared to PCA, NBS uses more base vectors to span approximately the same subspace. This is not a problem in reconstruction, but recognition favors fewer, more informative base vectors so that each image can be represented by a small number of coefficients. Because each test image will be projected onto each of these base vectors, fewer bases need less computation.

These problems with NBS have motivated us to find a better subspace representation that is as computationally efficient as NBS but with enhanced representation power for the recognition task.

Principal component analysis is a widely used subspace representation that captures the structure information of the data, and the PCA base vectors can often be easily interpreted. For example, the frontal face subspace base vectors resemble different facial appearance variants. These properties are just complement with NBS. The main computation in PCA is the dot product of a data vector with all the PCA base vectors. This can be computationally expensive, especially when the original data dimension is high because it involves many floating point multiplications. These inspired us to investigate the possibility of finding a subspace representation that is similar to PCA in terms of capturing the essential data structure while avoiding the costly floating point dot product operations.

4.1 Background—Principal Component Analysis

A PCA subspace is spanned by orthogonal eigenvectors corresponding to the largest eigenvalues of the data covariance matrix. The PCA technique has been used in Eigenface [8], subspace-based tracking [9], [43], [44], [45], adaptive face shape and appearance model (active shape model (ASM) and active appearance model (AAM)) [46], [47], motion estimation and segmentation [48], [49], 3D appearance space [50], and many other computer vision algorithms.

The intuition behind the PCA method is to find a set of base vectors $\mathbf{\Theta}_n = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n]$ that can explain the maximum amount of variance of the data. PCA can be defined with an incremental formulation. Suppose that we have determined the first $k-1$ principal components, the kth principal component $\mathbf{e}_k$ is determined as the principal vector of the residual

$$\arg\max_{\mathbf{e}} E\{[\mathbf{e}^T(\mathbf{X} - R_{\mathbf{\Theta}_{k-1}}(\mathbf{X}))]^2\}, \qquad (12)$$

$$subject\ to : \|\mathbf{e}\| = 1,$$

where $\mathbf{x} \in R^N$ denotes the data vector, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ denotes a set of data vectors with zero mean. The $\mathbf{e}_i$s are orthogonal to each other. $R_{\mathbf{\Theta}_{k-1}}(\mathbf{X}) = \mathbf{\Theta}_{k-1}\mathbf{\Theta}_{k-1}^T\mathbf{X}$ is the reconstruction of the data using the first $k-1$ principal components. It has been proven that $\mathbf{e}_i$s are the eigenvectors of the data covariance matrix $E(\mathbf{X}\mathbf{X}^T)$ that correspond to the ith largest eigenvalues.

Various extensions of PCA have been proposed. The probabilistic PCA (P-PCA) [51] is proposed to handle noisy data. The kernel PCA (K-PCA) [52] can be used to identify the nonlinear manifold of the data. The generalized PCA (G-PCA) [53] is proposed to handle cases where the data may lie on the union of different linear subspaces; an interesting extension of GPCA is a multiscale hybrid linear representation of images [54]. Manifold pursuit [55] is an extension of PCA, designed to handle the misaligned training images. To our knowledge, our work is the first that approximates the PCA using nonorthogonal binary box functions for the purpose of efficient computation.

4.2 The Approach

Similar to PCA, B-PCA tries to find a set of base vectors that encompasses most of the energy in a data set. However, an additional requirement for B-PCA is that each base vector has to be a linear combination of a small number of Haar-like box functions. Like PCA, we formulate the B-PCA in an incremental way: having first determined the $(k-1)$ B-PCA bases, the kth base vector ψ_k is determined as

$$\arg\max_{\psi} E\{[\psi^T(\mathbf{X} - R_{\mathbf{\Psi}_{k-1}}(\mathbf{X}))]^2\} - \beta cost(\psi), \qquad (13)$$

$$subject\ to : \|\psi\| = 1; \psi = \sum_{j=1}^{N_k} c_{j,k}\mathbf{b}_{j,k}; \mathbf{b}_{j,k} \in D,$$

where $\mathbf{\Psi}_{k-1} = [\psi_1, \ldots, \psi_{k-1}]$ is the set of B-PCA base vectors selected up to iteration $k-1$, and $R_{\mathbf{\Psi}_{k-1}}(\mathbf{X})$ is the reconstruction of the data $\mathbf{X}$ using the bases $\mathbf{\Psi}_{k-1}$. Since each B-PCA base vector is a linear combination of a small number of binary box functions, they may not be orthogonal to each other. As a result, the reconstruction process becomes $R_{\mathbf{\Psi}_{k-1}}(\mathbf{X}) = \mathbf{\Psi}_{k-1}(\mathbf{\Psi}_{k-1}^T\mathbf{\Psi}_{k-1})^{-1}\mathbf{\Psi}_{k-1}^T\mathbf{X}$. The term $\mathbf{X} - R_{\mathbf{\Psi}_{k-1}}(\mathbf{X})$ is the reconstruction residual. $cost(\psi)$ is the cost function associated with computing the dot product between a data vector and ψ; it is roughly proportional to the number of binary box functions N_k that are used to represent ψ_k. Similar to the NBS, we enforce approximation error constraints to avoid selecting too many base vectors. β is a weight to balance the two terms. The fact that each B-PCA base vector is represented in NBS is reflected in the constraint $\psi_k = \sum_{j=1}^{N_k} c_{j,k}\mathbf{b}_{j,k}$ and $\mathbf{b}_{j,k} \in D$. Note that the $c_{j,k}$ is different from the direct NBS approximation coefficients; they are normalized to enforce the constraint $\|\psi\| = 1$.

The intuition behind this objective function is that the new base vector ψ_i needs to capture most of the data variance (the first term) and, at the same time, be a linear combination of a few binary box functions $\mathbf{b}_{j,k}$, which makes it computationally efficient.

As can be observed, the main difference between PCA and B-PCA is the additional constraint that B-PCA base vectors need to be represented in NBS. These B-PCA base vectors are in general nonorthogonal. Fig. 4 illustrates the relation between a PCA subspace and a B-PCA subspace. The base vectors of PCA are orthogonal to each other; the B-PCA base vectors are generally nonorthogonal. Each B-PCA base vector is a linear combination of binary box functions. However, both of them represent roughly the same subspace.
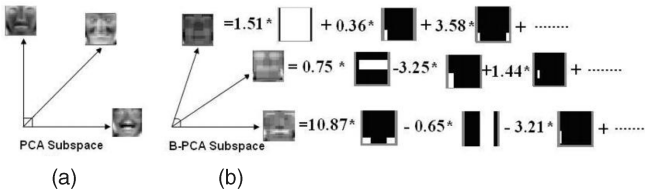
(a)

(b)

Fig. 4. Relation of the PCA subspace and the B-PCA subspace. (a) is the PCA base vectors, which are orthogonal to each other. (b) is the B-PCA base vectors, which are approximations of the PCA bases using NBS. We can observe block effect in the B-PCA base vectors, which is the consequence of such an approximation.
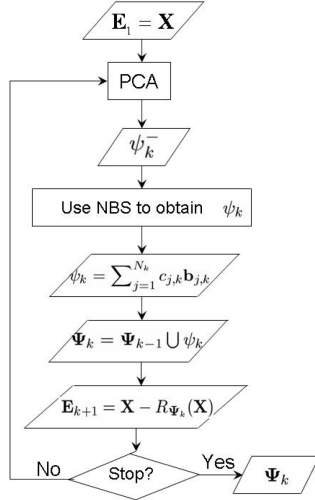

Fig. 5. Flow chart of the PCA-guided OOMP.

4.3 The Solution—PCA-Guided NBS

The search space for the optimization problem in (13) is extremely large because the solution can be any base vector that is a linear combination of box functions from the binary dictionary D. Even for a small image of size 24×24 used in our experiments, there are 134,998 box functions in D. Suppose that each B-PCA base vector is represented by 10 box functions, the number of possible choices of box functions for a single B-PCA base vector is $C_{134,998}^{10}$. It would be an understatement to say that finding the global optimal would be impractical.

One possible solution is to apply the PCA on the training data to obtain k principal components $[\mathbf{e}_1, \ldots, \mathbf{e}_k]$, then employ NBS to approximate each of these principal components with a given precision, and use the approximated vectors as the B-PCA base vectors $[\psi_1, \ldots, \psi_k]$. The problem with this solution is that the approximation errors $(\mathbf{e}_i - \psi_i)$ are generally not represented by any of the B-PCA base vectors; this leads to an inaccurate subspace.

To overcome this problem, we propose a PCA-guided NBS method to find a suboptimal solution efficiently. In the PCA-guided NBS, we denote the selected B-PCA base vectors up to iteration k as $\mathbf{\Psi}_k = [\psi_1, \psi_2, \ldots, \psi_k]$. This set is empty at the beginning. We start from the original PCA procedure to obtain the first principal component that captures the majority of the data variance. We call the first principal component the *Pre-BPCA* vector, denoted as ψ_1^-. NBS is then applied to approximate this vector as $\psi_1 = \sum_{j=0}^{N_1} c_{j,1} \mathbf{b}_{j,1}$. Then, in iteration k, the data $\mathbf{X}$ is projected to the subspace spanned by the already selected B-PCA bases $\mathbf{\Psi}_{k-1}$, and PCA
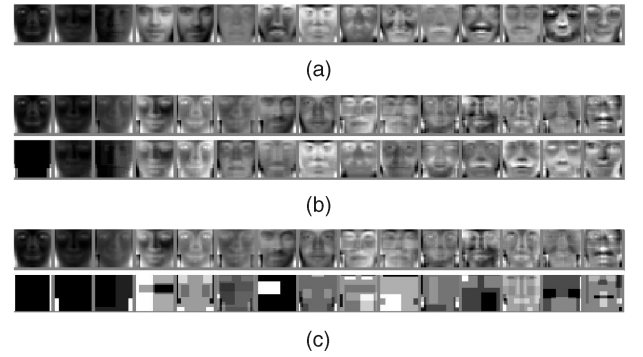

(a)

(b)

(c)

Fig. 6. Approximations of the first 15 Pre-BPCA base vectors using different thresholds ζ. (a) The PCA base vectors. (b) B-PCA base vectors with $\zeta = 0.2$ Pre-PCA base vectors (first row) and B-PCA base vectors (second row). (c) B-PCA base vectors with $\zeta = 0.85$, which has an obvious block effect.

is applied on the residual of the data $\mathbf{X} - R_{\mathbf{\Psi}_{k-1}}(\mathbf{X})$ to obtain the next *Pre-BPCA* ψ_k^-, which is again approximated using NBS. The approximation of *Pre-BPCA* at iteration k is called the kth *B-PCA base vector*. This procedure iterates until the desired number of B-PCA bases have been obtained or an error threshold is reached. The procedure of this optimization is shown in Fig. 5.

Generally, it takes a large number of box functions to represent each *Pre-BPCA* base vector perfectly. However, we prefer a solution with fewer box functions to make the representation computationally efficient. To make the optimization simpler, we enforce the computational cost constraint by finding the minimum number of box functions that satisfy

$$\frac{1}{N} \sum_{i=1}^{N} \left| \frac{(\psi_1^- - \overline{\psi_1^-})_i}{(\psi_1^-)_i} \right| \leq \zeta, \tag{14}$$

where $\overline{\psi_1^-}$ is the reconstruction of ψ_1^- using binary box functions. $\zeta \in [0, 1]$ is the approximation error threshold that controls the precision. A smaller value of ζ leads to produce a more accurate approximation. N is the dimension of the base vector, and $(\cdot)_i$ refers to the ith element of a vector. This constraint requires the elementwise approximation error ratio to be within a threshold. It is stricter than the norm constraint, which requires the difference in norm to be smaller than a threshold. A comparison of PCA, Pre-BPCA, and B-PCA base vectors are shown in Fig. 6.

4.4 Theoretical Analysis of the B-PCA Bases

As mentioned in the previous section, when the approximation error threshold ζ is 0, the B-PCA base vector is identical to the PCA base vector. When ζ increases, B-PCA base vectors deviate from the PCA bases and become more nonorthogonal. Nonorthogonality, which is often measured using *coherence*, will be defined in this section. We will prove that by approximating the original projection process $P_{\mathbf{\Psi}}(\mathbf{x}) = (\mathbf{\Psi}^T \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \mathbf{x}$ with the *direct dot nonorthogonal projection process* (DNP): $P_{\mathbf{\Psi}}(\mathbf{x}) = \mathbf{\Psi}^T \mathbf{x}$. The resultant distance error of $P_{\mathbf{\Psi}}(\mathbf{x})$ is related to coherence and, therefore, ζ. Based on this property, we conclude that when ζ is small, the information loss by using B-PCA and DNP is also small, and the computational complexity is reduced significantly. This was verified by our experiments on real data sets.

Definition 1. *A μ-coherent base vector set $\boldsymbol{\Psi}$ with $\|\psi_i\| = 1$ has coherence μ for $0 \leq \mu \leq 1$ if $|\langle \psi_i, \psi_j \rangle| \leq \mu$ for all distinct ψ_i, $\psi_j \in \boldsymbol{\Psi}$. Intuitively, for a μ-coherent dictionary, the angle between any pair of base vectors or the negation of the vectors has to be larger than $|\cos^{-1}\mu|$. A 0-coherent base vector set is orthogonal.*

Lemma 1. *If we denote $B = |\boldsymbol{\Psi}|$ and $\mu B \leq 0.5$, then there exists a set of vectors $\mathbf{e}_i, i = 1, \ldots, B$, such that*

- *the $\mathbf{e}_i$s form an orthonormal system;*
- *Span $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_B) = \mathrm{span}(\psi_1, \psi_2, \ldots, \psi_B)$; and*
- *$\|\mathbf{e}_i - \psi_i\|^2 \leq 8\mu^2 B$.*

This lemma states that when the coherence satisfies the above conditions, we can find an orthonormal system that has the same span as the nonorthogonal base vectors. In addition, these orthonormal base vectors are very close to the original nonorthogonal ones. The distance between corresponding base vectors is a function of coherence. The proof can be found in [36].

Lemma 2. *The angle θ_i between each nonorthogonal base vector ψ_i and its corresponding orthogonal base vector $\mathbf{e}_i$ is smaller than $\theta_{max} = 2\sin^{-1}(2\mu^2 B)^{1/2}$, where $B = |\boldsymbol{\Psi}|$.*

Proof. See the supplementary material, which can be found at http://computer.org/tpami/archives.htm. ☐

Theorem 1. *By approximating the original projection process $P_{\boldsymbol{\Psi}}(\mathbf{x}) = (\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^T\mathbf{x}$ with the direct dot nonorthogonal projection process (DNP) $\widehat{P}_{\boldsymbol{\Psi}}(\mathbf{x}) = \boldsymbol{\Psi}^T\mathbf{x}$, the resultant distance error of $P_{\boldsymbol{\Psi}}(\mathbf{x})$ is bounded by a function of μ, i.e.,*

$$\|\widehat{P}_{\boldsymbol{\Psi}}(\mathbf{x})\| - \|P_{\boldsymbol{\Psi}}(\mathbf{x})\| \leq g(\mu) = \left(\sqrt{\sum_i c_i'^2} - 1\right)\|\mathbf{x}\|$$

$$\leq \left(\sqrt{1 + 2(B-1)K + BK^2} - 1\right)\|\mathbf{x}\|,$$

where $K = \sqrt{8\mu^2 B(1 - 2\mu^2 B)}$.

Proof. See the supplementary material, which can be found at http://computer.org/tpami/archives.htm. ☐

4.5 Speed Improvement of B-PCA

Suppose that the image size is $m \times n$, T_{PCA} denotes the time for computing the PCA subspace projection coefficients, and N denotes the number of PCA base vectors. It will take $m \times n \times N$ floating point multiplications and $N \times (m \times n - 1) + (N-1)$ floating point additions to perform the projection operation

$$T_{PCA} = N \times m \times n \times T_{fm} + [N \times (m \times n - 1) + (N-1)] \times T_{fa}, \quad (15)$$

where T_{fm} is the time for a single floating point multiplication, and T_{fa} is the time for a single floating point addition.

For B-PCA, the time for a single projection is denoted as T_{BPCA}, which consists of two parts. One part is T_{ii}, the time to construct the integral image. For an $m \times n$ image, it will take $m \times n \times 2$ integer additions with recursive implementation. This is performed only once for each image. The other part is the time for the projection operation $P_{\boldsymbol{\Psi}}(\mathbf{x}) = (\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^T\mathbf{x}$. When the bases are nearly orthogonal to each other, we can approximate the projection coefficient using the direct dot

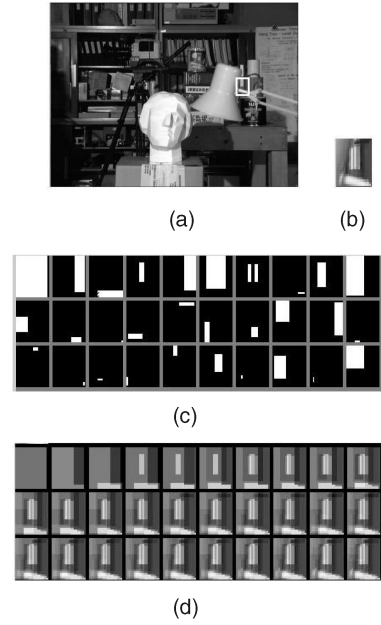Fig. 7. (a) The "Tsukuba" image taken from [56]. (b) A 25×20 image patch from this image. (c) The first 30 NBS base vectors used to represent (b). (d) The reconstruction of (b) using 1-30 NBS base vectors.

product $\boldsymbol{\Psi}^T\mathbf{x}$. The B-PCA base vector $\psi_i (1 \leq i \leq N)$ is represented as a linear combination of N_i box functions, $\psi_i = \sum_{j=1}^{N_i} c_j \mathbf{b}_j$. The projection of $\mathbf{x}$ to ψ_i can be written as $\langle \psi_i, \mathbf{x} \rangle = \sum_{j=1}^{N_i} c_{i,j} \langle \mathbf{b}_j, \mathbf{x} \rangle$. Each box function $\mathbf{b}_j$ has n_j boxes, where n_j can be one or two. The $\langle \mathbf{b}_j, \mathbf{x} \rangle$ can be performed using $3 \times n_j$ integer additions. Since c_j is a floating point, $\langle \psi_i, \mathbf{x} \rangle$ needs N_i floating point multiplications and $N_i - 1$ floating point additions.

$$T_{BPCA} = \sum_{i=1}^N \sum_{j=1}^{N_i} (3 \times n_j \times T_{ia} + N_i T_{fm} + (N_i - 1)T_{fa}), \quad (16)$$

where T_{ia} is the time for one integer addition. As we can observe, T_{BPCA} is only dependent on the number of binary box functions, which is often much less than the dimension of the image. For PCA, however, the time is proportional to the image dimension. Since the number of operations in B-PCA is much smaller than that in PCA, T_{BPCA} is much less than T_{PCA}, and the speedup is more dramatic with higher dimensional data.

Suppose that $m = n = 24$ and $N = 15$, then T_{PCA} needs $24 \times 24 \times 15 = 8,640$ floating point multiplications to compute the projection coefficients. Suppose that the total number of NBS base vectors used to represent all the B-PCA base vectors is 200, that is, $\sum_{i=1}^N N_i = 200$, then the B-PCA projection only needs between $\sum_{i=1}^N N_i = 200$ and $2 \times \sum_{i=1}^N N_i = 400$ floating point operations. The speedup is significant.

5 IMPLEMENTATION AND EXPERIMENT RESULTS FOR NBS

5.1 Representing a Single Image Using NBS

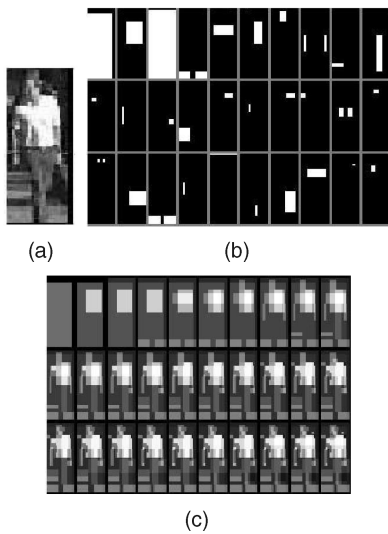To demonstrate the effectiveness and generality of the proposed NBS method, we applied it to model images

(a)   (b)

Fig. 8. (a) The "pedestrian" image of size 80×35 taken from [57]. (b) The first 30 NBS base vectors used to represent (a). (c) The reconstruction of (a) using 1-30 NBS base vectors.
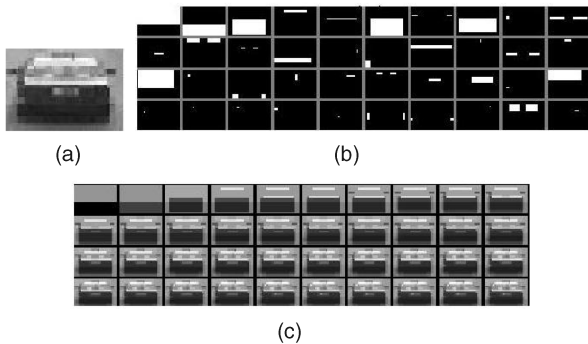

(a)   (b)
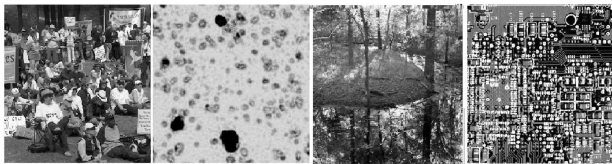
(c)

Fig. 9. (a) The original rear view vehicle image of size 20×30. (b) The first 30 NBS base vectors used to represent (a). (c) The reconstruction of (a) using 1-30 NBS base vectors.

Fig. 10. Examples in the data set for template matching.

widely used in computer vision applications. The test image in Fig. 7 is from Tsukuba data set [56], which is commonly used in stereo computation. In Fig. 7, we use the NBS method to model a 25×20 image patch in the image as shown in Fig. 7b. The box functions used to represent each image patch are shown in Fig. 7c. Fig. 7d shows the reconstruction of the image using the 1-30 NBS base vectors. Fig. 8 is the modeling of a pedestrian image taken from the Massachusetts Institute of Technology (MIT) pedestrian database [57]. Fig. 9 is the result for a rear view vehicle image. As we can observe, NBS is capable of modeling a wide variety of images effectively. With roughly 20 NBS base vectors, the images can be represented very well.

5.2 Image Matching Using NBS

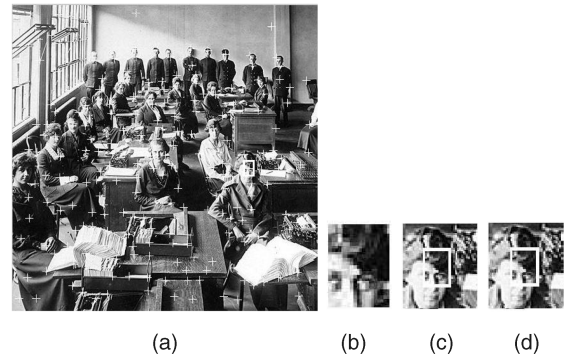As described in Section 2, to compute the numerator for NCC in (6), the template is first approximated using NBS. The

(a)   (b)   (c)   (d)

Fig. 11. (a) The original 480×480 image. (b) Selected 25×20 template patch at a corner. (c) The position with the highest normalized correlation score. (d) The result using NCC approximated by the proposed NBS method.

TABLE 1
Time Comparison between the Original NCC and NBS-NCC

-	Preprocessing(sec)	NCC (sec)	Total(sec)
Original NCC	0	3.72	3.72
FFT-NCC	0.016	0.843	0.859
NBS-NCC	0.016	0.108	0.124

numerator is computed using (7) and (8), with $|\Lambda| + 1$ multiplications and between $3|\Lambda|$ and $7|\Lambda|$ additions. In a straightforward implementation, $(2h + 1) \times (2w + 1)$ multiplications and $(2h + 1) \times (2w + 1) - 1$ additions are needed. To test the matching performance, experiments on real images were performed. Ten images of different scenes (indoor, outdoor, from microscope, and so forth) were used as the testing data set, and some of them are shown in Fig. 10 In each image, 40 patches were chosen to be located using NCC. In Fig. 11, an example is shown. Fig. 11a is the original image with 480×480 pixels. Fig. 11b is a 25×20 pixel image patch in the original image. It is approximated using up to 20 NBS base vectors. Figs. 11c and 11d show the normalized correlation matching result using the original image patch and using the NBS template representation. Both of them locate the correct object position. Table 1 shows the computation cost for matching a 25×20 patch in the test image of size 480×480 using NCC and NBS-NCC. The patch is represented using 20 box functions. The preprocessing time includes the computation of integral image, the squared integral image, the reconstruction of the patch ($\sum_{i \in \Lambda} c_i \mathbf{b}_i$), and the precomputation of $\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{y}(h + i, w + j)$ and $\sum_{i=-h}^{h} \sum_{j=-w}^{w} \mathbf{Y}(i, j)^2$, as discussed in Section 3.1. We compared the speed for original NCC, fast NCC [42], and our method. In the fast NCC, the integral image and squared integral image are used for acceleration, and the convolution is computed using FFT. According to our experiments, the NCC and FFT-based fast NCC have almost the same matching performance. The timing for fast NCC includes the computation for image padding, FFT/inverse FFT of template and image. The original NCC is the straightforward implementation without using any acceleration. The experiments are carried out on an Intel Pentium 4 3.2 GHz machine with a 1-Gbyte RAM and C++ implementation. As we can observe, a speedup factor of over 30 is achieved over the

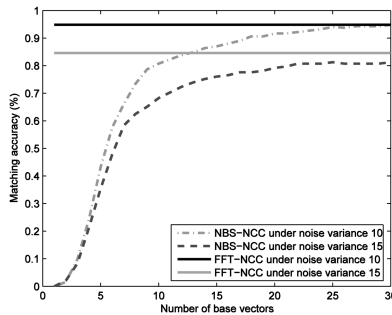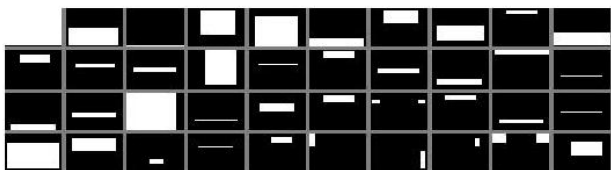Fig. 12. Testing image with added zero mean Gaussian noise ($\text{variance} = 10$).

Fig. 13. Matching performance comparison between NBS-NCC and FFT-NCC under Gaussian noise ($\text{variance} = 10$ and 15).

Fig. 14. Samples from the "rear view vehicle" data set with 1,200 images. Each image is of size 20×30.

(a)

(b)

Fig. 15. (a) The first 40 NBS base vectors computed using the rear view vehicle data set. (b) The reconstruction of a particular vehicle image, using 1-40 of the base vectors.

original NCC and over 7 times faster than the FFT-based fast NCC method for our testing scenario.

Appearances of a template in an image may vary due to noise, quantization, compression, and transformation errors. In the following experiment, various levels of Gaussian noise were added to images. Fig. 12 is a noisy image with a zero mean Gaussian noise of variance 15 (image intensity scale is 0-255). The template is matched against the noisy images
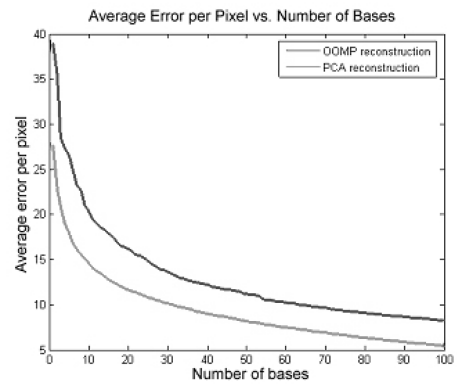

Fig. 16. Approximation error as a function of the number of base vectors for the rear view vehicle data set.

(a)

(b)

Fig. 17. (a) The first 30 nonorthogonal binary base vectors for frontal face subspace. (b) An input face image and the reconstructed face images using 1-30 of the base vectors.

using FFT-based fast NCC and our NBS-NCC method. The performance comparison is shown in Fig. 13.

5.3 Representing a Set of Images Using NBS

The NBS method described in Section 2 was implemented to construct the rear view vehicle subspace and frontal face subspace. For the vehicles data set, we used 1,200 aligned vehicle rear view images; each is of size 20×30. The data set is shown in Fig. 14. Then, the NBS subspace is trained using 800 images, and the rest are used for testing the reconstruction error. Fig. 15 shows the first 40 binary base vectors selected using NBS and the reconstruction of a particular image using the learned subspace. Note the difference of the base vectors with those in Fig. 9. The reconstruction performance of the rear view vehicles is shown in Fig. 16.

Face modeling is a very challenging task in computer vision, because of its extensive variations and due to the fact that humans are very sensitive to artifacts in face images. To model a face subspace, 500 spatially aligned frontal view images from the FERET database [58] were used. Each of these aligned images is then scaled down to 24×24 pixels. Using these 500 data samples, the NBS base vectors are computed. The first 30 of these vectors are shown in Fig. 17a. It can be observed that many of these base vectors are symmetric. The asymmetric ones may be caused by the illumination bias in the relatively small data set. The incremental reconstruction of a single face image using these 30 base vectors is shown in Fig. 17b. With each additional base vector, more details are added to the reconstructed image.

TABLE 2
Speed Comparison of MP and OOMP for Selecting 20 Features

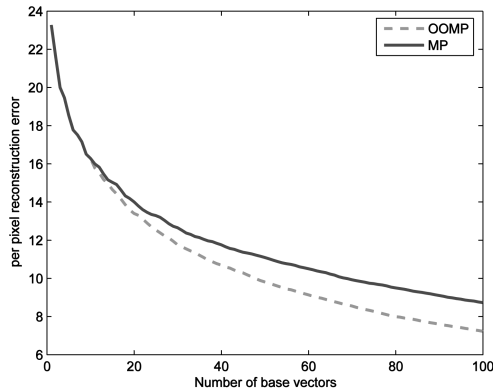| Template size	25×20	20×30	30×40	40×60
MP speed (sec)	0.063	0.093	0.297	1.156
OOMP speed (sec)	204.470	233.267	1845.605	–

Fig. 18. Reconstruction performance comparison of OOMP and MP.

5.4 Training Time

In each iteration of the OOMP feature selection process, all the remaining features in the dictionary have to be projected to the subspace spanned by the previously selected features. This process is very time consuming because it involves computing the matrix pseudoinverse. This limits its use in real-time applications. However, we found that MP [33] can be implemented much faster than OOMP without sacrificing much accuracy. This is because in each iteration of MP, the residual is projected to all the remaining features without considering the previously selected features; thus, no matrix pseudoinverse calculation is involved. In addition, this projection process can be accelerated by building an integral image for the residual image. The timing comparison is given in Table 2, and the performance comparison in terms of reconstruction error for the face data set is shown in Fig. 18. For time-critical applications such as tracking, we can use MP. On the other hand, for applications such as recognition that are not time critical on bases finding, we can use OOMP.

5.5 Face Recognition

As has been discussed in Section 3.4, it is possible to do recognition directly in NBS. We tested this in face recognition. The test database comprises 64 aligned face images in the FERET database. Each image is projected to the face NBS learned in the previous section. The projection of a face image in NBS takes the form $\mathbf{B}_\Lambda^T \mathbf{x}$. A simple nearest neighbor method is used for finding the best match as the recognition result. The NBS recognition performance is shown in Fig. 25 and compared with B-PCA method.

6 EXPERIMENTS FOR B-PCA

We applied the proposed B-PCA method to solving modeling and recognition problems of faces and vehicles. Promising results have been achieved and are demonstrated in this section.

6.1 Effectiveness of B-PCA for Reconstruction

The PCA guided NBS method described in Section 4 was implemented to obtain the B-PCA face subspace. The

TABLE 3
B-PCA Base Vectors Orthogonality with Different
Approximation Thresholds

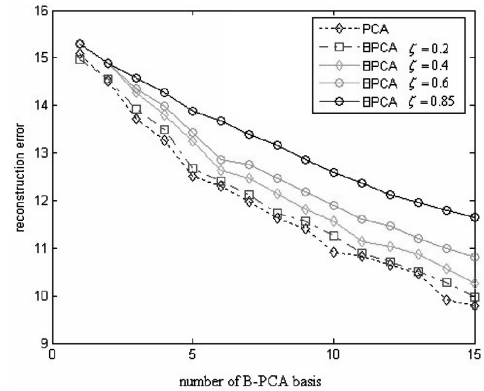| Coherence	0.0062	0.0889	0.3759
Approximation threshold ζ	0.2	0.5	0.8
Min-angle between bases(in degree)	89.6424	84.9034	68.4516

Fig. 19. Reconstruction error using different approximation errors for faces.

TABLE 4
Distortion for B-PCA Reconstruction Using
15 B-PCA Base Vectors with Different
Approximation Errors for the Frontal Face Data Set

Approximation threshold ζ	0.3	0.6	0.9
Distortion ($\|\mathbf{x} - \bar{\mathbf{x}}\|/\|\mathbf{x}\|$)	0.0937	0.1530	0.3312

training set is the same as that used in NBS subspace computation. Using 500 training samples, the B-PCA base vectors are computed. The first 15 of these vectors are shown in Fig. 6. It can be observed that like PCA, B-PCA base vectors can capture the face structure. Each individual base vector resembles some face shape. However, there is some blocky appearance in the B-PCA base vectors due to the approximation error using box functions.

The B-PCA bases coherence and reconstruction performance are directly influenced by the approximation threshold ζ in the PCA-guided NBS. With a higher threshold, which implies a less accurate approximation, the coherence will increase, and the bases become less orthogonal. When the base vectors are more orthogonal (smaller ζ), the reconstruction error will be smaller, because B-PCA base vectors become more similar to PCA base vectors. We have listed the coherence of the B-PCA base vectors with different approximation thresholds in Table 3. The reconstruction performance using different approximation thresholds are shown in Fig. 19.

To demonstrate the effectiveness of DNP, we compute the distortion, which is defined as the ratio of the reconstruction error and the orthogonal reconstruction $\|\mathbf{x} - \bar{\mathbf{x}}\|/\|\mathbf{x}\|$, where $\bar{\mathbf{x}}$ is the reconstruction using DNP, and $\mathbf{x}$ is the reconstruction using orthogonal base vectors—PCA bases. When the B-PCA base vectors are more orthogonal to each other, this distortion tends to be smaller. We tested the distortion using 15 B-PCA base vectors for frontal face image reconstruction. The distortion versus the approximation error ζ are listed in Table 4. As we can observe, the more orthogonal the B-PCA bases, the less is the distortion. This experimentally proved the theorem in Section 4.4.
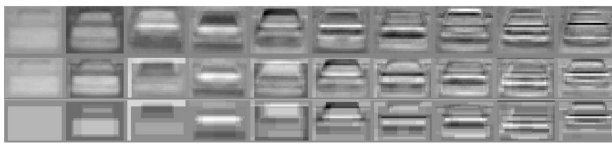
Fig. 20. Comparison of different bases (PCA, Pre-BPCA, and B-PCA) for rear view vehicles. The first row are the PCA base vectors, the second row are the Pre-BPCA base vectors, and the third row are the B-PCA base vectors ($\zeta = 0.4$).
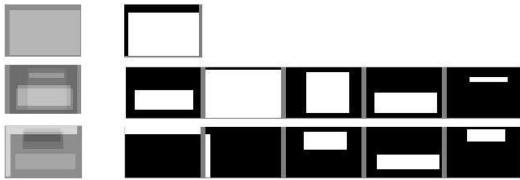

Fig. 21. B-PCA base vectors and their corresponding box functions for vehicle modeling.
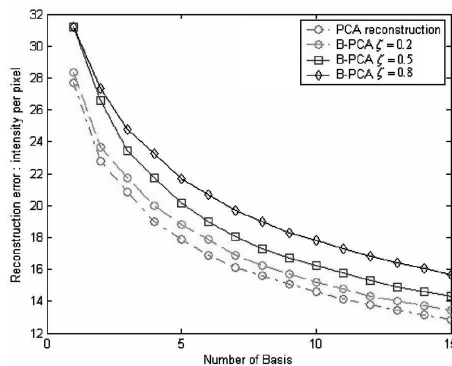

Fig. 22. Reconstruction error using different approximation errors.

To demonstrate the generality of the B-PCA subspace, we performed another experiment to model the rear views of vehicles. The data set is shown in Fig. 14. Then, we trained the B-PCA subspace using 800 images and used the other 400 images for testing the reconstruction error. The first 10 PCA, Pre-PCA, and B-PCA are visualized in Fig. 20, and Fig. 21 shows that the first three B-PCA base vectors and the corresponding box functions used to represent them. A comparison of the reconstruction performance between PCA and B-PCA with varying approximation errors is illustrated in Fig. 22.

6.2 B-PCA for Face Recognition

To demonstrate the effectiveness of the proposed B-PCA method, we applied it to face recognition. The result is compared with that of the "Eigenface." The test database comprises 64 aligned face images in the FERET database. Each image is projected to the PCA and B-PCA subspaces, and the projection coefficients of the image onto each base are used as the features; the best match in a nearest neighbor classifier is used as the recognition result. We compared the PCA with B-PCA using the original projection process that has the pseudoinverse $(\Psi^T\Psi)^{-1}\Psi^T\mathbf{x}$ and also that of DNP $\Psi^T\mathbf{x}$ for recognition; the results are shown in Figs. 23a and 23b. In Fig. 23a, the approximation threshold ζ is set to 0.9, and in Fig. 23b, ζ is 0.2. As can be observed, when $\zeta = 0.2$, which means the B-PCA base vectors are more orthogonal, the difference between recognition rates using pseudoinverse and DNP is very small. The DNP $\Psi^T\mathbf{x}$ is computationally
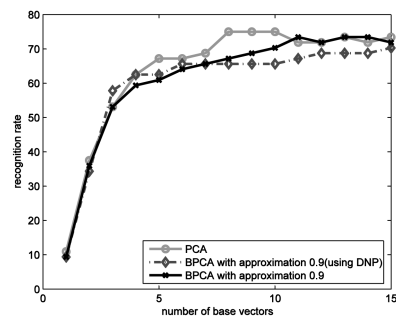
(a)
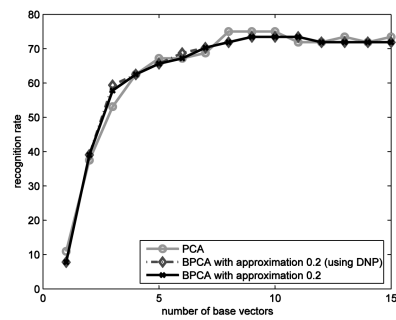
(b)

Fig. 23. (a) Recognition performance for the face data set with $\zeta = 0.9$. (b) Recognition performance for the face data set with $\zeta = 0.2$.
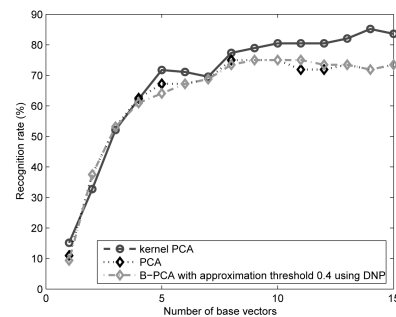

Fig. 24. Performance comparison between PCA, B-PCA, and K-PCA using different numbers of base vectors. In the B-PCA recognition, the approximation threshold is set to $\zeta = 0.4$ and DNP is used.

much cheaper because it only needs several additions. The difference between $(\Psi^T\Psi)^{-1}\Psi^T\mathbf{x}$ and $\Psi^T\mathbf{x}$ is small when the B-PCA base vectors are more orthogonal (smaller ζ), and it is larger when the base vectors are less orthogonal (larger ζ). As we can observe, when the approximation threshold ζ is small, the difference between B-PCA and PCA base vectors is small, the difference between DNP and pseudoinverse projection is also small.

We also compared the B-PCA performance with K-PCA, which has empirically shown to have superior performance in PCA-related recognition algorithms. A detailed description of K-PCA can be found in [52]. The performance comparison between PCA, B-PCA, and K-PCA (Gaussian kernel) is shown in Fig. 24. As can be observed, K-PCA has a better recognition rate than PCA and B-PCA; this is congruent with the result in [59]. However, the computation of K-PCA feature extraction is significantly higher than PCA and B-PCA because the algorithm needs to compute the kernel matrix, which involves the kernel computation between a testing image and all of the training images. As

TABLE 5
Comparison of the Computational Cost between PCA and B-PCA Projection Operations

Approximation threshold: ζ	0.1	0.4	0.6	0.85
# binary box functions	1278	409	214	68
T_{PCA} (a single subspace projection (ms))	0.315			
T_{ii} (integral image computation for each image)	4.72×10^{-2}			
T_{BPCA} (a single subspace projection (ms))	25.78×10^{-3}	8.11×10^{-3}	4.29×10^{-3}	1.45×10^{-3}
$Speedup(\frac{T_{PCA}}{T_{BPCA}+T_{ii}/N})$	10.84	27.97	42.34	68.47

the result, the K-PCA computation cost is not only related to the number base vectors but also to the number of training images. Usually, the K-PCA feature extraction is hundreds or thousands times slower than PCA depending on the number of training images used and the kernel type.

6.3   Speed of B-PCA

The experiment for speed improvement is carried out on a Pentium 4 3.2 GHz 1 Gbyte RAM machine using C++ code. Fifteen base vectors are computed for both PCA and B-PCA, and the time to project images onto each subspace is observed. The projection for PCA is $\langle \mathbf{e}_i, \mathbf{x} \rangle$; for B-PCA, it is $\langle \psi_i, \mathbf{x} \rangle$ $(1 \leq i \leq N)$. For B-PCA, the dot product between each of its NBS base vectors and the image $\mathbf{x}$ is computed using integral image with several additions. We tested the PCA projection and B-PCA projection with 1,000 images, and the time for a single projection is computed as the average. In Table 5, we can observe that the B-PCA projection process $\mathbf{\Psi}^T\mathbf{x}$ is over 50 times faster than direct PCA projection operation. The improvement is even more dramatic if more base vectors are used for comparison. Note that 1) for the PCA projection test, the image data is put in a continuous memory space to avoid unnecessary memory access overhead, 2) the integral image computation is very fast, because it can be computed efficiently with a recursive implementation using only integer additions, and 3) the T_{ii} is distributed into each base vector projection operation to make the comparison fair, because the integral image only needs to be computed once.

6.4   Parameter Choosing for B-PCA

For a given number of B-PCA base vectors, the only parameter in B-PCA is the approximation threshold ζ. A smaller ζ usually gives better results because the approximation is more accurate. However, on the other hand, more accurate approximations need more NBS base vectors and, thus, more computation. Therefore, the goal is to choose a ζ that can trade off between accuracy and efficiency. Although

different tasks may have different requirements, the B-PCA method provides a simple and flexible way to balance these two factors. For reconstruction tasks, accuracy is usually weighted more heavily, so a smaller ζ, for example, 0.2 might be chosen. However, for the recognition task, both accuracy and speed are of concern, and according to our experiments in Sections 6.2 and 6.4, $\zeta = 0.6$ is a good trade-off.

6.5   Relation between B-PCA and NBS

B-PCA and NBS are closely related; each B-PCA base vector is represented in NBS. Still, they do differ from each other. One of the main differences is the ability to represent global image structure information. NBS base vectors do not have a meaningful interpretation of the images, but B-PCA base vectors can capture the global image structure.

Another major difference is the orthogonality of the base vectors. As has been discussed in Section 3, B-PCA base vectors are nearly orthogonal to each other (that is, with small coherence), whereas NBS base vectors are highly nonorthogonal. To show the orthogonality of base vectors, we visualize the mutual coherence matrix for the NBS and that of B-PCA (with approximation threshold $\zeta = 0.2$) in Fig. 25. Each element at (i, j) of the matrix is the dot product between base vector ϕ_i and ϕ_j (for NBS) or ψ_i and ψ_j (for B-PCA). The orthogonality, as depicted, ranges from zero to one, where completely black elements represent orthogonal vectors, and identical vectors are completely bright (hence, the diagonal line).

The recognition task favors base vectors with low coherence, because it will make the distance closer to the euclidean distance. Since B-PCA base vectors are more orthogonal than NBS, with the same number of base vectors, B-PCA performs better than NBS in the task of recognition. To verify this, we made an experiment to compare the recognition rates between NBS and B-PCA, the test settings are the same as B-PCA recognition, the result is demonstrated in Fig. 26.
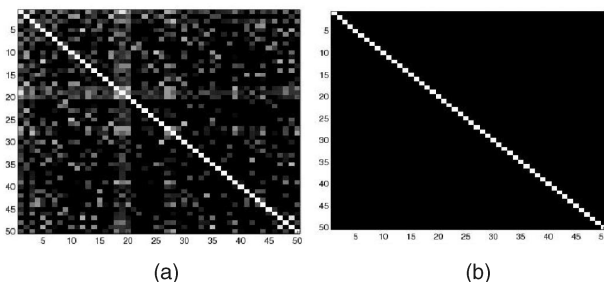

Fig. 25. (a) Mutual coherence matrix of NBS with 50 base vectors for the face data set. (b) Mutual coherence matrix for the B-PCA base vectors using the same training data.
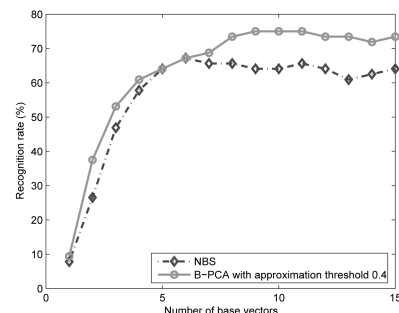

Fig. 26. Recognition performance comparison for B-PCA and NBS with different numbers of base vectors. B-PCA is computed using the approximation threshold 0.4 with direct dot product.
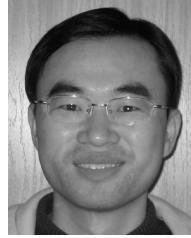
7 CONCLUSIONS AND FUTURE WORK

We have developed, in this paper, novel efficient representations for a single image or a set of images using Haar-like box functions. Using the integral image trick, the inner product with these 2D box functions can be performed using only several additions. This property makes the proposed NBS and B-PCA representations suitable for speed critical applications. The proposed NBS method can effectively represent images for a variety of matching and reconstruction tasks including NCC and SSD with a significant speedup. The proposed B-PCA subspace inherits the properties of PCA in terms of capturing the main structure of the data set while taking advantages of the computational efficiency of nonorthogonal binary bases. A PCA-guided OOMP method is proposed to obtain the B-PCA base vectors. We have applied the B-PCA method to the image reconstruction and recognition tasks. Promising results are demonstrated in this paper. Some of our future work includes the following:

- We would experiment on setting different approximation thresholds for different B-PCA base vectors. This is motivated by the observation that the first several B-PCA base vectors mainly capture the general structure of the data, whereas later base vectors capture the high-frequency details. The high-frequency information requires more binary base vectors to approximate. An adaptive scheme is needed to adjust the approximation threshold to obtain the best B-PCA both in terms of approximation accuracy and computational complexity.

- We would study the effectiveness of the proposed B-PCA method on different types of objects. Human faces and rear view of vehicles are tested in this paper. More experimental results on pedestrians and other types of objects will establish the proposed method as a valid alternative to the extensively used PCA method.

- We would try to apply more sophisticated classification methods to enhance the recognition performance since only a simple nearest neighbor classifier is used in the B-PCA recognition experiment.

REFERENCES

[1] H. Davis, *Fourier Series and Orthogonal Functions.* Dover Publications, 1989.

[2] C. Chui, L. Montefusco, and L. Puccio, *Wavelets: Theory, Algorithms, and Applications.* Academic Press, 1994.

[3] I. Jollife, *Principal Component Analysis.* Springer, 1986.

[4] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* Wiley Interscience, 2001.

[5] B. Olshausen and D. Field, "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by v1?" *Vision Research,* vol. 37, no. 23, pp. 3311-3325, Dec. 1997.

[6] D. Lee and H. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature,* vol. 401, no. 6755, pp. 788-791, Oct. 1999.

[7] H. Tao, R. Crabb, and F. Tang, "Non-Orthogonal Binary Subspace and Its Applications in Computer Vision," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 864-870, 2005.

[8] M. Turk and A. Pentland, "Face Recognition Using Eigenface," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 586-591, 1991.

[9] M. Black and A. Jepson, "Eigentracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *Proc. Fourth European Conf. Computer Vision,* pp. 329-342, 1996.

[10] R. Basri and D. Jacobs, "Lambertian Reflectance and Linear Subspaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 2, pp. 218-233, Feb. 2003.

[11] D. Belhumeur and D. Kriegman, "What Is the Set of Images of an Object under All Possible Lighting Conditions?" *Int'l J. Computer Vision,* vol. 28, no. 3, pp. 245-260, July 1998.

[12] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. I: 511-I: 518, 2001.

[13] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. II: 734-II: 741, 2003.

[14] O. Veksler, "Fast Variable Window for Stereo Correspondence Using Integral Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. I: 556-I: 561, 2003.

[15] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient Visual Event Detection Using Volumetric Features," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. I: 166-I: 173, 2005.

[16] H. Schweitzer, J. Bell, and F. Wu, "Very Fast Template Matching," *Proc. Seventh European Conf. Computer Vision,* pp. 358-372, 2002.

[17] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-Like Features for Face Detection," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. II: 1619-II: 1626, 2005.

[18] A. Bell and T. Sejnowski, "The Independent Components of Natural Scenes Are Edge Filters," *Vision Research,* vol. 37, no. 23, pp. 3327-3338, Dec. 1997.

[19] P.O. Hoyer, "Non-Negative Matrix Factorization with Sparseness Constraints," *J. Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.

[20] P. Georgiev, F. Theis, and A. Cichocki, "Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures," *IEEE Trans. Neural Networks,* vol. 16, no. 4, pp. 992-996, July 2005.

[21] M. Heiler and C. Schnorr, "Learning Non-Negative Sparse Image Codes by Convex Programming," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. II: 1667-II: 1674, 2005.

[22] T. Hazan, S. Polak, and A. Shashua, "Sparse Image Coding Using a 3D Non-Negative Tensor Factorization," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. I 50-I 57, 2005.

[23] S. Chen, D. Donoho, and M. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM J. Scientific Computing,* vol. 20, no. 1, pp. 33-61, Aug. 1998.

[24] D. Donoho and M. Elad, "Maximal Sparsity Representation via ℓ^1 Minimization," *Proc. Nat'l Academy of Sciences,* pp. 2197-2202, 2003.

[25] M. Elad and M. Aharon, "Image Denoising via Learned Dictionaries and Sparse Representation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 329-342, 2006.

[26] J. Friedman and J. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Trans. Computers,* vol. 23, no. 9, pp. 881-890, Sept. 1974.

[27] T. Kolda and D. Leary, "Computation and Uses of the Semidiscrete Matrix Decomposition," *ACM Trans. Math. Software,* vol. 26, no. 3, pp. 415-435, Sept. 2000.

[28] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Academic Press, 1990.

[29] J. Shanks, "Computation of the Fast Walsh-Fourier Transform," *IEEE Trans. Computers,* vol. c, no. 18, pp. 457-459, May 1969.

[30] Y. Meyer, *Wavelets.* SIAM, 1993.

[31] R. Gonzalez and P. Wintz, *Digital Image Processing,* second ed. Prentice Hall, 2002.

[32] G. Davis, S. Mallat, and M. Avellaneda, "Greedy Adaptive Approximation," *J. Constructive Approximation,* vol. 13, no. 1, pp. 57-98, Mar. 1997.

[33] S. Mallat and Z. Zhang, "Matching Pursuit with Time-Frequency Dictionaries," *IEEE Trans. Signal Processing,* vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[34] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal Matching Pursuits: Recursive Function Approximation with Applications to Wavelet Decomposition," *Proc. 27th Asilomar Conf. Signals, Systems, and Computers,* pp. 40-44, 1993.

[35] L. Rebollo-Neira and D. Lowe, "Optimized Orthogonal Matching Pursuit Approach," *IEEE Signal Processing Letters,* vol. 9, no. 4, pp. 137-140, Apr. 2002.

[36] A. Gilbert, S. Muthukrishnan, and M. Strauss, "Approximation of Functions over Redundant Dictionaries Using Coherence," *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms,* pp. 243-252, 2003.

[37] M. Andrle and L. Rebollo-Neira, "A Swapping-Based Refinement of Orthogonal Matching Pursuit Strategies," *Signal Processing,* special issue on sparse approximations in signal and image processing, vol. 86, no. 3, pp. 480-495, Mar. 2006.

[38] L. Di Stefano and S. Mattoccia, "Fast Template Matching Using Bounded Partial Correlation," *Machine Vision and Applications,* vol. 13, no. 4, pp. 213-221, 2003.

[39] S. Kilthau, M. Drew, and T. Moller, "Full Search Content Independent Block Matching Based on the Fast Fourier Transform," *Proc. IEEE Int'l Conf. Image Processing,* pp. I: 669-I: 672, 2002.

[40] Y. Hel-Or and H. Hel-Or, "Real-Time Pattern Matching Using Projection Kernels," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. II 1430-II 1445, 2003.

[41] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or, "The Gray-Code Filter Kernels," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 3, pp. 382-393, Mar. 2007.

[42] J. Lewis, "Fast Template Matching," *Vision Interface,* pp. 120-123, 1995.

[43] J. Ho, K. Lee, M. Yang, and D. Kriegman, "Visual Tracking Using Learned Linear Subspaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. I 782-I 789, 2004.

[44] K. Lee, J. Ho, M. Yang, and D. Kriegman, "Visual Tracking and Recognition Using Probabilistic Appearance Manifolds," *Computer Vision and Image Understanding,* vol. 99, no. 3, pp. 303-331, Sept. 2005.

[45] D. Ross, J. Lim, and M. Yang, "Adaptive Probabilistic Visual Tracking with Incremental Subspace Update," *Proc. Eighth European Conf. Computer Vision,* vol. II, pp. 470-482, 2004.

[46] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, Jan. 1995.

[47] T. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," *Proc. Fifth European Conf. Computer Vision,* pp. II 484-II 498, 1998.

[48] M. Irani, "Multi-Frame Correspondence Estimation Using Subspace Constraints," *Int'l J. Computer Vision,* vol. 48, no. 3, pp. 173-194, July 2002.

[49] K. Kanatani, "Motion Segmentation by Subspace Separation and Model Selection," *Proc. Eighth IEEE Int'l Conf. Computer Vision,* pp. II: 586-II: 591, 2001.

[50] H. Murase and S. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *Int'l J. Computer Vision,* vol. 14, no. 1, pp. 5-24, June 1995.

[51] M. Tipping and C. Bishop, "Probabilistic Principal Component Analysis," *J. Royal Statistical Soc.,* vol. 61, no. 3, pp. 611-622, 1999.

[52] B. Scholkopf, A. Smola, and K. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, no. 5, pp. 1299-1319, July 1998.

[53] R. Vidal, Y. Ma, and S. Sastry, "Generalized Principal Component Analysis (GPCA)," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1-15, Dec. 2005.

[54] W. Hong, J. Wright, and Y. Ma, "Multi-Scale Hybrid Linear Models for Lossy Image Representation," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 674-771, 2005.

[55] A. Shashua, A. Levin, and S. Avidan, "Manifold Pursuit: A New Approach to Appearance Based Recognition," *Proc. 16th Int'l Conf. Pattern Recognition,* pp. 11-15, 2002.

[56] http://cat.middlebury.edu/stereo, Sept. 2007.

[57] http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html, Sept. 2007.

[58] http://www.itl.nist.gov/iad/humanid/feret/, Sept. 2007.

[59] S. Li and A. Jain, *Handbook of Face Recognition,* chapter 7, Springer, 2004.

Feng Tang received the BS and MEng degrees from the Computer Science Department of Zhejiang University in 2001 and 2004, respectively. He is a PhD student in the Computer Engineering Department, University of California, Santa Cruz. His research interests are in image representation, tracking, and machine learning. He is a student member of the IEEE.

Ryan Crabb received the BS degree in computer science from Harvey Mudd College in 2003. He is currently in the PhD program of the Department of Computer Engineering, University of California, Santa Cruz. His research interests include visual tracking and object recognition. He is a student member of the IEEE.

Hai Tao received the BS and MS degrees in automation from Tsinghua University in 1991 and 1993, respectively. He received the MS degree in electrical engineering from Mississippi State University in 1995 and the PhD degree in electrical engineering from the University of Illinois, 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, University of California, Santa Cruz, where he is now an assistant 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 the associate editor for the journal *Machine Vision and Applications* and the journal *Pattern Recognition.*

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.