Sparse Representation for Coarse and Fine Object Recognition

Thang V. Pham and Arnold W.M. Smeulders, *Senior Member*, *IEEE*

Abstract—This paper offers a sparse, multiscale representation of objects. It captures the object appearance by selection from a very large dictionary of Gaussian differential basis functions. The learning procedure results from the matching pursuit algorithm, while the recognition is based on polynomial approximation to the bases, turning image matching into a problem of polynomial evaluation. The method is suited for coarse recognition between objects and, by adding more bases, also for fine recognition of the object pose. The advantages over the common representation using PCA include storing sampled points for recognition is not required, adding new objects to an existing data set is trivial because retraining other object models is not needed, and significantly in the important case where one has to scan an image over multiple locations in search for an object, the new representation is readily available as opposed to PCA projection at each location. The experimental result on the COIL-100 data set demonstrates high recognition accuracy with real-time performance.

Index Terms—B-spline, Gaussian derivatives, matching pursuit, multiscale, PCA, polynomial approximation, sparse representation.

1 Introduction

Automatic learning of object models from pictorial examples plays an important role in recent computer vision systems. For object manipulation and robot navigation, as well as content-based image retrieval, the ability to recognize the object view is essential.

We make a distinction between coarse and fine recognition. Coarse recognition is the ability to distinguish between different objects. For this purpose, one can construct various classifiers including boundary-based methods, for example [20]. On the other hand, fine recognition is the case where the classes are defined on a gradual or continuous scale such as aging or pose estimation. In fine recognition, discriminative approaches such as support vector classification [34], boosting [4], or one-class learning [28] cannot be used in their current application manner. In this paper, we focus on the problem of both coarse and fine recognition of object views.

This paper presents a novel representation for object models, which is the foundation for accuracy and efficiency. The representation is built from sparse, significant details rather than complete image arrays. The representation is made up from literal view fragments. It is incrementally accurate, where one may employ degrees of accuracy depending on the task at hand. The representation is extensible in that new objects can be learned incrementally without involvement of the other objects. The representation is readily available when one has to scan over an image at multiple scales and locations in search for an object.

Learning models in the new representation is based on sparse function approximation from a large dictionary of Gaussian derivative bases. It is related to the classic Njet representation [10] at multiple locations by the truncated Taylor expansion. The advantage of the approach is that, from a rich repertoire of local signs, those that best characterize the object are selected. We will show that object models can be learned efficiently from examples with the matching pursuit algorithm [13].

The new representation is related to the work of Schmid and Mohr [25], which uses combinations of Gaussian derivative filter responses invariant to certain image transformations and imaging conditions. In this method as well as others employing invariant descriptors [23], the input image is matched against known views only. On the other hand, our method provides a mechanism for matching against in-between views that are not present in the training set and, therefore, is capable of fine classification. In addition, the selection of salient points to achieve sparsity in the invariant approaches is left open. In our approach, the sparsity property has a direct link to the approximation accuracy.

Another related method is presented in [17], where the author adapts the matching pursuit algorithm to learn a linear combination of second order Gaussian derivative bases for face recognition. A major drawback of this method is that all objects share the same set of bases, which is not suited where the object appearances differ sharply from one to another. In our approach, each object is projected on a different set of bases, which allows efficient representation and expansion to a variety of objects.

The paper is organized as follows: In Section 2, we review previous approaches and formulate the problem. Section 3 presents our solution to the problem. We describe the experiments with the new approach in Section 4. Finally, Section 5 concludes the paper.

• *The authors are with Intelligent Sensory Information Systems, Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. E-mail: {vietp, smeulders}@science.uva.nl.*

Manuscript received 22 July 2004; revised 28 July 2005; accepted 11 Aug. 2005; published online 14 Feb. 2006.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0371-0704.

2 Formulation of the Problem

The intensity images taken from different view points of an object d, $d \in \{1, \ldots, D\}$, are samples of a three-dimensional function $f^{(d)}$ of intensity

$$f^{(d)}(x, y, \phi) : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad (1)$$

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

where x and y denote the two spatial axes while ϕ denotes the orientation axis. By sampling the orientation ϕ by P samples and the x and y direction by N samples, the set of images for object d can be represented as a matrix $F^{(d)}$

$$F^{(d)} = \left[\boldsymbol{f}_1^{(d)} \quad \boldsymbol{f}_2^{(d)} \quad \dots \quad \boldsymbol{f}_P^{(d)} \right], \tag{2}$$

where the pth column of the matrix $\boldsymbol{f}_p^{(d)}$ is a row-wise vectorization of the pixel values of the object d at view ϕ_p, $f^{(d)}(x_i, y_j, \phi_p)$ for $i = 1, \dots, N$ and $j = 1, \dots, N$.

Let $\boldsymbol{f} \in \mathbb{R}^{N \times N}$ be a vector representing an input image. For fine recognition, from the training data $F^{(d)}$, one tries to learn a mapping $\varphi^{(d)}(\boldsymbol{f}) : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$ to estimate the pose of the input object. For coarse recognition, given a training set of D objects $\{F^{(d)}\}$, one tries to learn a mapping $\omega(\boldsymbol{f}) : \mathbb{R}^{N \times N} \rightarrow \{1, \dots, D\}$ to identify the object.

2.1 Related Work

Poggio and Edelman [18] pioneer research in view-based representation using function approximation with radial basis function networks [19]. Specifically, the approximation of the function $\varphi^{(d)} : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$ is

$$\varphi^{(d)}(\boldsymbol{f}) = \sum_{k=1}^{K} \alpha_k G\Big(||\boldsymbol{f} - \bar{\boldsymbol{f}}_k^{(d)}|| \Big), \tag{3}$$

where the K coefficients α_k and the centers $\bar{\boldsymbol{f}}_k^{(d)}$ are found in the learning phase and $G(\cdot)$ is an appropriate basis function, typically Gaussian [19]. In this case, the image $\boldsymbol{f}$ of the object is represented by the similarities to the centers $\bar{\boldsymbol{f}}_k^{(d)}$ which can be seen as object image prototypes. The problem with this approach is that one has to store the prototypes $\bar{\boldsymbol{f}}_k^{(d)}$, each of which is of the size of the input image. The coarse recognition is carried out by generating a standard view for the input image by learning one function for each pixel, similar to the pose function in (3) and, subsequently, comparing the generated view with the standard views of all objects in the database. This is computationally expensive as the image generation and comparison is done in the high-dimensional space of $\boldsymbol{f}$.

There is a link between the type of basis function G and a priori information about the function to be approximated. In [19], the authors show that function $\varphi^{(d)}$ in (3) minimizes the functional

$$H[\varphi] = \sum_{p=1}^{P} \ell\Big(\phi_p - \varphi\Big(\boldsymbol{f}_p^{(d)} \Big) \Big) + \lambda ||\mathcal{P}\varphi||^2, \tag{4}$$

where $\varphi(\cdot)$ is the function we search for and ϕ_p is the true value. The function $\ell(\cdot)$ is a loss function, $\mathcal{P}\varphi$ is a constraint operator, and λ is a regularization parameter. The first term in (4) addresses the proper fitting of the data points while the second term penalizes overfitting. The regularization term reflects the a priori information about the function. For instance, when G in (3) is a Gaussian function, it is shown in [19] that the regularization term penalizes all derivatives of φ to obtain smooth solutions. The derivation of G for other types of a priori knowledge is not straightforward.

The support vector estimation of functions also approximates functions by a linear combination of bases as in (3) with the prototypes being a subset of the training data

$$\varphi^{(d)}(\boldsymbol{f}) = \sum_{p=1}^{P} \alpha_p G\Big(||\boldsymbol{f} - \boldsymbol{f}_p^{(d)}|| \Big), \tag{5}$$

where $G(\cdot)$ is a kernel function. The link between various kernel functions and a priori information is established in [27] via the regularization approach as in (4). This approximation employs the so-called ϵ-sensitive loss function ℓ

$$\ell(z) = \begin{cases} |z| & \text{if} \quad |z| \geq \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where z is the difference between the true value and the approximated value in (4). The interesting aspect of this loss function is that the solution $\varphi^{(d)}$ of (4) has a small number of nonzero coefficients α_p. Thus, only the corresponding support vectors $\boldsymbol{f}_p^{(d)}$, $\alpha_p \neq 0$, need to be kept. Similar to the radial bases function approach, one problem with the support vector method is the number of data points that have to be stored, especially when the function to be approximated is nonlinear.

The work of Murase and Nayar [15] is representative for the class of methods that employ dimension reduction by linear projection prior to recognition. The approach uses principle component analysis (PCA), similar to the eigenface method in [30]. Specifically, let $\bar{\boldsymbol{f}}$ denote the average of all training vectors

$$\bar{\boldsymbol{f}} = \frac{1}{DP} \sum_{d,p} \boldsymbol{f}_p^{(d)} \tag{7}$$

and M denote the projection matrix learned by PCA. The training set $F^{(d)}$ is projected to the low-dimensional space spanned by the row vectors of M

$$F_{\text{PCA}}^{(d)} = M\Big\{ F^{(d)} - \bar{\boldsymbol{f}} \mathbf{1}_P^T \Big\}, \tag{8}$$

where $\mathbf{1}_P^T$ is the transpose of a vector of P components of ones. Murase and Nayar use cubic-spline interpolation among the training points $F_{\text{PCA}}^{(d)}$ to learn smooth curves $\boldsymbol{\Gamma}^{(d)}$ parameterized the pose parameter ϕ

$$\boldsymbol{\Gamma}^{(d)}(\phi) = \boldsymbol{a}_{p,0}^{(d)} + \boldsymbol{a}_{p,1}^{(d)} \phi + \boldsymbol{a}_{p,2}^{(d)} \phi^2 + \boldsymbol{a}_{p,3}^{(d)} \phi^3 \quad \text{for} \quad \phi_p \leq \phi < \phi_{p+1}. \tag{9}$$

The curve $\boldsymbol{\Gamma}^{(d)}(\phi)$ is a vector-valued function with the number of components equal to the number of dimensions of the eigenspace. The coefficients $\boldsymbol{a}_{p,r}^{(d)}$ are computed by solving systems of linear equations derived from the smoothness condition at the known points, including the periodic condition.

At runtime, the input image $\boldsymbol{f}$ is also projected to the low-dimensional space

$$\boldsymbol{f}_{\text{PCA}} = M\{\boldsymbol{f} - \bar{\boldsymbol{f}}\}. \tag{10}$$

The pose estimation function $\varphi^{(d)}(\boldsymbol{f})$ is defined as the pose of the closest point to the curve $\boldsymbol{\Gamma}^{(d)}(\phi)$ in terms of the sum of squared differences

$$\varphi^{(d)}(\boldsymbol{f}) = \arg\min_{\phi} ||\boldsymbol{f}_{\text{PCA}} - \boldsymbol{\Gamma}^{(d)}(\phi)||^2 \tag{11}$$

and, for coarse recognition,

$$\omega(\boldsymbol{f}) = \arg\min_d \left\{ \min_\phi ||\boldsymbol{f}_{\text{PCA}} - \boldsymbol{\Gamma}^{(d)}(\phi)||^2 \right\}. \quad (12)$$

To solve the minimization problem in (11), the authors sample the curve $\boldsymbol{\Gamma}^{(d)}(\phi)$ densely and, subsequently, search for the closest one among the sampled points. Clearly, this is not efficient in both memory storage and recognition time. The problem in (12) is solved by minimizing (11) for all d.

There is an alternative approach for recognition in the eigenspace using radial basis function networks [14]. However, a more fundamental drawback resides in the representation of the object data. The pooled eigenspace can only work as long as all objects share common features. For a large data set, it may be too general to be successful. This also holds true for other techniques for linear dimension reduction; for example, the so-called optimal linear projection [11]. Another drawback of these approaches lies in the fact that object learning is not incremental. When new objects are added to the data set, to maintain the optimality condition of PCA, the eigenspace has to be recomputed, and so do the existing object models. One solution is to use object eigenspaces [15] that are specific for individual objects. However, one needs to overcome the problem of matching objects across different low-dimensional representations.

For the class of nonlinear reduction techniques such as kernel PCA [26], automatic selection of a proper nonlinear model is a hard problem in itself. For other nonlinear techniques that have been applied successful for data visualization such as local linear embedding [22] and isomap [29], both projection and reconstruction of a new data point (not in the training set) are nontrivial [1]. Hence, it is not yet clear how one may apply these methods to learn to recognize objects efficiently.

A major drawback of all methods described in this section is that they use the vector representation for images, which does not exploit an essential image property, namely, spatial coherence allowing for condensation in the image representation to a few readily computable points. This is demonstrated by the fact that permuting the pixels in the images does not change the results. In the rest of the paper, we present a new approach for recognition that approximates the function $f^d(x, y, \phi)$ in (1) directly, exploiting a rich amount of a prior information about $f^d(x, y, \phi)$, including the spatial coherence property.

2.2 Problem Statement

We wish to approximate the function $f^{(d)}(x, y, \phi)$ from the training examples $F^{(d)}$. To this end, let $f^{(d)}(x, y, \phi)$ be parameterized by $m_d \in M$, where M denotes a family of functions to be specified. The parameter m_d can be seen as an object model for the object d.

The problem of learning a model for object d includes the specification family M and, subsequently, the estimation of $m_d \in M$ from training data $F^{(d)}$. Choosing M is hard because there is no criterion for the goodness of a family of functions. It is often defined vaguely as one that has good approximation properties for the problem at hand and supports computational efficiency. In Section 3, we will propose the use of sparse function approximation.

Once the object models $\{m_d\}$ have been learned, we can estimate the optimal pose $\phi^{(d)}$ of each object d for a two-dimensional input image $f(x_i, y_j)$ with the sum of squared differences measure by solving the following minimization problem:

$$\phi^{(d)} = \arg\min_\phi \sum_{i,j} \left(f^{(d)}(x_i, x_j, \phi) - f(x_i, y_j) \right)^2. \quad (13)$$

For the coarse recognition task, the object identity d^* can be recognized by finding the closest view across the whole data set

$$d^* = \arg\min_d \left\{ \min_\phi \sum_{i,j} \left(f^{(d)}(x_i, x_j, \phi) - f(x_i, x_j) \right)^2 \right\}. \quad (14)$$

The minimization problem in (13) is nontrivial since the objective function is generally nonconvex. Thus, local optimization methods do not suffice. In addition, the optimization problem is carried out in a high-dimensional space that is equal to the number of pixels of the input image. In Section 3.4, we present a solution to this optimization problem by turning it into the problem of piecewise polynomial evaluation.

To minimize (14), we solve (13) repeatedly for all d, $1 \leq d \leq D$. The challenge of minimizing (14) without having to solve (13) for all d is beyond the scope of the current paper. Nevertheless, it is feasible, in real-time, for databases with as many as 100 objects.

In short, we have to specify a family of functions M and learn the functions in M from training data. In addition, we have to solve the minimization problem in (13) for recognition.

3 Model Learning and Recognition

We treat the problem of model learning for each object d as that of the sparse multivariate function approximation [3], [13]. The function $f^{(d)}(x, y, \phi)$ is approximated by a linear combination of bases

$$f^{(d)}(x, y, \phi) = \sum_{k=1}^{K} \alpha_k^{(d)} \psi_k(x, y, \phi), \quad (15)$$

where $\psi_k(x, y, \phi) \in \Psi$, a predefined dictionary of K bases and $\alpha_k^{(d)}$ are the coefficients. Consequently, the model m_d consists of the nonzero coefficients $\alpha_k^{(d)}$ and the indices of corresponding bases in the dictionary. The sparsity comes from the fact that only a small number of $\alpha_k^{(d)}$ differ from zero.

We first present our preferred dictionary Ψ in Sections 3.1 and 3.2. After that, we discuss the learning of object models with the chosen dictionary. Finally, in Section 3.4, we present an efficient algorithm to recognize novel objects using the learned models.

3.1 Gaussian Derivative Bases

The design of the dictionary Ψ reflects the a priori information about the object appearance. First of all, one can observe that the functions representing images have certain general properties. There are abrupt changes due to geometric concavities of the object and of the projection. Other causes include object sharp convex folds, albedo transition, and projected shadow. In spite of these abrupt changes in the

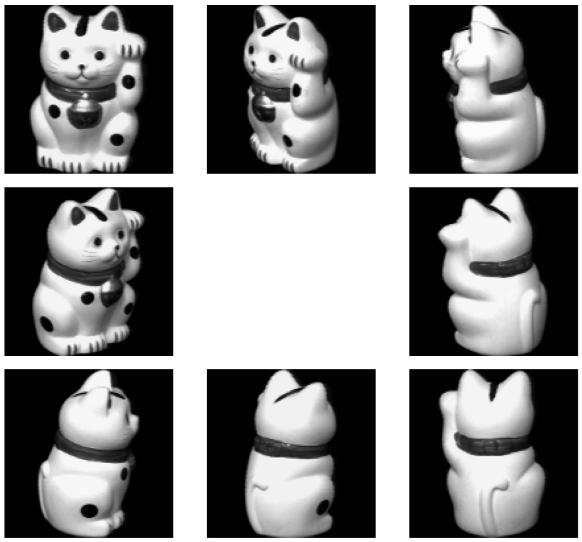

Fig. 1. Examples of an object seen from eight views, courtesy of [16].

image, the majority of pixels are smooth by the spatial coherence of object surfaces. The general challenge is to find a set of functions which match comfortably to the smooth regions almost everywhere and at the same time adapt to abrupt changes here and there, see also Fig. 2.

The dictionary Ψ consists of separable bases

$$\psi_k(x, y, \phi) = \psi_{\tau_k^x}(x)\psi_{\tau_k^y}(y)\psi_{\tau_k^\phi}(\phi), \tag{16}$$

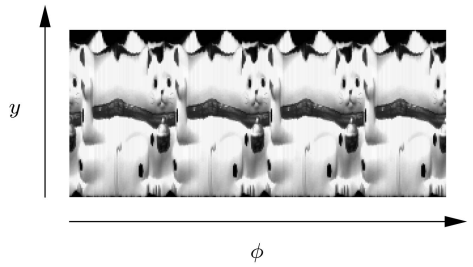

Fig. 2. The middle lines of 2D views of the object in Fig. 1 as the pose changes.

where $\psi_{\tau_k^x}$, $\psi_{\tau_k^y}$, and $\psi_{\tau_k^\phi}$ are the bases in each dimension, indexed by τ_k^x, τ_k^y, and τ_k^ϕ in the corresponding dictionary.

Our choice for the one-dimensional bases is the Gaussian derivatives $G_n(z; \sigma, \mu)$ of order n at scale σ and location μ

$$G_n(z; \sigma, \mu) = \frac{\partial^n}{\partial z^n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}. \tag{17}$$

Here, the analyzing scales are sampled exponentially. The analyzing locations are sampled equally at each scale. The reader is referred to [9] for discussion of stepping in scale and space. Figs. 3a, 3b, and 3c show examples of zeroth, first, and second order Gaussian derivative bases, respectively.

Under a reasonable assumption that no new image structure may appear at higher scale, Koenderink [9] shows that the Gaussian kernel is the only function to be used to probe an image at different scales. This leads to a representation for local image structures based on the Taylor expansion

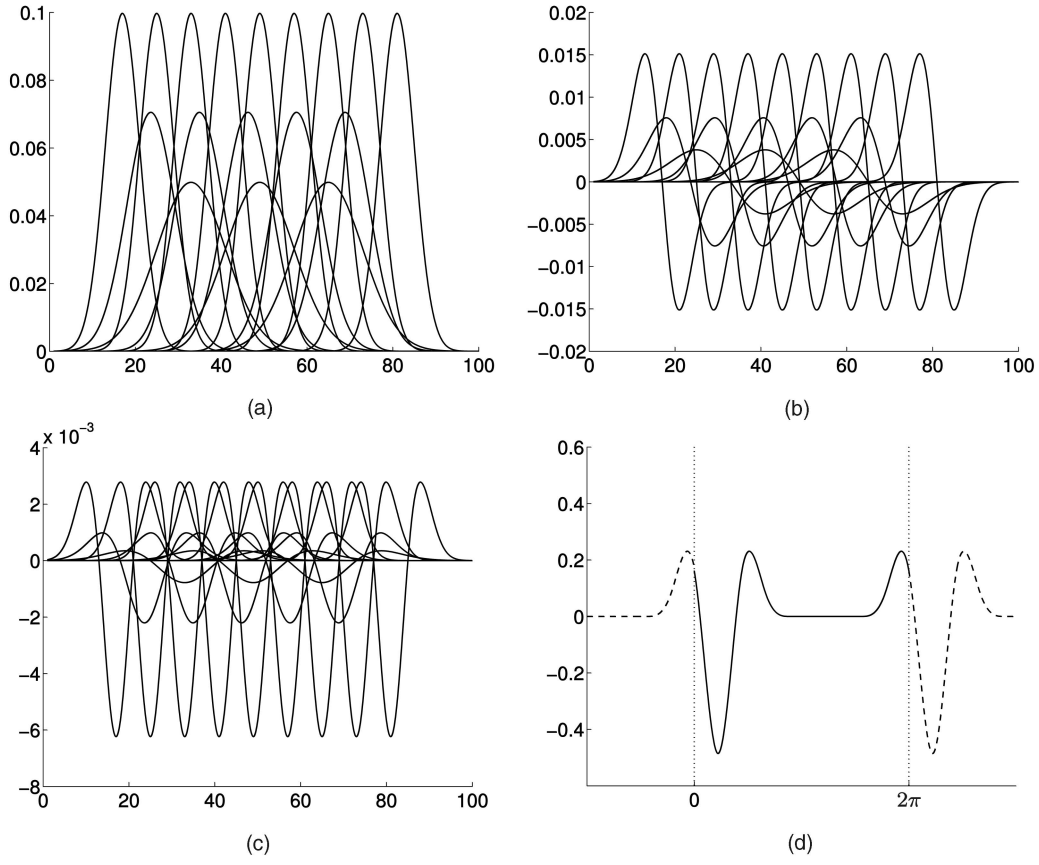

Fig. 3. Gaussian derivative bases (a) zero order, (b) first order, and (c) second order. The bases in the ϕ-axis are periodic over $[0, 2\pi)$, see (d).

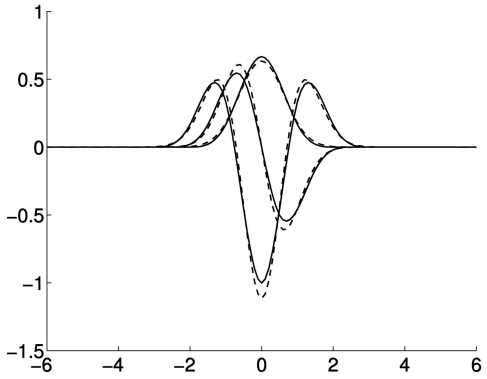

Fig. 4. The cubic B-splines (dash lines) well approximate the Gaussian derivatives (solid lines) up to the second order.

of the image using Gaussian derivatives (the Njets) [10]. By employing the Gaussian derivative bases, we will show in Section 3.4 that the recognition is based on the Njet coefficients of images, which is a well-founded approach to probe image content.

Finally, the addition of nonseparable bases can provide better image approximation, for example, along an oriented edge or around a complex structure. However, they introduce extra computational burden and are not considered in this paper.

3.2 Piecewise Polynomial Approximation

In practice, to acquire great computational advantages, we approximate the Gaussian derivatives, similar to recursive implementation [6], [33], by B-spline [31], [32], see Fig. 4. B-spline approximation to a Gaussian function has appeared in [37].

A B-spline of order n denoted by $\beta^n(z)$ is generated by convolving $n+1$ times $\beta^0(z)$ with itself

$$\beta^n(z) = \beta^{n-1}(z) * \beta^0(z) = \underbrace{\beta^0(z) * \beta^0(z) * \ldots * \beta^0(z)}_{n+1 \text{ times}}, \quad (18)$$

where $*$ denotes convolution and $\beta^0(z)$ is a centered normalized rectangle pulse

$$\beta^0(z) = \begin{cases} 1 & \text{if } |z| < \frac{1}{2} \\ \frac{1}{2} & \text{if } |z| = \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

The nth order derivative of the B-spline can be obtained from the $(n-1)$th order B-spline as follows:

$$\frac{\partial \beta^n(z)}{\partial z} = \beta^{n-1}\left(z + \frac{1}{2}\right) - \beta^{n-1}\left(z - \frac{1}{2}\right). \quad (20)$$

For our purpose, the cubic B-splines are represented by piecewise polynomials. Hence, the 1D basis functions $\psi_{\tau_k^x}(x)$, $\psi_{\tau_k^y}(y)$, and $\psi_{\tau_k^\phi}(\phi)$ are piecewise polynomials of degree three: $\beta^3(z)$, $\partial\beta^4(z)/\partial z$, or $\partial^2\beta^5(z)/\partial z^2$. We will show in Section 3.4 that efficient recognition is achieved by polynomial calculation.

3.3 Model Learning with Matching Pursuit

Once a family of functions Ψ has been chosen, the next step is to estimate a specific model from a set of training examples. Among the various methods for function approximation from a dictionary of bases, we use the matching pursuit algorithm of Mallat and Zhang [13] yielding a local optimum.

The matching pursuit algorithm aims to learn $f^{(d)}(x, y, \phi)$ in (15) as follows: At each step, it finds the basis most correlated with the residual. Let $\langle \cdot, \cdot \rangle$ denote inner product. Initialize the residual function $\rho^1(x, y, \phi)$

$$\rho^1(x, y, \phi) = f^{(d)}(x, y, \phi). \quad (21)$$

Then, loop over all bases and select the index k_r^* of the best basis function in Ψ

$$k_r^* = \arg\max_k \left\langle \rho^r, \psi_k \right\rangle \quad (22)$$

with a coefficient α along that dimension

$$\alpha_{k_r^*}^{(d)} = \left\langle \rho^r, \psi_{k_r^*} \right\rangle. \quad (23)$$

After that, update the residual function

$$\rho^{r+1} = \rho^r - \alpha_{k_r^*}^{(d)} \psi_{k_r^*} \quad (24)$$

and continue over the next basis. The number of bases R can be chosen a priori. Alternatively, we can stop the iteration when a good approximation has been reached by checking the residual

$$||\rho^r||^2 = \left|\left| f^{(d)}(x, y, \phi) - \sum_{k=1}^{K} \alpha_k^{(d)} \psi_k(x, y, \phi) \right|\right|^2. \quad (25)$$

The computation of our matching pursuit algorithm is intensive. In each iteration of the matching pursuit algorithm, one has to compute the project of the current residual ρ^r on all bases $\psi_k \in \Psi$, see (22). Fortunately, one can use the so-called network calculations for matching pursuit as in [12]. By taking an inner product with a basis ψ_k on each side of (24), we have

$$\left\langle \rho^{r+1}, \psi_k \right\rangle = \left\langle \rho^r, \psi_k \right\rangle - \alpha_{k_r^*}^{(d)} \left\langle \psi_{k_r^*}, \psi_k \right\rangle. \quad (26)$$

Hence, the correlation between the new residual ρ^{r+1} with a basis ψ_k can be computed from the existing correlation value and the correlation between the selected basis and ψ_k. By storing the correlation between all pairs of bases, (26) can be computed in a constant time.

The number of bases K can be quite large, equal to the product of the number of bases of the 1D dictionaries, far more than the capability of standard computer systems. Thanks to the separable bases, we can reduce the memory storage greatly. The correlation between two bases ψ_k and $\psi_{k'}$ can be computed as (using (16))

$$\begin{aligned} \left\langle \psi_k, \psi_{k'} \right\rangle &= \left\langle \psi_{\tau_k^x}(x)\psi_{\tau_k^y}(y)\psi_{\tau_k^\phi}(\phi), \psi_{\tau_{k'}^x}(x)\psi_{\tau_{k'}^y}(y)\psi_{\tau_{k'}^\phi}(\phi) \right\rangle \\ &= \left\langle \psi_{\tau_k^x}(x), \psi_{\tau_{k'}^x}(x) \right\rangle \left\langle \psi_{\tau_k^y}(y), \psi_{\tau_{k'}^y}(y) \right\rangle \\ &\quad \left\langle \psi_{\tau_k^\phi}(\phi), \psi_{\tau_{k'}^\phi}(\phi) \right\rangle. \end{aligned} \quad (27)$$

Thus, instead of storing one big table for the correlation between all pairs of bases in Ψ, we store three tables, one for each dimension. And, each correlation value of bases in Ψ is obtained by just two multiplications.

In summary, the learning of an object model in (15) is done efficiently by the matching pursuit algorithm. The network computation technique is used where the correlation between all pairs of bases are precomputed. This large table is implemented by storing three correlation tables, one for each dimension, and is accessed by two multiplications.

3.4 Recognition

The minimization problem in (13) is central for both coarse and fine recognition. In the following, we will show that it can be solved efficiently by polynomial computation.

Expanding (13), we have

$$\begin{aligned}\phi^{(d)} &= \arg\min_{\phi} \sum_{i,j} \Big(f^{(d)}(x_i, y_j, \phi) - f(x_i, y_j) \Big)^2 \\ &= \arg\min_{\phi} \Bigg\{ \sum_{i,j} f^{(d)}(x_i, y_j, \phi)^2 \\ &\qquad -2 \sum_{i,j} f^{(d)}(x_i, y_j, \phi) f(x_i, y_j) \Bigg\}. \end{aligned} \tag{28}$$

The first term in (28) is the energy

$$\begin{aligned}&\sum_{i,j} f^{(d)}(x_i, y_j, \phi)^2 = \\ &= \sum_{i,j} \left(\sum_{k=1}^{K} \alpha_k^{(d)} \psi_{\tau_k^x}(x_i) \psi_{\tau_k^y}(y_j) \psi_{\tau_k^\phi}(\phi) \right)^2 \\ &= \sum_{k'=1}^{K} \sum_{k''=1}^{K} \Bigg\{ \alpha_{k'}^{(d)} \alpha_{k''}^{(d)} \left(\sum_i \psi_{\tau_{k'}^x}(x_i) \psi_{\tau_{k''}^x}(x_i) \right) \\ &\quad \left(\sum_j \psi_{\tau_{k'}^y}(y_j) \psi_{\tau_{k''}^y}(y_j) \right) \times \quad \psi_{\tau_{k'}^\phi}(\phi) \psi_{\tau_{k''}^\phi}(\phi) \Bigg\} \\ &= \sum_{k'=1}^{K} \sum_{k''=1}^{K} \left\{ \alpha_{k'}^{(d)} \alpha_{k''}^{(d)} \left\langle \psi_{\tau_{k'}^x}, \psi_{\tau_{k''}^x} \right\rangle \left\langle \psi_{\tau_{k'}^y}, \psi_{\tau_{k''}^y} \right\rangle \psi_{\tau_{k'}^\phi}(\phi) \psi_{\tau_{k''}^\phi}(\phi) \right\}. \end{aligned} \tag{29}$$

Note that each term in (29) is a product of two piecewise polynomials, each of degree three. Hence, the energy is a piecewise polynomial of degree six at most. In addition, this piecewise polynomial can be precomputed since it does not involve the input image.

The second term is the cross correlation term (ignoring the constant factor)

$$\begin{aligned}&\sum_{i,j} f^{(d)}(x_i, y_j, \phi) f(x_i, y_j) = \\ &= \sum_{i,j} \left(\sum_{k=1}^{K} \alpha_k^{(d)} \psi_{\tau_k^x}(x_i) \psi_{\tau_k^y}(y_j) \psi_{\tau_k^\phi}(\phi) \right) f(x_i, y_j) \\ &= \sum_{k=1}^{K} \alpha_k^{(d)} \psi_{\tau_k^\phi}(\phi) \left(\sum_{i,j} \psi_{\tau_k^x}(x_i) \psi_{\tau_k^y}(y_j) f(x_i, y_j) \right). \end{aligned} \tag{30}$$

The sum within the brackets is an Njet coefficient corresponding to the basis $\psi_{\tau_k^x}$ in the x-axis and the basis $\psi_{\tau_k^y}$ in the y-axis. Let $\mathcal{N}^I_{\tau_k^x, \tau_k^y}$ denote this value. By grouping the Njets that belong the same basis in the ϕ-axis, we have

$$\sum_{i,j} f^{(d)}(x_i, y_j, \phi) f(x_i, y_j) = \sum_{\tau^\phi=1}^{K_\phi} \left(\sum_{k, \tau_k^\phi = \tau^\phi} \alpha_k^{(d)} \mathcal{N}^I_{\tau_k^x, \tau_k^y} \right) \psi_{\tau^\phi}(\phi), \tag{31}$$

where K_ϕ denotes the number of bases in the ϕ-axis. We have that (31) is the sum of piecewise polynomials of degree three. Hence, the result is also a piecewise polynomial of degree three.

Consider the minimization problem in (28). The first term (29) is a piecewise polynomial of degree six and the second term (31) is a piecewise polynomial of degree three. Hence, the sum does not result in a piecewise polynomial of degree greater than six. Let $g(\phi)$ be the resulting piecewise polynomial, which consists of Q polynomials $g_q(\phi)$, $1 \le q \le Q$, each of which is of degree six at most

$$g_q(\phi) = \sum_{\gamma=0}^{6} b_{q,\gamma} \phi^\gamma \quad \text{for} \quad \phi_q \le \phi < \phi_{q+1}, \tag{32}$$

where ϕ_q are the break points of the piecewise polynomial $g(\phi)$, and $b_{q,\gamma}$ are the polynomial coefficients of $g_q(\phi)$. As a consequence, the minimization problem (28) can be turned into the Q minimization of $g_q(\phi)$.

$$(\phi^{(d)}, q^*) = \arg\min_{\phi, q} \sum_{\gamma=0}^{6} b_{q,\gamma} \phi^\gamma \quad \text{for} \quad \phi_q \le \phi < \phi_{q+1}. \tag{33}$$

A simple approach to minimize (33) is to evaluate $g_p(\phi)$ at a set of densely sampled points, hence, it is equivalent to the approach of Murase and Nayar to solving (11). However, the evaluation of a polynomial is very fast. The evaluation of (33) at an arbitrary point $\phi = v$ requires only six additions and six multiplications using Horner's rule [8]

$$\begin{aligned}&g_q(v) = \\ &\big(\big(\big(\big(\big(b_{q,6} v + b_{q,5}\big) v + b_{q,4}\big) v + b_{q,3}\big) v + b_{q,2}\big) v + b_{q,1}\big) v + b_{q,0}. \end{aligned} \tag{34}$$

Fig. 5 depicts the computation of (28). The input image is transformed into the Njet representation. This can be done efficiently by convolutions exploiting the separability and the polynomial representation of the bases. The Njet coefficients $\mathcal{N}^I_{\tau_k^x, \tau_k^y}$, weighted by the corresponding $\alpha_k^{(d)}$, are grouped according to the model in (31). The resulting scalars are, in turn, the weights of the corresponding bases in the ϕ-axis. The final result $g^{(d)}(\phi)$ to be minimized is the sum of the weighted bases, which is a piecewise polynomial of degree six.

Our recognition strategy compares favorably to the approach of Murase and Nayar in both storage space and time performance. In terms of storage space, the PCA method has to store all sampled points in the eigenspace. For instance, 3,600 points are required for each object for the accuracy of 0.1 degree. On the contrary, the new method stores the coefficients of the polynomials, which is marginal. Furthermore, it is independent of the accuracy; that is, higher accuracy is obtained at no extra storage cost.

Fig. 6a shows the time performance of the two methods with respect to the number of objects. For the case where one object fills the image, it takes more time to compute the Njet representation than PCA projection in this case. However, the new method solves the minimization

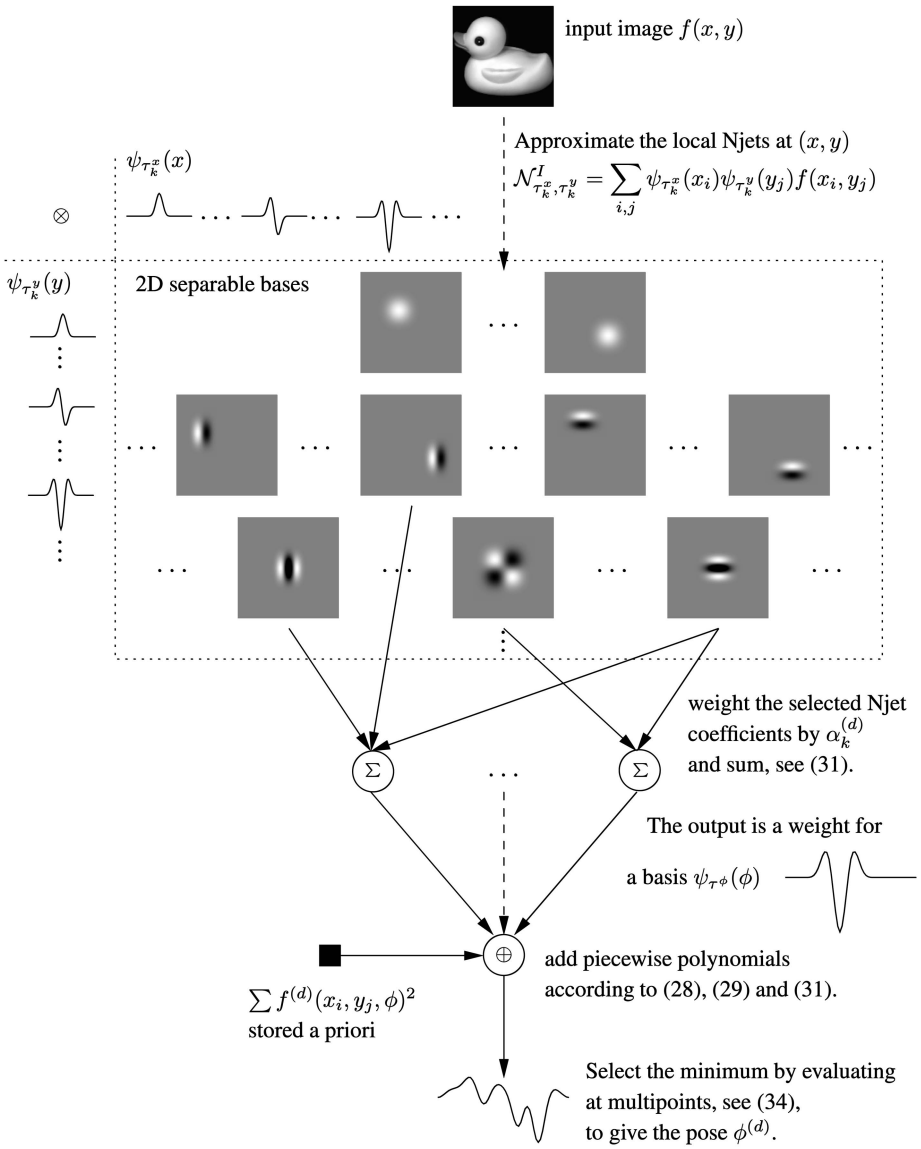

Fig. 5. The recognition algorithm.

problem in (33) (including the computation of the polynomial coefficients $b_{q,\gamma}$) faster than the minimization in the eigenspace (11), as shown in Fig. 6b. As a result, for compatible accuracy for a data set of 100 objects the new method with 300 bases performs similar to PCA with six dimensions. With 1,000 bases, it is two times faster than the PCA method with 20 dimensions. The differences will become more pronounced for larger data sets.

In another important case where one has to scan an input image over multiple scales and locations in search for an object, our method is advantageous because the Njet representation is computed only once for the whole image, as opposed to the PCA approach which has to do projection for each location and scale separately. As the input image is scanned, the basis functions in (30) move. However, the sum within the brackets in (30) need not be carried out once the Njet representation of the input image has been computed. This strategy differs from operating on each window one-by-one in that computing Njets of the input image is involved with a set of convolution operators that can be optimized efficiently.

4 Experiments

We contrast the performance of the new approach to the approach of Murase and Nayar using PCA on the COIL-100 data set [16]. This data set consists of images of 100 objects of size 128×128, each of which is captured at 72 poses (five degrees apart). We convert all images into gray scale. Fig. 7 shows 20 objects randomly taken from this data set.

First of all, we followed the experiment setup of Murase and Nayar [15]. We divided the data set into two partitions of the same size. The training set has 36 views for each object, at 10 degrees apart. The remaining views are used for testing. Therefore, there are 3,600 object views in the training set and 3,600 object views in the test set.

The parameters of the method are set as follows to ensure a reasonable learning time. The dictionary Ψ consists of three 1D dictionaries. The dictionaries in the two spatial directions x and y have three scales $\sigma \in \{4, 8, 16\}$; in each scale, the space μ are sampled at 4, 8, and 16 (pixels) apart, respectively. In total, there are 120 bases in each spatial 1D dictionary. In the ϕ direction, we examine at scale $\sigma \in \{1, 2, 4, 8\}$. The space is also sampled correspondingly at 1, 2, 4, and 8 ($\times 10$ degrees).

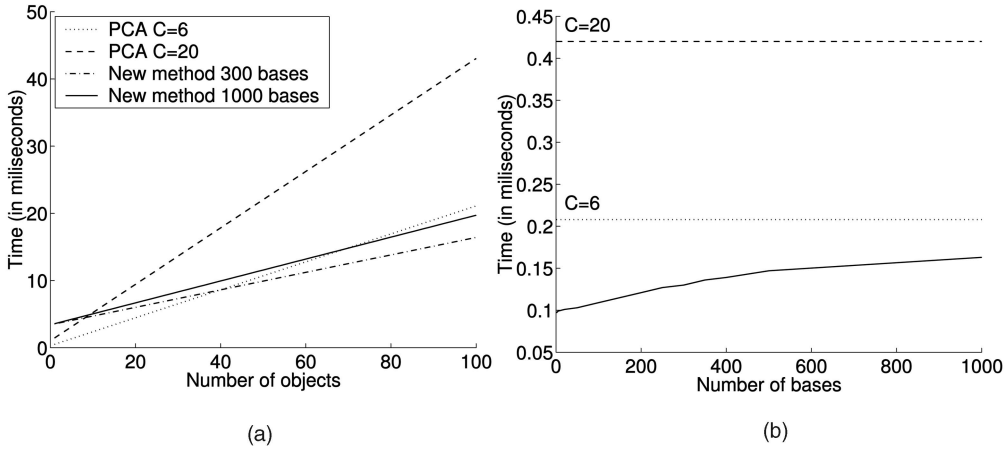

Fig. 6. Time performance of the new method versus PCA with the number of principle components C on a computer with a Pentium 4 CPU 2.8 GHz for object images of size 128×128. For both methods, 3,600 points are sampled along the ϕ-axis for each object. (a) The total recognition time with respect to the number of objects. (b) Minimization time for (33) versus that for (11) in the PCA approach.

Fig. 7. Twenty objects randomly taken from the COIL-100 data set [16].

The number of discrete points sampled for our minimization problem (33) and the PCA approach (11) is 3,600, equivalent to 0.1 degree in the orientation directions.

4.1 Compactness of the Representation

First of all, we examine the visual quality of the reconstructed images using the new representation in comparison with PCA.

Fig. 8a shows examples of generated images by the new representation using 1,000 bases for each object. For this representation, each basis requires four storage locations, one for the coefficient and three for the indices of the three one-dimensional bases. For efficient recognition as discussed in Section 3.4, we also store the energy in (29) and a number of bases in their polynomial representation (one piecewise polynomial for each scale and derivative order in the x and y direction for convolution, and the complete dictionary in the ϕ direction). Thus, the average number of storage locations required for each of the D objects is

$$\frac{B}{D} + 4R + E, \tag{35}$$

where B is the storage space for the bases, R is the number of bases, and E is the storage space for the energy. In this experiment, B is approximately 8,000 (locations). The storage space of the energy depends on the number of bases in the object model. For R equals 300 and 1,000, E is approximately 400 and 600 (locations), respectively.

For PCA, one has to store the mean and the eigenvectors which has the same size as the input dimensionality ($N \times N$), and the representation in the eigenspace with dimensionality C, see Figs. 8b and 8c. The number of storage locations required for each of the D objects is

$$\frac{(1+C)N^2}{D} + CS, \tag{36}$$

where S is the number of points to be stored for each object in the eigenspace. The storage space for PCA in this experiment is much larger than that for the new representation.

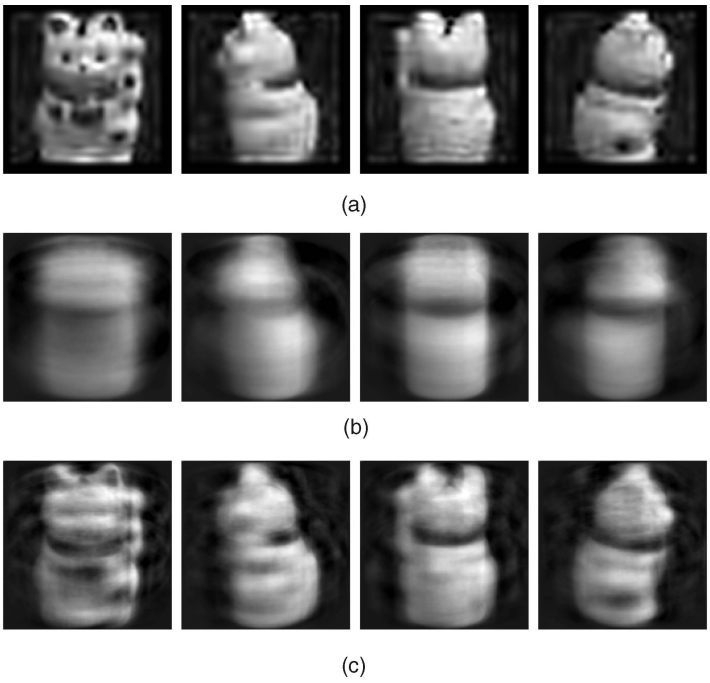

Fig. 8. Generated images from different representation learned on the COIL-100 training set. (a) The new method with 1,000 bases. (b) PCA with 20 dimensions ($T = 20$). (c) PCA with 100 dimensions ($T = 100$). The new method gives better results at a lower storage cost by exploiting the spatial correlation in images.

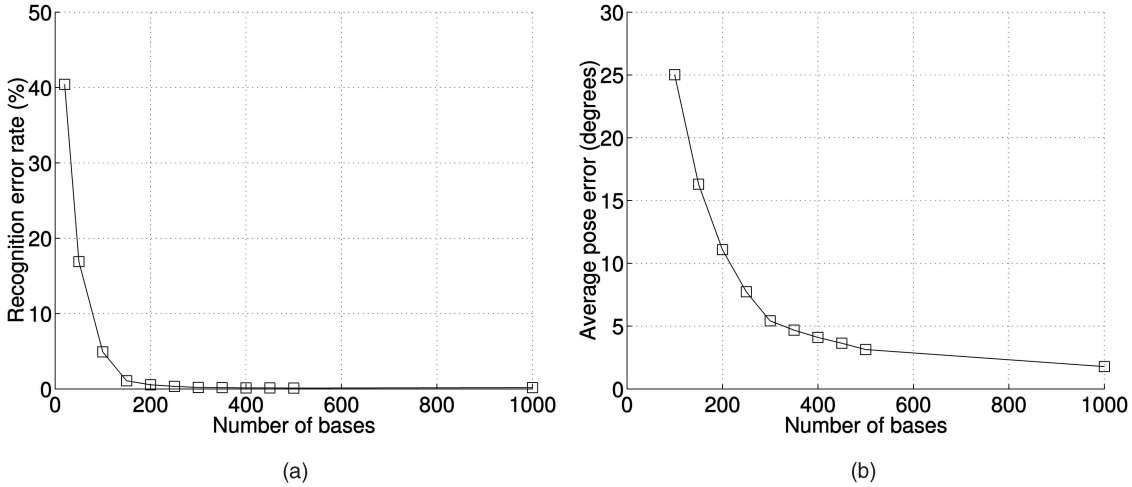

Fig. 9. Recognition and pose estimation results on the COIL-100 test set. The number of evaluation points (34) for each object is 3,600 (0.1 degree accuracy). (a) Recognition rate (%). (b) Pose estimation error (in degrees).

4.2 Recognition Performance

Fig. 9a shows the coarse recognition error rate on the test set. The method obtains an error rate of 0.17 percent with about 300 bases only. Given the variety, object discrimination typically does not require precise descriptions. For fine recognition, the accuracy increases as more bases are used as shown in Fig. 9b. The pose error appears saturated after approximately 1,000 bases.

We can observe similar behavior with the PCA approach. Fig. 10a shows the recognition error rate on the test set as a function of dimensionality of the eigenspace. The recognition error goes down to 0.19 percent using only six-dimensional eigenspace for coarse classification. For a 20-dimensional eigenspace, the average pose error showed in Fig. 10b decreases to an equal number for the new method with 1,000 bases. Hence, in terms of accuracy, the two methods are comparable when a 20-dimensional eigenspace is compared to 1,000 sparse bases.

In the next experiment, we vary the size of the training set for $D = 10, 20, \ldots, 100$, by sampling from the COIL-100 data set. We split each data set into a training set and a test set. We repeat the experiment 20 times and average the results. Fig. 11a shows the result for coarse recognition and Figs. 11b, 11c, and 11d show results for fine recognition with error in orientation within 45, 10, and 2 degrees, respectively. As expected, the results show that, while a small number of bases is sufficient for object discrimination, more bases are required for an accurate estimation of orientation. The decrease in accuracy as the number of objects grows, however, is not clearly observed, especially when more than 200 bases are used. One needs a larger data set of objects to estimate this performance degradation.

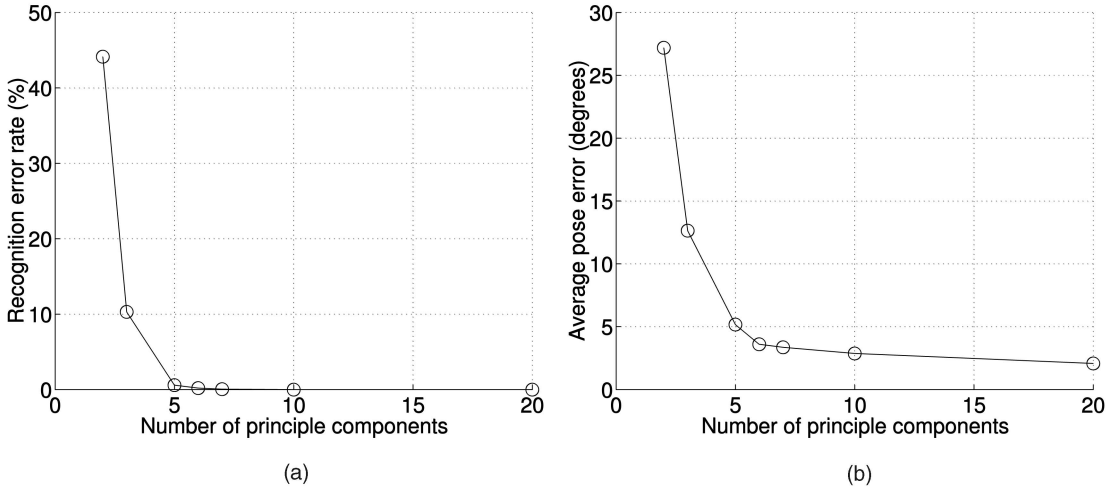

Fig. 10. The number of discrete points stored for each object is 3,600 (0.1 degree accuracy). (a) Recognition rate (%). (b) Pose estimation error (in degrees). Recognition and pose estimation results of PCA on the COIL-100 test set.

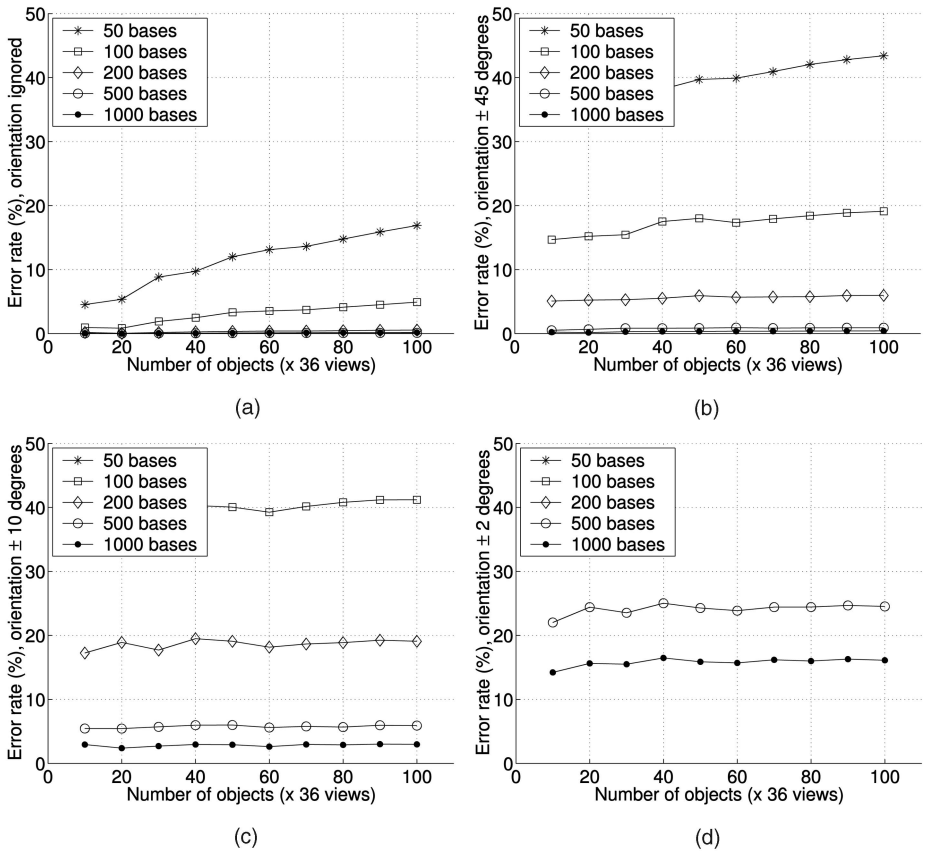

Fig. 11. The recognition rate on data sets of different sizes. Each result is obtained by averaging over 20 subsets randomly taken from the COIL-100 data set. The error in orientation is (a) ignored, (b) within 45 degrees, (c) within 10 degrees, or (d) within 2 degrees.

4.3 On Generalization of the Algorithm with Fewer Training Views

In this experiment, we examine the generalization performance of the algorithm when the number of training views decreases. For this purpose, we followed the experimental setup as in [21], [35]. The number of training views per objects (v) is varied. The rest of the views ($72 - v$) are used for testing.

Table 1 contrasts the generalization performance of the new algorithm and those obtained in [21] and [35], in which two methods, SNoW and support vector machine, have generalization performance guarantees. The result of the new algorithm is comparable to support vector machine with a Gaussian kernel. It is worse than support vector machine with the Kullback-Leibler kernel as reported in [35] in case of four and eight training views per object, and is equivalent for the other cases. In comparison to the other learning algorithms, one observes that the new algorithm does not perform as well when only four training views per object are available. This condition is, however, unfavorable for the new method

TABLE 1
Comparison of the Recognition Rates (%) of Various Learning Algorithms on the COIL-100 Data Set

	one-against-one	one-against-all	Number of views per object			
			36	18	8	4
SNoW_1 [21]	✓		95.81	92.31	85.13	81.46
SNoW_2 [21]		✓	90.52	84.50	81.85	76.00
L-SVM [21]	✓		96.03	91.30	84.80	78.50
G-SVM [35]	✓		99.17	97.04	90.13	75.54
KL-SVM [35]	✓		98.67	98.65	95.22	84.32
NN [21]		✓	98.50	87.54	79.52	74.63
New algorithm		✓	99.83	98.43	90.97	74.25

SNOW_1 *with a one-against-one scheme [21],* SNOW_2 *with a one-against-all scheme [21], support vector machine with a linear kernel (L-SVM) [21], with a Gaussian kernel (G-SVM) [35], and with a Kullback-Leibler kernel (KL-SVM) [35], the nearest neighbor method (NN) reported in [21] and, finally, the new algorithm with 1,000 bases.*

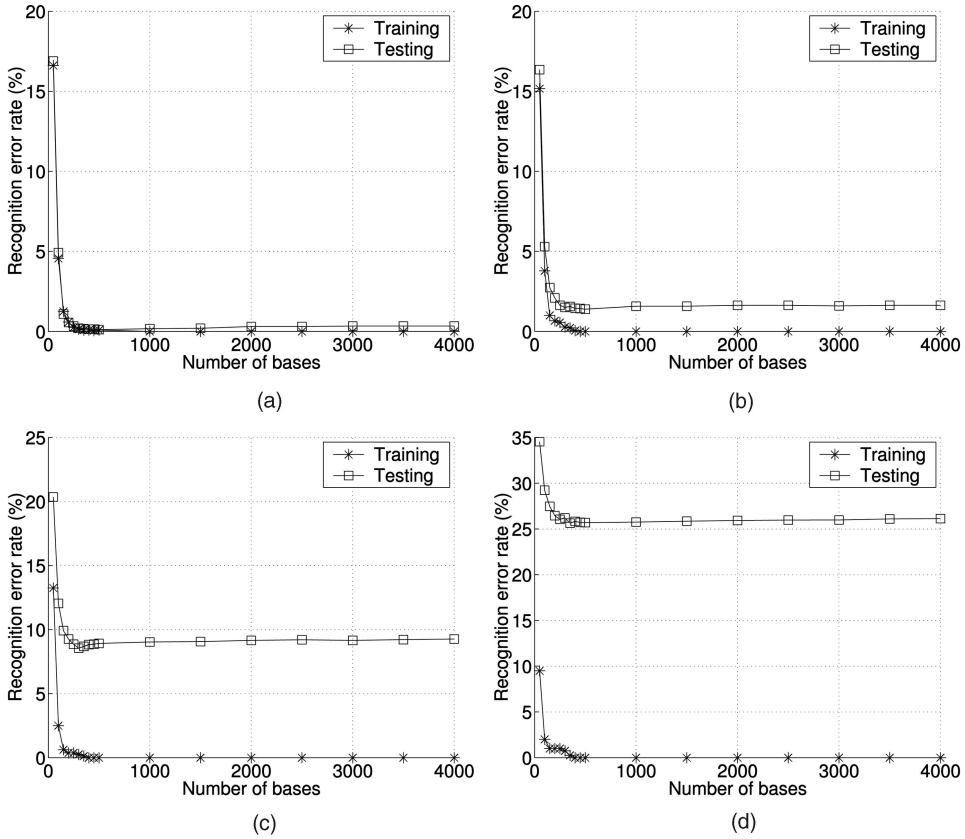

Fig. 12. Generalization performance as a function of number of bases. (a) Thirty-eight training views, (b) 18 training views, (c) eight training views, and (d) four training views per object.

because it learns object representation rather than discrimination, which enables fine recognition. The new algorithm compares favorably against all other methods for 36, 18, and 8 training views per object. In particular, in comparison to SNoW with a one-against-all scheme that learns a representation for each object, the new algorithm outperforms SNoW from 9 to 14 percent of correct recognition for 36, 18, and 8 training views.

Figs. 12a, 12b, 12c, and 12d examine the effect of overfitting for the four cases: 36, 18, 8, and 4 training views per object, respectively. One observes that mild overfitting occurs. The decrease in performance on the test set is insignificant. This is in agreement with experiments of a related class of learning algorithms, namely, boosting methods [4], where it is found that the greedy stage-wise fitting strategy employed in boosting algorithms, which is also the strategy of the matching pursuit algorithm, are resistant to overfitting [5], [24]. In particular, the boosting algorithm $L_2\mathrm{Boosting}$ [2] minimizes the same criterion as the matching pursuit algorithm (25). This reference also provides an explanation for the "overfitting resistance," where the number of bases serves as a regularization parameter. Other theoretical study of generalization is beyond the scope of this paper. The reader is referred to [7], [36] for connection between sparse approximation and support vector machine and other kernel methods.

5 Discussion and Conclusion

This paper addresses the problem of coarse and fine object recognition. We have considered three main issues in this problem, namely, object representation, model learning from examples, and efficient object matching.

We proposed a new representation by sparse function approximation in both spatial dimensions together with the orientation dimension. Thus, the generalization to unseen views is naturally obtained. This is in contrast to previous approaches that use low-dimensional representation by a linear projection, such as principle component analysis, or the view-based representation where the spatial coherence principle is ignored. The new representation is able to exploit a rich amount of a priori information about the object views in general.

We have also presented a solution to the problem of object view matching in a high-dimensional space by polynomial computation. In particular, we showed that the minimization problem involved in recognition phase can be solved by evaluating polynomials of degree six at multipoints. Hence, the computation of the new algorithm equals the PCA approach with six principle components.

We performed experiments on the COIL-100 data set. The results show the high visual quality of the new representation in comparison with PCA. This is because the new representation takes into account the spatial correlation. The experimental results also show that the new method performs as well as the approach using PCA in terms of accuracy.

Experimental results show that the algorithm performs well when eight or more training views are available for each object. In particular, the algorithm exhibits resistance to overfitting, as often observed with the class of greedy stage-wise learning algorithms.

There are three significant advantages of the new method over PCA. First, the storage of sampled points for recognition is not required for the new approach. Instead, a compact object model based on a polynomial representation is employed. Second, the addition of new objects into an existing data set in the new approach is trivial because, unlike PCA retraining, other object models are not needed. Third, in case one has to scan the input image for recognition, the Njet representation is computed only once, as opposed to PCA projection at each location.

Currently, the algorithm handles pose variation in the horizontal plane only. Generalization to other parameters, such as vertical pose variation and illumination direction, is possible by extending the separable bases to the new dimensions. Further experimentation is required to determine the feasibility in higher-dimensional parameter spaces. Another challenge is for the algorithm to be insensitive to some variations while capable of fine recognition in others.

In summary, we propose a new representation for object recognition where the object model can be learned efficiently with a rich amount of a priori information. The new representation also allows fast recognition by polynomial computation. Overall, the method is comparable to the PCA approach in terms of accuracy, and more efficient in terms of storage space and recognition time. Real-time performance is achieved for a data set of 100 objects.

Acknowledgments

This research is supported by the TNO Institute of Applied Physics and MultimediaN.

References

[1] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems,* vol. 16, 2004.

[2] P. Buhlmann and B. Yu, "Boosting with the L_2 Loss: Regression and Classification," *J. Am. Statistical Assoc.,* vol. 98, pp. 324-340, 2001.

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

[4] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *The Annals of Statistics,* vol. 38, no. 2, pp. 337-374, 2000.

[6] J.M. Geusebroek, A.W.M. Smeulders, and J. van de Weijer, "Fast Anisotropic Gauss Filtering," *IEEE Trans. Image Processing,* vol. 12, no. 8, pp. 938-943, 2003.

[7] F. Girosi, "An Equivalence between Sparse Approximation and Support Vector Machines," *Neural Computation,* vol. 10, pp. 1455-1480, 1998.

[8] D.E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms,* vol. 2, third ed., Addison-Wesley, 1997.

[9] J.J. Koenderink, "The Structure of Images," *Biological Cybernetics,* vol. 50, pp. 363-370, 1984.

[10] J.J. Koenderink and A.J. van Doorn, "Representation of Local Geometry in the Visual System," *Biological Cybernetics,* vol. 55, pp. 367-375, 1987.

[11] X. Liu, A. Srivastava, and K. Gallivan, "Optimal Linear Representations of Images for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 662-666, May 2004.

[12] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

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

[14] S. Mukherjee and S.K. Nayar, "Automatic Generation of RBF Networks Using Wavelets," *Pattern Recognition,* vol. 29, pp. 1369-1383, 1996.

[15] H. Murase and S. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *Int'l J. Computer Vision,* vol. 14, pp. 5-24, 1995.

[16] S.A. Nene, S.K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," Technical Report CUCS-006-96, Columbia Univ., 1996.

[17] P.J. Phillips, "Matching Pursuit Filters Applied to Face Identification," *IEEE Trans. Image Processing,* vol. 7, no. 8, pp. 1150-1164, 1998.

[18] T. Poggio and S. Edelman, "A Network that Learns to Recognize Three-Dimensional Objects," *Nature,* vol. 343, pp. 263-266, 1990.

[19] T. Poggio and F. Girosi, "Networks for Approximation and Learning," *Proc. IEEE,* vol. 78, no. 9, pp. 1481-1497, 1990.

[20] M. Pontil and A. Verri, "Support Vector Machines for 3-D Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 6, pp. 637-646, June 1998.

[21] D. Roth, M-H. Yang, and N. Ahuja, "Learning to Recognize Three-Dimensional Objects," *Neural Computation,* vol. 14, pp. 1071-1103, 2002.

[22] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.

[23] F. Schaffalitzky and A. Zisserman, "Viewpoint Invariant Texture Matching and Wide Baseline Stereo," *Proc. Int'l Conf. Computer Vision,* pp. 636-643, 2001.

[24] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics,* vol. 26, no. 5, pp. 1651-1686, 1998.

[25] C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 5, pp. 530-534, May 1997.

[26] B. Schölkopf, A.J. Smola, and K.R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.

[27] A.J. Smola and B. Schölkopf, "From Regularization Operators to Support Vector Kernels," *Advances in Neural Information Processing Systems,* vol. 10, pp. 343-349, 1998.

[28] D.M.J. Tax and R.P.W. Duin, "Support Vector Domain Description," *Pattern Recognition Letters,* vol. 20, nos. 11-13, pp. 1191-1199, 1999.
[29] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.
[30] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.
[31] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part I—Theory," *IEEE Trans. Signal Processing,* vol. 41, no. 2, pp. 821-833, 1993.
[32] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part II—Efficient Design and Applications," *IEEE Trans. Signal Processing,* vol. 41, no. 2, pp. 834-848, 1993.
[33] L.J. van Vliet, I.T. Young, and P.W. Verbeek, "Recursive Gaussian Derivative Filters," *Proc. 14th Int'l Conf. Pattern Recognition,* vol. I, pp. 509-514, 1998.
[34] V.N. Vapnik, *Statistical Learning Theory.* John Wiley and Sons, 1998.
[35] N. Vasconcelos, P. Ho, and P. Moreno, "The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition," *Proc. Eighth European Conf. Computer Vision,* vol. III, pp. 430-441, 2004.
[36] P. Vincent and Y. Bengio, "Kernel Matching Pursuit," *Machine Learning,* vol. 48, no. 1, pp. 165-187, 2002.
[37] W.M. Wells, "Efficient Synthesis of Gaussian Filters by Cascaded Uniform Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 2, pp. 234-239, 1986.

Thang V. Pham received the BSc degree in computer science from the RMIT University in 1998 and the PhD degree from the University of Amsterdam in 2005. He is a postdoctoral fellow at the Intelligent Systems Lab Amsterdam. His current research interests are in object matching, biometrics, and video surveillance.

Arnold W.M. Smeulders graduated from the Technical University of Delft in physics in 1977 (MSc) and in 1982 from Leiden University in medicine (PhD) on the topic of visual pattern analysis. He is the scientific director of the Intelligent Systems Lab Amsterdam of the MultimediaN, the national public-private partnership, and of the ASCI national research school. He participates in the EU-Vision, DELOS, and MUSCLE networks of excellence. He is fellow of the International Association of Pattern Recognition. His research interests are in cognitive vision, content-based image retrieval, learning and tracking, and the picturelanguage question. He has graduated 28 PhD students. The ISIS research group concentrates on theory, practice, and implementation of multimedia information analysis including image databases and computer vision. The group has an extensive record in cooperation with Dutch institutions and industry in the area of multimedia and video analysis. Currently, he is an associated editor of the *International Journal for Computer Vision* as well as the *IEEE Transactions on Multimedia*. He is a senior member of the IEEE.