



Appearance-based object recognition using optimal feature transforms[☆]

Joachim Hornegger*, Heinrich Niemann, Robert Risack

Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg, Martensstr. 3, D-91058 Erlangen, Germany

Received 9 December 1997; accepted 7 January 1999

Abstract

In this paper we discuss and compare different approaches to appearance-based object recognition and pose estimation. Images are considered as high-dimensional feature vectors which are transformed in various manners: we use different types of *non-linear* image-to-image transforms composed with *linear* mappings to reduce the feature dimensions and to beat the curse of dimensionality. The transforms are selected such that special objective functions are optimized and available image data provide some invariance properties. The paper mainly concentrates on the comparison of preprocessing operations combined with different linear projections in the context of appearance-based object recognition. The experimental evaluation provides recognition rates and pose estimation accuracy. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Appearance-based object recognition; Pose estimation; Feature transform; Manifold models; Statistical modeling

1. Introduction

Even up to these days, the efficient and robust recognition and localization of arbitrary 3-D objects in gray-level images is a generally open problem. There exists no unified technique which allows the reliable recognition of arbitrary-shaped objects in cluttered scenes. The available algorithms are mostly restricted to special types of objects. Standard identification and pose estimation techniques use segmentation operations in order to detect geometrical features like corners or lines [1]. The classification itself is based on geometrical relationships between observations and suitable models like geometric representations which use, for instance, wire frame or

CAD models [1]. The main problems of these approaches are due to the automatic generation of models using training samples, and the robust detection of geometric features.

Recently, appearance-based methods have become more and more popular, and are used to deal with object recognition tasks [2,3]. These techniques consider the appearance of objects in sensor signals instead of the reconstruction of geometrical properties. This overcomes quite a lot of problems related to standard approaches as, for example, the geometric modeling of fairly complex objects and the required feature segmentation. Preliminary comparative studies prove the power and the competitiveness of appearance-based approaches to solve recognition problems [2] and suggest further research and experiments. Now well-known and classical pattern recognition algorithms can be used for computer vision purposes: feature selection methods [4,5], feature transforms [4,6], or even more recent results from statistical learning theory [7]. This paper will consider and compare different transforms of high-dimensional feature vectors for object recognition and pose estimation purposes.

*Corresponding author. Tel.: +49-9131-852-7894; fax: +49-9131-303811.

E-mail address: jh@robotics.stanford.edu (J. Hornegger)

[☆]This work was funded partially by the Deutsche Forschungsgemeinschaft (DFG) under grant numbers Ho 1791/2-1 and GRK 244/1-96. The authors are solely responsible for the contents.

2. Contribution and organization of the paper

Appearance-based approaches treat images as feature vectors. If we consider $M \times N$ images, the dimension of associated feature vectors is $m := NM$. Obviously, these high-dimensional feature vectors will not allow the implementation of efficient recognition algorithms [8] and the curse of dimensionality will prohibit classification [5]. For that reason, transforms are necessary to reduce the dimensions of features. Commonly used transforms are the principal component analysis [9–11] or in more recent publications the Fisher transform [12]. Variations of feature vectors dependent on different illumination or pose parameters are modeled by interpolating between different feature vectors and considering the resulting manifolds as object models [9]. These models are called *eigenfaces* or *Fisherfaces* – dependent on the chosen transform.

This work extends the existing appearance-based methods with respect to different *linear transforms* of feature vectors. The considered linear transformations are based on optimization criteria which are basically known from standard pattern recognition literature [8]. In addition, we also consider various types of non-linear preprocessing operations which eliminate, for instance, noise or dependencies of illumination. The main contribution of this paper is therefore twofold and includes

- the comparison of different preprocessing operations, and
- the application of various feature transforms for the reduction of dimensions.

The experimental evaluation provides an extensive characterization of distinct feature transforms. We summarize several methods for improving the recognition rates and pose estimation accuracy of existing algorithms for 3-D object recognition. The final judgement of methods depends on recognition rates and pose estimation errors.

The paper is organized as follows: the next section gives a brief overview of related work and discusses parallels and differences to the main contributions of this paper. Before we introduce mathematical and technical details, we clarify and specify the general formal framework (Section 4). The restriction of already published approaches to the principal component analysis for the reduction of features' dimensions motivates to consider and to compare experimentally different types of linear projections from high-dimensional image into lower-dimensional feature spaces. Feature transforms and the efficient solution of optimization problems related to these projections form the main part of Sections 5 and 6. Instead of using gray-level images as features, some non-linear image transforms, which can be applied within a preprocessing stage, are summarized in Section 5.2. Computational aspects of the involved algorithms are

included in Section 7. The experimental evaluation of introduced concepts is summarized in Section 9: the recognition and pose estimation experiments are evaluated with various combinations of image transforms. The paper ends with a summary, draws some conclusions, and gives several hints to further unsolved research problems concerning appearance-based recognition. Mathematical details, which are less essential for the basic understanding of the proposed techniques, are provided in the appendix.

3. Related work

Appearance-based approaches discussed in the literature are mostly restricted to the principal component analysis to map gray-level images to low-dimensional feature vectors and neglect preprocessing operations [2]. Here, the considered feature transforms are incorporated into an optimization-based framework, which allows geometrical interpretations within the feature space. The mathematical tools which are required for a practical implementation are provided by Murase and Lindenbaum [13]. Fields of application are medical image processing, face recognition [14], or 3-D object recognition and pose estimation. The major problems related to appearance-based methods are due to unknown background objects and occlusion. Classification in cluttered scenes is discussed and sufficiently solved in Ref. [15]. The application of appearance-based methods in the presence of occlusion is considered in Ref. [16], whereas the influence of varying illumination to eigenfaces is experimentally evaluated in Ref. [17]. The authors of Ref. [17] show that 5-D vectors in the eigenspace are sufficient for modeling different lighting conditions. For that reason, this work does not discuss various methods which work with cluttered background and occlusion, but concentrates on the comparison of different optimization criteria with respect to the computation of linear projections.

4. General framework

A digital image \mathbf{f} is mathematically represented by a matrix $[f_{i,j}]_{1 \leq i \leq N, 1 \leq j \leq M}$, where the range of $f_{i,j}$ is determined by the quantization of intensity values. The parameters N and M denote the number of rows and columns. Let us assume, we have K different classes $\Omega_1, \Omega_2, \dots, \Omega_K$ of objects. Examples of different objects are shown in Fig. 1. These objects are assigned to the pattern classes $\Omega_1, \Omega_2, \dots, \Omega_5$. The classification procedure is thus a discrete mapping which assigns an image, showing one of these objects, to the pattern class the present object corresponds to. If we compute the pose parameters, the position and orientation of the object



Fig. 1. Object classes considered in the experiments (Columbia images).

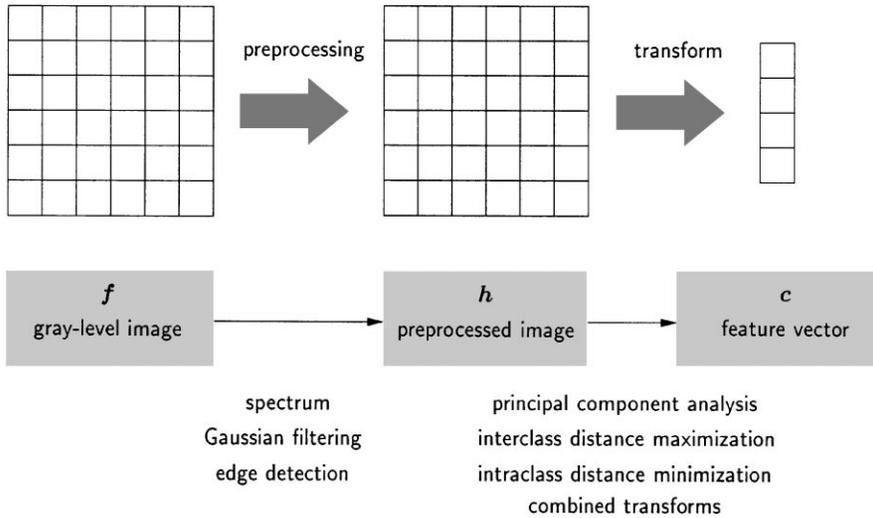


Fig. 2. Structure of feature computations: images and preprocessed sensor data are transformed into feature vectors.

with respect to the world coordinate system are calculated. Usually, there exists no closed-form analytical description of these mappings. Most systems decompose this function into a sequence of – mostly independent – procedures [18].

It is suggesting to consider images $[f_{i,j}]_{1 \leq i \leq N, 1 \leq j \leq M}$ as feature vectors $\mathbf{f} \in \mathbb{R}^m$, where $m = NM$. Because of the geometric nature of objects, however, this is not self-evident. Due to the dimension of $(N \times M)$ -images, classifiers using these high-dimensional features directly will not provide efficient algorithms for several reasons: in high-dimensional vector spaces the definition of similar vectors is somehow difficult, since nearly all vectors are considered to be neighbors [18]. Furthermore, the comparison of vectors is the most often used operation within the classification module and should be as efficient as possible. The use of high-dimensional feature vectors contradicts this requirement [8]. To reduce the data, it is important to select or project features from a gray-level image. Especially for object recognition, traditional methods use the segmentation of (hopefully discriminat- ing) features in the image, like edges or corners of an

object. These features allow the explicit use of geometrical relationships between 3-D models and 2-D observations. The geometry of object transforms and the projection of 3-D models into the 2-D image plane are well-understood and mathematically formalized [19]. But the usage of segmentation results shows some substantial disadvantages, which partially confine their practical use:

- the quality of segmentation highly depends on the chosen algorithm and on illumination conditions as well as selected viewpoints,
- robust, optimal, and reliable detection of features is far from its implementation, and finally
- the huge amount of data reduction induces a decrease of information, which might also decrease the discriminative power of resulting classifiers.

Appearance-based approaches to object recognition, however, prohibit the use of geometry, but the algorithms do not depend on a reliable and accurate detection of points or lines. The computation of features directly from gray-level images can be done by different types of mappings:

1. The first stage might transform the gray-level image into another image which shows special properties. The discrete Fourier transform and spectrum [20], for example, results in features which are invariant with respect to translations of the objects in the image plane and thus reduce the search space with respect to pose parameters. Other filtering operations, like high-pass filters, abate dependencies on illumination. These transforms again lead to large feature vectors for input signals and do not reduce dimensions.
2. For efficient algorithms, however, it is essential that features are projected or selected to obtain small but still discriminating feature vectors. Because selecting the subset of best features is an NP complete problem, only an approximation of the optimal set can be computed within practical applications [18,21].

The transforms which reduce the dimension of features can have two different motivations: one might be the application of some heuristic principles which project the feature vectors and show satisfying recognition results. Here we will consider a restricted class of transforms which have the property to be optimal with respect to objective functions. The applied objective functions

- are based on basic assumptions concerning the distribution of feature vectors,
- are comparatively easy to calculate, and
- induce efficient algorithms for the analytical computation of optimal feature transforms.

Fig. 2 summarizes the general idea the subsequent analysis is based on.

5. Gray-level features, non-linear preprocessing, and feature selection

The classification and pose estimation task is generally formalized as a sequence of mappings which assign an image $\mathbf{f} \in \mathbb{R}^m$ showing one object to a preprocessed image $\mathbf{h} \in \mathbb{R}^m$, then to a feature vector $\mathbf{c} \in \mathbb{R}^n (n \ll m)$, and finally to the class Ω_k of the observed pattern class. Furthermore, related pose parameters, which are defined by three rotations and three components defining the translation vector, have to be computed. The classification and localization of objects crucially depends on *postulates* which are the basic requirements of most pattern recognition algorithms. These postulates – as far as they are relevant for our application – are summarized in the following subsection, and they form the base of all subsequent linear feature transforms.

5.1. Postulates

Usually, feature vectors suitable for 3-D recognition are expected to show a high discriminating power and to

allow reliable classification as well as pose estimation. For that reasons, features have to satisfy basic *postulates* for decision making.

- *Similarity*: objects belonging to the same pattern class show similar feature vectors independent of the associated classes.
- *Distinction*: objects of distinct pattern classes have different feature vectors, which provide a high discriminating power.
- *Smoothness*: small variations in pose or illumination induce small variations in associated features.

Using these basic assumptions for the construction of *good* features, we derive different types of linear feature transforms from high-dimensional into lower-dimensional feature spaces. The basic idea here is to select the transform such that the resulting features are optimal with respect to above postulates. In detail we will consider transforms which:

- maximize the distance of all features among each other independent of pattern classes,
- maximize the distance of features belonging to different pattern classes (*interclass distance*),
- minimize the distance of features belonging to the same pattern class (*intraclass distance*), and
- optimize combinations of the above measures.

However, it should be clear to the reader that linear transforms will not improve the recognition rate of classifiers, even if we choose $m = n$.

5.2. Non-linear image transforms

Before we transform the image matrix into lower-dimensional vectors, we transform sensor data into images, which show distinguished properties [22]. Examples for preprocessing operations are high-pass filters, low-pass filters, the application of the 2-D Fourier transform or the use of segmented images. The application of segmented images gives also a fundamental hint how recognition results are influenced by segmentation. A comparison of gray-level and feature-based identification as well as pose estimation is possible based on this approach (cf. Section 9). Other approaches to object recognition do not allow comparative studies of that kind. Within this work we use the following preprocessing operations: the absolute values of the 2-D discrete Fourier transform (spectrum), the result of Gaussian filtering, the absolute values of second derivatives (Laplace), the edge strength of pixels computed by the operators due to Nevatia–Babu and Sobel. Finally, we also use binary images, where edge pixels are black and the rest white (edge images).

5.3. Optimization-based feature selection and transform

The basic idea of feature transforms is that we are looking for mappings which reduce the dimension of feature vectors and optimize optimality criteria related to the cited postulates of pattern recognition systems. The search for the *optimal transform* requires the restriction to special parametric types of transforms. It seems computationally prohibitive to search for the best transform without any constraints to its properties. For that reason, we restrict the subsequent discussion to linear transforms which map the m -dimensional vectors ${}^i\mathbf{f}$ of the sample set $\omega = \{{}^i\mathbf{f} \in \mathbb{R}^m | i = 1, \dots, r\}$ from \mathbb{R}^m to the n -dimensional features ${}^i\mathbf{c} \in \mathbb{R}^n$. The linear transform is obviously completely characterized by the matrix $\Phi \in \mathbb{R}^{n \times m}$ which maps the m -dimensional preprocessed image vector \mathbf{h} to the n -dimensional feature vector \mathbf{c} , i.e.,

$$\mathbf{c} = \Phi \mathbf{h}. \tag{1}$$

The nm components of the matrix are considered to be the variables of the search process, and thus the search space for suitable transforms is strongly restricted to an nm -dimensional vector space. This makes the search problem tractable and – as we will see in the following subsections – induces optimization problems which can be solved using basic and well-understood techniques of linear algebra.

The computation of the *optimal* linear transform Φ^* makes the use of objective functions necessary which have to be optimized with respect to the parameters, i.e., the components of Φ . In the following, we define different objective functions

$$s_i: \begin{cases} \mathbb{R}^{n \times m} \rightarrow \mathbb{R}, \\ \Phi \mapsto s_i(\Phi), \end{cases} \tag{2}$$

where $i = 1, 2, \dots$, according to the postulates summarized in Section 5.1. A transform Φ_i^* is called optimal with respect to $s_i(\Phi)$, if it holds

$$\Phi_i^* = \underset{\Phi}{\operatorname{argmax}} s_i(\Phi), \tag{3}$$

presupposed s_i has to be maximized, and

$$\Phi_i^* = \underset{\Phi}{\operatorname{argmin}} s_i(\Phi) \tag{4}$$

if s_i has to be minimized. Since scaling of the matrix Φ_i would also affect the value of the objective, the matrices are restricted to those composed of unit length vectors. In the following we use illustrative motivations for different objectives by considering distributions of feature vectors.

5.4. Principal component analysis

The most often used linear transform of this type results from the principal component analysis and is the so-called *Karhunen–Loève transform* (KLT), [10,23].

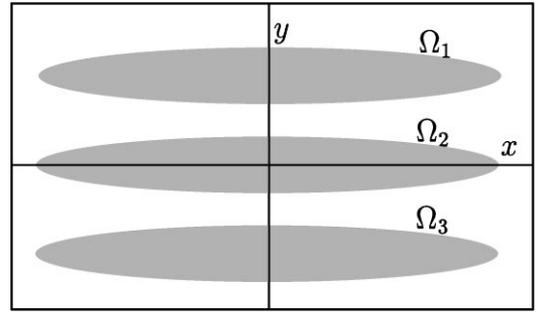


Fig. 3. KLT and the ADIDAS-problem.

The idea of this transform is based on the reduction of the dimension of original image vectors \mathbf{h} using a linear mapping Φ such that the resulting feature vectors \mathbf{c} show pairwise maximum distance. For this transform Φ , the objective function $s_1(\Phi)$ thus is the mean squared distance of all sample feature vectors ${}^i\mathbf{c} = \Phi {}^i\mathbf{h}$ to each other, i.e.,

$$s_1(\Phi) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N ({}^i\mathbf{c} - {}^j\mathbf{c})^T ({}^i\mathbf{c} - {}^j\mathbf{c}). \tag{5}$$

The use of KLT provides both advantages and disadvantages: the computation of the optimal linear transform Φ^* with respect to $s_1(\Phi)$ does not require the classification of sample vectors. Furthermore, feature vectors resulting from KLT allow the reconstruction of images with minimal mean quadratic error [9]. Problems, however, occur if the distribution of features is such that the principal axes of all classes are parallel to each other. A 2-D example, where we project the features onto the x -axis, is shown in Fig. 3 (ADIDAS-problem, Ref. [24]). Obviously, the projected features will allow no discrimination of these classes. For this situation, the optimal linear mapping related to s_1 will not induce discriminating features, whereas the projection on the y -axis would.

This simple example shows that other objective functions than $s_1(\Phi)$ seem to be useful or necessary for reducing the features' dimension and for providing a higher discriminating power.

5.5. Maximizing interclass distance

Another plausible optimization criterion, which does not show the disadvantages of KLT, results from the distinction property. Features of one and the same pattern class should have maximum distance to features of the other pattern classes. In contrast to the KLT, however, this transform requires a classified sample set. The original sample data are partitioned, i.e., $\omega = \dot{\cup} \omega_k$, where $\omega_k = \{{}^i\mathbf{f}_k | i = 1, \dots, r_k\}$ consists of all samples belonging to pattern class Ω_k . Thus the following objective function

can be applied only for those sample sets, where such a labeling is available. Let $i\mathbf{c}_\kappa$ denote the i th sample vector which belongs to class Ω_κ . Of course, the number r_κ of sample data of each class may be different, i.e., $r_\kappa \neq r_\lambda$. The associated objective function based on the above motivated criterion is defined by

$$s_2(\Phi) = \frac{2}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} \frac{1}{r_\kappa r_\lambda} \times \sum_{i=1}^{r_\kappa} \sum_{j=1}^{r_\lambda} (i\mathbf{c}_\kappa - j\mathbf{c}_\lambda)^T (i\mathbf{c}_\kappa - j\mathbf{c}_\lambda), \quad (6)$$

where K denotes the number of pattern classes.

Now we use the classified sample data and also define a criterion which combines the ideas of s_1 and s_2 . For each class Ω_κ we compute the mean vector μ_κ , $\kappa = 1, 2, \dots, K$, and substitute the feature vectors of s_1 by the mean vectors. The objective s_3 is thus defined by

$$s_3(\Phi) = \frac{2}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\mu_\kappa - \mu_\lambda)^T (\mu_\kappa - \mu_\lambda). \quad (7)$$

If we optimize s_3 with respect to the linear transform Φ , the distance between the class centers is maximized. Consequently, the above ADIDAS-problem can be solved using both s_2 and s_3 .

5.6. Minimizing intraclass distance

The objective functions discussed so far maximize distances of features. In order to take the similarity postulate into account, we define an objective which yields a measure for the density of features belonging to the same pattern class. Features of the same pattern class should have a minimum distance and therefore we suggest to minimize the intraclass distance defined by

$$s_4(\Phi) = \frac{1}{K} \sum_{\kappa=1}^K \frac{1}{r_\kappa^2} \sum_{i=1}^{r_\kappa} \sum_{j=1}^{r_\kappa} (i\mathbf{c}_\kappa - j\mathbf{c}_\kappa)^T (i\mathbf{c}_\kappa - j\mathbf{c}_\kappa). \quad (8)$$

The use of this objective function also requires a set of classified training data. The optimal feature transform w.r.t. s_4 results from solving

$$\Phi_4^* = \underset{\Phi}{\operatorname{argmin}} s_4(\Phi). \quad (9)$$

The trivial solution $\Phi = 0$ is excluded, because Φ_4 has to be composed of unit length vectors.

As we see later, the matrix Φ_4^* will be composed by eigenvectors of a kernel matrix $\mathbf{Q}^{(i)}$. In this application the number of sample image vectors $i\mathbf{f}_\kappa$ will be much smaller than the dimension of these vectors. Therefore, the matrix $\mathbf{Q}^{(i)}$ will have a fairly large and therefore non-trivial null space. Projection to this null space will minimize the objective with $s_4 = 0$.

In this space, as Fig. 4 shows, each class will be represented by a single point. If further feature reduction has to be applied, a proper subspace must be selected to

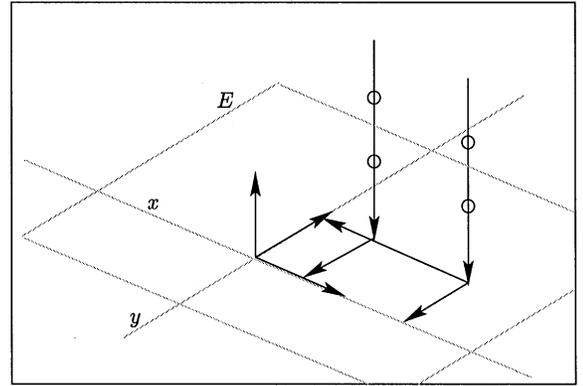


Fig. 4. Projection to null space.

allow good separation of class points. This can be done by another KLT. Due to the high dimension of the null space and numerical problems in evaluating eigenvectors to eigenvalue zero, we only consider combined objectives which maximize the inter- and minimize the intraclass distance at the same time.

5.7. Combination of inter- and intraclass distance

The simplest way of combining the inter- and intraclass distance measure is the use of fractions or linear combinations of s_2 , s_3 , and s_4 , i.e., we could, for instance, define

$$s_5(\Phi) = s_3(\Phi) + \theta s_4(\Phi), \quad (10)$$

$$s_6(\Phi) = s_2(\Phi) + \theta s_4(\Phi) \quad \text{or} \quad s_7(\Phi) = \frac{s_2(\Phi)}{s_4(\Phi)}, \quad (11)$$

and compute linear transforms using these objectives. Herein, the weighting factor θ is a free variable which has to be chosen by the user. The following considerations are restricted to the first definition s_5 . An experimental comparison of s_7 (Fisher transform) and s_1 can be found in Ref. [12].

6. Optimization of objective functions

The optimal linear transforms Φ_i^* , $i = 1, 2, 3, 4, 5$, with respect to the introduced objective functions s_1, s_2, \dots, s_5 are not obvious considering the complicated sums in Eqs. (5)–(8) and (10). Of course, we could start a brute force exhaustive optimization procedure, but concerning above objectives a simplification of the related optimization tasks results from a reorganization of summations and multiplications. Indeed, all objective functions can be written in the following sum of quadratic forms:

$$s_i(\Phi) = 2 \sum_{l=1}^m \varphi_l^T \mathbf{Q}^{(i)} \varphi_l, \quad (12)$$

where the *kernel matrix* $\mathbf{Q}^{(i)}$ corresponding to the i th objective function $s_i(\Phi)$ is implicitly defined, and $\varphi_b, b = 1, 2, \dots, m$, denote the column vectors of the transform Φ , i.e., we obtain $\Phi = (\varphi_1^T, \varphi_2^T, \dots, \varphi_m^T)$ where $\varphi_l \in \mathbb{R}^n$.

The introduction of kernel matrices shows one crucial advantage: the optimization of the introduced objectives s_i is reduced to the computation of eigenvectors and eigenvalues due to the quadratic forms involved in Eq. (12). It is a well-known result from linear algebra that quadratic terms are minimal (resp. maximal) if the vectors φ_l are eigenvectors corresponding to the minimal (resp. maximal) eigenvalues.

The computation of the optimal scatter matrix Φ^* thus proceeds as follows:

1. we compute the eigenvalues and eigenvectors of the involved kernel matrices $\mathbf{Q}^{(i)}$,
2. sort the eigenvalues,
3. define the n rows of the scatter matrix Φ^* to be the n eigenvectors related to the n eigen values; herein we take the n highest eigenvalues, if the objective function has to be maximized, otherwise we use the vectors corresponding to the smallest eigenvalues.

The remaining problem is the explicit definition of kernel matrices, and for the implementation of the proposed feature transforms, however, some numerical aspects and computational considerations are required.

We prove the validity of Eq. (12) exemplary for s_3 in the Appendix by explicitly computing the kernel matrix $\mathbf{Q}^{(3)}$. The technical aspects of computations of other kernel matrices are quite similar and left to the reader. In the following, we present only the final kernel matrices related to above objective functions, since these will be required for formalizing the optimization algorithms.

6.1. Kernel matrix of s_1

Elementary algebraic transforms show that using the objective function s_1 the kernel matrix $\mathbf{Q}^{(1)}$ is simply the covariance matrix of the sample set, which is defined by

$$\mathbf{Q}^{(1)} = \frac{1}{r} \sum_{j=1}^r j\mathbf{f}j\mathbf{f}^T - \left(\frac{1}{r} \sum_{j=1}^r j\mathbf{f} \right)^2 - \frac{1}{r} \sum_{j=1}^r (j\mathbf{f} - \boldsymbol{\mu})(j\mathbf{f} - \boldsymbol{\mu})^T, \quad (13)$$

where $\boldsymbol{\mu}$ denotes the mean vector of the non-classified sample data, i.e.,

$$\boldsymbol{\mu} = \frac{1}{r} \sum_{j=1}^r j\mathbf{f}. \quad (14)$$

6.2. Kernel matrices of s_2 and s_3

Considering the interclass distance and the related objective function s_2 , we get the explicit kernel matrix

$$\begin{aligned} \mathbf{Q}^{(2)} &= \frac{1}{K} \sum_{\kappa=1}^K \frac{1}{r_\kappa} \sum_{j=1}^{r_\kappa} j\mathbf{f}_\kappa j\mathbf{f}_\kappa^T \\ &\quad - \frac{1}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} \left(\frac{1}{r_\kappa r_\lambda} \sum_{j=1}^{r_\kappa} j\mathbf{f}_\kappa \sum_{j=1}^{r_\lambda} j\mathbf{f}_\lambda^T \right. \\ &\quad \left. + \frac{1}{r_\kappa r_\lambda} \sum_{j=1}^{r_\lambda} j\mathbf{f}_\lambda \sum_{j=1}^{r_\kappa} j\mathbf{f}_\kappa^T \right) \\ &= \frac{2}{K(K-1)} \sum_{\kappa=1}^K (\boldsymbol{\mu}_\kappa - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_\kappa - \bar{\boldsymbol{\mu}})^T, \end{aligned} \quad (15)$$

where

$$\boldsymbol{\mu}_\kappa = \frac{1}{r_\kappa} \sum_{j=1}^{r_\kappa} j\mathbf{f}_\kappa \quad \text{and} \quad \bar{\boldsymbol{\mu}} = \frac{1}{K} \sum_{\kappa=1}^K \boldsymbol{\mu}_\kappa. \quad (16)$$

The reorganization of arithmetic operations in s_3 results in the explicit kernel matrix:

$$\mathbf{Q}^{(3)} = \frac{1}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda)(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda)^T. \quad (17)$$

Obviously, the matrix $\mathbf{Q}^{(3)}$ is the kernel matrix of a KLT based on mean vectors instead of feature vectors. In contrast to other kernels, the rank of this matrix is not bounded by the cardinality of the feature set, but by the class number.

6.3. Kernel matrix of s_4

The kernel matrix $\mathbf{Q}^{(4)}$ is given by

$$\mathbf{Q}^{(4)} = \frac{1}{K} \sum_{\kappa=1}^K \frac{1}{r_\kappa} \left(\sum_{j=1}^{r_\kappa} j\mathbf{f}_\kappa j\mathbf{f}_\kappa^T - \boldsymbol{\mu}_\kappa \boldsymbol{\mu}_\kappa^T \right). \quad (18)$$

This result shows a connection between the kernel matrices of s_2, s_3 and s_4 . Indeed the identity

$$\mathbf{Q}^{(4)} = \mathbf{Q}^{(2)} - \mathbf{Q}^{(3)} \quad (19)$$

is valid.

6.4. Kernel matrices of s_5

The explicit kernel matrices for linear combinations of objective functions are trivial. Due to the linear nature of involved mappings the kernel matrices are linear combinations of the kernel matrices of its summands

$$\mathbf{Q}^{(5)} = \mathbf{Q}^{(2)} + \theta \mathbf{Q}^{(4)} = \mathbf{Q}^{(3)} + \tilde{\theta} \mathbf{Q}^{(4)}, \quad (20)$$

where θ and $\tilde{\theta}$ denote the weighting factors of $\mathbf{Q}^{(4)}$.

7. Computational considerations

The direct calculation of the eigenvalues and vectors of $\mathbf{Q}^{(i)}$ is computationally prohibitive due to the storage requirements. If we, for instance, assume images of size 128×128 , which is a quite low resolution for common computer vision purposes, we get $m = 16\,384$ rows and columns for kernel matrices $\mathbf{Q}^{(i)}$. If we suppose, for example, that the entries of the matrix are double valued (i.e., eight byte for each entry), this matrix has storage requirements of about 2 GB. This simple numerical example shows that there is a need for more sophisticated methods to compute the optimal linear transforms related to the objective functions s_1, s_2, s_3, s_4 , and s_5 as well as associated kernel matrices.

7.1. Implicit computation of eigen vectors

The storage requirements can be reduced using a result of singular-value decomposition theory. Let us assume we have to compute the eigenvalues of a matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$ which can be factorized as follows:

$$\mathbf{Q} = \mathbf{F}\mathbf{F}^T, \quad (21)$$

where $\mathbf{F} \in \mathbb{R}^{m \times p}$, $p < m$. As already mentioned, the size of the matrix is intractable for the main memory of our computer. We are interested in computing the eigenvectors and eigenvalues, but a straightforward computation is prohibited. Instead of considering \mathbf{Q} directly, we define according to Murase and Lindenbaum [25] the *implicit matrix*

$$\hat{\mathbf{Q}} = \mathbf{F}^T\mathbf{F}, \quad (22)$$

and observe that there is a remarkable relation between the eigenvalues and eigenvectors of \mathbf{Q} and $\hat{\mathbf{Q}}$.

Let $\hat{\phi}_l$ denote the eigenvectors and $\hat{\lambda}_l$ the eigenvalues of the implicit matrix $\hat{\mathbf{Q}}$. The eigenvectors and values are defined by

$$\hat{\mathbf{Q}}\hat{\phi}_l = \hat{\lambda}_l\hat{\phi}_l; \quad (23)$$

using (22) we thus get

$$\mathbf{F}^T\mathbf{F}\hat{\phi}_l = \hat{\lambda}_l\hat{\phi}_l. \quad (24)$$

In the next step, we multiply both sides by \mathbf{F} , this yields

$$\mathbf{F}\mathbf{F}^T(\mathbf{F}\hat{\phi}_l) = \hat{\lambda}_l(\mathbf{F}\hat{\phi}_l) \quad (25)$$

and thus we get

$$\mathbf{Q}(\mathbf{F}\hat{\phi}_l) = \hat{\lambda}_l(\mathbf{F}\hat{\phi}_l). \quad (26)$$

The last equation shows that each eigenvalue of \mathbf{Q} is also an eigenvalue of $\hat{\mathbf{Q}}$, and the eigenvectors are related by the linear transform \mathbf{F} .

This result proves that the eigenvalues and eigenvectors of the kernel matrices $\mathbf{Q}^{(i)}$ can be computed with low memory requirements presupposed $p \ll m$ and matrices can be factorized in the form of [26]

$$\mathbf{Q}^{(i)} = \mathbf{F}^{(i)}\mathbf{F}^{(i)T}. \quad (27)$$

For that reason, the following subsections will derive the required factorizations of involved kernel matrices.

7.2. Reorganization of $\mathbf{Q}^{(1)}$

The kernel matrix $\mathbf{Q}^{(1)}$ is the covariance matrix of the given sample data, i.e.,

$$\mathbf{Q}^{(1)} = \frac{1}{r} \sum_{i=1}^r (\mathbf{i}\mathbf{f} - \boldsymbol{\mu})(\mathbf{i}\mathbf{f} - \boldsymbol{\mu})^T. \quad (28)$$

We define

$$\mathbf{F}^{(1)} = \sqrt{\frac{2}{r}} (\mathbf{1}\mathbf{f} - \boldsymbol{\mu}, \dots, \mathbf{r}\mathbf{f} - \boldsymbol{\mu}) \in \mathbb{R}^{m \times r}, \quad (29)$$

and it is obvious that

$$\mathbf{Q}^{(1)} = \mathbf{F}^{(1)}\mathbf{F}^{(1)T}. \quad (30)$$

Before we compute the factorization of $\mathbf{Q}^{(2)}$, it is advantageous to consider the decomposition of $\mathbf{Q}^{(3)}$ and $\mathbf{Q}^{(4)}$ (see Eq. (19)). This concrete example shows that there is a trade of between the storage requirements of implicit kernel matrices and the size of sample sets: here we have $r = p$, i.e., the higher r , the higher is the reliability of resulting models. Higher p -values, however, increase the storage requirements.

7.3. Reorganization of $\mathbf{Q}^{(3)}$

Analogous to $\mathbf{Q}^{(1)}$ we get for class centers the decomposition

$$\mathbf{Q}^{(3)} = \mathbf{F}^{(3)}\mathbf{F}^{(3)T}, \quad (31)$$

where

$$\mathbf{F}^{(3)} = \frac{\sqrt{2}}{r_k} (\boldsymbol{\mu}_1 - \bar{\boldsymbol{\mu}}, \dots, \boldsymbol{\mu}_k - \bar{\boldsymbol{\mu}}) \in \mathbb{R}^{m \times K}. \quad (32)$$

The scaling factor $\sqrt{2}/\sqrt{K}$ is important if we use the combined distance measures. Otherwise this factor can be neglected.

7.4. Reorganization of $\mathbf{Q}^{(4)}$

The kernel matrix

$$\mathbf{Q}^{(4)} = \frac{1}{K} \sum_{\kappa=1}^K \frac{1}{r_{\kappa}} \sum_{j=1}^{r_{\kappa}} (\mathbf{i}\mathbf{f}_{\kappa} - \boldsymbol{\mu}_{\kappa})(\mathbf{i}\mathbf{f}_{\kappa} - \boldsymbol{\mu}_{\kappa})^T \quad (33)$$

can also be factorized in the required manner. The similarity to $\mathbf{Q}^{(1)}$ is evident, and analogous to Eq. (29) we define the class-dependent matrices

$$\mathbf{F}_\kappa = \frac{\sqrt{2}}{\sqrt{r_\kappa}} (\mathbf{1}_{\mathbf{f}_\kappa} - \boldsymbol{\mu}_\kappa, \dots, r_\kappa \mathbf{f}_\kappa - \boldsymbol{\mu}_\kappa) \in \mathbb{R}^{m \times r_\kappa}, \quad (34)$$

where $\kappa = 1, 2, \dots, K$. The factor $\sqrt{2}/\sqrt{r_\kappa}$ is necessary, because r_κ varies for different classes Ω_κ . The summation of matrix products can be written using matrix multiplication, i.e.,

$$\begin{aligned} \mathbf{Q}^{(4)} &= \frac{1}{K} \sum_{\kappa=1}^K \mathbf{F}_\kappa \mathbf{F}_\kappa^T = \frac{1}{K} (\mathbf{F}_1, \dots, \mathbf{F}_K) \begin{pmatrix} \mathbf{F}_1^T \\ \vdots \\ \mathbf{F}_K^T \end{pmatrix} \\ &= \frac{1}{K} \mathbf{F}^{(4)} \mathbf{F}^{(4)T}. \end{aligned} \quad (35)$$

7.5. Reorganization of $\mathbf{Q}^{(2)}$

Using Eq. (19) we obviously get

$$\mathbf{Q}^{(2)} = \mathbf{F}^{(2)} \mathbf{F}^{(2)T} = (\mathbf{F}^{(3)}, \mathbf{F}^{(4)}) (\mathbf{F}^{(3)}, \mathbf{F}^{(4)})^T. \quad (36)$$

7.6. Reorganization of $\mathbf{Q}^{(5)}$

The kernel matrix of the combined objective s_5 is

$$\begin{aligned} \mathbf{Q}^{(5)} &= \mathbf{Q}^{(4)} + \theta \mathbf{Q}^{(3)} = (\mathbf{F}^{(4)}, \sqrt{\theta} \mathbf{F}^{(3)}) (\mathbf{F}^{(4)}, \sqrt{\theta} \mathbf{F}^{(3)})^T \\ &= \mathbf{F}^{(5)} \mathbf{F}^{(5)T}. \end{aligned} \quad (37)$$

The weight factor θ has to be positive, because of the square root in the definition of $\mathbf{F}^{(5)}$.

The theoretical part has introduced objective functions which are used to compute optimal linear transforms and which are motivated by the basic postulates of pattern recognition. The required linear mapping is efficiently computed reducing objectives to quadratic forms and solving eigenvalue problems. Related problems with storage requirements of involved computations were solved by the introduction of implicit matrices. In the following section we will compare these transforms and techniques experimentally. Before, it is necessary to define the used models and decision rules the experimental evaluation is based on.

8. Models and decision rules

The classification of objects is based on the introduced features of the eigenspace. Samples of the training set are represented by points within the eigenspace. Due to the fact that lighting conditions, position and orientation of objects vary, feature vectors differ in the eigenspace. Here we distinguish two different types of models:

- manifold models as suggested by Murase and Nayar [9], and
- Gaussian models.

More recent classification methods using support vector machines are omitted [3].

8.1. Manifold models

Objects have several degrees of freedom. Different rotation angles, for instance, result in different feature vectors. It is suggesting to use parametric models (curves) with respect to these variables. Manifold models result from sample feature vectors by interpolation. Fig. 5 shows 3-D feature vectors and the interpolated manifold model. These manifolds are computed for each object class. The class decision is based on the minimization of the distance between an observed feature vector and the manifold models. The parameter vector associated with the manifold point, which has the lowest distance to the observation, defines the pose parameters.

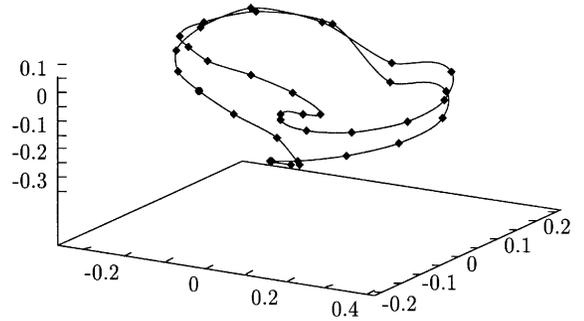


Fig. 5. Example of a manifold model with one degree of freedom (rotation angle) resulting from KLT features. The model corresponds to the second object shown in Fig. 1. The gray-level image is not preprocessed.

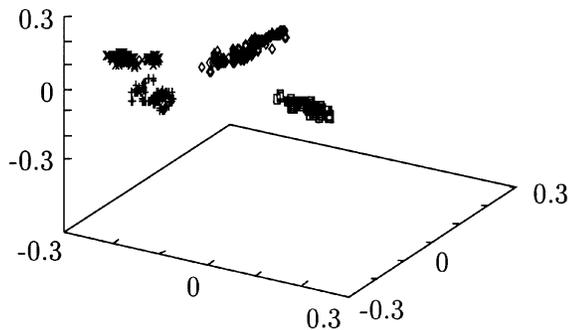


Fig. 6. Clusters of features belonging to four classes shown in Fig. 7. Feature transforms use the combined objective s_5 where $\theta = 10^{-4}$. Herein, the original image matrix was transformed into the Fourier spectrum.

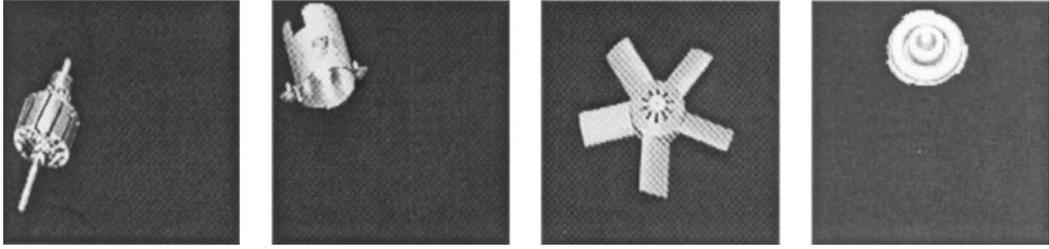


Fig. 7. Industrial objects captured by uncalibrated cameras.

8.2. Gaussian densities

Simpler modeling schemes characterize all sample features assigned to one class by a single probability density function. Here we use multivariate Gaussians for modeling and decide for that class with the highest a-posteriori probability. Of course, these statistical models do not allow for pose estimation. Therefore, these models are especially useful for those applications or training samples, where no pose information is required or available for model generation. Fig. 6 shows four clusters of features belonging to object classes shown in Fig. 7. Within the chosen probabilistic framework, each cluster is characterized by a 3-D Gaussian density.

9. Experimental results

The experimental evaluation provides a comparative empirical study of the introduced transforms Φ_i^* . Before we describe detailed results, we give a brief overview of the experimental set up and the used image data.

9.1. Experimental setup and image data

The experimental evaluation is done on a HP 9000/735 (99 MHz, 124 MIPS) using 128×128 images. Within the experiments we use two different image databases. To provide the capability of comparing the introduced feature transforms with other methods and different approaches to 3-D object recognition, we discuss some experiments using the standard images of the Columbia University image database.¹ We restrict these recognition experiments to the five object classes, which were already shown in Fig. 1. For each object 36 training and 36 test views are available. The images show single 3-D objects with homogeneous background rotated by 5° . Rotations of 0, 10, 20, ..., 360° are used for training. The

recognition experiments run on images showing rotations 5, 15, ..., 355° . Training and test sets are disjoint, and contain images showing objects of varying pose. Occlusion, except self-occlusion, does not occur. For each training and test view the pose parameters, i.e. the single rotation angle, are known. Illumination conditions are constant for all samples.

In addition to these idealized images (homogeneous black background) we also consider industrial parts from a real application using an uncalibrated camera.² We use four objects which are shown in Fig. 7. Of each object 200 different views are available, including also partially occluded objects. Planar rotations and translations as well as lighting are chosen randomly. The set of 2-D views is partitioned into training and test sets of equal cardinalities. In contrast to the above-mentioned image database, the pose parameters are not available.

9.2. Varied parameters and evaluation criterions

The computation of features has several degrees of freedom. Within the experiments, we varied the following parameters:

- dimension of used features,
- different preprocessing methods, and
- different objective functions.

The basic criteria for experimental evaluations are the *recognition rates*, *errors in pose estimates*, and the *runtime*. The used models are both manifold models as suggested in Ref. [9], which also consider the pose parameters, and simple statistical models. Statistical models assume normally distributed feature vectors for each class, and do not use pose information within the training data. The experiments related to pose estimation accuracy are restricted to manifold models and therefore to images of the Columbia database.

¹ See <http://www.cs.columbia.edu/CAVE/coil-20.html>.

² These images are available via URL: <http://www5.informatik.uni-erlangen.de>.

9.3. Pose estimation results

We tested the pose estimation accuracy using manifold models. The considered object transforms are rotations around a single coordinate axis. The features are transformed by linear mappings induced by the discussed objective functions. Table 1 summarizes the obtained errors with respect to rotations around the z-axis of the world coordinate system. Obviously, the best results are achieved by the combined objectives. The overall improvement with respect to the standard principal component analysis, however, is minor.

Table 2 summarizes the errors based on different preprocessing operations and a subsequent principal component analysis in 10 dimensions. If no bijective mappings are used, we expect a reduction of accuracy. Indeed, the experiments show that the best pose estimates result from the immediate use of the gray-level image. The worst accuracy is obtained by using edge images. These examples prove that the appearance-based approach does not provide reliable pose estimates if segmented images are used. Using images containing lines only decreases the accuracy of pose estimates

drastically. Appearance-based pose estimation techniques should not be applied to this type of preprocessed images.

9.4. Recognition results

In the following experiments we compare various preprocessing operations and linear transforms with respect to the resulting recognition rates.

9.4.1. Columbia images

Using the Columbia images (see Fig. 1, five classes) we compare manifold models and statistical models based on simple multivariate Gaussians. The graphs shown in Fig. 8 summarize the recognition results for varying linear transforms and different dimensions of used feature vectors. These experiments prove that the recognition rate is 100% for all transforms, if the dimension of eigen vectors is at least 3 and manifold models are used. For lower-dimensional features, s_3 dominates both with respect to manifold and Gaussian models. The recognition results using combined objectives with different weights are summarized in Fig. 9.

Table 1

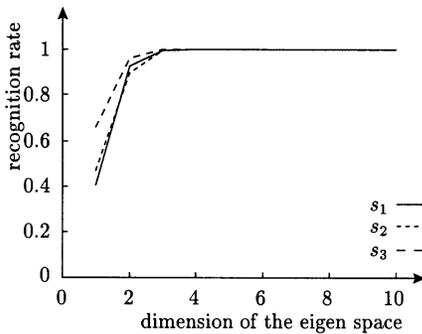
Mean errors and deviations in estimated rotation angles based on 10-D feature vectors for s_1, s_2 and s_5 , and 4-D feature vectors for s_3

Objective function	Mean error (deg)	Standard deviation (deg)
s_1	0.71	0.78
s_2	0.71	0.79
s_3	8.45	43.32
$s_5(\theta = 10^{-4})$	0.69	0.77
$s_5(\theta = 0.1)$	0.70	0.78
$s_5(\theta = 0.5)$	0.67	0.74

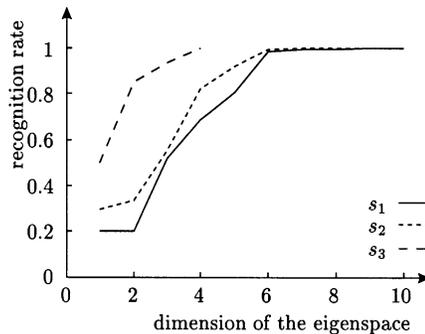
Table 2

Mean error in pose estimates using s_1 as objective function and different preprocessing operations. The chosen dimension of eigenvectors is 10

Filter	Error (deg)
No filtering	0.70
Spectrum	0.96
Gaussian filtering	0.74
Edge detection	14.84
Laplace	3.81
Nevatia	2.96
Sobel	1.73



(a) manifolds



(b) Gaussian distributions

Fig. 8. Comparison of different linear image transforms using s_1, s_2 , and s_3 (Columbia images) and different models: manifold models (left) and Gaussian models (right).

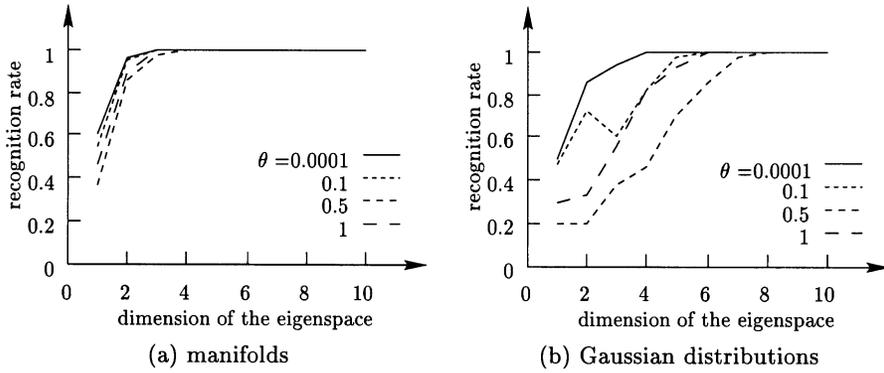


Fig. 9. Combined objective $s_5 = s_3 + \theta s_4$, where $\theta = 10^{-4}, 10^{-1}, \frac{1}{2}, 1$ (Columbia images).

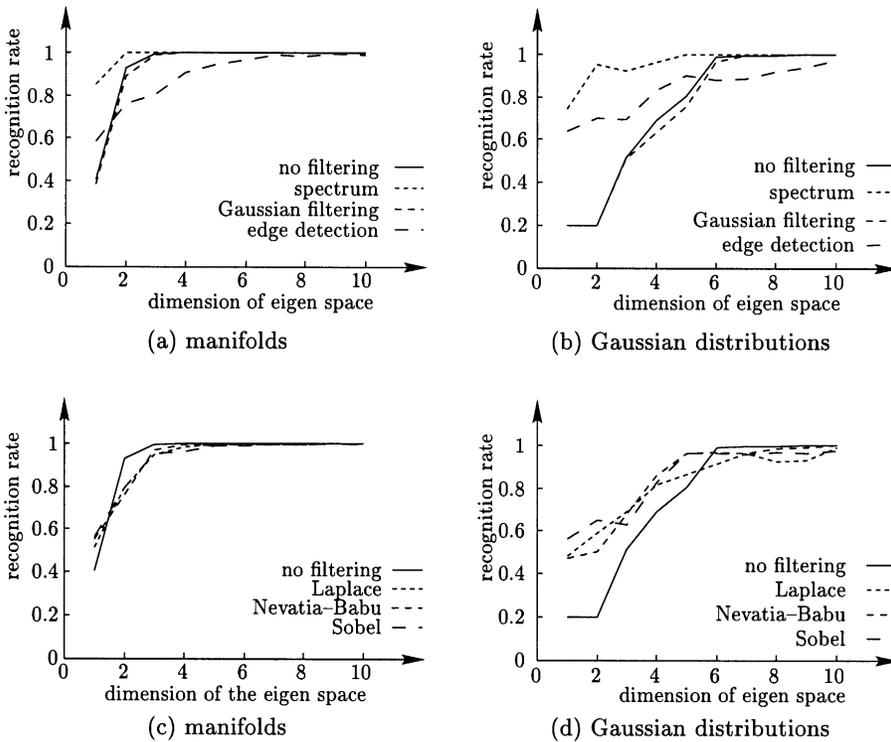


Fig. 10. Recognition rates using s_1 and different preprocessing operations.

Recognition results using different preprocessing operations are summarized in Fig. 10 using objective s_1 and Fig. 11, where we have used s_3 . It is conspicuous that the optimization criterion s_3 combined with the spectrum shows the highest recognition rates independently of the selected model. The main reason for that is the invariance of the spectrum with respect to object translations in the image plane.

All examples show that manifold models provide higher recognition rates than Gaussian models. How-

ever, manifold models require pose information within the training samples, probabilistic models using multivariate Gaussians do not.

9.4.2. Industrial objects

The next experiments use images, where no pose information is available (see Fig. 7, four classes). Therefore, we only consider probabilistic models and analyze the recognition in the presence of occlusion. The recognition rates do also vary with the dimension of eigenvectors. In

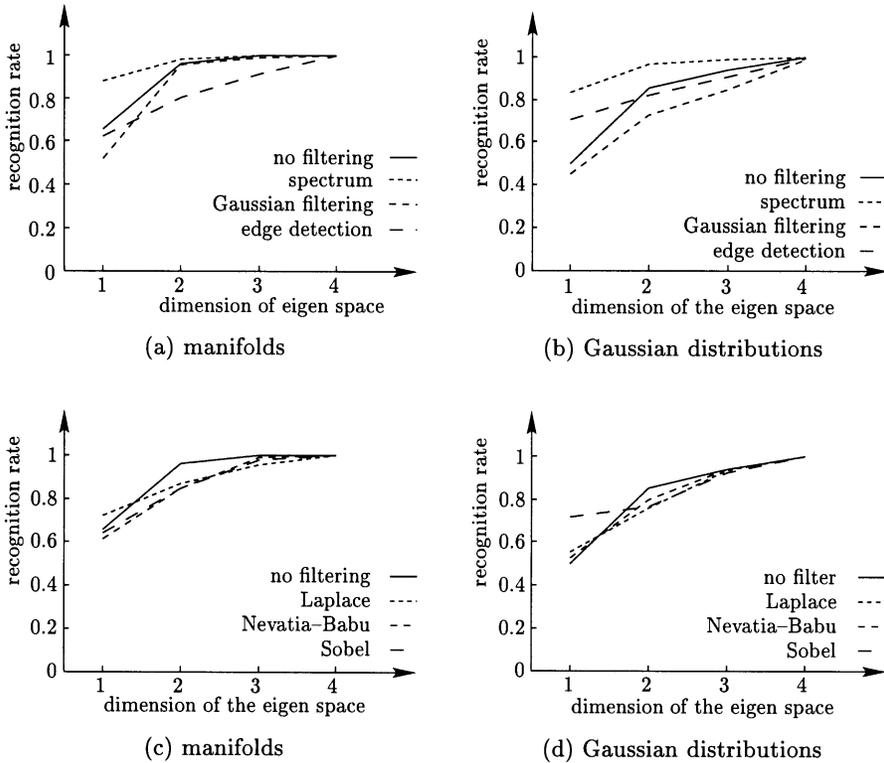


Fig. 11. Recognition rates using s_3 and different preprocessing operations.

Table 3
Recognition rates using images of industrial objects shown in Fig. 7. The dimension of the eigenspace is 20. The linear transform is based on s_1 , and the columns show recognition rates using no preprocessing, Gaussian filtering (GF) and segmenting the background (BG)

Class	No occlusion			Occlusion		
	—	GF	BG	—	GF	BG
Ω_1	25	41	40	10	10	10
Ω_2	87	85	80	30	20	30
Ω_3	1	0	3	0	0	0
Ω_4	57	48	64	80	80	80
Average	43	44	47	30	28	30

contrast to previous experiments, we restrict the dimension of the eigenspace to 10 and 20. Table 3 shows the low recognition rates for the industrial objects based on linear transforms using s_1 , even if different preprocessing operations are used. Obviously, partially occluded objects cannot be classified using the high-dimensional eigenvectors and this approach. Recognition rates are

comparable to random estimates. The use of linear transforms introduced above also does not essentially improve the recognition results. Tables 4 and 5 also show the curse of dimensionality: an increasing dimension of feature vectors does not necessarily increase recognition results. The main reason for low recognition rates is the presence of translations in the image plane. If we detect the object and consider only pixels belonging to the object, we observe a remarkable improvement of recognition rates. We get 100% even in the presence of occlusion. Therefore, we use the spectrum of images (absolute values of the 2-D Fourier transform), which is known to be invariant with respect to translations. These experiments show that rotations do not influence the accuracy of recognition in contrast to translations. The segmentation of objects, i.e. the bi-partition of image points into object and background pixels, or usage of Fourier transform for object classification is advantageous for recognition, if no pose information is available within the training data.

9.5. Run time

The run time behavior of the complete system is summarized in Tables 6–9. All numbers are based on the

Table 4
Recognition rates based on 10-dimensional eigen vectors. The images are preprocessed such that background and object pixels are separated

Class	No occlusion				Occlusion			
	s_1	s_2	$s_5(0.5)$	$s_5(10^{-4})$	s_1	s_2	$s_5(0.5)$	$s_5(10^{-4})$
Ω_1	90	84	88	61	30	30	30	40
Ω_2	99	98	98	88	100	100	100	100
Ω_3	92	93	92	62	60	60	60	40
Ω_4	87	87	87	62	100	100	100	100
Average	92	90	91	68	72	72	72	70

Table 5
Recognition rates based on 20-dimensional eigenvectors. The images are preprocessed such that background and object pixels are separated

Class	No occlusion				Occlusion			
	s_1	s_2	$s_5(0.5)$	$s_5(10^{-4})$	s_1	s_2	$s_5(0.5)$	$s_5(10^{-4})$
Ω_1	40	35	47	61	10	10	10	30
Ω_2	80	80	83	88	30	30	30	40
Ω_3	3	9	25	62	0	0	0	0
Ω_4	64	65	63	62	80	90	100	80
Average	47	47	54	68	30	33	35	38

Table 6
Recognition rates using 20-dimensional feature vectors

Method	Recognition rate (%)	
	No occlusion	Occlusion
Non-invariant features	47	30
Separated object/background pixels	99	73
Spectrum	100	100

Columbia image database including 180 training images of size 128×128 . Table 6 shows the time required for training using all images of the training set. Most of the time is obviously required for the computation of eigenvectors. Table 8 shows the relation between the time for classification and the dimension of the eigenspace.

10. Summary and conclusions

Standard linear feature transforms which are broadly used in pattern recognition and speech recognition are

Table 7
Run time of the learning stage dependent on the dimension of used eigenspaces (180 images)

Dimension of eigenvectors	Computation of eigenvectors (min:s)	Training (ms)	
		Gauss	Manifold
5	3:34	< 10	< 10
10	3:55	< 10	< 10
20	4:18	40	< 10

Table 8
Run time of eigenvalue computations (10-dimensional eigenspace) dependent on the number of training images

Number of images	Time [min:s]
45	0:38
90	1:37
135	2:34
180	3:55

Table 9
Run time of the classification module. The images are represented as vectors

Dimension of eigenspace	Projection	Classification	
		Gauss (ms)	Manifold (ms)
5	30	< 10	560
10	60	< 10	650
20	110	30	790

successfully applied to solve object recognition and pose estimation problems in the field of 3-D computer vision using gray-level images.

This paper has summarized various objective functions for the computation of optimal feature transforms: the principal component analysis, the interclass distance, the intraclass distance, and various combinations. We have shown how the associated optimization problems are reduced to the computation of eigenvectors. A two-stage re-organization of considered objective functions leads to computational practical solutions:

1. the transform of the objective functions into sums of quadratic forms that reduces the optimization problem to the computation of eigenvectors, and
2. the factorization of kernel matrices into products of matrices and its transposes which induces lower storage requirements for computing eigenvalues and eigenvectors.

The experimental evaluation provides a comparison of new types feature transforms. Based on a standard image database, we prove empirically that the best pose estimation results are provided by a transform which maximizes a combination of intra- and interclass distances. The recognition results show highest accuracy, if the distance of class-specific mean vectors is maximized. Dependent on the selected dimension of feature vectors we have shown that a dimension of 4 already leads to recognition results of 100% correctness. Instead of manifolds, we have also tested the recognition rates using the assumption of normally distributed feature vectors. Using spectral features which are invariant to translations in the image plane we observed also recognition rates of 100% using industrial objects, where the training set includes *no* pose information.

Considering these results, we conclude that appearance-based object recognition systems can compete with standard geometrical approaches both with respect to recognition rates and run time behavior. The introduction of implicit kernel matrices has reduced storage requirements.

The problems which are not yet solved sufficiently are the explicit modeling of occlusion, the analysis of

multiple object scenes, and the construction of object models in the presence of background features. The application of the considered transforms to classify and localize objects with heterogeneous background is straightforward using the hierarchical framework introduced in Ref. [15].

Acknowledgements

The authors gratefully acknowledge S. Nene, H. Murase and K. Nayar for the friendly admission to use their Software Library for Appearance Matching (SLAM).

Appendix A

We consider the criterion s_3 and get the following quadratic form:

$$\begin{aligned}
 s_3 &= \frac{2}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\Phi \mu_{\kappa} - \Phi \mu_{\lambda})^T (\Phi \mu_{\kappa} - \Phi \mu_{\lambda}) \\
 &= \frac{2}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\mu_{\kappa} - \mu_{\lambda})^T \Phi^T \Phi (\mu_{\kappa} - \mu_{\lambda}) \\
 &= \frac{2}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} \text{tr} (\Phi^T \Phi (\mu_{\kappa} - \mu_{\lambda}) (\mu_{\kappa} - \mu_{\lambda})^T) \\
 &= 2 \text{tr} \left[\Phi^T \Phi \left(\frac{1}{K(K-1)} \right. \right. \\
 &\quad \left. \left. \times \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\mu_{\kappa} - \mu_{\lambda}) (\mu_{\kappa} - \mu_{\lambda})^T \right) \right] \\
 &= 2 \text{tr} [\Phi^T \Phi \mathbf{Q}^{(3)}] \\
 &= 2 \sum_{i=1}^r \varphi_i^T \mathbf{Q}^{(3)} \varphi_i. \tag{38}
 \end{aligned}$$

The kernel matrix for this case thus is [8]

$$\mathbf{Q}^{(3)} = \frac{1}{K(K-1)} \sum_{\kappa=2}^K \sum_{\lambda=1}^{\kappa-1} (\mu_{\kappa} - \mu_{\lambda}) (\mu_{\kappa} - \mu_{\lambda})^T. \tag{39}$$

References

- [1] A.K. Jain, P.J. Flynn (Eds.), Three-Dimensional Object Recognition Systems, Elsevier, Amsterdam, 1993.
- [2] J. Ponce, Zisserman, M. Hebert (Eds.), Object Representation in Computer Vision, Lecture Notes in Computer Science, vol. 1144, Springer, Heidelberg, 1996.
- [3] M. Pontil, A. Verri, Support vector machines for 3d object recognition, IEEE Trans. Pattern Anal. Machine Intell. (PAMI) 20 (1998) 637–646.
- [4] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.

- [5] C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- [6] E. Trucco, A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall, New York, 1998.
- [7] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Heidelberg, 1996.
- [8] H. Niemann, *Klassifikation von Mustern*, Springer, Heidelberg, 1983.
- [9] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, *Int. J. Comput. Vision* 14 (1) (1995) 5–24.
- [10] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, *Ann. Acad. Sci. Fenn. Ser. AI* (1947) 37.
- [11] Y.T. Chien, K.S. Fu, Selection and ordering of feature observations in a pattern recognition system, *Inform. Control* 12 (1968) 395–414.
- [12] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Trans. Pattern Anal. Machine Intell.* 19 (7) (1997) 711–720.
- [13] H. Murase, M. Lindenbaum, Spatial temporal adaptive method for partial eigenstructure decomposition of large images, Tech. Report 6527, Nippon Telegraph and Telephone Corporation, March 1992.
- [14] M. Turk, A. Pentland, Eigenfaces for recognition, *J Cognitive Neurosc.* 3 (1) (1991) 71–86.
- [15] H. Murase, S.K. Nayar, Detection of 3D objects in cluttered scenes using hierarchical eigenspace, *Pattern Recognition Lett.* 18 (5) (1997) 375–384.
- [16] H. Bischof, A. Leonardis, Robust recovery of eigenimages in the presence of outliers and occlusions, *Int. J. Comput. Inform. Technol.* 4 (1) (1996) 25–38.
- [17] R. Epstein, P.W. Hallinan, A.L. Yuille, 5 ± 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models, *Proceedings of IEEE Workshop on Physics Based Modeling in Computer Vision*, Boston, June 1995, pp. 108–116.
- [18] H. Niemann, *Pattern Analysis and Understanding*, Springer Series in Information Sciences, vol. 4, Springer, Heidelberg, 1990.
- [19] O.D. Faugeras, New steps toward a flexible 3d-vision system for robotics, *Proceedings of the Eighth International Conference on Pattern Recognition*, Montreal, 1987, pp. 796–805.
- [20] A.V. Oppenheim, R.W. Schaffer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [21] A.K. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, *IEEE Trans. Pattern Anal. Machine Intell.* 19 (2) (1997) 153–158.
- [22] W.K. Pratt, *The PIKS Foundation C Programmers Guide*, Manning, Greenwich, 1995.
- [23] K. Karhunen, I. Selin (trans.), *On linear methods in probability theory*. Translation to [10] T-131, The Rand Corporation, August 1960.
- [24] E.G. Schukat–Talamazzini, *Automatische Spracherkennung*, Vieweg, Wiesbaden, 1995.
- [25] H. Murase, M. Lindenbaum, Spatial temporal adaptive method for partial eigenstructure decomposition of large images, *IEEE Trans. Image Process.* 4 (5) (1995) 620–629.
- [26] H. Murakami, V. Kumar, Efficient calculation of primary images from a set of images, *IEEE Trans. Pattern Anal. Machine Intell.* 4 (5) (1982) 511–515.

About the Author—JOACHIM HORNEGGER graduated (1992) and received his Ph.D. degree in Computer Science (1996) at the Universität Erlangen-Nürnberg, Germany, for his work on statistical object recognition. Joachim Hornegger was research and teaching associate at Universität Erlangen-Nürnberg, a visiting scientist at the Technion, Israel, at the Massachusetts Institute of Technology, USA, and a visiting scholar at Stanford University, USA. His major research interests are 3D computer vision, 3D object recognition and statistical methods applied to image analysis problems. Joachim has taught computer vision and pattern recognition at the Universität Erlangen-Nürnberg, Germany, at the University of Seville, Spain, and at Stanford University, USA. He is the coauthor of a monography on pattern recognition and image processing in C + +. Currently Joachim is with Siemens Medical Systems working on 3-D reconstruction and a lecturer at the Universität Erlangen-Nürnberg, Germany.

About the Author—HEINRICH NIEMANN obtained the degree of Dipl.-Ing. in Electrical Engineering and Dr.-Ing. at Technical University Hannover in 1966 and 1969, respectively. During 1966/67 he was a graduate student at the University of Illinois, Urbana. From 1967 to 1972 he was with the Fraunhofer Institut für Informationsverarbeitung in Technik und Biologie, Karlsruhe, working in the field of pattern recognition and biological cybernetics. During 1973–1975 he was teaching at Fachhochschule Giessen in the department of Electrical Engineering. Since 1975 he has been Professor of Computer Science at the University of Erlangen-Nürnberg, since 1988 he is also head of the research group “Knowledge Processing” at the Bavarian Research Institute for Knowledge Based Systems (FORWISS) where he also was on the board of directors for six years. During 1979–1981 he was dean of the Engineering faculty of the University, in 1982 he was program chairman of the 6. International Conference on Pattern Recognition in München, Germany, in 1987 he was director of the NATO Advanced Study Institute on “Recent Advances in Speech Understanding and Dialog Systems”, in 1992 he was Program Chairman of the conference track on “Computer Vision and Applications” at the 11. International Conference on Pattern Recognition in The Hague, The Netherlands, and he was program co-chairman at the International Conference on Acoustics, Speech, and Signal Processing 1997 in München. His fields of research are speech and image understanding and the application of artificial intelligence techniques in these fields. He is on the editorial board of *Signal Processing*, *Pattern Recognition Letters*, *Pattern Recognition and Image Analysis*, and *Journal of Computing and Information Technology*. He is the author or coauthor of 6 books and about 250 journal and conference contributions as well as editor or coeditor of 23 proceedings and special issues. He is a member of ESCA, EURASIP, GI, IEEE, and VDE.

About the Author—ROBERT RISACK received his M.Sc. degree in Computer Science (Diplom Informatiker) from the Universität Erlangen-Nürnberg, Germany. Since May 1997 Robert has been a Ph.D. student at the Fraunhofer Institut für Informations- und Datenverarbeitung, Karlsruhe. He is working on the design and implementation of computer vision systems.