Ellipsoidal bias in learning appearance-based recognition functions

Josef Pauli and Gerald Sommer

Christian-Albrechts-Universität zu Kiel,
Institut für Informatik und Praktische Mathematik,
Preußerstraße 1–9, D-24105 Kiel,
www.ks.informatik.uni-kiel.de/~jpa{~gs},
jpa{gs}@ks.informatik.uni-kiel.de

Abstract. We present an approach for learning appearance-based recognition functions, whose novelty is the sparseness of necessary training views, the exploitation of constraints between the views, and a special treatment of discriminative views. These characteristics reflect the trade-off between efficiency, invariance, and discriminability of recognition functions. The technological foundation for making adequate compromises is a combined use of principal component analysis (PCA) and Gaussian basis function networks (GBFN). In contrast to usual applications we utilize PCA for an ellipsoidal interpolation (instead of approximation) of a small set of seed views. The ellipsoid enforces several biases which are useful for regularizing the process of learning. In order to control the discriminability between target and counter objects the coarse manifold must be fine-tuned locally. This is obtained by dynamically installing weighted Gaussian basis functions for discriminative views. Using this approach, recognition functions can be learned for objects under varying viewing angle and/or distance. Experiments in numerous real-world applications showed impressive recognition rates.

1 Introduction

Famous physiologists (e.g. Hermann von Helmholtz) insisted on the central role of learning in visual processes [2]. For example, object recognition is based on adequate a priori information which can be acquired by learning in the actual environment. The statistical method of principal component analysis (PCA) has been used frequently for this purpose, e.g. by Turk and Pentland for recognition of faces [8], or by Murase and Nayar for recognition of arbitrary objects [3]. The most serious problem in using PCA for recognition is the daring assumption of *one* multi-dimensional Gaussian distribution of the vector population, which is not true in many realistic applications. Consequently, approaches of *nonlinear dimension reduction* have been developed, in which the input data are clustered and local PCA is applied for each cluster, respectively. The resulting architecture is a Gaussian basis function network which approximates the manifold more accurately by a combination of local multi-dimensional Gaussian distributions [6]. However, large numbers of training views are required for approximating

the Gaussians. Furthermore, the description length has increased which makes the recognition function less efficient in application. Our concern is to reduce the effort of training and description by discovering and incorporating *invariances* among the set of object views.[1] Apart from characteristics of efficiency and invariance, the major criterion for evaluating a recognition function is the discriminability, i.e. the capability to discriminate between the target object and counter objects. Similar views stemming from different objects are of special interest for learning reliable recognition functions. This principle is fundamental for the methodology of *support vector machines*. At the border between neighboring classes a small set of critical elements must be determined from which to construct the decision boundary [9]. Although the border elements play a significant role it would be advantageous to additionally incorporate a statistical approximation of the distribution of training samples. Our approach takes special care for counter (critical) views but also approximates the distribution of all training views.

2 Foundation for object recognition

For the purpose of object recognition we construct an implicit function f^{im}, which approximates the manifold of appearance patterns under different viewing conditions.

$$f^{im}(\boldsymbol{A}, \boldsymbol{Z}) = 0 \tag{1}$$

Parameter vector $\boldsymbol{A}$ specifies a certain version subject to the type of the function, and measurement vector $\boldsymbol{Z}$ is the representation of an appearance pattern. In terms of the *Lie group theory of invariance* [4], the manifold of realizations of $\boldsymbol{Z}$ is the *orbit* of a *generator function* whose *invariant features* are represented in $\boldsymbol{A}$. Function f^{im} must be learned such that equation (1) holds more or less for patterns of the target object and clearly not holds for patterns of counter situations. Solely small deviations from the ideal orbit are accepted for target patterns and large deviations are expected for counter patterns. The degree of deviation is controlled by a parameter ψ.

$$\mid f^{im}(\boldsymbol{A}, \boldsymbol{Z}) \mid \leq \psi \tag{2}$$

The function f^{im} can be squared and transformed by an exponential function in order to obtain a value in the *unit interval*.

$$f^{Gi}(\boldsymbol{A}, \boldsymbol{Z}) := exp\left(-f^{im}(\boldsymbol{A}, \boldsymbol{Z})^2\right) \tag{3}$$

If function f^{Gi} yields value 0, then vector $\boldsymbol{Z}$ is infinite far away from the orbit, else if function f^{Gi} yields value 1, then vector $\boldsymbol{Z}$ belongs to the orbit. Equation (1) can be replaced equivalently by

[1] Our work in [5] replaces the concept of *invariance* realistically by the concept of *compatibility*.

$$f^{Gi}(\boldsymbol{A}, \boldsymbol{Z}) = 1 \tag{4}$$

For reasons of consistency, we also use the exponential function to transform parameter ψ into ζ in order to obtain a threshold for *proximities*, i.e. $\zeta := exp\left(-\psi^2\right)$. With this transformations, we replace equation (2) equivalently by

$$f^{Gi}(\boldsymbol{A}, \boldsymbol{Z}) \geq \zeta \tag{5}$$

3 Concept of canonical frames and ellipsoid basis function

Learning a recognition function requires the estimation of parameter vector $\boldsymbol{A}$ based on measurement vectors $\boldsymbol{Z}$. However, for an appearance-based approach to recognition the input space is high-dimensional as it consists of patterns, and frequently, also the parameter vector is high-dimensional. Consequently, first we project the high-dimensional space in a low-dimensional subspace (so-called *canonical frame*), and then we do the learning therein. The construction of canonical frames is based on so-called *seed images* which are representative for the object. The learning procedure is based on a *coarse-to-fine strategy* in which the *coarse* part does the subspace construction and is responsible for global aspects in the manifold of patterns. The subsequent refinement step treats local aspects in the manifold by taking more specific object views or counter situations into account, i.e. so-called *validation views*.

We impose three requirements to canonical frames. First, the implicit function will be defined in the canonical frame and should have a simpler description than in the original frame. Second, equation (4) must hold perfectly for all seed images which are represented as vector $\boldsymbol{Z}$, respectively. Therefore, the parameters $\boldsymbol{A}$ are invariant features of the set of all seed images. Third, the implicit function should consider generalization biases as treated in the theory of Machine Learning [10, pp. 349-363]. For example, according to the *enlarge-set bias* and the *close-interval bias*, the implicit function must respond continuous around the seed vectors and must respond nearly invariant along certain courses between successive seed images (in the space of images). The *minimal-risk bias* avoids hazardous decisions by preferring low degrees of generalization.

An appropriate canonical frame together with a definition of the implicit function can be constructed by principal component analysis (PCA). Remarkably, we use PCA for *interpolating* a small set of seed images by a hyper-ellipsoid function. As all seed images are equal significant we avoid approximations in order not to waste essential information. Based on the covariance matrix of the seed images, we take the normalized eigenvectors as basis vectors. The representation of a seed image in the canonical frame is by *Karhunen-Loéve expansion (KLE)*. Implicit function f^{im} is defined as a hyper-ellipsoid in normal form with the half-lengths of the ellipsoid axes defined dependent on the eigenvalues of the covariance matrix, respectively. As a result, the seed vectors are located on the orbit of this hyper-ellipsoid, and invariants are based on the half-lengths of the ellipsoid axes.

4 Construction of canonical frame and ellipsoid basis function

Let $\Omega := \{\boldsymbol{X}_i^s | \boldsymbol{X}_i^s \in R^m; i = 1, \cdots, I\}$ be the vectors representing the seed images of an object. Based on the mean vector $\boldsymbol{X}^c$ the *covariance matrix* $\mathcal{C}$ is obtained by

$$\mathcal{C} := \frac{1}{I} \cdot \mathcal{M} \cdot \mathcal{M}^T \; ; \quad \mathcal{M} := (\boldsymbol{X}_1^s - \boldsymbol{X}^c, \cdots, \boldsymbol{X}_I^s - \boldsymbol{X}^c) \tag{6}$$

We compute the eigenvectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_I$ and eigenvalues $\lambda_1, \cdots, \lambda_I$ (in decreasing order). The lowest eigenvalue is equal to 0 and therefore the number of relevant eigenvectors is at most $(I-1)$ (this statement can be proved easily). The I original vectors of Ω can be represented in a coordinate system which is defined by just $(I-1)$ eigenvectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{I-1}$ and the origin of the system. KLE defines the projection/representation of a vector $\boldsymbol{X}$ in the $(I-1)$-dimensional eigenspace.

$$\hat{X} := (\hat{x}_1, \cdots, \hat{x}_{I-1})^T := (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{I-1})^T \cdot (\boldsymbol{X} - \boldsymbol{X}^c) \tag{7}$$

Based on PCA and KLE we introduce the $(I-1)$-dimensional hyper-ellipsoid function.

$$f^{im}(\boldsymbol{A}, \boldsymbol{Z}) := \left(\sum_{l=1}^{I-1} \frac{\hat{x}_l^2}{\kappa_l^2} \right) - 1 \tag{8}$$

Measurement vector $\boldsymbol{Z} := \hat{X}$ is defined according to equation (7). Parameter vector $\boldsymbol{A} := (\kappa_1, \cdots, \kappa_{I-1})^T$ contains parameters κ_l, which are taken as half-lengths of the ellipsoid axes in normal form and are defined as

$$\kappa_l := \sqrt{(I-1) \cdot \lambda_l} \tag{9}$$

For the special case of assigning the KLE-represented seed vectors to $\boldsymbol{Z}$, respectively, we can prove that equation (1) holds perfectly for all seed vectors.[2] All seed vectors are located on the orbit of the hyper-ellipsoid defined above, and therefore, the half-lengths are an *invariant description* for the set of seed vectors. The question of interest is twofold, (i) why use ellipsoid interpolation, and (ii) why use PCA for constructing the ellipsoid?

Ad (i): The hyper-ellipsoid considers the enlarge-set and the close-interval biases, as demonstrated in the following. Let us assume three points $\boldsymbol{X}_1^s, \boldsymbol{X}_2^s, \boldsymbol{X}_3^s$ in $2D$, visualized as black disks in the left diagram of Figure 1. The 2D ellipse through the points is constructed by PCA. The right diagram of Figure 1 shows a constant value 1 when applying function f^{Gi} (as defined in equations (8) and (3)) to all orbit points of the ellipse. Therefore the generalization comprises all points on the ellipse (close-interval bias). The degree of generalization can be increased furthermore by considering the threshold ζ and accepting for f^{Gi} small

[2] The proof is given in the Appendix.

deviations from 1. The relevant manifold of points is enlarged, as shown by the dotted band around the ellipse in Figure 1 (enlarge-set bias).

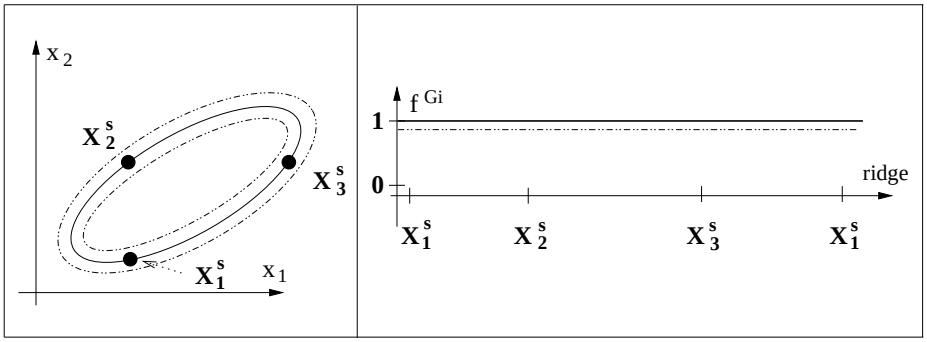

Fig. 1. (Left) Input space with three particular points from which a 2D ellipse is defined by PCA, small deviations from this ellipse are constrained by an inner and an outer ellipse; (Right) Result of function f^{Gi} along the ellipse, which is constant 1, accepted deviations are indicated by horizontal lines with offset $\mp\zeta$.

Ad (ii): In general, more than I points are necessary for fitting a unique $(I-1)$-dimensional hyper-ellipsoid. PCA determines the first principal axis by maximizing the variances which are obtained by an orthogonal projection of the sample points on hypothetical axes, respectively. Actually, this is the constraint which makes the fitting unique. Figure 2 shows two examples of ellipses fitting the same set of three points, the left one was determined by PCA, and the right one was fitted manually. As expected, the variance on the right is lower than on the left, which is measured along the dashed axes, respectively. Exemplary, it is also observed in the figure that the maximum variance (on the left) implies a minimal size of the ellipsoid. The size of the ellipsoid manifold correlates with the degree of generalization, and therefore, PCA produces moderate generalizations by avoiding large ellipsoids (minimal-risk bias).

5 Fine approximation based on validation views

The manifold defined by the hyper-ellipsoid must be refined in order to consider the discriminability criterion of recognition functions. We take an *ensemble of validation views* $\boldsymbol{X}_j^v$ into account (different from the ensemble of validation views) which in turn is subdivided into two classes. The first class $\mathcal{X}^{vp}$ (positive) of validation views is taken from the target object additionally and the second class $\mathcal{X}^{vn}$ (negative) is taken from counter objects or situations. Depending on certain results of applying the implicit function f^{Gi} to these validation views we specify spherical Gaussians and combine them appropriately with the implicit function. The purpose is to obtain a modified orbit which includes target views and excludes counter views.

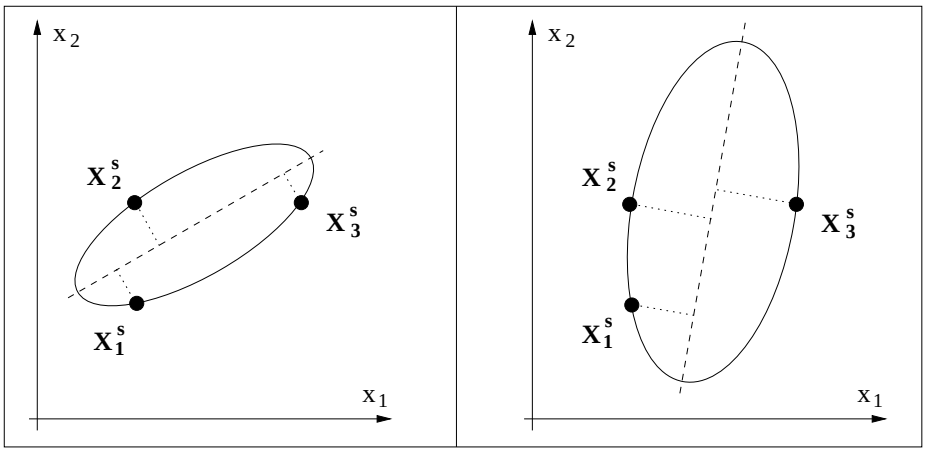

Fig. 2. Ellipses fitted through three points; (Left) Ellipse determined by PCA, showing first principal axis, determined by maximizing the variance; (Right) Ellipse determined manually with less variance along the dashed axis.

For each validation view $\boldsymbol{X}_j^v \in \mathcal{X}^{vp} \cup \mathcal{X}^{vn}$ the function f^{Gi} yields a *measurement of proximity* η_j to the hyper-ellipsoid orbit.

$$\eta_j := f^{Gi}(\boldsymbol{A}, \boldsymbol{X}_j^v) \tag{10}$$

For $\eta_j = 0$ the view $\boldsymbol{X}_j^v$ is far away from the orbit, for $\eta_j = 1$ the view belongs to the orbit. There are *two cases* for which it is reasonable to modify the implicit function. *First*, maybe a view of the target object is too far away from the orbit, *i.e.* $\boldsymbol{X}_j^v \in \mathcal{X}^{vp}$ *and* $\eta_j \leq \zeta$. *Second*, maybe a view of a counter situation is too close to the orbit, *i.e.* $\boldsymbol{X}_j^v \in \mathcal{X}^{vn}$ *and* $\eta_j \geq \zeta$. In the first case the modified function should yield a value near to 1 for validation view $\boldsymbol{X}_j^v$, and in the second case should yield a value near to 0. Additionally, we would like to reach generalization effects in the local neighborhood (in the space of views) of the validation view. The modification of the implicit function takes place by locally putting a spherical Gaussian f_j^{Gs} into the space of views, then multiplying a weighting factor to the Gaussian, and finally adding the weighted Gaussian to the implicit function. The mentioned requirements are reached with the following parameterizations. The center vector of the Gaussian is defined as $\boldsymbol{X}_j^v$.

$$f_j^{Gs}(\boldsymbol{X}) := exp\left(-\frac{1}{\tau} \cdot \|\boldsymbol{X} - \boldsymbol{X}_j^v\|\right) \tag{11}$$

For the two cases we define the weighting factor w_j dependent on η_j.

$$w_j := \begin{cases} 1 - \eta_j & : \quad first\ case\ (\ target\ pattern\ too\ far\ away\ from\ orbit\ ) \\ -\eta_j & : \quad second\ case\ (\ counter\ pattern\ too\ close\ to\ orbit\ ) \end{cases} \tag{12}$$

The additive combination of implicit function and weighted Gaussian yields a new function for which the orbit has changed, and in particular meets the requirements for the validation view $\boldsymbol{X} = \boldsymbol{X}_j^v$. In both cases, the Gaussian value is 1, and the specific weight plays the role of an increment respective decrement to obtain the final outcome 1 for the case $\boldsymbol{X}_j^v \in \mathcal{X}^{vp}$ respective 0 for vector $\boldsymbol{X}_j^v \in \mathcal{X}^{vn}$. The coarse-to-fine strategy of learning can be illustrated graphically (by recalling and modifying Figure 1). The course of proximity values obtained along the ellipse is constant 1 (see Figure 3, left and right), and along a straight line passing the ellipse perpendicular, the course of proximity values is a Gaussian (see left and middle).

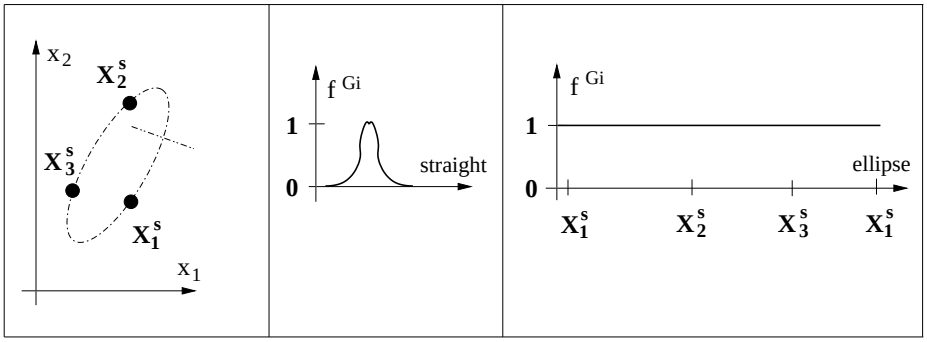

Fig. 3. (Left) Ellipse through three seed vectors and perpendicular straight line across the ellipse; (Middle) Gaussian course of proximity values along the straight line; (Right) Constant course of proximity values along the ellipse.

The first example considers a counter vector, *i.e.* $\boldsymbol{X}_2^v \in \mathcal{X}^{vn}$, which is too near to the ellipse. A Gaussian f_2^{Gs} is defined with $\boldsymbol{X}_2^v$ as center vector, and weight w_2 defined by η_2. Based on the additive combination of implicit function and weighted Gaussian the value decreases locally around point $\boldsymbol{X}_2^v$. Figure 4 (left and middle) shows the effect along the straight line passing through the ellipse, i.e. the summation of the two dashed Gaussians results in the bold curve. Figure 4 (left and right) shows the effect along the ellipse, i.e. the constancy is disturbed locally, which is due to diffusion effects originating from the added Gaussian.

The second example considers an additional view from the target object, *i.e.* $\boldsymbol{X}_3^v \in \mathcal{X}^{vp}$, which is far off the ellipse orbit. The application of f^{Gi} at $\boldsymbol{X}_3^v$ yields η_3. A Gaussian is defined with vector $\boldsymbol{X}_3^v$ taken as center vector, and the weighting factor w_3 is defined by $(1 - \eta_3)$. The combination of implicit function and weighted Gaussian is constant 1 along the course of the ellipse (for this example), and additionally the values around $\boldsymbol{X}_3^v$ are increased according to a Gaussian shape (see Figure 5).

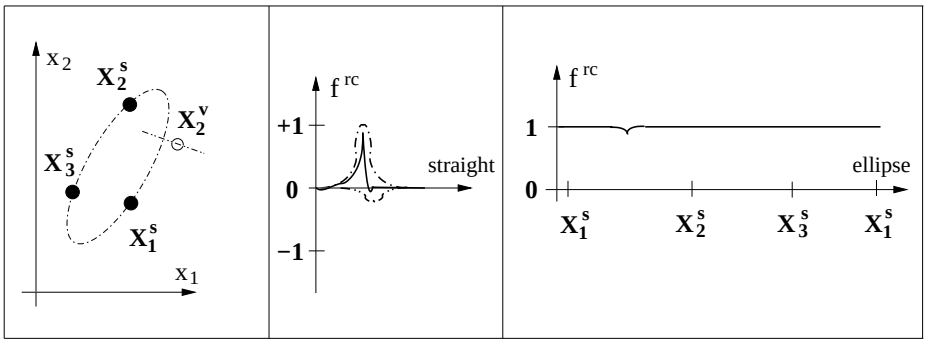

Fig. 4. (Left) Ellipse through three seed vectors and perpendicular straight line through a counter vector located near to the ellipse; (Middle) Along the straight line the positive Gaussian course of proximity values is added with the shifted negative Gaussian originating from the counter vector, such that the result varies slightly around 0; (Right) Along the ellipse the values locally decrease at the position near the counter vector.

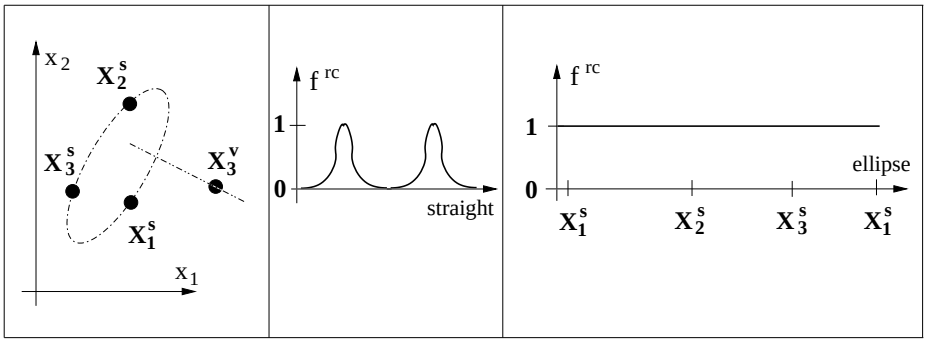

Fig. 5. (Left) Ellipse through three seed vectors and perpendicular straight line through a further target vector located far off the ellipse; (Middle) Along the straight line the positive Gaussian course of proximity values is added with the shifted positive Gaussian originating from the target vector, such that the result describes two shifted Gaussians; (Right) Along the ellipse the values are constant 1.

6 Construction of recognition functions

The recognition function f^{rc} is defined as sum of implicit function and linear combination of Gaussians.

$$f^{rc}(\boldsymbol{X}) := f^{Gi}(\boldsymbol{A}, \boldsymbol{X}) + \sum_{j=1}^{J} w_j \cdot f_j^{Gs}(\boldsymbol{X}) \tag{13}$$

Vector $\boldsymbol{X}$ represents an unknown view which has to be recognized. Parameter vector $\boldsymbol{A}$ is determined during the coarse approximation phase, and the set

of Gaussians is constructed during the fine approximation phase. Factor τ for specifying the extent of the Gaussians is obtained by the *Levenberg-Marquardt algorithm* [7, pp. 683-688].

This coarse-to-fine strategy of learning can be applied to any target object which we would like to recognize. If $k \in \{1, \cdots, K\}$ is the index for a set of target objects, then recognition functions f_k^{rc}, with $k \in \{1, \cdots, K\}$, can be learned as above. The final decision for classifying an unknown view $\boldsymbol{X}$ is by looking for the maximum value computed from the set of recognition functions f_k^{rc}, $k \in \{1, \cdots, K\}$.

$$k^* := \arg \max_{k \in \{1, \cdots, K\}} f_k^{rc}(\boldsymbol{X}) \tag{14}$$

For taking images under controlled viewing conditions, i.e. viewing angle or distance, the camera can be mounted on a robot arm and moved in any desired pose. The simplest strategy for selecting seed views is a regular discretization of the space of possible viewing poses. The selection of validation views may be done in a similar way, but considering pose offsets for views of the target objects and also taking images from counter situations. Several improvements for these strategies are conceivable, and the most important one concerns the treatment of validation views. For example, the fine approximation phase may be performed iteratively by checking for every validation view the necessity for modifying the emerging recognition function. Actually, we must install a new Gaussian only in the case of facing a recognition error according to the decision criterion given in equation (14). Every validation view is considered as a candidate, and only if the view is critical then the refinement may take place *dynamically*. This sophisticated strategy reduces the description length of the recognition function. Other interesting work has been reported belonging to the paradigm of *active learning* in which random or systematic sampling of the input domain is replaced by a *selective sampling* [1]. This paper doesn't focus on this aspect.

7 Experiments with the coarse-to-fine strategy of learning

The primary purpose is to obtain a recognition function for a target object of three-dimensional shape, which can be rotated arbitrary and can have different distances from the camera. According to this, both the gray value structure and the size of the target pattern varies significantly. Three objects are considered which look similar between each other, *i.e.* integrated circuit, chip carrier, bridge rectifier. Figure 6 shows a subset of three images from each object, respectively. Different sets of seed and validation ensembles will be used for learning. Exemplary, we only present the recognition results for the integrated circuit. A set of 180 testing images is taken which differs from the training images in offset values of the rotation angle and in the size of the patterns, as shown by three overlays in Figure 7.

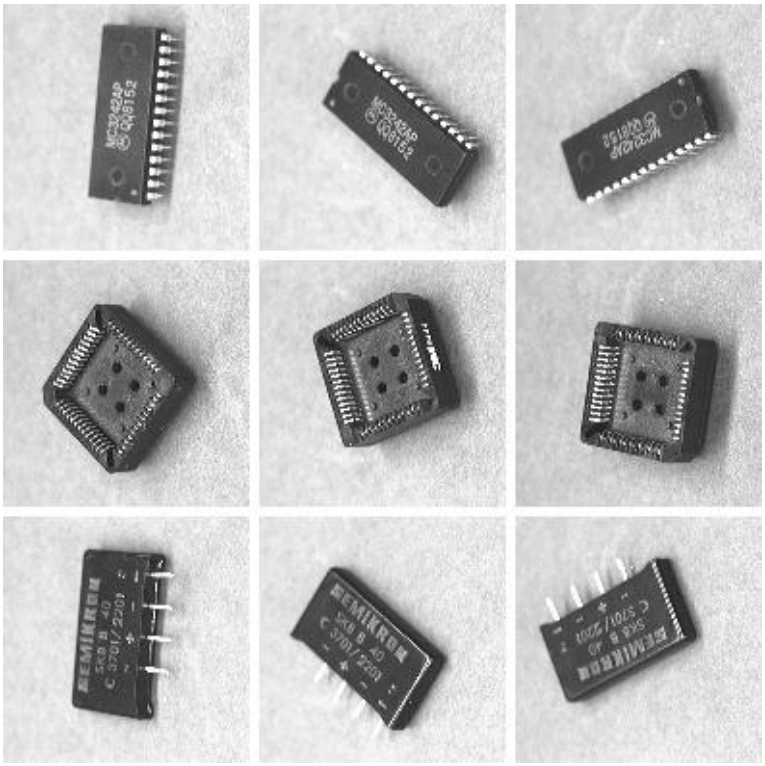

Fig. 6. Three seed images from three objects, respectively.

Fig. 7. Overlay between a training image and three testing images.

We determine recognition results, first by using a 1-nearest-neighbor approach, second by applying a coarse manifold approximation, and third compare them with those recognition results obtained from our coarse-to-fine strategy of learning. The approaches have in common that in a first step a testing pattern is projected into three canonical frames (CFs), which are the eigenspaces of the three objects, respectively. The second step of the approaches is the charac-

teristic one. In the first approach (CF_{1NN}, *1NN* for *1-N*earest-*N*eighbor) the recognition of a testing view is based on the proximity to all seed views from all three objects, and the relevant seed view determines the relevant object. In the second approach (CF_{ELL}, *ELL* for proximity to *ELL*ipsoids) the recognition of a testing view is based on the proximity to the three hyper-ellipsoids defined in the canonical frames, respectively. In the third approach (CF_{EGN}, *EGN* for proximity to *E*llipsoids extended with *G*aussian *N*etworks), the recognition of a testing view is based on a refinement of the coarse approximation of the pattern manifold by considering counter views with a network of GBFs, *i.e.* our favorite coarse-to-fine approach of learning. For validation views we simply take the seed views of the other objects, respectively. The decision for recognition is based on equation (14).

We make experiments with different numbers of seed views and thus obtain canonical frames of different dimensionality. Exemplary, 6, 12, 20, and 30 seed views are used which give dimension 5, 11, 19, and 29 of the canonical frames (denoted by NS_1, NS_2, NS_3, NS_4, respectively, NS for *N*umber of *S*eed views). Table 1 shows the results, i.e. the numbers of recognition errors, when applying the three approaches and taking four different dimensions into account. Approach CF_{ELL} clearly surpasses CF_{1NN}, and our favorite approach CF_{EGN} is clearly better than the other two. The course of recognition errors of CF_{EGN}, by increasing the dimension, shows the classical conflict between over-generalization and over-fitting. That is, the number of errors decreases significantly when increasing the dimension from NS_1 to NS_2, and remains constant or even increases when increasing the dimension further from NS_2 to NS_3 or to NS_4. Therefore, it is convenient to take the dimension NS_2 for the recognition function as compromise, which is both reliable and efficient. Qualitatively, all our experiments showed similar results (we omit to present them in this paper).

Errors	NS_1	NS_2	NS_3	NS_4
CF_{1NN}	86	59	50	49
CF_{ELL}	32	3	14	18
CF_{EGN}	24	1	2	3

Table 1. Recognition errors for a testing set which consists of 180 elements. The approaches of object recognition have been trained alternatively with 6, 12, 20, or 30 seed vectors, for the CF_{EGN} approach we take into account additionally 12, 24, 40, or 60 validation vectors.

According to the last row in Table 1 a slight increase of the number of recognition errors occurs when raising the number of seed views beyond a certain threshold, e.g. 20 or 30 seed views in our experiments. Therefore, the advantage of considering generalization biases (mentioned in Section 3) is weakened to a certain extent. This undesired finding is due to the fact that each new seed view will lead to an additional dimension and thus will cause a redefinition of the

canonical frame. The generalization induced by the higher dimensional hyper-ellipsoid may become more and more hazardous.

A more sophisticated approach is needed which increases the dimension on the basis of several (instead of just one) additional seed views. However, the purpose of this work has been to demonstrate the advantageous role of generalization biases in neural network learning, which is obtained by combining Gaussian basis funtion networks with hyper-ellipsoidal interpolations.

8 Discussion

The work presented a learning paradigm for appearance-based recognition functions. Principal component analysis (PCA) and Gaussian basis function networks (GBFN) are combined for dealing with the trade-off between efficiency, invariance, and discriminability. PCA is used for incorporating generalization biases which is done by a hyper-ellipsoid interpolation of seed views. GBFN is responsible for making the recognition function discriminative which is reached by a dynamic installation of weighted Gaussians for critical validation views. The combined set of training views is sparse which makes the learning procedure efficient and also results in a minimal description length. Apart from that, the discriminability of the learned recognition functions is impressive.

The presented learning paradigm is embedded in our methodology of developing Robot Vision systems. It works in combination with Active Vision strategies, i.e. we must exploit the agility of a camera in order to constrain the possible camera-object relations and thus reduce the complexity of the manifold. Specifically controlled camera movements enable the incorporation of further constraints, e.g. space-time correlations, log-polar invariants, which make the manifold construction more sophisticated.

We may extend the iterative learning procedure such that also canonical frames are constructed dynamically. This would be in addition to the dynamic installation of Gaussians. The purpose is to find a compromise between the dimension of canonical frames and the number of Gaussians, i.e. keep the product of both numbers as low as possible to reach minimum description length. The mentioned concept is a focus of future work.

Appendix

Let a hyper-ellipsoid be defined according to Section 4.

Theorem *All seed vectors* $\boldsymbol{X}_i^s \in \Omega, i \in \{1, \cdots, I\}$ *are located on the hyper-ellipsoid.*

Proof. There are several $(I-1)$-dimensional hyper-ellipsoids which interpolate the set Ω of vectors, respectively. PCA determines the principal axes $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{I-1}$ of a specific hyper-ellipsoid which is subject to maximization of projected variances along candidate axes. For the vectors in Ω we determine the set $\hat{\Omega}$ of

KLE-transformed vectors $\hat{X}_i := (\hat{x}_{i,1}, \cdots, \hat{x}_{i,I-1})^T$, $i \in \{1, \cdots, I\}$. All vectors in $\hat{\Omega}$ are located on a normal hyper-ellipsoid with constant Mahalanobis distance h form the origin. With the given definition for the half-lengths we can show that h is equal to 1, which will prove the theorem.

For the vectors in $\hat{\Omega}$ the variance v_l along axis $\boldsymbol{e}_l$, $l \in \{1, \cdots, I-1\}$ is given by $v_l := \frac{1}{I} \cdot (\hat{x}_{1,l}^2 + \cdots + \hat{x}_{I,l}^2)$. The variances v_l are equal to the eigenvalues λ_l. For each vector $\hat{X}_i$ we have the equation $\frac{\hat{x}_{i,1}^2}{\kappa_1^2} + \cdots + \frac{\hat{x}_{i,I-1}^2}{\kappa_{I-1}^2} = h$, because the vectors are located on a normal hyper-ellipsoid. Replacing κ_l^2 in the equation by the expression $\frac{I-1}{I} \cdot \left(\hat{x}_{1,l}^2 + \cdots + \hat{x}_{I,l}^2\right)$ yields the following equation $\frac{I}{I-1} \cdot \left(\frac{\hat{x}_{i,1}^2}{\hat{x}_{1,1}^2 + \cdots + \hat{x}_{I,1}^2} + \cdots + \frac{\hat{x}_{i,I-1}^2}{\hat{x}_{1,I-1}^2 + \cdots + \hat{x}_{I,I-1}^2}\right) = h$. Summing up all these equations for $i \in \{1, \cdots, I\}$ yields the equation $\frac{I}{I-1} \cdot (I-1) = I \cdot h$, which results in $h = 1$.

q.e.d.

References

1. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning **15** (1994) 201-221
2. Dayan, P., Hinton, G., Neal, R., Zemel, R.: The Helmholtz machine. Neural Computation **7** (1995) 889-904
3. Murase, H., Nayar, S.: Visual learning and recognition of 3D objects from appearance. International Journal of Computer Vision **14** (1995) 5-24
4. Papathomas, T., Julesz, B.: Lie differential operators in animal and machine vision. In Simon, J.: From Pixels to Features. Elsevier Science Publishers (1989) 115-126
5. Pauli, J.: Development of Camera-Equipped Robot Systems. Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, Technical Report **9904** (2000)
6. Poggio, T., Girosi, F.: Networks for approximation and learning. Proceedings of the IEEE **78** (1990) 1481-1497
7. Press, W., Teukolsky, S., Vetterling, W.: Numerical Recipes in C. Cambridge University Press (1992)
8. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3** (1991) 71-86
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
10. Winston, P.: Artificial Intelligence. Addison-Wesley Publishing Company (1992)