Generic Hierarchic Object Models and Classification based on Probabilistic PCA

Christopher Drexler*, Frank Mattern and Joachim Denzler
Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
E-mail: {drexler,mattern,denzler}@informatik.uni-erlangen.de
www: http://www5.informatik.uni-erlangen.de

Abstract

In this paper we tackle the problem of classifying objects, which are not known to the system but similar to some of the objects contained in the training set. This type of classification is referred to as generic object modeling and recognition and is necessary for applications were it is impossible to model all occurring objects. As no class for unknown objects exist, they are either rejected or assigned to the most similar class contained in the training set. Even in the case of soft assignments this can lead to wrong interpretation of the actual class membership.

We present a new approach for generating appearance based hierarchical object models based on probabilistic PCA for generic object recognition. During the training step a hierarchical set of mixtures of probabilistic PCA models is generated. This represents a coarse–to–fine gradation with respect to the reconstruction ability of the training views at each hierarchy level. So coarse parts of the training views are covered on higher levels whereas the lower levels cover more details of the encoded training views. The mixture components are calculated at each hierarchy in an unsupervised manner using the expectation–maximization algorithm.

1 Introduction

The main task of object recognition systems is the distinct classification of objects into trained classes, taking into account varying illumination, different object poses and partial occlusion. Under these conditions the way of constructing the object models is crucial. Moreover many application domains exist where it is impossible to model all possibly occurring objects. This includes, e.g., autonomous service robot scenarios, in which it is not possible to present all objects within the operational area in the training step. The actual success of commands like "Bring me the cup of water!" does not depend on recognizing a distinct, previously trained cup, but any cup is sufficient for a correct execution.

Generic object models allow for classifying unknown objects into categories which describe subsets of the training data with respect to common features. Organizing these categories in a hierarchical manner defines a coarse–to–fine model hierarchy where higher levels describe generic super classes and the lowest level distinct objects.

*This work was funded by the German Science Foundation (DFG) under grant DE 735/2–1. Only the authors are responsible for the content.

In order to achieve such an object recognition system, certain requirements have to be fulfilled.

An object model type has to be defined which allows for hierarchic partitioning of the training set and, in addition to the quality criteria for class assignments, a criteria must be available to distinguish unknown from known objects.

Former approaches [7, 5] have used geometric primitives as features for building the models, confining themselves to handle only objects describable by a small set of geometric elements. Our approach combines the advantages of generic and appearance based object modeling together with a Bayesian classification due to the underlying probabilistic model. In this paper the work from [3] is extended with the focus on generic classification.

2 Theory

In contrast to former approaches based on geometric primitives as features for generating generic object models [7, 5] we focus on a categorization scheme which is based on the appearance and not on the semantic of the objects using probabilistic PCA (PPCA) models [10].

The PCA or Karhunen–Loéve–Transformation is the starting point for our considerations which is already widely used in the object and face recognition community [6, 2, 11]. The idea of using view based generic models using PCA features is based on the property of the PCA to define an order on the information content that each basis vector of the transformation carries.

The disadvantages of the PCA are its global linearity assumption and a missing underlying statistical model that would allow for soft decisions about the membership of a certain object using probabilities. Using mixtures of PCA models would involve some kind of vector quantization in advance to get clusters for calculating local PCA features. As the clustering is done independently from the PCA, the resulting representation with respect to the reconstruction error is not optimal. With the application of mixtures of factor analyzers, a combined optimization of clusters and local PCA like dimensionality reduction is available [1].

In [4] it has been shown, how mixtures of factor analyzers (FA) can be calculated within an expectation–maximization framework and [10] explains the relationship between standard PCA and FA. This information is used to build "mixtures of probabilistic PCA models" (MPPCA), which are derived from FA.

2.1 Factor Analysis and PPCA

Factor analysis is based on a generative model, where an observation, e.g. an image vector, $\boldsymbol{t}_i \in \mathbb{R}^d$ is generated by a q–dimensional random vector $\boldsymbol{x}_i$, build from the so called factors, according to the mapping

$$\boldsymbol{t}_i = \boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon} \quad . \tag{1}$$

Here $\boldsymbol{\mu}$ is a constant displacement vector, $\boldsymbol{\epsilon}$ is a noise vector and $\boldsymbol{W}$ the so called factor loading matrix. The assumption is that $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_q)$ as well as $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$ are zero mean Gaussian distributed random vectors (with $\boldsymbol{I}_q$ being a $q \times q$–dimensional identity matrix and $\boldsymbol{\Psi}$ a $d \times d$–dimensional diagonal covariance matrix). Consequently the observation $\boldsymbol{t}_i$ is also Gaussian distributed.

Given a set of n observations $\boldsymbol{t}_i$ the unknown parameters of the factor model $\boldsymbol{W}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Psi}$ can be estimated using the EM algorithm. Details of the EM–computation can be found in [4].

The model from (1) can be easily extended to a mixture model of m Gaussian distributions. The observation vectors $\boldsymbol{t}_i$ are now modeled by

$$\boldsymbol{t}_i = \sum_{k=1}^{m} \omega_k (\boldsymbol{W}_k \boldsymbol{x}_i + \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_k) \tag{2}$$

with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_q)$ and $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Psi}_k)$. The quantity ω_k is the weight of the kth mixture component, $\boldsymbol{\Psi}_k$ again a diagonal covariance matrix of the observation noise. Again the reader is referred to [4] for a further discussion of how to extent the EM–algorithm for estimating the unknown parameters ω_k, $\boldsymbol{W}_k$, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Psi}_k$.

For approximating the PCA the diagonal covariance matrix $\boldsymbol{\Psi}$ is restricted to have identical elements ($\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}_d$) [10]. Moreover in the case of mixture analyzers *all* $\boldsymbol{\Psi}_k$ are restricted to have identical σ's. This restriction is based on the interpretation of the elements of $\boldsymbol{\Psi}_k$ as the sensor noise model. In the case of images as observations the elements of $\boldsymbol{\Psi}_k$ represent the noise model of each individual CCD–sensor element. Allowing only one σ, we assume the noise model of each sensor element to be the same and independent of the sensor reading value.

2.2 Training and classification

Figure 1 depicts the training algorithm. The hierarchical model generation takes three steps for each hierarchy level. At the beginning all input images of all objects are used to generate a low dimensional eigenspace [6] and the according eigenspace features for all input images. This is done to reduce the input dimension for the factor analysis to be numerically feasible, e.g. from 16384 for 128×128 pixel images to a maximum of 100 dimensions. Then the MPPCA model is generated based on the eigenspace features of the training images. Therefore the log–likelihood of the model

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{n} \ln p(\boldsymbol{t}_i) \quad \text{with} \\ p(\boldsymbol{t}_i) &= \sum_{k=1}^{m} \omega_k p(\boldsymbol{t}_i|k) \quad , \end{aligned} \tag{3}$$

with n observations and m Gaussian distributions, is maximized via the EM algorithm.

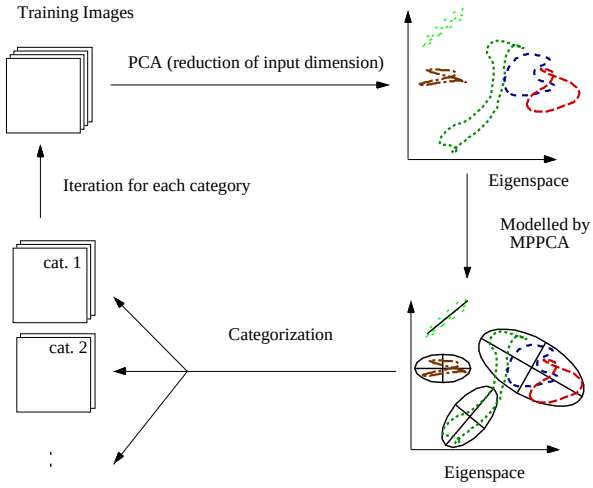


Figure 1: Iterative model generation at the training step.

As a third step the input images are assigned to one of the mixture components utilizing a Bayes classifier with the a posteriori probability

$$p(k|\boldsymbol{t}_i) = \frac{\omega_k p(\boldsymbol{t}_i|k)}{p(\boldsymbol{t}_i)} \quad \text{with} \tag{4}$$

$$p(\boldsymbol{t}_i|k) = (2\pi)^{-d/2} |\boldsymbol{C}_k^{-1/2}| \exp\left(-\frac{1}{2}(\boldsymbol{t}_i - \boldsymbol{\mu}_k)^T \boldsymbol{C}_k^{-1} (\boldsymbol{t}_i - \boldsymbol{\mu}_k)\right) \tag{5}$$

and

$$\boldsymbol{C}_k = \boldsymbol{W}_k \boldsymbol{W}_k^T + \sigma^2 \boldsymbol{I}_d \tag{6}$$

of the submodel k given the observation $\boldsymbol{t}_i$.

All input images assigned to one mixture component serve as a new category for the next iteration where the algorithm is repeated with the images from each category.

Classification is done similar to the training. At each hierarchy level, starting from the highest one, the test image is first projected into the eigenspace calculated from the training images at this level. According to the ML assignment to one of the mixture components the model for the next hierarchy level is selected. At the lowest level a nearest–neighbor classification is performed within the eigenspace.

The log–likelihood of a test image vector $\hat{\boldsymbol{t}}$ according to (3) at each hierarchy level serves as a quality criteria how well a test view is represented by the model. This is used to distinguish between views of objects contained in the training set and views of objects which are not. The behavior of this quality measure for different known and unknown views is shown in Section 3.

3 Experiments

All experiments for evaluating the approach were done using the COIL–20 and COIL–100 databases [9, 8] in order to have a widely used basis for comparison. For demonstrating our approach for generic object recognition we present results on standard, i.e. non–generic, and generic object recognition based on MPPCA models together with results from standard PCA, where appropriate, i.e. for the non–generic part.

feature space dimension	hierarchy level 0	level 1	level 2	std. PCA
3–D	80.6%	87.5%	90.3%	74.4%
5–D	93.9%	94.1%	95.3%	89.4%
10–D	98.1%	97.5%	96.8%	94.4%
15–D	98.7%	98.0%	97.3%	94.9%

Table 1: Recognition rates for COIL–100 database with disjunct 50% training and 50% test images with different input dimensions from PCA.

3.1 MPPCA vs. standard PCA

In order to compare our model with standard eigenspace approaches we calculated recognition rates using graylevel images of size 128×128 pixels from the COIL–100 database which consists of 72 views for each of the 100 objects. The object model uses 3 hierarchy levels with 5 mixture components at each level. In order to perform the actual classification at each hierarchy level the training views contained in a category are divided according to their class labels and for each set a standard eigenspace is calculated. Classification on one level is done by assigning the test view to one of the categories according to the Bayes scheme (c.f. (4)) and projecting the test view into each of the associated eigenspaces. The class label is selected by a nearest-neighbor classifier.

Table 1 summarizes the achieved recognition rates for the COIL–100 database. The database was divided into disjunct sets of 50% training and 50% test images. For each hierarchy level the recognition rates for the test set of known objects is given. The last column gives the result on a standard PCA nearest–neighbor classification for a PCA model at the given dimension. It can be seen that the MPPCA approach is superior to the standard PCA approach at each input dimension, even on the coarsest level 0.

Taking the very low dimension of the input features for the 3–D case into account, the recognition rate of 90.3% are a reasonable and promising result for larger databases. An increase of the featurespace dimension does not necessarily increase the recogntions rates at finer levels as not enough training images remain for proper eingenspace calculation.

3.2 Generic Classification

Achieving generic classification results is not as trivial as for the classification of trained objects. Precise numbers can not be presented as the categorization is done unsupervised, which does not necessarily lead to sensible results for a "human classifier".

As Section 3.1 proved that the categorization and classification scheme according to the a posteriori probabilities performs very well, this section focus on analyzing the properties of the log–likelihood (LL) criteria.

For testing the generic recognition capabilities, two objects of the COIL–20 and four of the COIL–100 training set have been completely removed, leaving only similar objects for model generation. The data of all other objects where divided into disjunct training and test set, both containing 50% of the images for each object (c.f. Figure 2).

The log–likelihood at each hierarchy level should give information on whether the test image is represented by the current MPPCA model or whether it is too different from the stored views to be represented by this model.

For known objects the log–likelihood should stay the same through all hierarchy levels or even increase. This is due to the fact that the finer models are able to capture the actual appearance better than the coarser models. Unknown objects, in contrast, should be recognizable by a decrease of the log–likelihood at a hierarchy level where the model gets too specialized.

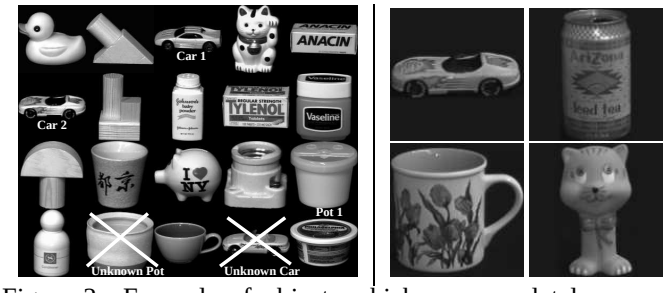


Figure 2: Example of objects which are completely removed from the training set for generic object recognition. On the left side: examples of all objects of the COIL–20 database with those excluded crossed. On right side: examples of the objects of COIL–100 database excluded from the training.

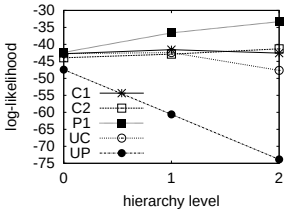


Figure 3: PPCA

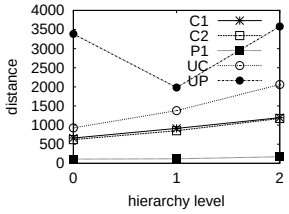


Figure 4: NN

Log–likelihood (Figure 3) and distance function plot within eigenspace (Figure 4) for the two cars (C1,C2) and the pot (P1) which are part of the training set as well as for the unknown car (UC) and the unknown pot (UP).

Figure 3 shows the log–likelihood for the hierarchy level 0 to 2 and for comparison Figure 4 does the same for the "distance in feature space" function used as quality criteria for a nearest–neighbor classification for standard PCA. Both diagrams show the averaged curves over all test images for the two unknown objects "uncovered pot" and "car3" as well as for the three known objects "half covered pot" and "car1/2".

In this case, for 20 classes, the log–likelihood criteria performs very well for generic classification. The two unknown objects can be identified by decreasing LL. The LL for the unknown car remains stable until level 1 and decreases at level 3, compared to the LL of the unknown pot, which decreases monotonically through all levels. This behavior reflects the fact, that for the car, two very similar objects were part of the training set, whereas for the pot, only one object, which exhibits more differences to the unknown object as the known cars to the unknown, remains in the training set.

Experiments with the COIL–100 database and the four excluded objects (Figure 2) show a similar behavior, but the results are not as clear as for the smaller data set.

Figure 5 and 6 show the averaged log–likelihood over all views of the four unknown objects and a subset of the remaining 96 known objects. The subset was chosen to contain objects which for humans visually similar to the unknown objects, e.g. eight cars, four cups, seven cans and one toy bear.

The basic properties of the COIL–20 results can be verified but for the unknown objects only the log–likelihood for

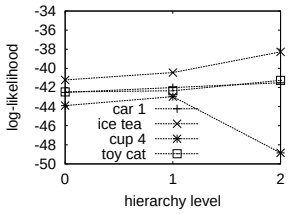


Figure 5: unknown objects

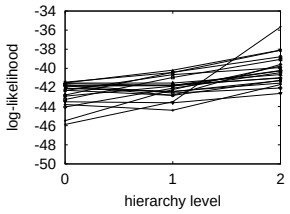
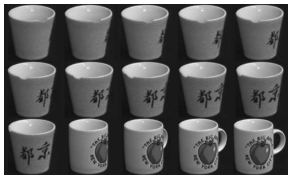


Figure 6: known objects

Averaged Log–likelihood of the four unknown objects (Figure 5) and for a subset of the remaining 96 known objects (Figure 6) of the COIL-100 database.



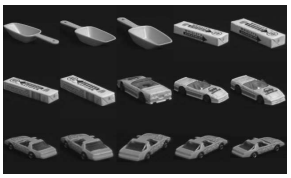


Figure 7: Examples of training views contributing the models at level 2 where most of the unknown cup views (left) and the unknown car views (right) were assgined to.

"cup 4" shows a proper behavior.

Due to the larger number of objects more hierarchy levels and/or mixture components at each level are required to achieve class specific information on the finest level. This leads to problems while calculating eigenspaces and MPPCAs because of the lack of training data. For 20 objects we achieve good separation results on level 2 with a sufficient number of images for estimating the distributions. Having 100 objects leads to categories which consists of similar views of a larger number of object classes. The consequence is, that unknown objects at this level have high LL's and require further hierarchy levels to separate the objects. Having approximately 50% of all images (=3600) as training data results in an average of 28 images per model at level 2 (3600 images divided by $5 \times 5 \times 5 = 125$ models). Further splitting is therefore not always possible.

Figure 3.2 and 3.2 show example views of object from two models at level 2 where most of the unknown cups and cars were assigned to. Whereas the model for the cup contains only two very similar classes the model for the car consists of 23 classes altogether (not all shown).

The models, where the other unknown objects are assigned to, show a similar distribution of object classes. This indicates that for the cup the lowest hierarchy level has been reached but further levels should be calculated in the case of the other models. For example examining two more levels derived from the car model from Figure 3.2, most of the car views are separated at level 4 (Figure 8). The left side of the figure contains nearly all training views contained in that model, showing that only cars are included, but still five different types. The unknown car of a view similar to these get therefore good log–likelihood values. More images of each car type would be necessary to generate more specialized models, further extending the hierarchy.

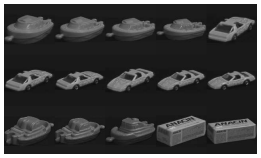


Figure 8: Example views of objects from models containing most of the cars at level 3 (left) and 4 (right).

A solution to this in the absence of a proper number of training images is to generate new views either by interpolating within the eigenspace, as used in [6] for building subspace models, by small affine transformations of the original images and by adding additional noise.

4 Conclusion

To summarize the results, we have shown, that our proposed hierarchical object model based on mixtures of probabilistic PCA's suits the need for generic object recognition. It can be seen that for standard object recognition problems the new models give reasonable results and that additionally a quality criteria is defined which can be exploited for generic object recognition.

Further work is done especially on evaluating the models using other databases, i.e. analyzing the behavior of the log-likelihood for a larger number of object classes and improving the hierarchy level generation with respect to solving the mentioned problems.

References

[1] D. Bartholomew. *Latent Variable Models and Factor Analysis*. Charles Griffin & Co. Ltd., London, 1987.

[2] N. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 711–720, July 1997.

[3] Ch. Drexler, F. Mattern, and J. Denzler. Appearance Based Generic Object Modeling and Recognition using Probabilistic Principal Component Analysis. In *Pattern Recognition — 24rd DAGM Symposium*, Berlin, September 2002. Springer.

[4] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CFG–TR–96j–1, Dept. of Computer Science, University of Toronto, February 1997.

[5] G.G. Medioni and A.R.J. Francois. 3–d structure for generic object recognition. In *Volume 1. Computer Vision and Image Analysis*, International Conference on Pattern Recognition, pages 30–37, 2000.

[6] H. Murase and S. Nayar. Visual Learning and Recognition of 3–D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[7] R. C. Nelson and A. Selinger. Large–scale tests of a keyed, appearance–based 3–D object recognition system. *Vision Research*, 38(15–16):2469–2488, August 1998.

[8] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL–100). Technical Report CUCS–006–96, Dept. fo Computer Science, Columbia University, 1996.

[9] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL–20). Technical Report CUCS–005–96, Dept. fo Computer Science, Columbia University, 1996.

[10] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[11] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. *IEEE Conference on Vision and Pattern Recognition*, pages 586–591, 1991.