The Missing Information Principle in Computer Vision

Joachim Hornegger and Heinrich Niemann

The following paper will appear in the
Proceedings of the 2nd German–Slovenian–Workshop

The Missing Information Principle in Computer Vision

J. Hornegger, H. Niemann
Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen–Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
email: {hornegger,niemann}@informatik.uni-erlangen.de

Abstract

Central problems in the field of computer vision are learning object models from examples, classification, and localization of objects. In this paper we will motivate the use of a classical statistical approach to deal with these problems: the missing information principle. Based on this general technique we derive the Expectation Maximization algorithm and deduce statistical methods for learning objects from invariant features using Hidden Markov Models and from non–invariant features using Gaussian mixture density functions. The derived training algorithms will also include the problem of learning 3D objects from two–dimensional views. Furthermore, it is shown how the position and orientation of a three–dimensional object can be computed. The paper concludes with some experimental results.

Keywords: Expectation Maximization algorithm, Hidden Markov Models, statistical object recognition

1 Introduction

Object recognition systems are expected to be robust with respect to instabilities of segmentation results. Moreover, those systems should also provide capabilities of learning, i. e. the algorithms should be able to acquire knowledge of a new object from sample data. The efficiency of an object recognition system is based on a reliable classification and localization of objects. These requirements suggest the use of statistical methods in a

natural manner. If the features are treated as random variables or vectors, their behavior in varying segmentation results can be described by probability functions. Thus, complete object models are represented as density functions. The process of learning corresponds to the computation of the parameters of the density function or the application of non–parametric estimation techniques in the non–parametric case. Reliability is achieved, if the Bayesian decision rule is applied, since it is known from decision theory that the Bayesian classifier is optimal with respect to the probability of misclassification.

Classical pattern recognition theory [8] is based on the assumption that each pattern of a class can be characterized by one feature vector of a fixed dimension. The statistical model of one object is thus described by a single density function. An object, in general, cannot be represented by a single feature vector; commonly, a sequence of features or a set of features is required for the demanded discriminental power. Hence, more general techniques and algorithms have to be used to deal with the training and recognition problem of objects.

This contribution motivates that many image recognition problems can be understood as an incomplete data estimation problem. We introduce a general mathematical framework to manage those issues. The described abstract algorithm is applied to three different problem domains: we introduce Hidden Markov Models for learning from feature sequences of varying length, we suggest a statistical approach to the problem of learning three–dimensional structure from 2D views, and finally, it is shown how the proposed recipe can be used to compute the position and orientation of a known object in a given scene. The paper concludes with some experimental results and additional remarks.

2 Incomplete Data Estimation Problems

The features which can be computed for a given image frequently do not provide the complete information. For instance, if we use the model based approach for object recognition, it is a priori not known which image feature corresponds to which model feature. In the case of three–dimensional vision problems from 2D views the range information is additionally missing. If there is an heterogeneous background, the partition of object and background features is also a component of hidden information for the classification algorithm.

These examples demonstrate, that a lot of image recognition problems can be decomposed using the following colloquial paraphrase of the *Missing Information Principle* [13]:

Observable Information = Complete Information − Missing Information

Obviously, this general statement provides no algorithms. We will use statistical principles and deduce an algorithmic framework, which admits to deal with incomplete data estimation problems. Let us assume that a parametric probability function is given by the parameter set $\boldsymbol{B} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$. We denote the observable data by the random variable $\boldsymbol{X}$ and the missing data by $\boldsymbol{Y}$. The goal of a training algorithm is the estimation of the parameters $\boldsymbol{B}$ using exclusively the observable information. Nevertheless, there exist relations between observable and hidden data, which might be advantageous for the learning process in some cases. Using a maximum likelihood approach to estimate the parameter set $\boldsymbol{B}$, the probability function

$$P(\boldsymbol{X} \mid \boldsymbol{B}) \;=\; \frac{P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{B})}{P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B})} \tag{1}$$

has to maximized. Frequently, it is computationally worthwhile to use the logarithm of the probability function $L(\boldsymbol{X}, \boldsymbol{B}) = \log\, P(\boldsymbol{X} \mid \boldsymbol{B})$ for the optimization process. Thus, we have

$$\log\, P(\boldsymbol{X} \mid \boldsymbol{B}) \;=\; \log\, P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{B}) - \log\, P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B}) \tag{2}$$

which indeed corresponds to a mathematical formalization of the missing information principle: the complete information is described by $\log\, P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{B})$ and $\log\, P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B})$ represents the missing part. An iterative algorithm for the computation of $\boldsymbol{B}$ can be derived if we use the conditional expectation of the logarithmic likelihood function (2) with respect to the actual estimate of $\boldsymbol{B}$ and the observable set of random variables $\boldsymbol{X}$. The reestimations are denoted by $\hat{\boldsymbol{B}}$. The application of the definition of the conditional expectation results in the following *key–equation*

$$E[L(\boldsymbol{X}, \hat{\boldsymbol{B}}) \,|\, \boldsymbol{X}, \boldsymbol{B}] = L(\boldsymbol{X}, \hat{\boldsymbol{B}}) = Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) - H(\boldsymbol{B}, \hat{\boldsymbol{B}}), \tag{3}$$

where

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \int P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B})\, \log\, P(\boldsymbol{X}, \boldsymbol{Y} \mid \hat{\boldsymbol{B}})\, d\boldsymbol{Y} \tag{4}$$

and

$$H(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \int P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B})\, \log\, P(\boldsymbol{Y} \mid \boldsymbol{X}, \hat{\boldsymbol{B}})\, d\boldsymbol{Y}. \tag{5}$$

Using Jensen's inequality [11] it can be shown that an increase of the Q–function (4) corresponds to a decrease of the H–function (5). Consequently, it is sufficient to optimize

Initialize $\boldsymbol{B}$	
WHILE $\boldsymbol{B} \neq \hat{\boldsymbol{B}}$	
	E–STEP: compute $Q(\boldsymbol{B}, \hat{\boldsymbol{B}})$
	M–STEP: compute $\hat{\boldsymbol{B}} = \arg\max_{\hat{\boldsymbol{B}}} Q(\boldsymbol{B}, \hat{\boldsymbol{B}})$

Figure 1: A structogram for the EM–Algorithm

merely the *Kullback–Leibler statistics* $Q(\boldsymbol{B}, \hat{\boldsymbol{B}})$. An iterative algorithm for the optimization of $L(\boldsymbol{X}, \hat{\boldsymbol{B}})$ is the *Expectation Maximization Algorithm* described in Figure 2. This algorithm was developed by Dempster e. a. [3]. The properties of the EM–algorithm are summarized in [3, 12, 17]. The positive characteristics of the EM–algorithm are based on the constant storage requirements and on the observation that in many applications a decomposition in much easier optimization problems occurs. The disadvantages of this iterative estimation procedure are the slow convergence rate and the restriction for the computation of local maxima. A comparison of maximum likelihood estimates with the iterative EM–algorithm can be found in [2].

The missing information principle can be summarized by the following steps: First of all, you have to define a suitable statistical model for the given problem to be solved. Then, the observable and hidden information can be derived and the computation of probability functions $P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{B})$ and $P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{B})$ has to be done. Before the EM–iterations are carried out, one has to choose an appropriate maximization technique of the Kullback–Leibler statistics.

The remaining parts of this contribution are dedicated to the application of this abstract algorithm to different computer vision problems.

3 Applications

This section describes three different applications of the missing information principle in the field of image processing. We introduce Hidden Markov Models, which are broadly used for the classification of speech signals, and derive training formulas for estimating the parameters. The second and third application describes recent results dealing with the problem of learning 3D objects from 2D views and the localization of a known 3D object in an observed scene.

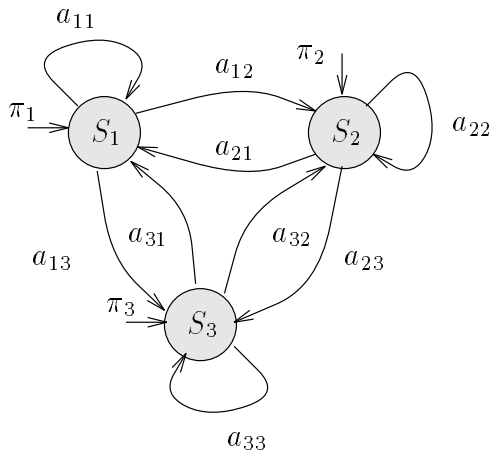

Figure 2: Ergodic Hidden Markov Model; each state emits probabilistic output symbols.

3.1 Learning from Feature Sequences

If an object in a scene can be associated with a feature sequence $\boldsymbol{O} =< \boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_n >$, a classification system is needed which can compute the a posteriori probability for observing the special feature sequence of length n. An established statistical model for dealing with the problem of classifying feature sequences are stochastic automata, especially Hidden Markov Models (HMMs). The stochastic automata consist of a set of states, transitions among these states, and emission probabilities for elements of a given alphabet. The probabilistic behavior of a HMM with N states $\{S_1, \ldots, S_N\}$ can be described by a triplet $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, where $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the vector of probabilities for the generation of a sequence of output elements to start at a special state. The state transition matrix $\boldsymbol{A} = (a_{i,j})_{0<i\leq N, 0<j\leq N}$ includes the probabilities $a_{i,j}$ to change from state S_i to state S_j. The third element of the triplet $\boldsymbol{\lambda}$ is a matrix $\boldsymbol{B} = (b_i(v_l))_{0<i\leq N, 0<l<L}$ including discrete probabilities for a finite output alphabet $\{v_1, v_2, \ldots, v_L\}$. An example of a Hidden Markov Model is shown in Figure 3.1. Let us assume that the observable feature sequence for a given object is produced by one automaton. The parameter set $\boldsymbol{B}$ of Figure 2 corresponds to the parameters $\boldsymbol{\pi}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ of the Hidden Markov Model. Following the missing information principle, we have to determine what is known and what is hidden for the training process. Obviously, the parameter estimation procedure is unsupervised inasmuch as the state sequence, which produces the sequence of output symbols, is not observable. Thus $\boldsymbol{X} = \boldsymbol{O}$ and $\boldsymbol{Y} = \boldsymbol{s}$, where $\boldsymbol{O}$ represents the sequence of observable output symbols and $\boldsymbol{s}$ is the non–observable state sequence. Thus, the needed

probabilities for the complete and the missing information are

$$P(\boldsymbol{s}, \boldsymbol{O} \mid \boldsymbol{\lambda}) = \pi_{s_1} \prod_{t=1}^{n-1} a_{s_t,s_{t+1}} \prod_{t=1}^{n} b_{s_t}(o_t), \tag{6}$$

and

$$P(\boldsymbol{s} \mid \boldsymbol{O}, \boldsymbol{\lambda}) = \frac{P(\boldsymbol{s}, \boldsymbol{O} \mid \boldsymbol{\lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\lambda})} = \frac{\pi_{s_1} \prod_{t=1}^{n-1} a_{s_t,s_{t+1}} \prod_{t=1}^{n} b_{s_t}(o_t)}{\sum_{\mathrm{s}} \pi_{s_1} \prod_{t=1}^{n-1} a_{s_t,s_{t+1}} \prod_{t=1}^{n} b_{s_t}(o_t)}. \tag{7}$$

Due to the fact that the unobservable data represent discrete state sequences, the integral (4) becomes a sum over all admissible state sequences.

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{\boldsymbol{s}} P(\boldsymbol{s} \mid \boldsymbol{O}, \boldsymbol{\lambda}) \log P(\boldsymbol{s}, \boldsymbol{O} \mid \hat{\boldsymbol{\lambda}}) \tag{8}$$

The calculation of the maximum of the Kullback–Leibler statistics in each iteration is a constraint optimization problem, because the parameters are discrete probabilities. We compute the zero crossings of the first derivative with respect to the parameters $\hat{\pi}_{s_i}$, $\hat{a}_{s_t,s_{t+1}}$ and $\hat{b}_{s_t}(o_t)$ by taking into consideration Lagrange multipliers.

$$\begin{aligned} \nabla_{\hat{\lambda}} Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) &= \sum_{\boldsymbol{s}} P(\boldsymbol{s} \mid \boldsymbol{O}, \boldsymbol{\lambda}) \nabla_{\hat{\lambda}} \log P(\boldsymbol{s}, \boldsymbol{O} \mid \hat{\boldsymbol{\lambda}}) \\ &= \sum_{\boldsymbol{s}} P(\boldsymbol{s} \mid \boldsymbol{O}, \boldsymbol{\lambda}) \nabla_{\hat{\lambda}} \left(\log \hat{\pi}_{s_1} + \sum_{t=1}^{n-1} \log \hat{a}_{s_t,s_{t+1}} + \sum_{t=1}^{n} \log \hat{b}_{s_t}(o_t) \right) \end{aligned} \tag{9}$$

Evidently, the derivatives separate different variables and we obtain a closed form solution of the reestimation procedure for the required parameters.

$$\hat{\pi}_i = \frac{P(s_1 = S_i, \boldsymbol{O} \mid \boldsymbol{\lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\lambda})} \tag{10}$$

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{n-1} P(s_t = S_i, s_{t+1} = S_j, \boldsymbol{O} \mid \boldsymbol{\lambda})}{\sum_{j=1}^{N} \sum_{t=1}^{n-1} P(s_t = S_i, s_{t+1} = S_j, \boldsymbol{O} \mid \boldsymbol{\lambda})} \tag{11}$$

$$\hat{b}_i(\boldsymbol{o}_j) = \frac{\sum_{t \in \{t \mid \boldsymbol{O}_t = \boldsymbol{O}_j\}} P(s_t = S_i, \boldsymbol{O} \mid \boldsymbol{\lambda})}{\sum_{t=1}^{n} P(s_t = S_i, \boldsymbol{O} \mid \boldsymbol{\lambda})}. \tag{12}$$

These formulas are the basis of the well known Baum–Welch algorithm [1] and can now be used for training the parameters of the HMM.

3.2 Learning 3D Objects from 2D Views

In the previous section we assumed that it is possible to associate with each object a feature sequence, independent of its localization in the image. Let us assume the more general case that a three–dimensional object is characterized by its 3D vertices, by means of rotation, translation and subsequent projection 2D–point–features can be observed (see Figure 3.2). In [7] it is shown that there exists no ordering on these projected points,

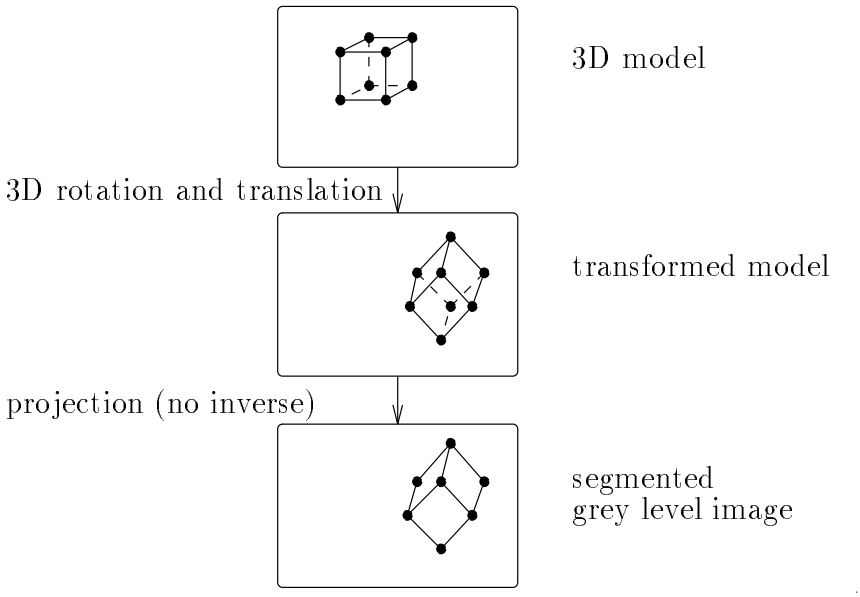

Figure 3: From the 3D model to the 2D scene

which is conform to the 3D ordering. Thus, the observable object of the j–th view ($1 \leq j \leq J$) has to be represented by a set of features $\boldsymbol{O}_j = \{\boldsymbol{O}_{j,1}, \boldsymbol{O}_{j,2}, \ldots, \boldsymbol{O}_{j,m_j}\}$. The set of the correlated model features is denoted by $\boldsymbol{C}_\kappa = \{\boldsymbol{C}_{\kappa,1}, \boldsymbol{C}_{\kappa,2}, \ldots, \boldsymbol{C}_{\kappa,n_\kappa}\}$. The matching function ζ_κ assigns to each observed feature $\boldsymbol{O}_{j,k}$ a model primitive $\boldsymbol{C}_{\kappa,i}$. Due to segmentation errors and noise, point features show some instabilities, which can be modeled by assuming that each point feature is normally distributed [15]. One observable point might be assigned to any model feature. Therefore, the probability of observing one special point $\boldsymbol{O}_{j,k}$ can be written as

$$P(\boldsymbol{O}_{j,k} \mid \boldsymbol{B}) = \sum_{i=0}^{n_\kappa} P(\zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i})\, P(\boldsymbol{O}_{j,k} \mid \zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i}, \boldsymbol{B}). \tag{13}$$

If statistical independency of point features is assumed, then the probability of observing a set of features can be computed by multiplying the single probabilities of (13).

Let us now postulate that the two–dimensional distribution of the described features is the result of the transformation and projection of the three–dimensional features $\boldsymbol{C}_\kappa$, i. e. a mapping of three–dimensional Gaussian distributed random vectors. If the projection is orthogonal, i. e. the complete mapping of a three–dimensional model variable $\boldsymbol{C}_{\kappa,i}$ into the observed scene feature $\boldsymbol{O}_{j,k}$ can be described by an affine transform

$$\boldsymbol{O}_{j,k} \;=\; \boldsymbol{R}\,\boldsymbol{C}_{\kappa,i} + \boldsymbol{t}, \tag{14}$$

where $\boldsymbol{R} \in \mathbb{R}^{2\times 3}$ and $\boldsymbol{t} \in \mathbb{R}^2$, the resulting random vector is again normally distributed. Let $\boldsymbol{\mu}_i$ and $\boldsymbol{K}_i$ be the mean and covariance of the i–th component of the 3D Gaussian mixture density. For the transformed random variables the following is valid: $\boldsymbol{O}_{j,i}$ is normally distributed with the mean $\boldsymbol{R}\boldsymbol{\mu}_i + \boldsymbol{t}$ and the covariance $\boldsymbol{D}_i := \boldsymbol{R}\boldsymbol{K}_i\boldsymbol{R}^T$ [11]. Now we have introduced a rudimentary statistical model for our learning problem: The projected features are modeled by mixture densities and each view j provides an affine mapping characterized by $\boldsymbol{R}_j$ and $\boldsymbol{t}_j$. The parameter set to be estimated are the weights $P(\zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i})$, means $\boldsymbol{\mu}_i$, and covariances $\boldsymbol{K}_i$ of each component of the mixture density function, $\boldsymbol{B} = \{P(\zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i}), \boldsymbol{\mu}_i, \boldsymbol{K}_i \,|\, 1 \le i \le n_\kappa\}$. The next step in the application of the missing information principle is the determination of the observable and missing information, and the definition of the probability functions needed for computing the Kullback–Leibler statistics for this application. Obviously, the set of 2D point features $\boldsymbol{O}_j$ is observable for each view. The known rotation and translation of the object in the image has not to be summarized as an observation, because they are not modeled as random variables. Hidden for each view is the the set of assignments of model and scene points. Consequently, we have

$$P(\boldsymbol{O}_{j,k}, \zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i} \,|\, \boldsymbol{B}) \;=\; \frac{P(\zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i})}{\sqrt{\det 2\pi\, \boldsymbol{D}_{i,j}}}\, e^u, \tag{15}$$

where $u = -\frac{1}{2}\,(\boldsymbol{O}_{j,k} - \boldsymbol{R}_j\boldsymbol{\mu}_i - \boldsymbol{t})^T\, \boldsymbol{D}_{i,j}^{-1}\, (\boldsymbol{O}_{j,k} - \boldsymbol{R}_j\boldsymbol{\mu}_i - \boldsymbol{t})$, and

$$P(\zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i} \,|\, \boldsymbol{O}_{j,k}, \boldsymbol{B}) \;=\; \frac{P(\boldsymbol{O}_{j,k}, \zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i} \,|\, \boldsymbol{B})}{\sum_{i=1}^{n_\kappa} P(\boldsymbol{O}_{j,k}, \zeta_\kappa(\boldsymbol{O}_{j,k}) = \boldsymbol{C}_{\kappa,i} \,|\, \boldsymbol{B})}. \tag{16}$$

Using the Dwyer–Macphail matrix derivative calculus [16], the zero crossings of the gradient of the associated Kullback–Leibler statistics can be computed with respect to the unknown parameters. As in the previous case of subsection 4.1, we get three types

of estimation formulas for the unknown parameters. For the weights and the means we obtain closed form solutions, see (17) and (18);

$$\hat{P}(\boldsymbol{C}_{\kappa,l}) \;=\; \frac{1}{J\,m_j}\sum_{j=1}^{J}\sum_{k=1}^{m_j} P(\boldsymbol{C}_{\kappa,l}\,|\,\boldsymbol{O}_{j,k},\boldsymbol{R}_j,\boldsymbol{t}_j,\boldsymbol{a}_{\kappa,l}), \tag{17}$$

$$\begin{aligned}\hat{\boldsymbol{\mu}}_i \;=\; & \left(\sum_{j=1}^{J}\sum_{k=1}^{m_j} P(\boldsymbol{C}_{\kappa,i}\mid \boldsymbol{O}_{j,k},\boldsymbol{a}_{\kappa,i})\boldsymbol{R}_j^T\boldsymbol{D}_{i,j}^{-1}\boldsymbol{R}_j\right)^{-1} \\ & \sum_{j=1}^{J}\sum_{k=1}^{m_j} P(\boldsymbol{C}_{\kappa,i}\mid \boldsymbol{O}_{j,k},\boldsymbol{a}_{\kappa,i})\boldsymbol{R}_j^T\boldsymbol{D}_{i,j}^{-1}\left(\boldsymbol{O}_{j,k}-\boldsymbol{t}_j\right).\end{aligned} \tag{18}$$

We define $\boldsymbol{S} = (\boldsymbol{O}_{j,k} - \boldsymbol{R}_j\boldsymbol{\mu}_i - \boldsymbol{t}_j)(\boldsymbol{O}_{j,k} - \boldsymbol{R}_j\boldsymbol{\mu}_i - \boldsymbol{t}_j)^T$ and get the following non–linear equation for the computation of the covariance matrices:

$$\sum_{j=1}^{J}\sum_{k=1}^{m_j} P(\boldsymbol{C}_{\kappa,i}\mid \boldsymbol{O}_{j,k},\boldsymbol{a}_{\kappa,i})\boldsymbol{R}_j^T\hat{\boldsymbol{D}}_{i,j}^{-1}(\hat{\boldsymbol{D}}_{i,j}-\boldsymbol{S})\hat{\boldsymbol{D}}_{i,j}^{-1}\boldsymbol{R}_j = 0. \tag{19}$$

These formulas admit the training of 3D objects from 2D views presupposed a good segmentation algorithm for detecting vertices is available and the capability of computing the objects pose for each view is given.

3.3 Estimation of Pose Parameters

Aside from the problem of learning from examples, the recognition and localization of objects is another central requirement of a recognition system. In the following, the issue of computing the position and orientation of a known object in a scene including heterogeneous background is done by applying the missing information principle. Analogous to the learning process, a three–dimensional object is modeled using a n–dimensional Gaussian mixture density function, where n is the dimension of the model features. Again, the features might vertices, for instance. In contrast to the previous section, the means, covariances, and weights are known parameters and the components of the rotation and translation constitute the parameter set $\boldsymbol{B}$, which has to be estimated throughout the EM–iterations. The observable image of the scene includes both features of the object and the background. All features corresponding to the background are assigned to the special model feature $\boldsymbol{C}_{\kappa,0}$. Following the results described in [15] the background features are

uniformly distributed; for the abstract mathematical formulation, we generalize that the background features underly an arbitrary distribution, which has to be independent of the rotation and translation parameters.

The known information is the set of two–dimensional point features. Unknown is again the matching of model and image features, which also indicates the partition of background and object features. The a priori probability of observing one element $\boldsymbol{O}_i$ of a set of image features thus is

$$\begin{aligned} P(\boldsymbol{O}_i \,|\, \boldsymbol{B}) &= \sum_{l=0}^{m_j} P(\boldsymbol{O}_i, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l} \,|\, \boldsymbol{B}) \\ &= P(\zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,0})\, P(\boldsymbol{O}_i \,|\, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,0}, \boldsymbol{B}) \\ &\quad + \sum_{l=1}^{m_j} P(\zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l})\, P(\boldsymbol{O}_i \,|\, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l}, \boldsymbol{B}), \end{aligned} \tag{20}$$

This results in the following Kullback–Leibler statistics

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{i=1}^{m} \sum_{l=0}^{n_\kappa} \frac{P(\boldsymbol{O}_i, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l} \,|\, \boldsymbol{B})}{P(\boldsymbol{O}_i \,|\, \boldsymbol{B})} \log P(\boldsymbol{O}_i, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l} \,|\, \hat{\boldsymbol{B}}). \tag{21}$$

The next fundamental problem is the selection of a suitable optimization technique. Due to the fact that the function for optimization has a lot of local extrema, local gradient techniques will a priori not be applicable for the computation of the maximum. Furthermore, the maximization problem does not fall into optimization problems in lower dimensional search spaces like in previous applications. Therefore, it is suggested to use global, iterative optimization techniques within the EM–iterations and the estimation of pose parameters can be decomposed into two iterations.

An interesting side effect results from the fraction $P(\boldsymbol{O}_i, \zeta(\boldsymbol{O}_i) = \boldsymbol{C}_{\kappa,l} \,|\, \boldsymbol{B}) / P(\boldsymbol{O}_i \,|\, \boldsymbol{B})$, which obviously yields a probability measure for the unknown matching between model and scene features.

4 Experimental Results

The application of the missing information principle results in the described iterative algorithms of section 3, which are implemented on a HP 735 in C++ for experimental evaluations using the object oriented image processing system ἵππος, described in [9]. The following subsections briefly summarize and discuss the achieved results.

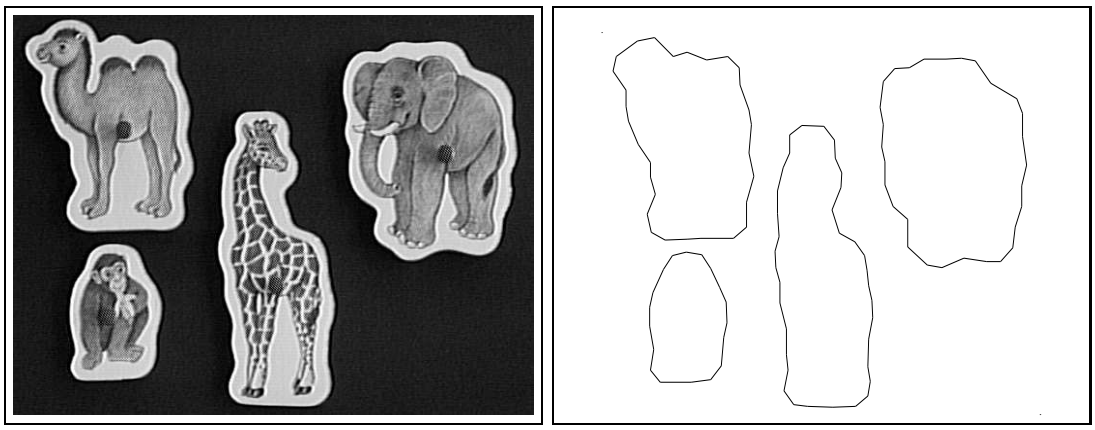

Figure 4: Original gray–level image (children toys) used for the experimental evaluation (left) and resulting closed polygons of the contour (right). Polygons are used for the computation of the affine invariant feature sequences.

object	number of states				
	3	4	5	6	7
monkey	9	8	9	8	8
giraffe	7	7	7	8	8
elephant	8	8	8	5	4
camel	5	5	5	3	5
rate in %	72	72	72	60	62

Table 1: Recognition results using Hidden Markov Models with different numbers of states

4.1 Hidden Markov Models for Object Recognition

In the first part of our experiments we implemented and tested Hidden Markov Models for 2D object recognition problems. The basic constraint for the use of Hidden Markov Models is the limitation to classification problems, where objects can be represented by feature sequences. We decided to use affine invariant features, described in [6], based on closed contour lines of 2D objects. Hence, the sequence of features does not change if the object is rotated and translated. We took four objects, shown in Figure 4, and trained Hidden Markov Models with differing numbers of states using 50 samples for each object. The classification results for 10 images of each object are shown in Table 4.1. For an efficient computation of the a priori probabilities for a given observation and a Hidden

Markov Model we use the forward–backward algorithm [10].

The disappointing recognition rates are not based on the chosen features. In [6] it is shown that the correct classifications increase, if we leave out the ordering on the feature sequence.

4.2 Training and Localization of 3D Objects

In contrast to Hidden Markov Models, the use of mixture density functions for object learning, recognition, and localization purposes is a new technique. The described algorithms are actually tested using synthetical data. For the estimation of parameter set including means, covariances, and weights, an initialization of the density function for each feature is required. The number of features and initial estimates of parameters have to be given before the EM–iterations can start (see Figure 2). Presently, we use views where no occlusion occurs. For simple polyedric objects the method produces satisfactory results, if we determine the number of needed object features using one view. The mean vectors are initialized by the observable 2D point features, where the depth value is defined to be zero. Empirically, 40–50 views are sufficient for learning an object, which is represented with 15 characteristic features. Although the convergence rate of the EM–algorithm was expected to be considerably low (see [3]), the learning process converges after 10 iterations, in average. The time needed for one iteration using a C++ implementation of the learning formula (18), which is suitable for arbitrary dimensions of feature vectors, takes 97.98 seconds with 50 training views. The memory requirements are constant for each iteration.

The experiences with methods for pose estimation using the computed density functions showed that the EM approach is only suitable for refinements of good initial pose parameters. For the localization of objects where no a priori information of the object's pose is available, the EM–algorithm yields translation and rotation parameters of no use, even if global optimization techniques are used within each EM–iteration. One conceivable application of EM–iterations for pose estimates might be the localization in image sequences, where the initial pose of an object is given by the object's pose in the previous image [4].

Promising results are achieved by applying an adaptive random search technique described in [14] to the log likelihood function (2) for the observable data. We trained the density function for 5 different objects with 15 features and used these results for evaluating the algorithms for computing the position and orientation. The pose estimates

for artificially rotated and translated objects succeeded in all tested cases. The actual implementation needs about 10 minutes to find the global maximum of the multivariate functions, which depend on three rotation angles and both components of the translation vector.

5 Future Work

The promising approach to treat the 3D object recognition problem using gray–level images will be used for realizing a system, which can learn and classify simple polyedric objects. For the implementation of the training stage we will use a robot, where a camera is mounted on its hand. This device will admit the computation of pose parameters for each training view. A brief introduction into the actual realized components of the system can be found in [5]. Theoretical work has to be done with respect to the optimization techniques of the estimation of pose parameters.

6 Conclusions

This contribution introduces the missing information principle and shows how this technique can be applied to different computer vision tasks. Characteristically, it is shown that the Hidden Markov Models, which are intensively used in the field of speech recognition, are based on the same theoretical foundation like the new statistical approach to deal with the 3D object recognition problem introduced in subsections 3.2 and 3.3. It should be emphasized that the learning procedure for 3D objects from two–dimensional avoids an explicit matching between features.

The experimental results show, that the EM–algorithm is not unlimited suitable for all appearing incomplete data estimation problems (subsection 3.3). For improving the three–dimensional object recognition system a more sophisticated statistical model for 3D objects might be useful, because statistical dependencies between several features and occlusion are not modeled in the actual framework. Finally, the extension to more general features like lines or polygons will be as necessary as the implementation of training and recognition formulas for perspective projection.

References

1. L. E. Baum and J. A. Eagon. An inequality with applications to statistical prediction for functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*,

73:360–363, 1967.

2. F. Campillo and F. Le Gland. MLE for partially observed diffusions: Direct maximization vs. the EM–algorithm. Technical Report 884, Institut National de Recherche en Informatique et en Automatique, Le Chesnay Cedex (France), August 1988.
3. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.
4. J. Denzler, R. Beß J. Hornegger, H. Niemann, and D. Paulus. Learning, tracking and recognition of 3D objects. In V. Graefe, editor, *International Conference on Intelligent Robots and Systems – Advanced Robotic Systems and Real World*, to appear September 1994.
5. J. Hornegger and H. Niemann. A Bayesian approach to learn and classify 3–D objects from intensity images. In *Proceedings of the* 12^{th} *International Conference on Pattern Recognition (ICPR)*, Jerusalem, to appear October 1994. IEEE Computer Society Press.
6. J. Hornegger, H. Niemann, D. W. R. Paulus, and G. Schlottke. Object recognition using Hidden Markov Models. In *Pattern Recognition in Practice IV*. Elsevier, Amsterdam, to appear July 1994.
7. D. W. Jacobs. *Recognizing 3D Objects using 2D Images*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts, 1992.
8. H. Niemann. *Klassifikation von Mustern*. Springer, Heidelberg, 1983.
9. D.W.R. Paulus. *Objektorientierte und wissensbasierte Bildverarbeitung*. Vieweg, Braunschweig, 1992.
10. L.R. Rabiner. Mathematical Foundations of Hidden Markov Models. In H. Niemann, M. Lang, and G. Sagerer, editors, *Recent Advances in Speech Understanding and Dialog Systems*, volume 46 of *NATO ASI Series F*, pages 183–205. Springer, Heidelberg, 1988.
11. C. R. Rao. *Linear Statistical Inference and its Applications*. Wiley Publications in Statistics. John Wiley & Sons, Inc., New York, 1973.
12. R.A. Redner and H.F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *Society for Industrial and Applied Mathematics Review*, 26(2):195–239, 1984.
13. M. A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, volume 67 of *Springer Series in Statistics*. Springer, Heidelberg, 2 edition, 1993.
14. A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Heidelberg, 1987.
15. W. M. Wells III. *Statistical Object Recognition*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Massachusetts, February 1993.
16. W. J. Wrobleski. *Extensions of the Dwyer–Macphail Matrix Derivative Calculus with Applications to Estimation Problems involving Errors–in–Variables and Errors–in–Equations*. PhD thesis, University of Michigan, Michigan, 1963.
17. C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.