A Novel Probabilistic Model for Object Recognition and Pose Estimation

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

Abstract

In this paper we consider the problem of object recognition and localization in a probabilistic framework. An objects is represented by a parametric probability density, and the computation of pose parameters is implemented as a nonlinear parameter estimation problem. The presence of a probabilistic model allows for recognition according to Bayes rule. The introduced probabilistic model requires no prior segmentation but characterizes the statistical properties of observed intensity values in the image plane. A detailed discussion of the applied theoretical framework is followed by a concise experimental evaluation which demonstrates the benefit of the proposed approach.

1 Introduction

In high–level computer vision applications *recognition of objects* is a standard problem which must be solved reliably and efficiently. In addition, it may be necessary to estimate the *position* and *orientation* of objects with respect to a world coordinate system. Both recognition and localization require models which characterize the objects' appearance in the image plane. These models should be generated from sample images automatically and they have to generalize to arbitrary views under varying illumination conditions and occlusions. Model generation, recognition as well as pose estimation given a probabilistic framework can be considered in terms of standard pattern recognition and statistics:

model generation and pose estimation correspond to regression problems, recognition to classification.

In this paper we will treat object recognition and pose estimation as *optimization problems*. Recognition is considered as a problem of statistical decision theory; localization is defined as a parameter estimation problem. In order to perform recognition and localization, an adequate *model* is required which is determined by two major components: the *structure* and the *parameters*. Besides recognition and localization also the automatic construction of models can be treated as an optimization problem.

Although there are quite a few different approaches to modeling relevant properties of objects, we will only consider *probabilistic models* here. The reasons are: sensor signals and associated features show a probabilistic behavior due to sensor noise, varying illumination conditions or segmentation errors; it allows a unified mathematical formulation by providing a framework for combining evidence; statistical decision theory and estimation theory have a sound basis and provide many useful results; probabilistic approaches had a certain impact to various areas, in particular in speech recognition. Some references are, for example, [1, 4, 6, 7].

2 Basic Vision Problems in Terms of Statistics

A digital image $\boldsymbol{f}$ is considered as a matrix of discrete intensity values $\boldsymbol{f} = [f_{j,k}]_{1 \leq j,k \leq M}$; it was assumed here that the number of pixels is M in x– and y–direction.

The task of object recognition, that is, the discrete mapping of images to pattern classes, is a composition of various labeling (respectively classification) processes. The mapping from the original image to discrete classes is mostly subdivided into the following stages:

1. *Preprocessing:* in the preprocessing stage images are filtered in order to remove noise and enhance useful properties of the image.

2. *Segmentation:* the segmentation maps the image matrix to a matrix which defines, for instance, geometric primitives, features, or in general observations $\boldsymbol{O} = \{\boldsymbol{o}_k | k = 1, \ldots, m\}$.

3. *Classification:* the final classification stage maps segmentation results to classes $\Omega_\kappa \in \Omega = \{\Omega_1, \ldots, \Omega_K\}$. Classes are obtained from the requirements of the task–domain.

It is recalled from statistical decision theory that the *Bayes classifier* allows classification with minimal error probability, provided an appropriate statistical model of the

classes is given. This model in general is a class–conditional probability density $p(\boldsymbol{O}|\boldsymbol{B}_\kappa)$ of a set of observations $\boldsymbol{O}$ given a class Ω_κ characterized by the parameters $\boldsymbol{B}_\kappa$. We call the parametric function $p(\boldsymbol{O}|\boldsymbol{B}_\kappa)$ model density. A simple example for a model density is the Gaussian density, where $\boldsymbol{B}_\kappa$ represents the mean vector and the covariance matrix.

Besides the recognition of objects, the position and orientation of objects provide important information. For instance, to make a robot grasp an object, the pose parameters have to be known. As introduced in [2], the probability density function which models an object and which should allow for both classification and pose estimation has to use pose dependent parameters and to include a second set of parameters. These additional degrees of freedom represent pose parameters. If the model also contains information about the localization (or pose) in a parameter $\boldsymbol{\theta}$, the model density has the general form $p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{\theta})$. The steps towards a complete recognition system would then be, first, to compute the maximum likelihood estimate of class specific parameters $\boldsymbol{B}_\kappa$ given the pose parameters from N observations, second, to estimate the pose parameter $\boldsymbol{\theta}_\kappa$ per class, third, to classify by Bayes rule. The resulting equations are:

$$\widehat{\boldsymbol{B}}_\kappa = \text{argmax}_{B_\kappa}\, p({}^1\boldsymbol{O}, \ldots, {}^N\boldsymbol{O}|\boldsymbol{B}_\kappa, {}^1\boldsymbol{\theta}_\kappa, \ldots, {}^N\boldsymbol{\theta}_\kappa)\ , \quad \kappa = 1, \ldots, K\ , \tag{2.1}$$

$$\widehat{\boldsymbol{\theta}}_\kappa = \text{argmax}_\theta\, p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{\theta})\ , \quad \kappa = 1, \ldots, K\ , \tag{2.2}$$

$$\Omega_\kappa = \text{argmax}_{\Omega_\lambda}\, p(\Omega_\lambda \mid \boldsymbol{O}) = \text{argmax}_{\Omega_\lambda}\, \frac{p(\boldsymbol{O}|\boldsymbol{B}_\lambda, \boldsymbol{\theta}_\lambda)p(\Omega_\lambda)}{p(\boldsymbol{O})}\ . \tag{2.3}$$

For numerical reasons and to avoid number underflows, it is more convenient to maximize the logarithm of probability densities.

3 Probabilistic Modeling of Images

One important question in probabilistic modeling is, where the noise appears. In [5] it is shown that the noise model has to operate in the image plane. Probabilistic models of images mostly make use of the Hemmersley Clifford theorem [9] and use Gibbs distributions to characterize images by Markov random fields. The automatic estimation of the Gibbs distribution from sample images and the application of Markov random fields for pose estimation and recognition is mostly unsolved still part of intensive research. As an alternative we introduce a new modeling scheme for images which is substantially based on products of mixture densities.

The basis for statistical modeling of images is to consider the image matrix $\boldsymbol{f} = [f_{j,k}]$ as a random matrix. Each entry $f_{j,k}$ is characterized by the components position in the image and pixel value (gray or color value). Similar to mathematical morphology we

interpret the image matrix as a set of random vectors

$$S = \left\{ [j, k, f_{j,k}]^T \middle| \, 1 \leq j, k \leq M \right\} \quad . \tag{3.1}$$

The 2–D grid points and intensity values are defined as potential random measures.

The random vectors can be characterized by a conditional probability density function, which depends on the present pattern class Ω_κ. Since the appearance of objects in the image plane changes with the objects' pose, the density will also be parameterized regarding the pose $\boldsymbol{\theta}$. For the whole image the probability density is

$$p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{\theta}) = p\left(\left\{ [j, k, f_{j,k}]^T \middle| \, 1 \leq j, k \leq M \right\} \middle| \, \kappa; \boldsymbol{\theta} \right) \ . \tag{3.2}$$

According to the intended generality, the set of observations $\boldsymbol{O}$ in the above equation is the set S. The parameter vector $\boldsymbol{B}_\kappa$ is replaced by κ since no parametric form was assumed yet.

This model density is far too general, not computationally feasible, and too abstract for applications. Nevertheless, it can be specialized to a broad class of model densities. The introduction of additional constraints, the consideration of dependencies of bounded order, the incorporation of specializations, and the usage of continuous instead of discrete random variables are basic tools which will induce feasible models and reduced parameter sets. We consider the intensity values to be discrete random measures. The pixel coordinates j and k are assumed to be continuous. Now we consider the probability density functions of image points dependent on intensities. A possible decomposition of (3.2) is now

$$p([j, k, f_{j,k}]^T|\kappa, \boldsymbol{\theta}) = p(f_{j,k}|\kappa)\, p(j, k|f_{j,k}; \kappa, \boldsymbol{\theta}) \quad , \tag{3.3}$$

where the term $p(f_{j,k}|\kappa)$ represents the discrete probability to observe the intensity value $f_{j,k}$. The other factor $p(j, k|f_{j,k}; \kappa, \boldsymbol{\theta})$ characterizes the bivariate probability density of image points dependent on a given intensity value $f_{j,k}$. Obviously, besides the class index κ this probability has to incorporate the pose parameter $\boldsymbol{\theta}$) because the spatial distribution of gray level varies with the object's pose. Given the assumption that relative frequencies of intensity values do not depend on pose parameters, this probability is independent of the position and orientation defined by $\boldsymbol{\theta}$. Assuming mutually independent intensity values and image points, we get the overall density

$$p(\boldsymbol{f}|\kappa, \boldsymbol{\theta}) = \prod_{j=1}^{M} \prod_{k=1}^{M} p(f_{j,k}|\kappa)\, p(j, k|f_{j,k}; \kappa, \boldsymbol{\theta}) \tag{3.4}$$

for the complete image.

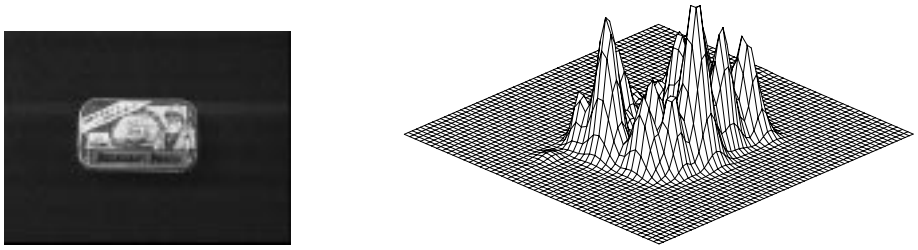

Figure 3.1: Gray–level image and the multi–modal probability density function of a selected intensity value.

Let us assume that L_f intensity levels $f^{(l)}, l = 1, \ldots, L_f$ are distinguished. The intensity level $f^{(l)}$ may be an interval of gray values or also a single gray value. In the experiments we will use fairly large intervals resulting in only 4 intensity levels. They are determined by standard vector quantization using entropy based objectives. Since a selected intensity level $f_{j,k} = f^{(l)}$ often appears in different places of the image, the density $p(j,k|f_{j,k} = f^{(l)}; \kappa, \boldsymbol{\theta})$ is expected to be multi–modal. Density functions with multiple modes are most commonly approximated by mixtures, i.e. convex combinations of uni–modal densities [6].

Figure 3.1 gives an example of one factor of such a model density as defined by (3.6). The parametric form used for $p(j,k|f_{j,k} = f^{(l)}; \kappa, \boldsymbol{\theta})$ in this example is a normal mixture density consisting of $i = 1, \ldots, L_g^{(l)} = 17$ Gaussian densities $\mathcal{N}(\boldsymbol{\mu}_{\kappa,l,i}, \boldsymbol{\Sigma}_{\kappa,l,i})$ for a class Ω_κ at an intensity level $f^{(l)}$. If pose parameters consist of a translation, represented by the vector $\boldsymbol{t} \in \mathbb{R}^2$, and a rotation in the plane represented by an orthogonal matrix $\boldsymbol{R}$, we obtain the conditional density

$$p(j,k|f_{j,k} = f^{(l)}; \kappa, \boldsymbol{\theta}) = \sum_{i=1}^{L_g^{(l)}} p_{\kappa,l,i} \mathcal{N}\left([j,k]^T; \boldsymbol{R}\boldsymbol{\mu}_{\kappa,l,i} + \boldsymbol{t},\ \boldsymbol{R}\boldsymbol{\Sigma}_{\kappa,l,i}\boldsymbol{R}^T\right) . \tag{3.5}$$

Since the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$ is uniquely defined by a single rotation angle α, the space of pose parameters is three–dimensional.

Assuming now mutual independence of random vectors $[j, k, f_{j,k}]$, the joint probability density function of the complete image showing class Ω_κ is given by the following product

$$p(\boldsymbol{f}|\kappa, \boldsymbol{\theta}) = \prod_{j=1}^{M}\prod_{k=1}^{M}\prod_{l=1}^{L_f} p(f_{j,k} = f^{(l)}|\kappa)\, p(j,k|f_{j,k} = f^{(l)}; \kappa, \boldsymbol{\theta}) \tag{3.6}$$

or in terms of logarithms:

$$\log\, p(\boldsymbol{f}\,|\kappa,\boldsymbol{\theta}) = \sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{L_f} \log\left(p(f_{j,k}=f^{(l)}|\kappa)\, p(j,k|f_{j,k}=f^{(l)};\kappa,\boldsymbol{\theta})\right). \qquad (3.7)$$

This probabilistic model characterizes 2–D objects adequately. It provides a simple and effective way to model irregular and textured objects (e.g. a cactus), and to localize and recognize objects where usually algorithms based on geometric models fail.

4 Model Generation

The explicit construction of model densities by human interaction is intractable for practical applications. Model densities should be generated from sample data automatically. The introduced model (3.6) provides several degrees of freedom. For instance, if mixture densities are used for modeling, we have to fix the quantization of intensity levels and to compute the number of mixture components for each interval of gray values. Once these measures are known, we have to estimate the parameters of mixtures. This example demonstrates that the learning of probabilistic models includes both automatic acquisition of the *structure* and the *parameters* of the model based on empirical data. In the above example the structure is basically defined by the number of mixture components, and parameter estimation corresponds to the computation of the mixture parameters.

The common principle of structural and parametric learning can be summarized as follows:

- define a (discrete or continuous) search space,
- choose an objective function which scores the actual structure or parameter set,
- use a search algorithm which guides the navigation in the search space, and
- terminate learning, if no improvement occurs or the improvement is below a certain threshold.

In spite of the existence of this general framework, the overall complexity of structural and parametric learning is completely different. While the estimation of parameters usually corresponds to optimization problems of continuous functions (see e.g. (2.1)), structural optimization implies a search problem in a combinatorial space of exponential size. In the following we will consider both structural and parametric learning based on model density (3.6).

4.1 Structural Learning

Structural learning requires the computation of the number of mixture components. For that purpose we apply a method which is widely applied in speech recognition [4]: vector quantization. Using the k–means algorithm we can determine the clusters of intensity values; the number of clusters corresponds to the number of mixture components. This number is denoted by l. Vector quantization is based on selecting randomly l initial cluster centers. These initial centers are updated in such a way that after a number of iterations they represent the clusters in the data as much as possible. An obvious disadvantage of this strategy is that the number of clusters is fixed. Once l is defined the algorithm will always return l cluster centers. We have to remove redundant clusters. Whenever a cluster center is not assigned enough samples, it is canceled and merged with the closest cluster. This results in a set of centers which define a more or less optimal number of clusters. The problem of choosing the initial number of clusters still remains. Defining the initial l large enough depends on the given sample set, but is usually no problem.

An illustration of the Voronoi diagram resulting from the vector quantization step for automatic model generation can be found in Figure 4.1. The number of mixture components in this is example is 10. The other 7 clusters are assigned to the background.

Figure 4.1: Voronoi diagram resulting from vector quantization (left) and the convex combination of Gaussians (right) for a single intensity value

4.2 Parameter Learning

Given the number of intensity values and the structure of mixtures, learning in the second stage reduces to a *parameter estimation* problem. Since we use statistical models as

introduced in (3.5) and (3.6), the learning of model densities requires the estimation of the following parameters for *each* class Ω_κ and *each* intensity level $f^{(l)}$:

- discrete probabilities $p(f^{(l)}|\kappa)$,
- $L_g^{(l)}$ discrete probabilities $p_{\kappa,l,i}$,
- $L_g^{(l)}$ mean vectors $\boldsymbol{\mu}_{\kappa,l,i}$, and
- $L_g^{(l)}$ covariance matrices $\boldsymbol{\Sigma}_{\kappa,l,i}$.

The discrete probabilities $p(f|\kappa)$ are just relative frequencies. The other parameters which characterize the mixture density (3.5) are initialized using the cluster which result from vector quantization (see also Section 4.1). The iterative refinement of the initial estimates applies the the *expectation maximization algorithm* (EM algorithm) [6]. Here, the EM algorithm is required because the assignment of observations to mixture components is not part of the training data. The re-estimation formulas can be found in [6].

5 Maximum–Likelihood Localization

Once the model parameters are known, the statistical models can be used to localize and classify objects according to (2.2) and (2.3). The localization of objects in the chosen probabilistic framework corresponds to a maximum–likelihood estimation problem. This is especially considered to be advantageous compared to standard least square or least median methods. Least square or median estimators are known to imply biased estimates in most cases. In contrast, maximum–likelihood estimators guarantee consistency, *i.e.*, the expectation of estimates converges (at least theoretically) against its true value for large sample size.

Despite of this theoretically proven advantage, the practical computation of the maximum–likelihood estimate is hard since we are looking for a global maximum of a multi–modal (log) likelihood function (see also Figure 6.2). An exhaustive grid search to estimate the pose parameters is computationally infeasible. Here we suggest to use marginals of model densities to speed up the global search. Projections of random variables reduce the dimensionality of the search space. The optimization of the multi–modal model density is based on a three–stage maximization process as originally introduced in [2].

The considered random vectors are triples $[j, k, f_{j,k}]$. Their probability density function is defined by (3.6). For a selected gray–level f we can compute marginals which allow the definition of hierarchical probabilistic models. The considered marginals are:

$$p([j,f]|\kappa,\boldsymbol{\theta}) \;=\; \int_k p([j,k,f]|\kappa,\boldsymbol{\theta})\,dk \quad , \quad \text{and} \tag{5.1}$$

$$p([k,f]|\kappa,\boldsymbol{\theta}) \;=\; \int_j p([j,k,f]|\kappa,\boldsymbol{\theta})\; dj \quad . \tag{5.2}$$

Remarkably, these marginals with respect to image point coordinates j and k induce the invariance of densities with respect to parts of the pose parameters. The marginal (5.1) does not depend on translations along the y–axis, and the integral (5.2) eliminates the parameter t_x of the original translation vector. We make use of this important observation within the optimization module and implement a three–stage maximization procedure:

1. We compute a set H of local maxima (α, t_y) of the bivariate density (5.1).

   For this global optimization problem we apply a grid search technique based on 40×70 equidistant sample points of the 2–D parameter space. At each grid location we start local optimizations using the downhill simplex algorithm.

2. We take the rotation angles of $(\alpha, t_y) \in H$, maximize (5.2) with respect to t_x, and get a list L of triples (α, t_x, t_y).

3. The elements of L are used as initializations for local optimizations of the original model density (3.6).

The following experimental evaluation will show the efficiency of this global optimization algorithm which use marginals to decompose the search space.

6 Experiments

The description of the experimental evaluation of the proposed probabilistic approach is divided up into four subsections. First we describe the experimental setup. This includes the definition of considered pattern classes, training and test sets, applied measurements, and the used hardware. The other sections treat the three stages model generation, pose estimation, and classification as defined by (2.1), (2.2), and (2.3):

6.1 Experimental Setup

In the experiments we consider four different objects. Each object is assigned to a different class. We have selected objects which allow no robust segmentation of point features. Recognition algorithms which are based on geometric transformations of point features, for instance, will definitely fail for these objects [3, 2]. Figure 6.1 shows images of the objects used in the experiments. For the estimation of model parameters we have 50 images of each object. The background is homogeneous. The localization and classification

experiments use another 200 images, 50 of each object. Of course, training and test sets are disjoint. The correct pose parameters of the test images are known, because we have used a calibrated turn table to generate different views. The given reference values are used for the evaluation of the localization algorithm. The resolution of the gray–level images is 320×240 pixels. The originally 256 gray–levels are quantized to four intensity values. One intensity value characterizes the background, the other three gray–levels are assigned to the object. The quantization is done by the maximization of an entropy based criterion (c.f. [8], Chapter 20). The training modules use all pixels to estimate the model parameters. The localization algorithm, however, gets only 128 image points for the first series of experiments, and 512 for the second. The recognition experiments are based on 512 sampling points. The selection of image points is done uniformly using 128 resp. 512 equidistant grid points. Due to this sampling, the runtime system uses *only* 0.16% resp. 0.66% of the available image data. All experiments run on a Silicon Graphics O2 (R10000, 195 MHz).

Figure 6.1: Objects used for localization and recognition experiments: toy car, cactus, candy box, beetle

6.2 Estimation of Model Parameters

The parameters of the model densities are computed using the vector quantization method and the parameter estimation techniques as introduced in Section 4. Each model density is a product of M^2 weighted mixtures of Gaussians according to (3.6). Due to the chosen quantization of gray–levels, we have to compute for each of the four gray–levels $f^l, l = 1, \ldots, 4$ the weight factors $p(f^l|\kappa)$ and the mixtures $p(j, k|f^l; \kappa, \boldsymbol{\theta})$. Table 1 summarizes the total number of 2–D Gaussians used in the probabilistic models. Figure 4.1 illustrates the result of vector quantization and of the final density estimation for a single gray–level. In this case, we need 10 Gaussians to model the spatial distribution of a certain gray–level.

Reliable estimates of model parameters depend on sufficient sets of training samples which are available in our case: Given the image resolution and the available set of train-

object	Gaussians
toy car	24
cactus	53
candy box	46
beetle	85

Table 1: Number of mixture components per object

ing images for each class, the total number of samples used for parameter estimation of each class is $50 \cdot 320 \cdot 240 \approx 4 \cdot 10^6$. In case of the beetle, these observations are used to compute $4 + 85(1 + 2 + 3) = 514$ parameters.

6.3 Localization

Object localization corresponds to the problem of computing the position and orientation. Here we use the automatically generated models and determine position and orientation by a maximum–likelihood estimate (2.2). The computation of the rotation angle and the translation vector is based on the three–stage search procedure as introduced in Section 5. Figure 6.2 shows the multi–modal functions which have to be optimized. Obviously we need global maximization techniques to compute the parameters we are looking for.

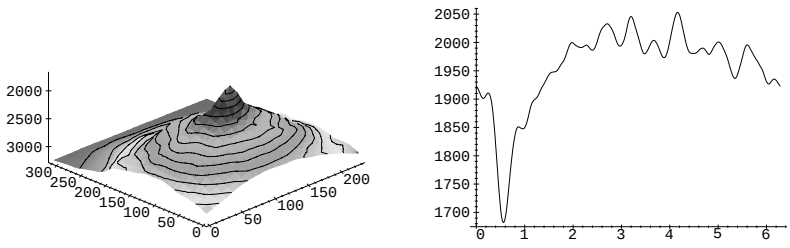

Figure 6.2: Objective functions for localization: the optimal translation parameters are defined by the maximum of the bivariate function on the left, the rotation angle by the global maximum on the right

Table 2 shows the standard deviation σ_α of the estimated rotation angle α, which

is measured in degrees, and the standard deviations σ_{t_x} and σ_{t_y} of the components of translation vector $\boldsymbol{t} = (t_x, t_y)^T$. The right column summarizes the mean runtime t_{mean} of pose estimations measured in seconds. A visualization of estimated pose parameters for one example is shown in Figure 6.3.

object	σ_α	σ_{t_x}	σ_{t_y}	t_{mean}
toy car	2.79	0.96	1.29	7.9
cactus	78.41	4.64	8.46	11.0
candy box	2.84	1.77	1.77	13.1
beetle	78.4	5.23	3.75	18.8

object	σ_α	σ_{t_x}	σ_{t_y}	t_{mean}
toy car	2.15	0.57	0.96	24.6
cactus	1.51	0.40	0.39	63.3
candy box	1.42	1.11	0.66	37.7
beetle	58.00	2.63	0.99	68.1

Table 2: Localization results using 128 (left) and 512 (right) equally sampled gray–levels

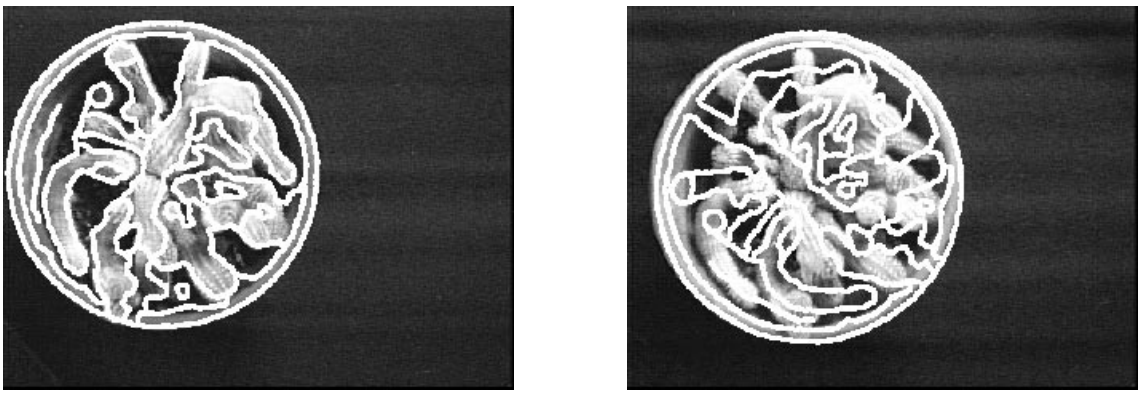

Figure 6.3: Correct and wrong results of the localization module; for visualization purposes a reference segmentation result is re–projected into the image using the estimated parameters.

The localization using 128 sample image points is remarkably precise for the toy car and the candy box. The low accuracy of the rotation angles of the cactus and the beetle show that the number of sample points is too low. An increase of the number of image points for localization (right part of Table 2) clearly decreases the variance of angles for the cactus and the beetle. The symmetry properties of the beetle, however, are the reason for the high deviations of the rotation angle estimates. Only a more detailed sampling of the image in those areas which resolve ambiguities in rotations will lead to estimates of higher accuracy.

A highly interesting question is the relationship between the number of mixture components, the number of pixels used for localization, and the accuracy of the resulting estimate. We consider the candy box. Figure 6.4 plots the average deviation of the estimated rotation angle in dependency of the number of mixtures and the cardinality of sample points. This plot shows that at least 128 uniformly distributed sample points are required for the estimation of the rotation angle.

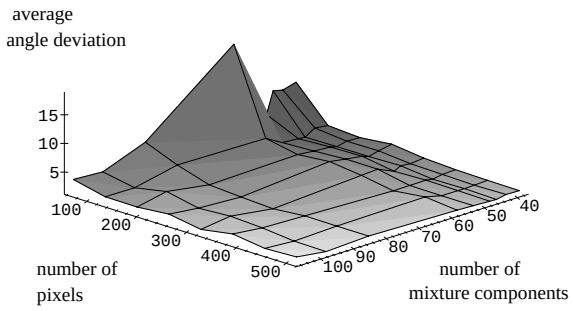

Figure 6.4: Localization results for the candy box using a different number of mixture components and sampling points

6.4 Recognition

The recognition experiments using 200 test images, 50 of each object, show a recognition rate of 100 % if four gray–levels are used. The overall runtime is less than three minutes (mean runtime 174.5 sec), where the computationally expensive part of the recognition module is the estimation of pose parameters. The most probable position and orientation of objects is required to evaluate the a posteriori probabilities (see (2.3)).

7 Summary and Conclusion

We have introduced a new modeling scheme which allows for statistical object recognition and localization. The proposed *Bayesian classifier* allows the unified incorporation of prior knowledge and class specific densities. It is remarkable that the implemented classifier requires no explicit computation of geometric features like points or lines. Just

object	recognition rate [%]
toy car	100
cactus	100
candy box	100
beetle	100

object	recognition rate [%]
toy car	100
cactus	100
candy box	100
beetle	90

Table 3: Recognition rates using 512 uniformly sampled gray–levels and three (left) resp. four intensity values (right)

the spatial distribution of intensity values is used. The theoretical results of this paper provide algorithms for automatic model generation, for pose computation, and for classification. The experimental evaluation proves that the proposed approach allows accurate estimations of pose parameters and recognition rates of 100 % for the given examples. An obvious and still open problem is incorporation of the projection mapping and the generalization of the proposed approach from 2–D to 3–D object recognition and pose estimation.

Acknowledgments

The authors gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG), SFB 603. Only the authors are responsible for the content of this paper.

References

[1] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory in Pattern Recognition*, volume 31 of *Applications of Mathematics, Stochastic Modelling and Applied Probability*. Springer, Heidelberg, 1996.

[2] J. Hornegger and H. Niemann. Statistical learning, localization, and identification of objects. In *Proceedings of the* 5^{th} *International Conference on Computer Vision (ICCV)*, pages 914–919, Boston, June 1995. IEEE Computer Society Press.

[3] D.P. Huttenlocher. Recognition by alignment. In A. K. Jain and P. J. Flynn, editors, *Three–Dimensional Object Recognition Systems*, pages 311–324. Elsevier, Amsterdam, 1993.

[4] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts, 1998.

[5] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*, volume 18 of *Machine Intelligence and Pattern Recognition*. Elsevier, Amsterdam, 1996.

[6] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, 1996.

[7] H. Niemann. *Pattern Analysis and Understanding*, volume 4 of *Springer Series in Information Sciences*. Springer, Heidelberg, 1990.

[8] D. Paulus and J. Hornegger. *Applied pattern recognition: A practical introduction to image and speech processing in C++*. Advanced Studies in Computer Science. Vieweg, Braunschweig, 2 edition, 1998.

[9] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Applications of Mathematics*. Springer, Heidelberg, 1995.