Statistical Learning, Localization, and Identification of Objects

Joachim Hornegger and Heinrich Niemann

The following paper appeareded in
Proceedings of the 5th International Conference on Computer Vision (ICCV)
Boston,
June 20–23 1995

Statistical Learning, Localization, and Identification of Objects

J. Hornegger and H. Niemann

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

Abstract

This work describes a statistical approach to deal with learning and recognition problems in the field of computer vision. An abstract theoretical framework is provided, which is suitable for automatic model generation from examples, identification, and localization of objects. Both, the learning and localization stage are formalized as parameter estimation tasks. The statistical learning phase is unsupervised with respect to the matching of model and scene features. The general mathematical description yields algorithms which can even treat parameter estimation problems from projected data. The experiments show that this probabilistic approach is suitable for solving 2D and 3D object recognition problems using grey–level images. The method can also be applied to 3D image processing issues using range images, i.e. 3D input data.

1 Introduction

Many object recognition systems are discussed in the literature [7, 8]. The huge variety of approaches are occupied with the development and realization of algorithms for learning, classification and localization of three–dimensional objects from different types of optical sensory data. Usually, specific paradigms are restricted to 3D, $2\frac{1}{2}$D, or 2D data [7]. An unconstrained application of these algorithms for arbitrary input data is, in general, not possible. Decision rules and model generation techniques often cannot be applied to different dimensions. Aside from these data dependent details, object recognition systems have to deal with instabilities and should be robust with respect to uncertainty, which is, for instance, caused by varying illumination, occlusion, noise, or segmentation errors. They are also expected to provide accurate classification results. Naturally, Bayesian classifiers consider uncertainties and fulfill an optimality criterion referring to misclassifications.

This paper describes a general uniform statistical framework for learning, localization, and classification of 2D or 3D objects using different optical data sources. The method enables even the automatic computation of object models from projections. However, instead of geometrical models, objects are represented by parameterized density functions of their features. Both, the learning and pose computation stage correspond to parameter estimation problems. The calculation of the involved parameters is either solved by the Expectation Maximization (EM) algorithm [3] or direct maximum likelihood estimations. In addition to the statistical modeling, we present a technique for decomposing the search space for pose estimates, which is based on transformations of the model density functions and the observations into several one–dimensional sub–spaces. A combinatorial explosion of the search space caused by an increasing number of observed scene features does not occur, since no explicit matching between scene and model primitives is required. The complexity of the pose estimation process is determined by the number of possible transforms from the model into the image space. The resulting image recognition system realizes a Bayesian – i.e. an optimal – classifier.

First, we briefly discuss related work. The abstract explanation of the mathematical framework is followed by a concrete application: algorithms for learning and localization of objects using normally distributed point features under orthographic projection are explicitly derived. The paper concludes with a summary discussion of the presented approach.

2 Related Work

Since the beginning of image analysis, object models were used for recognition purposes. First, they were coded implicitly as assumptions about features at certain locations; later on, the explicit structural models were used [2]. Statistical methods for knowledge acquisition and recognition of objects using non–parametric estimates are described in [10]. The referred work relates learning of 3D objects to the automatic computation of an aspect graph of 3D objects, and does not explicitly model the 3D structure in the sense that the 3D coordinates of an object are included. The description of segmentation errors and feature deviations in different intensity images by Gaussian distributions is discussed in detail in [13]. The parametric density function and the derived optimization problems are, in general, used for 2D object recognition from grey–level images. The involved density functions are not automatically learned by training samples; instead, the parameters are adjusted manually. In [13] it is suggested to apply the technique of linear combination of views [12] to extend the theory for 3D recognition purposes. The required matching among features of different views has to be computed. Wells [13] uses the EM algorithm for pose estimation

objectives. One approach, where statistical learning is done using samples of images, is the work of He and Kundu [6]. They suggest the use of Hidden Markov Models and implement a system which has the capability of learning two–dimensional objects from samples of closed contours.

3 Statistical Object Recognition

Most object recognition systems – especially for 3D applications – make use of a model based approach. Geometric models are rotated, translated, and finally projected from the model space into the image plane. Distance measures judge the localization and classification results. Various approaches can be compared and classified by the methods of model generation, distance computation, and decision making.

In the proposed statistical approach, k different objects are represented by parameterized density functions. Types and parameters of these functions may vary between objects and applications. In general, a statistical object recognition system should provide the following three stages:

1. *training stage*, where the parameters $\boldsymbol{B}_\kappa$, $1 \leq \kappa \leq k$, of the model density functions have to be estimated from a sample set $\{{}^\varrho\boldsymbol{O} | 1 \leq \varrho \leq N\}$ of views,
2. *localization stage*, where the pose, i.e. the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$, is computed, and
3. *identification stage*, where the class number κ of the observed object is determined.

Let $\boldsymbol{c}_l$ be an D_m–dimensional feature and $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{n_\kappa}\}$ be a set of features in the model space. In the chosen statistical framework features are considered as random variables and $p(\boldsymbol{C}|\boldsymbol{B}_\kappa)$ represents the density function of an object of class Ω_κ characterized by the parameters $\boldsymbol{B}_\kappa$. In the experiments described in section 6 we use a 2D model space in section 6.2, a 3D model space in section 6.3, and point features in both cases. The $\boldsymbol{c}_l$ is a model vertex, and $\boldsymbol{C}$ the set of features characterizing an object. The position of an object can change in the model space. This has also to be represented within the model density. Thus, both the rotation and translation of the object in the model space results in additional parameters of the model density functions. If $\boldsymbol{R}$ denotes the rotation matrix and $\boldsymbol{t}$ is the translation vector, we get the model density $p(\boldsymbol{C}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t})$ by the computation of a density transform. If features invariant to rotation and translation are used, the application of a density transform with respect to a changing position will not affect the density function, i.e. in this case we have $p(\boldsymbol{C}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = p(\boldsymbol{C}|\boldsymbol{B}_\kappa)$. For three–dimensional object recognition problems using two–dimensional views we have additionally the projection from model space into the image plane determined by the underlying camera. Of course, this requires information about the camera parameters. The influence of various types of projections can be integrated within the model density using once more a density transform. Since the range information is lost in the course of projection, a marginal density has to be computed. Consequently, the set of observable image features ${}^\varrho\boldsymbol{O} = \{{}^\varrho\boldsymbol{o}_1, {}^\varrho\boldsymbol{o}_2, \ldots, {}^\varrho\boldsymbol{o}_{{}^\varrho m}\}$ contains vectors of not necessarily lower dimension D_o than the model space. The image features used in the experiments of section 6 are 2D point features.

The influence of transformations and projections on model densities is straight forward. Now, a fundamental request is, which density function proves suitable for modeling objects. Due to occlusion, some features will not occur in the segmentation result causing ${}^\varrho m < n_\kappa$; due to segmentation errors and features belonging to the background, ${}^\varrho m > n_\kappa$ may occur. So in general we only know $n_\kappa \neq {}^\varrho m$. Additionally, for an observed scene, the correspondence of image and model primitives is not known. This matching must also be handled by the proposed statistical model. Furthermore, one should take into consideration that for purposes of image analysis the statistical modeling of background features and multiple object scenes should be possible.

4 Statistical Object and Scene Models

The least mentioned requirements for a statistical density function of an object are versatile. In this work we suggest the use of transformed mixture density functions for modeling objects. An object of class Ω_κ is associated with a set of statistical processes $\{S_1, S_2, \ldots, S_{n_\kappa}\}$. Each process, characterized by a state S_l, generates zero, one, or more output symbols ${}^\varrho\boldsymbol{o}_k$ and the complete set of n_κ statistical processes produces the ${}^\varrho m$ observable features of an object. For instance, an observed point feature ${}^\varrho\boldsymbol{o}_k$ of the ϱ–th view ${}^\varrho\boldsymbol{O}$ of an object is assumed to be an output symbol of exactly one out of n_κ stochastic processes. Within this context, $p({}^\varrho\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t})$ describes the emission density of the state S_l. Since for an observed image feature ${}^\varrho\boldsymbol{o}_k$ it is a priori unknown which state has emitted the primitive, each S_l is weighted by an a priori probability $p_{\kappa,l}$, which is indeed the discrete probability for the l–th state to emit a symbol. Consequently, the weights have to satisfy the condition $\sum_{l=1}^{n_\kappa} p_{\kappa,l} = 1$. The probability for observing a single feature ${}^\varrho\boldsymbol{o}_k$ is given by the marginal density over all states

$$p({}^\varrho\boldsymbol{o}_k|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \sum_{l=1}^{n_\kappa} p_{\kappa,l}\, p({}^\varrho\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t}),$$

where obviously the set of parameters is $\boldsymbol{B}_\kappa = \{p_{\kappa,l}, \boldsymbol{a}_{\kappa,l} | 1 \leq l \leq n_\kappa\}$. Under the idealized assumption that all ${}^\varrho m$ observed features out of a set of observed primitives ${}^\varrho\boldsymbol{O}$ are pairwise independent, the probability for observing a set of image features is computed by the product:

$$p({}^\varrho\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \prod_{k=1}^{{}^\varrho m} p({}^\varrho\boldsymbol{o}_k|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t})$$

$$= \prod_{k=1}^{{}^\varrho m} \sum_{l=1}^{n_\kappa} p_{\kappa,l}\, p({}^\varrho\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t}).$$

The marginal density over all states provides the probability that the given set of stochastic processes

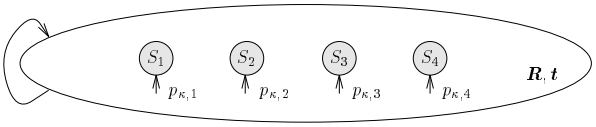

Fig. 1: A graph describing a statistical object model; for k different objects there are k such models

has produced the observable feature. Thus, the statistical model accomplishes the requirements of a probabilistic description of the missing alignment between scene features and model components. Fig. 1 shows a set of four statistical processes which can produce output symbols, i.e. observable features of the image space; the arc symbolizes that the generation of one output symbol is repeated for all ${}^{\varrho}m$ observable image features. Occlusion is represented as far as not every state is forced to produce output symbols. But it is also possible that one state generates more than one primitive. The advantage of this kind of modeling is that the observed features need not have any ordering and that the cardinality ${}^{\varrho}m$ of the set ${}^{\varrho}\boldsymbol{O}$ can be arbitrary.

Background scene features can be included into the probabilistic representation in a modular manner; one simply expands the mixture density model with an additional state S_H for the background. Fig. 2 visualizes a set of stochastic processes. Herein, the output probability of state S_H, which is weighted by p_H, is independent of the pose parameters $\boldsymbol{R}$ and $\boldsymbol{t}$, because it models background features, which in fact do not depend on object transforms. The estimation of the discrete probability $1 - p_H$ can either be done by applying the EM algorithm or by the fraction $n_\kappa / {}^{\varrho}m$ of the number of model states and observed features.

The probability for observing an object in a scene including background features is

$$p({}^{\varrho}\boldsymbol{O}|\boldsymbol{B}_H, \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \prod_{k=1}^{{}^{\varrho}m} \Big(p_H p({}^{\varrho}\boldsymbol{o}_k|\boldsymbol{a}_H) + (1-p_H) \sum_{l=1}^{n_\kappa} p_{\kappa,l}\, p({}^{\varrho}\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t}) \Big),$$

where $\boldsymbol{B}_H = \{p_H, \boldsymbol{a}_H\}$ is the set of parameters characterizing background features. The computation of the probability measure for pose parameters of an observed scene is bounded by $\mathcal{O}({}^{\varrho}m n_\kappa)$. In contrast, classical approaches, which consider all possible assignments between model and scene primitives, will take $\mathcal{O}({}^{\varrho}m^{n_\kappa})$.

The training stage is unsupervised in the sense that no correspondence between scene features and mixture density components is required. In the localization stage, the matching problem again occurs: it is unknown which features belong to the background or to the object. The EM algorithm introduced in [3] is an established technique which can be used for this type of incomplete data estimation problems. A comprehensive discussion of the EM algorithm and its applications can be found in [11], chapter 4.

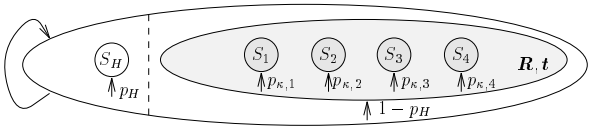

Fig. 2: A graph describing a statistical scene model consisting of one object plus the background

5 Gaussian Distributed Features and Orthographic Projection

The preceding sections provide a statistical framework for modeling objects and observed scenes, which meets the discussed requirements for model based object recognition systems with respect to 2D and 3D recognition tasks [5]. However, for a concrete realization of learning and recognition procedures, the statistical distributions $p({}^{\varrho}\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t})$ associated with each state S_l have to be worked out.

Empirical and statistical tests for point features justify the assumption that each state, which corresponds to the object in the statistical model, generates normally distributed output symbols [13]. The probability for observing a background feature is supposed to be uniform and independent of the pose parameters of the object searched for.

Let the transform in the D_m–dimensional model space and the subsequent projection into the D_o–dimensional image plane be characterized by an affine mapping with the matrix $\boldsymbol{R} \in \mathbb{R}^{D_o \times D_m}$ and the translation vector $\boldsymbol{t} \in \mathbb{R}^{D_o}$. From statistics it is known that the affine transform of a multivariate Gaussian distributed random vector is again normally distributed with the mean $\boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{t}$ and the covariance $\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^T$ [1], where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and covariance matrix of the original distribution.

5.1 Learning Stage

Due to the missing matching, the parameter set $\boldsymbol{B}_\kappa = \{p_{\kappa,l}, \boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l} | 1 \le l \le n_\kappa\}$ is estimated by the application of the iterative EM algorithm. The training data consist of N sample views of the object and the associated affine transforms which are assumed to be known, i.e. $\{{}^{\varrho}\boldsymbol{O}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t} | 1 \le \varrho \le N\}$. The ϱ–th view contains ${}^{\varrho}m$ features, ${}^{\varrho}\boldsymbol{O} = \{{}^{\varrho}\boldsymbol{o}_1, {}^{\varrho}\boldsymbol{o}_2, \ldots, {}^{\varrho}\boldsymbol{o}_{{}^{\varrho}m}\}$. The estimation problem lies in the unsupervised computation of the parameters referring to the D_m–dimensional model space from the projected D_o–dimensional observations.

For that purpose, the Kullback–Leibler statistics $Q(\widehat{\boldsymbol{B}}_\kappa | \boldsymbol{B}_\kappa)$ (see [3]), where $\widehat{\boldsymbol{B}}_\kappa$ represents the re–estimation of $\boldsymbol{B}_\kappa$, and its gradients concerning the parameter set $\widehat{\boldsymbol{B}}_\kappa$ have to be determined. By applying the incomplete data estimation algorithm to our problem domain, we get closed form re–estimation formulas for the weights and means of the i–th mixture density components using transformed observations

$$\widehat{p}_{\kappa,i} = \frac{1}{\sum_{\varrho=1}^{N} {}^{\varrho}m} \sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \frac{p_{\kappa,i}\, p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,i}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})}{p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{B}_\kappa, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})},$$

$$\widehat{\boldsymbol{\mu}}_{\kappa,i} = \left(\sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \frac{p_{\kappa,i} p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,i}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})}{p({}^{\varrho}\boldsymbol{o}_k \mid \boldsymbol{B}_\kappa, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})} \; {}^{\varrho}\boldsymbol{R}^T \, {}^{\varrho}\boldsymbol{D}_{\kappa,i}^{-1} \, {}^{\varrho}\boldsymbol{R} \right)^{-1}$$

$$\sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \frac{p_{\kappa,i} p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,i}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})}{p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{B}_\kappa, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})} \; {}^{\varrho}\boldsymbol{R}^T \, {}^{\varrho}\boldsymbol{D}_{\kappa,i}^{-1} \left({}^{\varrho}\boldsymbol{o}_k - {}^{\varrho}\boldsymbol{t} \right),$$

where ${}^{\varrho}\boldsymbol{D}_{\kappa,i} = {}^{\varrho}\boldsymbol{R} \boldsymbol{\Sigma}_{\kappa,i} {}^{\varrho}\boldsymbol{R}^T$. For the estimation of the covariances no closed form solution exists. The gradient of the Kullback–Leibler statistics has to be used for a local optimization technique within the EM iterations. For a clear representation, we introduce ${}^{\varrho}\widehat{\boldsymbol{S}}_{\kappa,i,k} = \left({}^{\varrho}\boldsymbol{o}_k - {}^{\varrho}\boldsymbol{R}\widehat{\boldsymbol{\mu}}_{\kappa,i} - {}^{\varrho}\boldsymbol{t} \right) \left({}^{\varrho}\boldsymbol{o}_k - {}^{\varrho}\boldsymbol{R}\widehat{\boldsymbol{\mu}}_{\kappa,i} - {}^{\varrho}\boldsymbol{t} \right)^T$, ${}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,i} = {}^{\varrho}\boldsymbol{R}\widehat{\boldsymbol{\Sigma}}_{\kappa,i} {}^{\varrho}\boldsymbol{R}^T$ and get

$$\nabla_{\widehat{\boldsymbol{\Sigma}}_{\kappa,i}} Q(\boldsymbol{B}_\kappa, \widehat{\boldsymbol{B}}_\kappa) = \sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \frac{p_{\kappa,i} p({}^{\varrho}\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,i}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})}{p({}^{\varrho}\boldsymbol{o}_k \mid \boldsymbol{B}_\kappa, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})} \; {}^{\varrho}\boldsymbol{R}^T \, {}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,i}^{-1} \left({}^{\varrho}\widehat{\boldsymbol{S}}_{\kappa,i,k} - {}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,i} \right) {}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,i}^{-1} \, {}^{\varrho}\boldsymbol{R} \quad .$$

This new class of estimation formulas for Gaussian mixture density functions constitute a generalization of the well–known estimation formulas for mixture density functions described in [4]. But the derived algorithms are also applicable to lower dimensional observation sets, because the affine mapping describes transformations from the D_m–dimensional model space into the D_o–dimensional image space, where $D_m \geq D_o$.

The realization of the training stage is characterized by three steps: First, the number of mixture components has to be determined. In a second stage, a suitable initialization of the parameters has to be done, and finally, we have to update the initial estimates by an iterative maximization of the Kullback–Leibler statistics, until the re–estimation converges.

5.2 Localization

The density function for localization of an object expects that the object's class is known and we search for the optimal position of this object. Since the matching between mixture components and available features is missing, the application of the EM algorithm seems natural. But, the initialization of the EM iterations is crucial for its success. Thus, it is generally preferable to use global optimization techniques for the computation of the pose parameters via

$$\max_{\boldsymbol{R},\boldsymbol{t}} p({}^{\varrho}\boldsymbol{O} | \boldsymbol{B}_H, \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}),$$

because it is unlikely to get an initialization close to the global maximum. For optimization procedures, gradients of the logarithmic Gaussian density function $L({}^{\varrho}\boldsymbol{o}_k) = \log N({}^{\varrho}\boldsymbol{o}_k | \, \boldsymbol{R}\boldsymbol{\mu}_{\kappa,l} + \boldsymbol{t}, \boldsymbol{R}\boldsymbol{\Sigma}_{\kappa,l}\boldsymbol{R}^T)$ will be useful. The gradients regarding the affine mapping given by $\boldsymbol{R}$ and $\boldsymbol{t}$ are

$$\nabla_t L({}^{\varrho}\boldsymbol{o}_k) = -\boldsymbol{D}_{\kappa,l} \left({}^{\varrho}\boldsymbol{o}_k - \boldsymbol{R}\boldsymbol{\mu}_{\kappa,l} - \boldsymbol{t} \right),$$

and

$$\nabla_R L({}^{\varrho}\boldsymbol{o}_k) = \boldsymbol{D}_{\kappa,l}^{-1} \left(\boldsymbol{S}_{\kappa,l,k} - \boldsymbol{D}_{\kappa,l} \right) \boldsymbol{D}_{\kappa,l}^{-1} \boldsymbol{R}\boldsymbol{\Sigma}_{\kappa,l} + \boldsymbol{U}_{\kappa,l,k},$$

where the component of the i–row and j–th column of $\boldsymbol{U}_{\kappa,l,k}$ is defined by $(\boldsymbol{U}_{\kappa,l,k})_{i,j} = (\boldsymbol{\mu}_{\kappa,l})_j \left(\boldsymbol{D}_{\kappa,l}^{-1} \left({}^{\varrho}\boldsymbol{o}_k - \boldsymbol{R}\boldsymbol{\mu}_{\kappa,l} - \boldsymbol{t} \right) \right)_i$.

In contrast to conventional pose estimation techniques where a feature matching is needed. The search space is determined by the degrees of freedom of the affine mapping and does not enlarge with the increase of observed features, For instance, under orthographic projection from a 3D model space into a 2D image plane the search space has five dimensions, that are three rotation angles and the components of the two–dimensional translation vector. Within the training stage we succeeded in separating the search space for several parameters by applying the EM algorithm. In the localization phase we can force a decomposition of the search space by breaking down the affine transform in D_o distinctive mappings into one–dimensional sub–spaces. The associated densities for the projected, one–dimensional feature sets can easily be computed using a standard density transform. A D_o–dimensional scene feature results from a transform

$$\boldsymbol{o} = \boldsymbol{R}\boldsymbol{c} + \boldsymbol{t} = \begin{pmatrix} \sum_{j=1}^{D_m} r_{1,j} c_j + t_1 \\ \sum_{j=1}^{D_m} r_{2,j} c_j + t_2 \\ \vdots \\ \sum_{j=1}^{D_m} r_{D_o,j} c_j + t_{D_o} \end{pmatrix}.$$

Each component of the feature vector is

$$o_i = (\boldsymbol{R})_i \, \boldsymbol{c} + t_i \quad , \quad i \in \{1, 2, \ldots, D_o\}$$

where $(\boldsymbol{R})_i = (r_{i,1}, r_{i,2}, \ldots r_{i,D_m})$. The i–th components of the observable feature vectors depend only on the components of $(\boldsymbol{R})_i$ and t_i. The original $(D_m D_o + D_m)$–dimensional search space falls into D_o parts of the dimension $D_m + 1$.

The covariance matrices of the projected, one–dimensional features are real numbers and thus the matrix inversion and the computation of the determinant within the evaluation of the Gaussian densities are unnecessary. The advantages of the suggested decomposition of the affine mapping are the separation of the search space and a more efficient computation of the density functions.

Fig. 3 illustrates this idea in a practical situation. Orthographic projection of two–dimensional point features onto the x–axis is not affected by a translation of the 3D object along the y–axis and by a rotation around the x–axis.

5.3 Identification

The identification stage makes the decision which object occurs in the image. The classification applies the Bayesian decision rule, which decides for the object class with the highest a posteriori probability.

$$\kappa = \arg\max_{\lambda} \left\{ \max_{\boldsymbol{R},\boldsymbol{t}} \frac{p(\Omega_\lambda) p(\boldsymbol{O} | \boldsymbol{B}_H, \boldsymbol{B}_\lambda, \boldsymbol{R}, \boldsymbol{t})}{p(\boldsymbol{O})} \right\}.$$

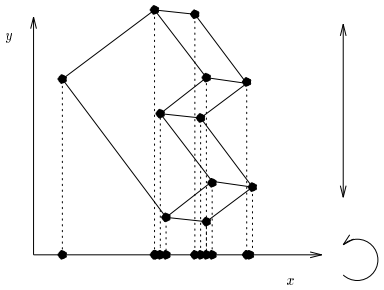

Fig. 3: Projection of corners to one coordinate axis; the arrows indicate translations and 3D rotations which do **not** affect the image of the projection.

Since we do not use invariant features, the classification step obviously implies the computation of the pose parameters.

6 Experimental Results

In our experiments we apply the statistical approach introduced above to learning, localization, and recognition of two– and three–dimensional objects using grey–level images. The chosen features are 2D point features. With each point we associate a state in our mixture modeling producing normally distributed 2D vectors. In all localization experiments we apply the search space decomposition as introduced in section 5.2. On the average, this separation entails an acceleration by a factor of five for the computation of the global maximum. The evaluation time of the one–dimensional features' density functions is 30 percent less than for two–dimensional points. All times refer to HP 735 workstations (99 MHz, 124 MIPS). The point features were computed looking at the curvature of the chain code representation of detected edges [9]. The time needed for segmentation is not taken into account.

6.1 The Training Stage

During the off–line training stage, the parameters of the mixture densities have to be estimated. If the dimension of model and image spaces are equal, the training can proceed without considering rotation and translation; the sample views can be generated with respect to different illumination conditions. If the model parameters of the model density have to be estimated from projected data, for each learning view ${}^{\varrho}\boldsymbol{O}$ the knowledge of the rotation matrix ${}^{\varrho}\boldsymbol{R}$ and the translation vector ${}^{\varrho}\boldsymbol{t}$ will be expected. For that purpose we use a calibrated camera which is mounted on a robot's hand. This device can be used for the generation of training views with its pose parameters. For the estimation of means, covariances, and weights, a parameter initialization of the density function for each feature is required. The number of features and initial estimates of means, covariance matrices, and weights have to be established. For simple polyhedral objects the method works, if we determine the number of features using one view and add the occluded features by hand. The mean vectors are initialized by the observable 2D point features, with the depth value set to zero.

Fig. 4: The grey–level image of an industrial part with homogeneous background, the segmentation result, and the visualization of the computed position

Fig. 5: The grey–level image of the scene, the segmentation, and the result of the localization

Empirically, 40–50 views are sufficient for learning an object with 15 characteristic point features. Although the convergence rate of the EM algorithm was expected to be considerably low (see [3]), the learning process converged, on average, after 10–15 iterations. The time needed for one iteration, using a C++ implementation of the learning formula for 3D mean vector estimations from projected observations takes 98 seconds with 50 training views. The memory requirements are constant for each iteration.

6.2 2D Object Recognition using Grey–level Images

Fig. 4 shows a grey–level image and the segmented point features where the computation of the rotation and translation is based on. The computed position is visualized in the right image. The computation time is 13 seconds. The same object is localized in the scene of Fig. 5 within 180 seconds. In this scene, partial occlusion takes place. Nevertheless, the computed position is correct.

For recognition experiments we took the parts shown in Fig. 6. Within 10 examples (5 for each object, homogeneous background) nine objects were correctly classified. The computed position was correct in all examples, and the classification took 30 seconds on average.

6.3 3D Object Localization using 2D Images

A much harder problem is the use of statistical model densities for pose estimations of three-dimensional objects in segmented grey–level images. We restrict our experiments to orthographic projection; the dimension of the pose space is five. Two simple polyhedral objects are represented by a transformed mixture densities with eight and twelve states. The computation time for localization of the stump in the grey–level image with homogeneous background shown in Fig. 7 was 86 seconds. The positioning of the polyhedral ob-

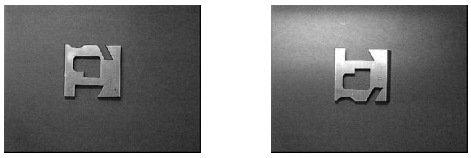

Fig. 6: Two 2D objects which cannot be transformed into each other by applying rotations and translations in the image plane

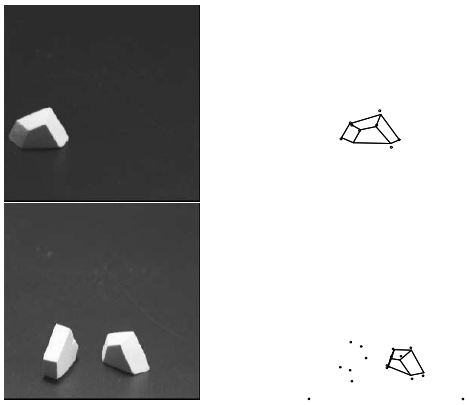

Fig. 7: Examples for grey–level images of 3D scenes and the extracted point features used for localization

ject in the two object scene took 95 seconds. The results of Fig. 8 show that both the L–piece and the stump were correctly detected. For visualization purposes of the computed pose parameters the 3D graphic tool of MAPLE V is used. These examples demonstrate that the statistical modeling works for the 3D case as well. Partial occlusion and multiple detection of features did not affect the localization process of the examples.

7 Summary and Conclusions

We presented a statistical object recognition system which includes an off–line training, a localization, and a classification stage. The experiments prove that the introduced mathematical framework is suitable for 2D and 3D computer vision purposes. Even the computation of object models out of a set of training samples, which include projections of the object, is possible. In contrast to classical geometrical approaches, the explicit solution of the correspondence problem is avoided.

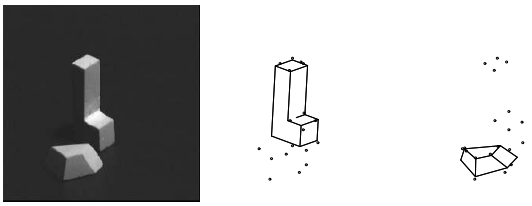

Fig. 8: Illustration of the computed 3D positions

It is part of the unsupervised parameter estimation algorithm.

Further research should concentrate on suitable initialization and more efficient parameter estimation techniques. Furthermore, the method should be made applicable for features of higher complexity like lines or polygons. The consideration of statistical dependencies and the explicit modeling of self–occlusion will also increase the robustness of the system. Probably, Markov models can be used for the embedding of these dependencies. The partitioning of the search space for pose estimation is fairly easy. Thus, a parallelization of the localization process will improve the recognition time.

8 References

1. T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Wiley Publications in Statistics. John Wiley & Sons, Inc., New York, 1958.
2. P. J. Besl and R. C. Jain. Three–dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, March 1985.
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. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, Inc., New York, 1973.
5. W.E.L Grimson and D.P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. In O. Faugeras, editor, *European Conference on Computer Vision 1990, Springer LNCS 427*, pages 489–498, Heidelberg, 1991. Springer.
6. Y. He and A. Kundu. 2-D Shape Classification Using Hidden Markov Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, 1991.
7. A. Jain and P. J. Flynn, editors. *Three–Dimensional Object Recognition Systems*, Amsterdam, 1993. Elsevier.
8. T. Kanade, editor. *Three–Dimensional Machine Vision*, Boston, 1987. Kluwer Academic Press.
9. R. Nevatia and R. Babu. Line feature extraction and description. *Computer Vision, Graphics and Image Processing (CVGIP)*, 13:257–269, 1980.
10. A.R. Pope and D.G. Lowe. Learning object recognition models from images. In *Proc. 4. Int. Conf. on Computer Vision*, pages 296–301, Berlin, May 1993. IEEE Press.
11. M. A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* Springer Series in Statistics. Springer, Heidelberg, 1993.
12. S. Ullman and R. Basri. Recognition by Linear Combinations of Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, October 1991.
13. W. M. Wells III. Statistical Object Recognition. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Massachusetts, February 1993.