Optimization Problems in Statistical Object Recognition

Joachim Hornegger, Heinrich Niemann
Lehrstuhl für Mustererkennung (Informatik 5)

hornegger@informatik.uni-erlangen.de
niemann@informatik.uni-erlangen.de

Reprint of the proceedings

International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition

Venice, Italy, 21. – 23. May 1997

Joachim Hornegger, Heinrich Niemann: *Optimization Problems in Statistical Object Recognition*, in Marcello Pelillo, Edwin R. Hancock (Hrsg.): *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, Berlin, Mai 1997, S. 311–326.

Contents

Optimization Problems in Statistical Object Recognition

Joachim Hornegger and Heinrich Niemann

Lehrstuhl für Mustererkennung (Informatik 5),
Martensstraße 3, D–91058 Erlangen, Germany,
email: {hornegger,niemann}@informatik.uni-erlangen.de
Tel.: 0049/9131/85–7826 Fax: 0049/9131/303811

Abstract This paper treats the application of statistical principles for 3D computer vision purposes. Both the automatic generation of probabilistic object models, and the localization as well as the classification of objects in compound scenes result in complex optimization problems within the introduced statistical framework. Different methods are discussed for solving the associated optimization problems: the Expectation–Maximization algorithm forms the basis for the learning stage of stochastic object models; global optimization techniques — like adaptive random search, deterministic grid search or simulated annealing — are used for localization. The experimental part utilizes the abstract formalism for normally distributed object features, proves the correctness of 3D object recognition algorithms, and demonstrates their computational complexity in combination with real gray–level images.

1 Introduction

Optimization problems are very much a part of pattern recognition and computer vision [28]. Within the field of object recognition and scene analysis one of the most challenging tasks is the computation of symbolic descriptions which optimally fit together sensor data and models as well as available domain knowledge [4]. The acquisition of object models and domain knowledge should be done automatically out of a set of representative training views [24, 27]. The methods for solving these hard computer vision problems are quite different and vary for particular approaches.

Recent trends show that the use and the extension of classical statistical pattern recognition algorithms [18] apply to image processing with an increasing interest. Examples for successfully tested probabilistic methods in the field of computer vision are:

- Bayesian parameter estimation techniques for surface segmentation and the computation of lower bounds for achievable errors [5],
- theory of probabilistic relaxation for matching symbolic structures [17, 20],
- Markov random field based image segmentation [14] and model–based image interpretation [23], and
- Bayesian networks for image labeling and interpretation [6, 21, 26].

Here we combine probabilistic and optimization methods for solving high–level vision tasks. We introduce statistical concepts for transforming sensor data into a symbolic description and show that knowledge acquisition, object localization, and classification

are associated with complicated maximizations. In contrast to [33], both the matching and the projection from the model into the image space are part of statistical models.

This paper is divided up into seven sections: The introduction is followed by the suggestion of a probabilistic formalism which combines continuous and discrete probability density functions for object and scene modeling. The stochastic object models are parameterized with respect to feature– and pose–specific parameters. The third section provides a mathematical representation of the involved optimization problems; model generation as well as object localization are defined as parameter estimation problems. The computation of feature–specific parameters corresponds to a parameter estimation problem with *incomplete data*. This is due to the loss of range information during the projection into the image space and the missing matching between image and model primitives. Just as the model generation, the localization reduces to the estimation of an optimal set of pose parameters. Methods and strategies for solving the introduced optimization problems are discussed in section four and five. The experimental evaluation illustrates how the investigated theory solves 3D computer vision problems. The paper ends with a brief summary of the main results, draws some conclusions, and gives some hints for future research.

2 Statistical Modeling of Objects

A wide range of different object recognition strategies has been proposed. Recognition systems, for instance, distinguish into sensor data, into features, into localization and classification algorithms, or into representation formalisms for object models and domain knowledge [29]. Because of the objects' geometric nature, most approaches to object modeling prefer geometric representations, like CAD models, wire frame models, or simply 3D line and point features [22]. In general, those pure geometric models do not or not sufficiently consider the probabilistic behavior of features available from sensor data, although these primitives constitute the input data for classification and localization. For a suitable and complete statistical description of an object and its appearance in images, the following components require an adequate probabilistic modeling [15]:

- statistical behavior of single features, for example, point or line features,
- object rotation and translation,
- projection from the model into the image space,
- occlusion,
- correspondence between image and model features, and
- relations between features, for instance, neighborhood relationships.

Classical pattern recognition theory and statistical classifiers expect non–transformed feature vectors of fixed dimension for each pattern [18]. Therefore an extension of common principles becomes necessary. An explicit or implicit statistical formalization of above items is introduced in the following subsections.

2.1 Statistical Modeling of Features

Object features occur in different domains, the model and the image space. Let D_{model} and D_{image} denote the dimensions of model and image spaces, respectively. For 3D object

recognition in gray–level images, for example, we set $D_{\text{model}} = 3$ and $D_{\text{image}} = 2$. The set $\{\Omega_\kappa | 1 \leq \kappa \leq K\}$ contains the considered object classes, where K denotes the number of models. Let $\boldsymbol{C}_\kappa = \{\boldsymbol{c}_{\kappa,1}, \boldsymbol{c}_{\kappa,2}, \ldots, \boldsymbol{c}_{\kappa,n_\kappa}\}$ be the set of model features of an object from class Ω_κ including n_κ different D_{model}–dimensional feature vectors $\boldsymbol{c}_{\kappa,l}$, where $l = 1, 2, \ldots, n_\kappa$. Usually, objects differ in the number of model features, i.e. $n_\kappa \neq n_\lambda$ for almost all $\kappa \neq \lambda$. With each single model feature $\boldsymbol{c}_{\kappa,l}$ we associate a random vector; thus a feature can be considered as a probabilistic event, which underlies a parametric distribution $p(\boldsymbol{c}_{\kappa,l}|\boldsymbol{a}_{\kappa,l})$. Here $\boldsymbol{a}_{\kappa,l}$ is the parameter set belonging to $\boldsymbol{c}_{\kappa,l}$. A rotation and translation of a rigid object within the model space is mathematically described by a bijective, affine transform given by the (parametric) matrix $\boldsymbol{R}_{rot} \in \mathbb{R}^{D_{\text{model}} \times D_{\text{model}}}$ and the vector $\boldsymbol{t}_{rot} \in \mathbb{R}^{D_{\text{model}}}$. If model primitives are attached features, i.e. for the transformed feature $\boldsymbol{c}'_{\kappa,l}$ we have $\boldsymbol{c}'_{\kappa,l} = \boldsymbol{R}_{rot}\boldsymbol{c}_{\kappa,l} + \boldsymbol{t}_{rot}$, then we get the density function $p(\boldsymbol{c}'_{\kappa,l}|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}_{rot}, \boldsymbol{t}_{rot})$ including two different types of parameters: *feature*– and *pose*–specific parameters. The integration of pose specific parameters can be done by a standard density transform [1]. The same holds for the extension of the model density with respect to the projection from the model into the image space [15].

Object rotation, translation, and orthographic projection, for instance, causes for each D_{model}–dimensional model feature $\boldsymbol{c}_{\kappa,l_k}$ a corresponding D_{image}–dimensional image feature $\boldsymbol{o}_k$. The relational dependency of image and model features

$$\boldsymbol{o}_k = \boldsymbol{R}\boldsymbol{c}_{\kappa,l_k} + \boldsymbol{t} \quad , \tag{1}$$

is given by an affine transform, where $\boldsymbol{R} \in \mathbb{R}^{D_{\text{image}} \times D_{\text{model}}}$ and $\boldsymbol{t} \in \mathbb{R}^{D_{\text{image}}}$. If the random variable in the model space is normally distributed with mean vector $\boldsymbol{\mu}_{\kappa,l_k}$ and covariance matrix $\boldsymbol{\Sigma}_{\kappa,l_k}$, the observed feature vector $\boldsymbol{o}_k$ is also Gaussian with the modified mean vector $\boldsymbol{R}\boldsymbol{\mu}_{\kappa,l_k} + \boldsymbol{t}$ and covariance matrix $\boldsymbol{R}^T\boldsymbol{\Sigma}_{\kappa,l}\boldsymbol{R}$ (see [1], p. 25).

If the corresponding pairs $[\boldsymbol{o}_k, \boldsymbol{c}_{\kappa,l_k}]$ of model and image features are known, the density function for observing a set $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m\}$ of image features for a given transformation is

$$p(\boldsymbol{O}|\{\boldsymbol{a}_{\kappa,1}, \boldsymbol{a}_{\kappa,2}, \ldots, \boldsymbol{a}_{\kappa,n_\kappa}\}, \boldsymbol{R}, \boldsymbol{t}) = \prod_{k=1}^{m} p(\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l_k}, \boldsymbol{R}, \boldsymbol{t}) \quad , \tag{2}$$

provided that all features are pairwise statistically independent.

2.2 Statistical Modeling of Matching

In practice, the major problem results from the missing assignment of image and model features. But the obvious computation or estimation of an assignment function applying heuristic or threshold methods would be a contradiction to the intended goal of a closed statistical framework for object modeling. For that reason, we define a hidden correspondence function

$$\zeta_\kappa : \begin{cases} \boldsymbol{O} \to \{1, 2, \ldots, n_\kappa\} \\ \boldsymbol{o}_k \mapsto \quad l_k \end{cases} , \quad k = 1, 2, \ldots, m \tag{3}$$

Each set of correspondence pairs $\{[\boldsymbol{o}_k, \boldsymbol{c}_{\kappa,l_k}] | 1 \leq k \leq m\}$ including elements of $\boldsymbol{O} \times \boldsymbol{C}_\kappa$ can be associated with an integer vector

$$\boldsymbol{\zeta}_\kappa = \begin{pmatrix} \zeta_\kappa(\boldsymbol{o}_1) \\ \zeta_\kappa(\boldsymbol{o}_2) \\ \vdots \\ \zeta_\kappa(\boldsymbol{o}_m) \end{pmatrix} = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{pmatrix} \in \{1, 2, \ldots, n_\kappa\}^m \quad , \tag{4}$$

and each correspondence $\boldsymbol{\zeta}_\kappa$ can be understood as a random vector. Thus, the discrete probability $p(\boldsymbol{\zeta}_\kappa)$ can be computed, giving stochastic measures for correspondence functions ζ_κ [33].

An example shows Figure 1. The assignment ζ_κ, illustrated by arrows, induces the random vector $\boldsymbol{\zeta}_\kappa = (3, 3, 2, 1, 5, 4, 4, 5)^T$.

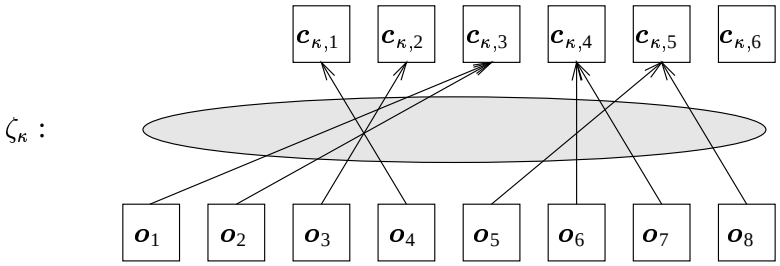


Figure1. Assignment of image and model features

2.3 Construction of Model Densities

With the suggested statistical modeling of assignment functions and the probabilistic characterization of image features, the density function for observing a set of features $\boldsymbol{O}$ with an assignment function ζ_κ and a transform, given by $\boldsymbol{R}$ and $\boldsymbol{t}$, is

$$p(\boldsymbol{O}, \boldsymbol{\zeta}_\kappa | \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = p(\boldsymbol{\zeta}_\kappa) \prod_{k=1}^{m} p(\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,\zeta_\kappa(o_k)}, \boldsymbol{R}, \boldsymbol{t}) \quad . \tag{5}$$

Herein, $\boldsymbol{B}_\kappa$ subsumes all involved parameters of the density, i.e. $\boldsymbol{B}_\kappa = \{p(\boldsymbol{\zeta}_\kappa), \boldsymbol{a}_{\kappa,1}, \boldsymbol{a}_{\kappa,2}, \ldots, \boldsymbol{a}_{\kappa,n_\kappa}\}$. Usually, the matching ζ_κ is not part of the observation. Within the statistical context the missing matching is considered by computing the marginal density of (5) with respect to $\boldsymbol{\zeta}_\kappa$, i.e. ζ_κ is integrated out, and we get

$$p(\boldsymbol{O} | \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \sum_{\boldsymbol{\zeta}_\kappa} p(\boldsymbol{O}, \boldsymbol{\zeta}_\kappa | \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \sum_{\boldsymbol{\zeta}_\kappa} p(\boldsymbol{\zeta}_\kappa) \prod_{k=1}^{m} p(\boldsymbol{o}_k | \boldsymbol{a}_{\kappa,\zeta_\kappa(o_k)}, \boldsymbol{R}, \boldsymbol{t}) \tag{6}$$

Since $p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t})$ represents a statistical description of the object of class Ω_κ appearance in the image plane with an eliminated correspondence function, it is called a *model density*. The complexity of evaluating the model density (6) for a given observation $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m\}$ is bounded by $\mathcal{O}(n_\kappa^m)$.

This exponential run–time behavior can be reduced by special independency assumptions for the components of $\boldsymbol{\zeta}_\kappa$. Let, for instance, all components of $\boldsymbol{\zeta}_\kappa$ be pairwise statistically independent, then we have

$$p(\boldsymbol{\zeta}_\kappa) = \prod_{k=1}^{m} p(\boldsymbol{\zeta}_\kappa(\boldsymbol{o}_k) = l_k) = \prod_{k=1}^{m} p_{\kappa,l_k} \quad , \tag{7}$$

and (6) combined with (7) results in

$$p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \prod_{k=1}^{m} \sum_{l=1}^{n_\kappa} p_{\kappa,l}\, p(\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t}) \quad , \tag{8}$$

with the linear complexity $\mathcal{O}(mn_\kappa)$. In general, it can be shown that a statistical dependency of order g implies the complexity $\mathcal{O}(mn_\kappa^{g+1})$ for the evaluation of (6).

3 Optimization Problems and Their Complexity

Model densities $p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t})$ as introduced in the previous section are characterized by two different types of parameters. Within the model generation stage the *model–specific* parameter set $\boldsymbol{B}_\kappa$ has to be estimated. The localization of objects corresponds to the computation of $\boldsymbol{R}$ and $\boldsymbol{t}$. Both stages are — in the mathematical sense — parameter estimation problems, which can be solved by different techniques [30]. One established and widely used method is, for instance, the maximum–likelihood approach.

3.1 Estimation of Model Parameters

For model generation purposes it is assumed that N different views are available and that for each view ϱ, $1 \leq \varrho \leq N$, the set of features ${}^{\varrho}\boldsymbol{O} = \{{}^{\varrho}\boldsymbol{o}_1, {}^{\varrho}\boldsymbol{o}_2, \ldots, {}^{\varrho}\boldsymbol{o}_{\varrho m}\}$ and the transformation parameters ${}^{\varrho}\boldsymbol{R}$ and ${}^{\varrho}\boldsymbol{t}$ are known. The maximum–likelihood estimation (ML estimation) of $\boldsymbol{B}_\kappa$ expects a set of statistically independent observations and solves the optimization problem

$$\widehat{\boldsymbol{B}}_\kappa = \underset{\boldsymbol{B}_\kappa}{\operatorname{argmax}} \prod_{\varrho=1}^{N} p({}^{\varrho}\boldsymbol{O}|\boldsymbol{B}_\kappa, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t}) \quad . \tag{9}$$

The number of unknown parameters depends on the existing model features, on the chosen parametric density functions for single feature distributions, and on the dependencies of the correspondence function ζ_κ. If, for instance, normally distributed 3D vectors are used as model features and statistically independent assignments are present, the number of unknown parameters is $n_\kappa + 3n_\kappa + n_\kappa \cdot 3(3+1)/2$. For a 3D cube with eight ($n_\kappa = 8$) corners this results in 80 parameters. Consequently, the ML estimation has to be done in a fairly high dimensional parameter space from possibly projected data, since $D_{\text{model}} \geq D_{\text{image}}$, and unsupervised with respect to feature correspondences. The parameter estimation has to deal with incomplete sample data.

3.2 Estimation of Pose Parameters

The localization of one object in a single image results in the global optimization problem

$$\{\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{t}}\} = \underset{\boldsymbol{R}, \boldsymbol{t}}{\operatorname{argmax}}\, p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) \quad , \tag{10}$$

where the parameters $\boldsymbol{R}$ and $\boldsymbol{t}$ of the transform from the model into the image space are estimated. In contrast to the estimation of model–specific parameters $\boldsymbol{B}_\kappa$ the set of data for parameter estimation is here restricted to a single view. The number of pose parameters does not depend on the number of model features, but on the parametric function which maps model features to image features. Assuming that objects are rotated, translated, and projected into the image plane by a perspective projection. Independent of the model features' present distribution the search space is six–dimensional, fixed by rotation angles ϕ_x, ϕ_y and ϕ_z and components t_1, t_2 and t_3 of the translation vector.

3.3 Classification of Objects

If the optimal pose parameters for all model densities are known, the class decision is based on the Bayesian decision rule, i.e. the discrete optimization problem

$$\kappa = \underset{\lambda}{\operatorname{argmax}}\, p(\Omega_\lambda|\boldsymbol{O}, \boldsymbol{B}_\lambda, \boldsymbol{R}, \boldsymbol{t}) \tag{11}$$

has to be solved for the observed feature set. This maximization process is bounded by $\mathcal{O}(K)$, where K denotes the number of object classes.

The previous discussion introduced three different types of optimization problems involved in the object recognition process. While the solution of the class decision problem (11) is obvious, model generation (9) and pose estimation (10) are incomparably hard problems. The presentation of possible solutions of both maximization tasks is the challenge of subsequent sections.

4 Model Generation: Incomplete Data Estimation

Object models including statistical properties of primitives should be learned out of a set of training views including non–corresponding features. The manual and painstaking construction of object models should be avoided and computers should learn the appearance of object features in images automatically.

4.1 The Expectation Maximization Algorithm

First, we consider the model generation problem using model densities as introduced in section 2, before we turn to the discussion of practical examples. We have already mentioned the incompleteness of the available training data and the infeasibility of a direct ML estimation due to the high dimension of the search space. The basic idea of the *Expectation Maximization algorithm* (EM algorithm) [8] is the augmentation of

the observable data with latent data to simplify the parameter estimation algorithm. This technique leads — in most applications — to a reduction of one complicated optimization problem into a series of independent simpler maximizations. In an informal and colloquial manner we describe the available information for parameter estimation by the difference

observed information = complete information − missing information .

For simplicity, let us assume that the observable random variables are $\boldsymbol{X}$ and the missing random variables are $\boldsymbol{Y}$. If the associated densities are parameterized with respect to $\boldsymbol{B}$, we have $p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{B}) = p(\boldsymbol{X}|\boldsymbol{B})p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{B})$. Taking the logarithm on both sides, we get an information theoretic formalization of above difference:

$$(-\log p(\boldsymbol{X}|\boldsymbol{B})) = (-\log p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{B})) - (-\log p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{B})) \quad . \tag{12}$$

By multiplying with $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{B})$ and integrating out the latent random variable $\boldsymbol{Y}$ it results the *key–equation* of the EM algorithm [8]

$$E[\log p(\boldsymbol{X}|\widehat{\boldsymbol{B}}^{(i+1)})|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i)}] = Q(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}) - H(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}) \quad , \tag{13}$$

where $\widehat{\boldsymbol{B}}^{(i+1)}$ is the re–estimation of the parameter set $\widehat{\boldsymbol{B}}^{(i)}$ in the $(i+1)$–st iteration,

$$\log p(\boldsymbol{X}|\widehat{\boldsymbol{B}}^{(i+1)}) = \int_{\boldsymbol{Y}} p(\boldsymbol{Y}|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i)}) \log p(\boldsymbol{X}|\widehat{\boldsymbol{B}}^{(i+1)})\, d\boldsymbol{Y} \quad , \tag{14}$$

$$Q(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}) = \int_{\boldsymbol{Y}} p(\boldsymbol{Y}|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i)}) \log p(\boldsymbol{X},\boldsymbol{Y}|\widehat{\boldsymbol{B}}^{(i+1)})\, d\boldsymbol{Y} \quad , \tag{15}$$

and

$$H(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}) = \int_{\boldsymbol{Y}} p(\boldsymbol{Y}|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i)}) \log p(\boldsymbol{Y}|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i+1)})\, d\boldsymbol{Y} \quad . \tag{16}$$

The main properties of equations (13)–(16) can be summarized as follows: The integral of the left hand side of (13) is equal to the original log–likelihood function $\log p(\boldsymbol{X}|\widehat{\boldsymbol{B}}^{(i+1)})$. Changes in the parameter set $\widehat{\boldsymbol{B}}^{(i+1)}$ induce a decrease of $H(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)})$ (Jensen's inequality [8]), thus an increase of the *Kullback–Leibler statistics* $Q(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)})$ causes a reduction of $H(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)})$. Consequently, a maximum–likelihood estimation can be *simulated* by an iterative maximization of $Q(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)})$. The final success of the EM iterations depends on the initial estimate $\widehat{\boldsymbol{B}}^{(0)}$, because the EM algorithm is a local optimization technique and provides a linear convergence behavior [34]. The already mentioned advantage of EM iterations instead of a straightforward ML estimation is that in most applications dealing with missing data the search space splits into independent lower dimensional sub–spaces. Furthermore, due to its iterative procedure the storage requirements remain constant. An impressive example out of the field of computer vision will be discussed in the following subsection.

4.2 Modeling of Line and Point Features

A D_{model}–dimensional line feature $\boldsymbol{c}_{\kappa,l_k}$ is identified by a sequence $[\boldsymbol{c}_{\kappa,l_k,1},\ldots,\boldsymbol{c}_{\kappa,l_k,q}]$ of D_{model}–dimensional points. Each component vector $\boldsymbol{c}_{\kappa,l_k,s} \in \mathbb{R}^{D_{\text{model}}}$ $(s = 1,2,\ldots,q;$ $l_k = 1,2,\ldots,n_\kappa)$ represents a supporting point of the D_{model}–dimensional polygon $\boldsymbol{c}_{\kappa,l_k}$ (see Figure 2), and is assumed to be normally distributed with mean vector $\boldsymbol{\mu}_{\kappa,l_k,s}$ and covariance matrix $\boldsymbol{\Sigma}_{\kappa,l_k,s}$. In accordance with subsection 2.1 the density for a feature sequence is

$$p(\boldsymbol{c}_{\kappa,l_k}|\boldsymbol{a}_{\kappa,l_k}) = \prod_{s=1}^{q} p(\boldsymbol{c}_{\kappa,l_k,s}|\boldsymbol{a}_{\kappa,k,s}) = \prod_{s=1}^{q} \mathcal{N}(\boldsymbol{c}_{\kappa,l_k,s}|\{\boldsymbol{\mu}_{\kappa,l_k,s},\boldsymbol{\Sigma}_{\kappa,l_k,s}\}) \quad . \tag{17}$$

Obviously, if we set $q = 1$, feature $\boldsymbol{c}_{\kappa,l_k}$ degenerates to a point feature.

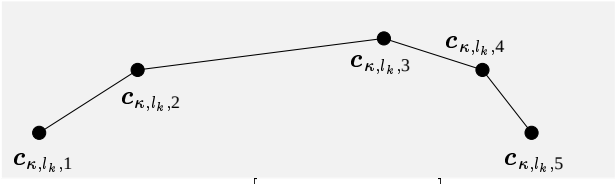


Figure2. A polygon represented as a sequence of supporting points

Let the mapping from the D_{model}–dimensional model into the D_{image}–dimensional image space be an affine transform, represented by $\boldsymbol{R} \in \mathbb{R}^{D_{\text{image}} \times D_{\text{model}}}$ and $\boldsymbol{t} \in \mathbb{R}^{D_{\text{image}}}$. The sequence of observable image points is $\boldsymbol{o}_k = [\boldsymbol{o}_{k,1},\boldsymbol{o}_{k,2},\ldots,\boldsymbol{o}_{k,q}]$, where $\boldsymbol{o}_{k,s} \in \mathbb{R}^{D_{\text{image}}}$ is also normally distributed. In addition to the lost dimensions during the projection process, the orientation of an observable sequence $\boldsymbol{o}_k$ is also latent. Mathematically, we formalize this change of ordering by the introduction of a permutation $\tau \in \Upsilon$, which acts on the sequence elements and assigns each s of $\boldsymbol{c}_{\kappa,k,s}$ to the index $\tau(s) = 1,\ldots,q$ of the corresponding image element $\boldsymbol{o}_{l_k,\tau(s)}$. In general, an identification of initial and final sequence points is impossible without a given matching ζ_κ of model and image features.

The permutation τ and the matching function ζ_κ are part of the missing data, where each permutation and matching is weighted by discrete probabilities $p(\tau)$ and $p(\boldsymbol{\zeta}_\kappa)$. The available training data consist of a N sample views $\{{}^{\varrho}\boldsymbol{O}|1 \le \varrho \le N\}$ and the corresponding transformation parameters $\{{}^{\varrho}\boldsymbol{R},{}^{\varrho}\boldsymbol{t}|1 \le \varrho \le N\}$.

The computation of the Kullback–Leibler statistics (15) and the calculation of the zero crossings of its gradient vector results in the following iteration formulas for estimating the discrete probabilities of the assignment function (cmp. [15])

$$\widehat{p}_{\kappa,l}^{(i+1)} = \frac{1}{\sum_{\varrho=1}^{N} {}^{\varrho}m} \sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \frac{\widehat{p}_{\kappa,l}^{(i)} p({}^{\varrho}\boldsymbol{o}_k|\widehat{\boldsymbol{a}}_{\kappa,l}^{(i)},{}^{\varrho}\boldsymbol{R},{}^{\varrho}\boldsymbol{t})}{p({}^{\varrho}\boldsymbol{o}_k|\widehat{\boldsymbol{B}}_{\kappa}^{(i)},{}^{\varrho}\boldsymbol{R},{}^{\varrho}\boldsymbol{t})} \quad , \tag{18}$$

and the mean vectors

$$\widehat{\boldsymbol{\mu}}_{\kappa,l,s}^{(i+1)} = \left(\sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \sum_{\tau \in \Upsilon} p({}^{\varrho}\boldsymbol{o}_k | l, \tau, \widehat{\boldsymbol{B}}_{\kappa}^{(i)}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})\ {}^{\varrho}\boldsymbol{R}^T ({}^{\varrho}\boldsymbol{R} \widehat{\boldsymbol{\Sigma}}_{\kappa,l,s}^{(i+1)} {}^{\varrho}\boldsymbol{R}^T)^{-1} {}^{\varrho}\boldsymbol{R} \right)^{-1} \tag{19}$$
$$\sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \sum_{\tau \in \Upsilon} p({}^{\varrho}\boldsymbol{o}_k | l, \tau, \widehat{\boldsymbol{B}}_{\kappa}^{(i)}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t}) {}^{\varrho}\boldsymbol{R}^T ({}^{\varrho}\boldsymbol{R} \widehat{\boldsymbol{\Sigma}}_{\kappa,l,s}^{(i+1)} {}^{\varrho}\boldsymbol{R}^T)^{-1} ({}^{\varrho}\boldsymbol{o}_k - {}^{\varrho}\boldsymbol{t}) \quad ,$$

where $1 \leq l \leq n_\kappa$ and $1 \leq s \leq q$. Unfortunately, no closed–form solution exists for the estimation of the covariance matrices. The zero crossings of the Kullback–Leibler statistics gradient

$$\nabla_{\widehat{\boldsymbol{\Sigma}}_{\kappa,l,s}^{(i+1)}} Q(\widehat{\boldsymbol{B}}_{\kappa}^{(i+1)} | \widehat{\boldsymbol{B}}_{\kappa}^{(i)}) = - \sum_{\varrho=1}^{N} \sum_{k=1}^{{}^{\varrho}m} \sum_{\tau \in \Upsilon} p({}^{\varrho}\boldsymbol{o}_k | l, \tau, \widehat{\boldsymbol{B}}_{\kappa}^{(i)}, {}^{\varrho}\boldsymbol{R}, {}^{\varrho}\boldsymbol{t})\ \widehat{\boldsymbol{M}}_{\kappa,k,l,\tau,s}^{(i+1)} \ , \tag{20}$$

results in nonlinear equations, where we set

$${}^{\varrho}\widehat{\boldsymbol{S}}_{\kappa,k,l,\tau,s}^{(i+1)} = \left({}^{\varrho}\boldsymbol{o}_{k,\tau(s)} - {}^{\varrho}\boldsymbol{R} \widehat{\boldsymbol{\mu}}_{\kappa,l,s}^{(i+1)} - {}^{\varrho}\boldsymbol{t} \right) \left({}^{\varrho}\boldsymbol{o}_{k,\tau(s)} - {}^{\varrho}\boldsymbol{R} \widehat{\boldsymbol{\mu}}_{\kappa,l,s}^{(i+1)} - {}^{\varrho}\boldsymbol{t} \right)^T \ , \tag{21}$$

$${}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,l,s}^{(i+1)} = {}^{\varrho}\boldsymbol{R} \widehat{\boldsymbol{\Sigma}}_{\kappa,l,s}^{(i+1)} {}^{\varrho}\boldsymbol{R}^T \ , \tag{22}$$

and

$$\widehat{\boldsymbol{M}}_{\kappa,k,l,\tau,s}^{(i+1)} = {}^{\varrho}\boldsymbol{R}^T \left({}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,l,s}^{(i+1)} \right)^{-1} \left({}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,l,s}^{(i+1)} - {}^{\varrho}\widehat{\boldsymbol{S}}_{\kappa,k,l,\tau,s}^{(i+1)} \right) \left({}^{\varrho}\widehat{\boldsymbol{D}}_{\kappa,l,s}^{(i+1)} \right)^{-1} {}^{\varrho}\boldsymbol{R} \tag{23}$$

For an iterative computation of the zero crossings of (20) numerical methods like the algorithm of Fletcher and Powell must be applied. In this case the EM algorithm yields a two stage iteration procedure: an iterative maximization of Kullback–Leibler statistics within each single EM iteration is necessary.

Above iteration algorithms allow the estimation of density parameters of the higher dimensional model space using projected observations. The search space is splitted up into separate optimization tasks for the estimation of matching parameters, mean vectors, and covariance matrices. It is remarkable that the parameter estimation step using the EM algorithm requires *no* explicit computations of the assignment functions. The model generation works unsupervised with respect to feature matching.

Specializations of the introduced estimation formulas were already published in [16], where $q = 1$, or can be found in [11], where $q = 1$ and no feature transform is considered, i.e. ${}^{\varrho}\boldsymbol{R} = \mathbf{1}$ and ${}^{\varrho}\boldsymbol{t} = \mathbf{0}$.

parameter	3D features [sec]	2D features [sec]	1D features [sec]
assignement (18)	10	7	4
mean vector (19)	28	27	23
covariance Matrix (20)	37	150	125
total	75	184	152
# **iterations**	5	20	25

Table1. Computation time for EM iterations using 400 training views of synthetic point features ($q = 1$) of different dimension. Each image includes 10 point features.

4.3 Experimental Results

The EM based model generation routines were implemented in C++ and tested on a HP 735 (100 MHz, 64 MB, 124 MIPS). The iteration formulas allow the estimation of 3D distribution parameters out of 3D, 2D, and 1D data. Table 1 summarizes the computation times and the required number of iterations. For the training of 3D model densities out of 2D views, we use a calibrated robot [9], such that for each random view the position of the camera is known (extrinsic camera parameters). The initialization of mean vectors is based on 2D features of a reference view. All range values are set to zero. The disadvantage of the introduced model generation is that n_κ, the number of model features, has to be known in advance. Up to now, there exists no reliable, robust, and feasible method for the automatic computation of n_κ.

5 Object Localization: Global Optimization

In contrast to geometric based methods, we do not hypothesize a matching of corresponding image and model features for the analytical computation of object's pose, but solve a parameter estimation problem for a smooth density function (see eq. (10)).

The dimension of the search space depends on the used projection model and only the mapping of the model into the image space influences the number of pose parameters. For instance, in the case of perspective projection the object's pose has six degrees of freedom, whereas the orthographic projection results in a five dimensional search problem. The translation parallel to the optical axis is omitted. The density function parameterized in pose parameters is a highly multimodal function. In general, local optimization techniques will not succeed in computing the global maximum. An initialization close to the global maximum is not possible without any additional knowledge. The reasonably low dimensional search space and the necessity of global optimization techniques do not suggest the use of the local optimizing EM algorithm for pose estimation. Nevertheless, in [33] an EM based localization procedure for pose refinement is discussed. The initialization of pose parameters is done by indexing techniques and geometric relations.

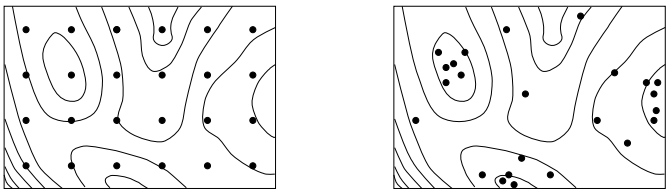


Figure3. Deterministic and probabilistic search in a 2D contour map. The points show the chosen initial values for a local search.

5.1 Deterministic Localization

Widely applied deterministic procedures for the computation of global maxima are grid search techniques, where equidistant initial points are distributed over the finite search space. Figure 3 (left) illustrates this for the 2D case. At the inserted initial points, local optimization algorithms can be started for a pose refinement. Of course, the success of a grid search method for object localization extremely depends on the chosen mesh, i.e. the distance of sample points.

A simple counter example proves the limited use of deterministic search algorithms, even if local optimization steps are left out: Let us assume, we have orthographic projection without scaling. The parameter space has five dimensions, three rotation angles $0^\circ \leq \phi_x, \phi_y, \phi_z \leq 360^\circ$ and two components $0 \leq t_1, t_2 \leq 100$ of the translation vector. Let the step–size for angles be 10° and the step–size for the translation 10. If the density function is evaluated at these grid points only and if one function evaluation takes 7 ms (cmp. Section 5.3), the computation of the global maximum will take $4.6 \cdot 10^6$ function evaluations, i.e. approx. 9 hours. This is intolerable for practical applications, and it furthermore proves the necessity of coarse–to–fine grid search or randomized optimization algorithms, which select prospective areas of the search space.

5.2 Probabilistic Localization

A recommended overview of probabilistic optimization techniques can be found in [10]. For solving the object localization problem, we choose the following basic idea: we randomly select a certain number of uniformly distributed initial points and evaluate the density function at these points. In a second step, additional random points are picked out, but their distribution should depend on hitherto computed density values. The random process, which controls the generation of new points, should take the observed density values into account and adapt to the density function's contour. A higher density value of a selected point implies a higher probability for the generation of a sample close to this area (see Figure 3, right). This process is repeated until a termination criterion is satisfied. Figure 4 shows a sketch of this adaptive random search technique. A frequently used termination criterion results from the usage of thresholds. For example, the difference of the lowest and highest entry of the included ordered list should reduce to a predefined threshold. The adaptive behavior of the random generator

/* **Adaptive Random Search** */
INPUT: model density, observed features
evaluate the model density at a randomly chosen initial points in the search space; store the best b of these points in a sorted list
as long as no stop criterion is satisfied repeat
generate new points in the view of the fact that the elements of b were already observed
add the new points to the ordered list
eliminate the worst points out of the list
adapt the parameters which guide the random process for the generation of points
select the global maximum and its position
OUTPUT: coordinates of the global maximum

Figure4. The principle of probabilistic search

can be controlled, for example, by a mixture density of Gaussians, which is iteratively updated by the observed function values.

5.3 Experimental Results

Within the experimental evaluation of different optimization techniques, we compare the following global search techniques:

V1: adaptive random search [12],
V2: adaptive random search combined with a locally operating downhill simplex algorithm [13],
V3: simulated annealing for continuous functions [3, 7],
V4: multi–start algorithm [31, 32],
V5: grid simplex algorithm [25], and
V6: pure probabilistic search [2, 31].

The average number of function evaluations required for the detection of the global maximum, and the average runtime on a monoprocessor system is shown in Table 2, (a). In our experiments with synthetic data ($D_{\text{model}} = 3, D_{\text{image}} = 2, n_\kappa = m = 10, q = 1$), the adaptive random search technique (V1) and modifications of this algorithm (V2) yield the best results. The probabilistic algorithm could find the global maximum with 87% success.

For a sample set of 400 real images the pose parameters were correctly found in average for 47% using adaptive random search (cmp. Table 2, (b)). The used objects can be found in Figure 5. Since the model densities do not take into account self occlusion of objects, both pose estimates shown in Figure 6 result in similar density values. The difference between the rotation angles around one coordinate axis is equal to π for both sets of pose parameters, because models are treated as transparent objects. But above experimental evaluation considers only one parameter set to be physically correct. Indeed, if we formally interpret both pose parameters to be equivalent, the correct localization rate increases to 78% for real data using the adaptive random search technique.

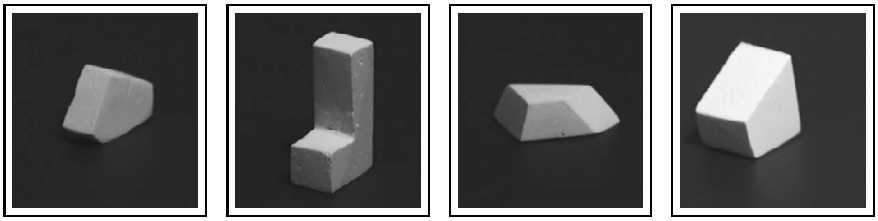
alg.	# eval.	time [sec]
V1	10010	75
V2	8560	64
V3	41300	310
V4	585000	4380
V5	1820000	13600
V6	10000000	74500

(a) Global optimization

3D object	correct pose [%]		comp. time [sec]	
	$q=1$	$q=2$	$q=1$	$q=2$
Q1	55	52	88	384
Q2	42	47	112	484
Q3	51	47	76	257
Q4	43	37	67	296
mean	48	46	86	355

(b) Localization results

Table2. Comparison of global optimization algorithms and pose estimation results



Q1 Q2 Q3 Q4

Figure5. Polyhedral 3D objects (Q1–Q4) used for recognition and pose estimation experiments.

6 Object Recognition

If the pose parameters for each model density are known, the classification is simply done by computing and by comparing the a posteriori probabilities. Since we have seen in previous experiments that the probabilistic search for pose parameters is responsible for about 22% localization errors, the expected recognition rate will be bounded by 78%.

Table 3 shows the classification results for point and straight line features ($q = 1, 2$) using 1600 randomly chosen images with varying illumination. The recognition rate of approximately 70% is due to the fact that the segmentation results are sometimes insufficient for classification, and the statistical behavior of point features is approximated by Gaussians. The chosen test objects show few corners, and segmentation errors often lead image features of low discriminating power.

7 Conclusions and Future Research Problems

This paper has introduced a general uniform statistical framework for object modeling, localization, and recognition. The statistical models include the assignment of features, the transformation of objects, and the projection from the model into the image space.

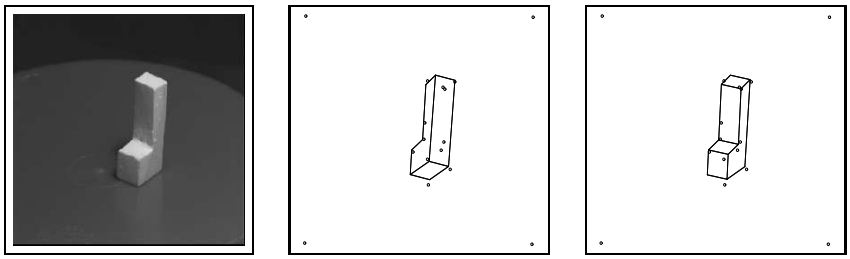


Figure6. The two best pose hypotheses for a polyhedral object. If objects are considered to be transparent, both pose parameter sets are considered to be equal.

	recognition [%]		**time [sec]**	
3D–object	$q = 1$	$q = 2$	$q = 1$	$q = 2$
Q1	47	44	466	1882
Q2	78	82	485	2101
Q3	58	36	465	1933
Q4	89	76	471	1520
mean	68	59	472	1859

Table3. Recognition rates and runtime on a HP 735 for 3D experiments (1600 real gray–level images using point ($q = 1$) and line ($q = 2$) features).

In contrast to many object recognition systems, all three stages of model generation, pose estimation, and class decision are defined as optimization problems. These were solved using different maximization algorithms.

The estimation of model parameters has shown to be a search problem in a high dimensional parameter space. Even for simple polyhedral objects the model generation has to deal with incomplete training data. The learning problem was solved here by the application of the EM algorithm with reasonable costs. This is the first approach in computer vision which deals with the estimation of model parameters from projected data without taking the matching into consideration.

The pose estimation is related to a global optimization problem. The EM algorithm could be applied, but requires an initialization within the area of attraction of the global maximum [33]. For that reason and due its low convergence rate, we did the localization without the EM technique. A comparison of different optimization algorithms showed that an adaptive random search combined with the downhill simplex algorithm gives best results regarding the runtime behavior. Especially in this application, probabilistic optimization techniques beat deterministic algorithms.

For future research the use of different views for classification and localization might improve and speed up the algorithms. The more data are available, the more reliable are the estimated parameters. Additionally, a parallelization of search algorithms can easily be done by partitioning the search space. It will lead to faster recognition modules. For

an improvement of the recognition rates, however, it will be necessary to choose features different from point features, which will also require some extensions and modifications within the introduced theoretical framework.

Acknowledgement

The authors wish to thank the German Research Foundation (DFG), who partially funded the work reported here under grant SFB 182.

References

1. T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley Publications in Statistics. John Wiley & Sons, Inc., New York, 1958.
2. C. G. E. Boender, A. H. G. Rinnoy Kan, G. T. Timmer, and L. Stougie. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.
3. M. E. Bohachevsky, M. E. Johnson, and M. L. Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28(3):209–217, 1986.
4. T. Caellei, M. Johnston, and T. Robinson. 3D object recognition: Inspirations and lessons from biological vision. In Jain and Flynn [19], pages 1–16.
5. B. Cernuschi–Frias, D. P. Cooper, Y. P. Hung, and P. N. Belhumeur. Toward a model–based Bayesian theory for estimating and recognizing parameterized 3–D objects using two or more images taken from different positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1028–1052, October 1989.
6. P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3):185–210, June 1990.
7. A. Corana, M. Marchesi, and S. Ridella. Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm. *ACM Transactions on Mathematical Software*, 13(3):209–217, 1987.
8. 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.
9. 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*, volume 1, pages 89–96, München, 1994.
10. L. C. W. Dixon and G. P. Szegö, editors. *Towards Global Optimisation*, volume 2, Amsterdam, 1978. North–Holland.
11. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York, 1973.
12. S. M. Ermakov and A. A. Zhiglyavskij. On random search of global extremum. *Probability Theory and Applications*, 28(1):129–136, 1983.
13. F. Gallwitz. Lokalisierung von 3D–Objekten in Grauwertbildern. Technical report, Diploma thesis, Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg, Erlangen, 1994.
14. S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. *Proceedings of the International Congress of Mathematicians*, pages 1496–1517, August 1986.

15. J. Hornegger. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten*. Shaker, Aachen, 1996.
16. 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.
17. R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:267–287, 1983.
18. A. K. Jain. Advances in statistical pattern recognition. In P. A. Devijver and J. Kittler, editors, *Pattern Recognition Theory and Applications*, volume 30 of *NATO ASI Series F: Computer and System Sciences*, pages 1–19. Springer, Heidelberg, 1987.
19. A. K. Jain and P. J. Flynn, editors. *Three–Dimensional Object Recognition Systems*, Amsterdam, 1993. Elsevier.
20. J. Kittler, W. J. Christmas, and M. Petrou. Probabilistic relaxation for matching problems in computer vision. In *Proceedings of the* 4^{th} *International Conference on Computer Vision (ICCV)*, pages 666–673, Berlin, May 1993. IEEE Computer Society Press.
21. W. B. Mann and T. O. Binford. An example of 3–D interpretation of images using Bayesian networks. In *Proceedings of Image Understandig Workshop*, pages 793–801, San Diego, California, January 1992. Morgan Kaufmann Publishers, Inc.
22. D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, San Francisco, 1982.
23. J.W. Modestino and J. Zhang. A Markov random field model–based approach to image interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):606–615, June 1992.
24. H. Niemann, H. Brünig, R. Salzbrunn, and S. Schröder. A knowledge–based vision system for industrial applications. In *Machine Vision and Applications*, pages 201–229, New York, 1990. Springer Verlag.
25. W.H. Press, B.P. Flannery, S. Teukolsky, and W.T. Vetterling. *Numerical Recipes - the Art of Numerical Computing, C Version*. Cambridge University Press, Cambridge, 1988.
26. S. Sarkar and K. L. Boyer. Integration, inference, and management of spatial information using Bayesian networks: Perceptual organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):256–274, March 1993.
27. J. Segen. Model learning and recognition of nonrigid objects. In *Proceedings of Computer Vision and Pattern Recognition*, pages 597–602, San Diego, June 1989. IEEE Computer Society Press.
28. Y. Shang and B. W. Wah. Global optimization for neural network training. *Computer*, 29(3):45–54, 1996.
29. P. Suetens, P. Fua, and A. J. Hanson. Computational strategies for object recognition. *ACM Computing Surveys*, 24(1):5–61, March 1992.
30. M. A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*. Springer Series in Statistics. Springer, Heidelberg, 1993.
31. G. T. Timmer. Global optimization: A stochastic approach. PhD thesis, Rotterdam, 1984.
32. A. A. Törn. A programm for global optimization, multistart with clustering (msc). In P. A. Samet, editor, *European Conference on Applied Information Technology, International Federation for Information Processing (EURO IFIP)*, pages 427–434, London, September 1979. Amsterdam North-Holland Publ. Co.
33. W. M. Wells III. Statistical Object Recognition. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts, February 1993.
34. C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.