From *Mustererkennung 1996*, Informatik FB, Springer, Heidelberg, 1996, (to appear).

Semantic Networks Meet Bayesian Classifiers

J. Hornegger, E. Nöth, V. Fischer, H. Niemann

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

Abstract This paper presents a statistical approach to object recognition and scene analysis, and is motivated by semantic networks, a knowledge representation formalism that allows to represent world knowledge at different levels of abstraction. We show how this explicit knowledge representation scheme and statistical methods can be used to model objects, object groups, and scenes. The theoretical part deals with the construction of statistical models, and preliminary results demonstrate the use of these model densities for object recognition and localization in practice.

1 Introduction

Several reasons motivate the development and the realization of statistical classifiers and the application of probabilistic methods for modeling, localization, and classification of objects or the analysis of complex scenes. The decision theory provides the *optimality of Bayesian classifiers.* Furthermore, image features which are used to classify and localize objects, show *instabilities* with respect to sensor noise and varying illumination conditions. Thus, geometric primitives have a probabilistic behavior which enforces an adequate mathematical description, i.e. statistical object and scene models. A fundamental advantage of stochastic models results from *estimation theory* where a broad range of parameter estimation techniques exists. These methods might be applied for model generation from observations.

The main problem is the construction of statistical descriptions for objects and complex scenes, since there are different levels of abstraction and a variety of phenomena to be modeled. One single feature vector and its associated density function — as usually used in pattern recognition theory [2] — is not sufficient. A successfully applied, but non–statistical technique for object and scene modeling are semantic networks [5]. The following contribution transfers basic ideas from this knowledge representation scheme into the statistical framework and shows the application of probabilistic models for learning and classification. For that reason, we briefly describe the principles for object and scene modeling using the semantic network concept. We introduce the mathematical framework for statistical object and scene modeling, and derive different model densities by specialization. In section 4 we discuss the automatic training of model parameters by applying the *missing information principle* and the associated EM–algorithm [1]. The experimental part shows preliminary results for both 2D– and 3D–examples. The paper ends up with a brief summary of the main points, draws some conclusions, and gives several hints for future research.

2 Knowledge Based Pattern Analysis

For knowledge based pattern analysis a representation scheme is needed to represent 'the world'. A system which analyzes, for example, traffic scenes has to know about houses, streets, cars, etc. One scheme which has been widely used in pattern analysis, are *semantic networks*. Semantic networks were introduced in [5] as a model of human memory. Several formalisms have been developed to use semantic networks for expert and pattern analysis systems [4].

A semantic network is a directed, labeled graph consisting of nodes (*concepts*) for the representation of facts or objects and links that provide relations between concepts. Concepts may be related to others by different types of links: *Part* links allow the decomposition of a concept into more simple concepts, its constituents. *Specialization* links (or *is_a, a_kind_of* links) are used to establish inheritance of properties and parts from general concepts to more special ones. *Concrete* links are introduced in the Erlangen semantic network system ERNEST[4] to allow the representation of different levels of abstraction which must be considered in pattern analysis, e.g. pixels, segmentation objects or real world objects.

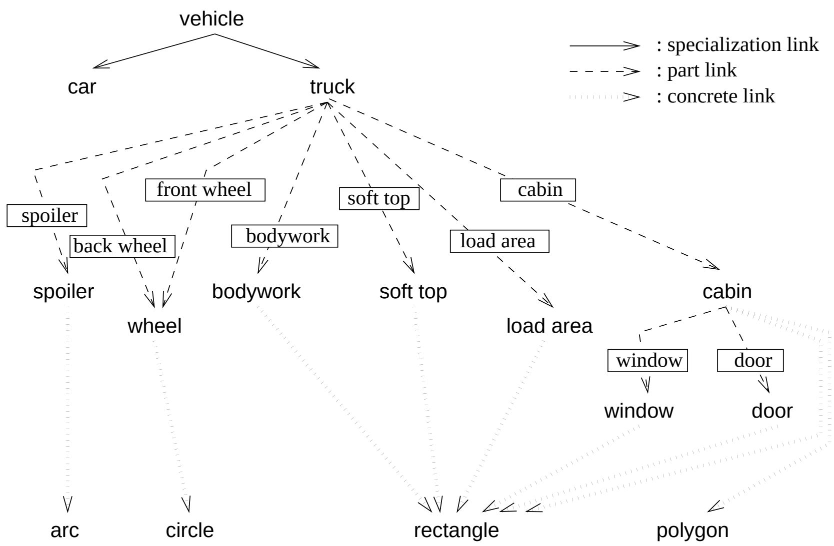

Figure1. Representation of the concept 'truck' in a semantic network formalism

Figure 1 shows the representation of the concept 'truck' in a semantic network formalism. A truck is a vehicle (specialization link), it consists of parts like wheels (part), which show up in the captured digital images as circles (concrete link). If the system has to answer the question 'Is there a truck in the image?', it has to assign results from segmentation to concepts which have a link to the geometric level. Then it has to compose the truck out of its parts.

The example shows that the system needs to know a lot about concepts (like wheels or doors) just to be able to hypothesize the concepts of interest (like cars and trucks). Besides, the combinations of parts which are optional (like the spoiler) and obligatory (like the wheels) have to be modeled. Things get worse, if occlusion occur and the obligatory parts cannot be detected in the sensory input.

To avoid these difficulties, we suggest statistical concept detectors for all the concepts that we need to know about in a given application, i.e. in the example we need a concept detector for a truck, a car, and a vehicle (as well as for houses and streets). The concept detector for a truck could replace the truck's subnet in Figure 1, if no details about trucks are needed in the application. If the system should be able to answer questions like Does the truck have a spoiler?, the subnet has to be expanded and concept detectors for the parts of the truck have to be trained as well.

Statistical models will not completely include the structural description like semantic networks, and we cannot derive a one–to–one correspondece between semantic networks and statistical models. But using a statistical framework the concept detectors for more general concepts (like vehicle) can easily be derived from detectors for more specialized concepts (like car and truck) by the introduction of marginal density functions. The same applies for the composition of concepts out of its parts.

3 Statistical Modeling

The above concepts are the foundation for the introduction and definition of statistical models (*model densities*) which partially solve some of the mentioned problems — at least from the theoretical point of view. For purposes of simplification, we restrict model densities to object groups, to elements of object groups (specialization link), to parts of each object (part link), and to the concrete representation of object parts (concrete link). Composition rules will be discussed as well as training algorithms. Figure 2 shows two similar 2D–objects

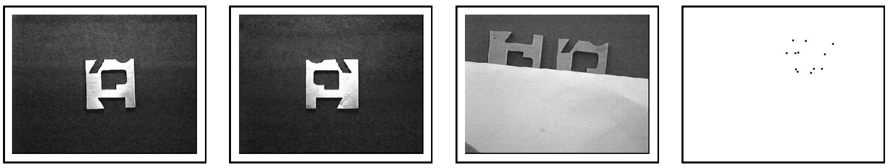

Figure2. Gray–level images showing two objects P1 and P2, their combination in a scene, and the point features obtained for P1

that will be used in the experiments described below, their combination to a scene (partially occluded), and the observable point features that belong to the object P1 in the scene. A possible model representation of such scenes containing geometric objects in a semantic network formalism is shown in Figure 3.

A statistical description of an object of class Ω_κ $(1 \leq \kappa \leq K)$ corresponds to a density function $p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t})$, where not only one feature vector, but a set of features $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m\}$ of varying size might be observed; each element of $\boldsymbol{O}$ is understood as a random variable. The parameters $\boldsymbol{B}_\kappa$ are *model–specific* and $\boldsymbol{R}$ and $\boldsymbol{t}$ are *pose–specific* parameters, which are necessary, since objects might have different positions in the world coordinate system. If the parameters $\boldsymbol{B}_\kappa$, $\boldsymbol{R}$, $\boldsymbol{t}$ of the model densities are known for a given set of observations, the classification is based on the Bayesian decision rule $\kappa = \text{argmax}_\lambda\, p(\Omega_\lambda)p(\boldsymbol{O}|\boldsymbol{B}_\lambda, \boldsymbol{R}, \boldsymbol{t})$. Now two problems arise: What does the structure of model densities for object groups, single objects, and parts of objects look like? How can model and pose

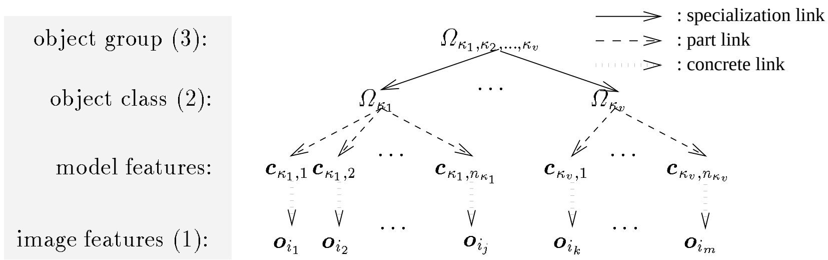

Figure3. Representation of the example object scene in a semantic network formalism (the numbers in the parentheses refer to the equations in the text)

parameters for different levels be estimated? For answering these questions, we start with the lowest level, the density for single observable object features.

Statistical Modeling of Features. An image feature $\boldsymbol{o}_k$ $(1 \leq k \leq m)$ has a corresponding model feature $\boldsymbol{c}_{\kappa,l_k}$ $(1 \leq l_k \leq n_\kappa)$, where $\zeta_\kappa(\boldsymbol{o}_k) = l_k$ is an alignment function of a scene feature to the index of a corresponding model feature. The density of features matched to $\boldsymbol{c}_{\kappa,l_k}$ is the parameterized function $p(\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l_k}, \boldsymbol{R}, \boldsymbol{t})$, where $\boldsymbol{a}_{\kappa,l_k}$ are *feature–specific* parameters. If the assignment ζ_κ of image features is known and the observations are pairwise statistically independent, we have the conditional density which is even suitable for feature sets of varying size:

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

Statistical Modeling of Objects. Usually, an object consists of a set of perhaps related primitives and additionally the assignment of image and model features is not part of the observation. These missing data cause problems when evaluating Eq. (1), because ζ_κ has to be known. By the definition of a discrete m–dimensional random vector $\boldsymbol{\zeta}_\kappa = (\zeta_\kappa(\boldsymbol{o}_1), \zeta_\kappa(\boldsymbol{o}_2), \ldots, \zeta_\kappa(\boldsymbol{o}_m))^T \in \mathbb{R}^m$, we associate with each assignment ζ_κ a discrete probability $p(\boldsymbol{\zeta}_\kappa)$; thus, the density for observing a set of features $\boldsymbol{O}$ and a correspondence ζ_κ can be computed. The matching ζ_κ is not observable and can be eliminated within the chosen statistical framework by marginalization

$$p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \sum_{\zeta_\kappa} p(\boldsymbol{\zeta}_\kappa) \prod_{k=1}^{m} p(\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,\zeta_\kappa(\boldsymbol{o}_k)}, \boldsymbol{R}, \boldsymbol{t}) \quad , \tag{2}$$

where $\boldsymbol{B}_\kappa$ includes all feature–specific parameters and the discrete probabilities $p(\boldsymbol{\zeta}_\kappa)$.

A density function for an object thus is composed by feature densities and the discrete probabilities for the matching.

Statistical Modeling of Scenes. The next level of abstraction is the modeling of object groups. Let $\Omega_{\kappa_1,\ldots,\kappa_v} = \bigcup_{i=1}^{v}\{\Omega_{\kappa_i}\}$; we want to know, whether an observation corresponds to the object group $\Omega_{\kappa_1,\ldots,\kappa_v}$. In the case of object groups there are two stages of alignment: the observed features have to be assigned by $\zeta_{\kappa_1,\ldots,\kappa_v}$ to an element of the object group, i.e. $\zeta_{\kappa_1,\ldots,\kappa_v}(\boldsymbol{o}_k) = \kappa_i \in \{\kappa_1,\ldots,\kappa_v\}$, and the matching ζ_{κ_i} between the image and the model features. The interpretation of these matching functions results in the density function

$$p(\boldsymbol{O}|\boldsymbol{B}_{\kappa_1,\ldots,\kappa_v}, \boldsymbol{R}_{\kappa_1}, \boldsymbol{t}_{\kappa_1}, \ldots, \boldsymbol{R}_{\kappa_v}, \boldsymbol{t}_{\kappa_v}) = \prod_{k=1}^{m} \sum_{\zeta_{\kappa_1,\ldots,\kappa_v}} p(\boldsymbol{\zeta}_{\kappa_1,\ldots,\kappa_v}) p(\boldsymbol{o}_k|\boldsymbol{B}_{\zeta_{\kappa_1,\ldots,\kappa_v}(\boldsymbol{o}_k)}, \boldsymbol{R}_{\zeta_{\kappa_1,\ldots,\kappa_v}(\boldsymbol{o}_k)}, \boldsymbol{t}_{\zeta_{\kappa_1,\ldots,\kappa_v}(\boldsymbol{o}_k)}) \quad (3)$$

for an object group, where all objects might have different pose parameters.

In summary we have three types of densities which are suitable for the classification of *object groups* (3), *objects* (2), and for the *alignment of image and model features* (1).

4 Missing Information Principle

The density types mentioned are also of interest within the learning phase. We have to estimate model parameters from available data. In general, the observation for the estimation of model parameters may consist of features with an unknown match. In the worst case, the parameter estimation is based on objects of an object group without knowing the object classes. Consequently, the training data is determined by incomplete data, i.e. the observable information is the difference of the complete and the hidden information. A translation of this principle into statistics results in the EM–algorithm, an iterative parameter estimation technique which can deal with these incomplete training sets. Let $\boldsymbol{X}$ be the observable and $\boldsymbol{Y}$ the hidden data. If the densities of these random variables are known, we have $p(\boldsymbol{X}|\boldsymbol{B}) = p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{B})/p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{B})$ and thus we get $\log p(\boldsymbol{X}|\boldsymbol{B})^{-1} = \log p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{B})^{-1} - \log p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{B})^{-1}$, which is the mathematical definition of the observable information. Let $\widehat{\boldsymbol{B}}^{(i)}$ be the estimate for $\boldsymbol{B}$ after the i–th iteration step. We compute the expectation conditioned by $\boldsymbol{X}$ over the complete data and apply Jensen's inequality [1]. We can show that an increase of the expectation

$$Q(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}) = \int p(\boldsymbol{Y}|\boldsymbol{X},\widehat{\boldsymbol{B}}^{(i)}) \log p(\boldsymbol{X},\boldsymbol{Y}|\widehat{\boldsymbol{B}}^{(i+1)})\, d\boldsymbol{Y} \quad (4)$$

implies a growth of the log likelihood for the observed data. The EM–algorithm utilizes this observation and maximizes iteratively (4) with respect to $\widehat{\boldsymbol{B}}^{(i+1)}$. If we have to deal with an incomplete data estimation problem, now the so–called *Kullback–Leibler statistics* (4) is computed and an iterative update of the parameters is started.

5 An Example

Up to now, the model densities were introduced in an abstract manner without any concrete applications. For illustration we discuss practical examples for 2D– and 3D–object recognition. The objects used for experiments are P1 and P2 in Figure 2 and the objects shown in Figure 4.

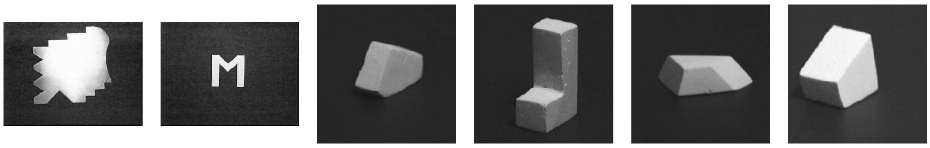

Figure4. 2D– (P3, P4) and 3D–objects (Q1–Q4) used for experiments

Conditions. Assuming that an object of class Ω_κ is described by a set of feature sequences $\boldsymbol{C}_\kappa = \{\boldsymbol{c}_{\kappa,l} | 1 \leq l \leq n_\kappa\}$ in the model space, where $\boldsymbol{c}_{\kappa,l} = [\boldsymbol{c}_{\kappa,l,1}, \boldsymbol{c}_{\kappa,l,2}, \ldots, \boldsymbol{c}_{\kappa,l,q}]$ and $\boldsymbol{c}_{\kappa,l,s} \in \mathbb{R}^{D_m}$. These sequences might be lines, which are represented as pairs including start and end points and thus $q = 2$, or single points where $q = 1$. An affine mapping, not necessarily invertible, is given by $\boldsymbol{R} \in \mathbb{R}^{D_o \times D_m}$ and $\boldsymbol{t} \in \mathbb{R}^{D_o}$ and defines a transform from the D_m–dimensional model into the D_o–dimensional image space. The resulting sequences of observable image features $\boldsymbol{O} = \{\boldsymbol{o}_k | 1 \leq k \leq m\}$, with $\boldsymbol{o}_k = [\boldsymbol{o}_{k,1}, \boldsymbol{o}_{k,2}, \ldots, \boldsymbol{o}_{k,q}]$ and $\boldsymbol{o}_{k,s} \in \mathbb{R}^{D_o}$, are the basis for learning, localization, and classification. During the transform into the image space, the ordering of features is modified by elements τ of a non–observable set of permutations Υ. For example, if the line features, as mentioned above, are used, the identification of start and end point gets lost during the projection; permitted permutations are the identity and the transposition. The object recognition is based on scenes. Beside object features, these images contain also background features.

In addition to these constraints, the statistical modeling assumes pairwise independent assignments of image to object and background features and pairwise independent matchings of object features to the object's components. All elements $\boldsymbol{o}_{k,s} \in \mathbb{R}^{D_o}$ $(1 \leq k \leq m,\ 1 \leq q \leq s)$ of the object's feature sequences are assumed to be the result of affine transformed normally distributed random vectors of the model space, where $\boldsymbol{\mu}_{\kappa,l_k,s}$ $(1 \leq l_k \leq n)$ is the mean vector and $\boldsymbol{\Sigma}_{\kappa,l_k,s}$ the covariance matrix of the s–th element of the corresponding l_k–th feature sequence. The parametric distribution of background features is assumed to be uniform and independent of the object's pose determined by $\boldsymbol{R}$ and $\boldsymbol{t}$.

Model Density. The components of random vectors induced by the involved matching functions are assumed to be statistical independent. Thus, for a given observation $\boldsymbol{O}$ we have $p(\zeta_\kappa) = \prod_{k=1}^{m} p(\zeta_\kappa(\boldsymbol{o}_k) = l_k) = \prod_{k=1}^{m} p_{\kappa,l_k}$ for all features corresponding to the object. The probability to observe an element concerning to the background is $p(\zeta_{\kappa,H}(\boldsymbol{o}_k) = l_k) = p_H$.

For simplification, let us first assume that all image features correspond to the object. The density function for an observed sequence is the marginalization over all alignments ζ_κ and all permutations τ, i.e.

$$p(\boldsymbol{O}|\boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \sum_{l=1}^{n_\kappa} \prod_{k=1}^{m} p_{\kappa,l} p(\tau) \sum_{\tau \in \Upsilon} \prod_{s=1}^{q} p(\boldsymbol{o}_{k,\tau(s)} | \{\boldsymbol{\mu}_{\kappa,l,\tau(s)}, \boldsymbol{\Sigma}_{\kappa,l,\tau(s)}\}, \boldsymbol{R}, \boldsymbol{t}) \quad (5)$$

In scenes, where background features occur, a given feature sequence $\boldsymbol{o}_k$ has to be aligned to the object or to the background. For this second level of abstraction the marginal density over $\zeta_{\kappa,H}$ and ζ_κ is computed, and we get the model density

$$p(\boldsymbol{O}|\boldsymbol{B}_H, \boldsymbol{B}_\kappa, \boldsymbol{R}, \boldsymbol{t}) = \prod_{k=1}^{m} \left(p_H p(\boldsymbol{o}_k|\boldsymbol{a}_H) + (1-p_H) \sum_{l=1}^{n_\kappa} p_{\kappa,l} p(\boldsymbol{o}_k|\boldsymbol{a}_{\kappa,l}, \boldsymbol{R}, \boldsymbol{t}) \right) \quad (6)$$

where $\boldsymbol{B}_H = \{p_H, \boldsymbol{a}_H\}$ are background–specific parameters and $\boldsymbol{a}_{\kappa,l}$ the parameters of the sequence elements. This model density is suitable for the localization and the classification of objects, and allows different types of sensors for varying levels of abstraction: objects can be localized and classified without knowing the correspondence of object, background and model features, the subset of object and image features can be determined without computing the correspondence of object and model features, and the correspondence of object and model features can be computed without knowing the permutation τ.

Training Stage. The first problem is the estimation of model parameters $\boldsymbol{B}_\kappa = \{p_{\kappa,l}, \boldsymbol{\mu}_{\kappa,l,s}, \boldsymbol{\Sigma}_{\kappa,l,s} | 1 \leq s \leq q, 1 \leq l \leq n_\kappa\}$. Assuming that the feature sets ${}^1\boldsymbol{O}, {}^2\boldsymbol{O}, \ldots, {}^N\boldsymbol{O}$ of the training images include only features of a known object. Each set ${}^\varrho\boldsymbol{O}$ has the cardinality ${}^\varrho m$, and the corresponding affine transformation from the model into the image space is given by ${}^\varrho\boldsymbol{R}$ and ${}^\varrho\boldsymbol{t}$. The non–observable part of the training data consists of the missing matching ζ_κ and permutation τ operating on the feature sequences. Using this knowledge, the application of the missing information principle and the EM–Algorithm results in the learning formulas

$$\widehat{p}_{\kappa,l}^{(i+1)} = \frac{1}{\sum_{\varrho=1}^{N} {}^\varrho m} \sum_{\varrho=1}^{N} \sum_{k=1}^{{}^\varrho m} \frac{\widehat{p}_{\kappa,l}^{(i)} p({}^\varrho\boldsymbol{o}_k | \widehat{\boldsymbol{a}}_{\kappa,l}^{(i)}, {}^\varrho\boldsymbol{R}, {}^\varrho\boldsymbol{t})}{p({}^\varrho\boldsymbol{o}_k | \widehat{\boldsymbol{B}}_\kappa^{(i)}, {}^\varrho\boldsymbol{R}, {}^\varrho\boldsymbol{t})} \quad (7)$$

for the probabilistic description of the matching function, and

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

$$\sum_{\varrho=1}^{N} \sum_{k=1}^{{}^\varrho m} \sum_{\tau \in \Upsilon} p({}^\varrho\boldsymbol{o}_k | l, \tau, \widehat{\boldsymbol{B}}_\kappa^{(i)}, {}^\varrho\boldsymbol{R}, {}^\varrho\boldsymbol{t}) {}^\varrho\boldsymbol{R}^T ({}^\varrho\boldsymbol{R} \widehat{\boldsymbol{\Sigma}}_{\kappa,l,s}^{(i+1)} {}^\varrho\boldsymbol{R}^T)^{-1} ({}^\varrho\boldsymbol{o}_k - {}^\varrho\boldsymbol{t}) \quad (8)$$

for the re–estimation of mean vectors [3], which allows the unsupervised estimation of mean vectors from projected features. For the estimation of covariance matrices there exists no closed–form re–estimation formula. Numerical optimization techniques can be used for the maximization of these Kullback–Leibler statistics.

Localization and Classification. The introduced model densities are applied for 2D and 3D experiments, which use point ($q = 1$) and line ($q = 2$) features. Figure 5 shows impressive examples for 2D–object localization where partial occlusion and background features occur. Classification experiments with 1000 images including 2D– and 1600 images with 3D–objects result in recognition rates of 93 and 68 resp. 59 percent (see Table 1). These examples proof the correctness of the chosen statistical approach and suggest future research in the application of probabilistic object modeling and recognition.

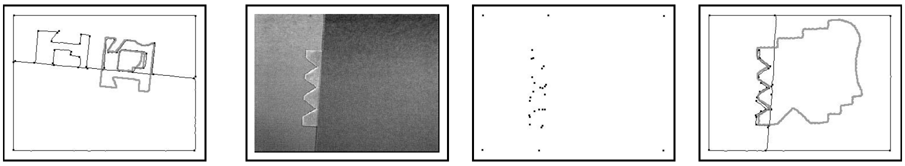

Figure5. The localization result for P1 in the gray–level image of Figure 2; example for partially occluded object P3 (gray–level image, observed point features, localization)

2D–object	**recognition [%]**		**time [sec]**	
	$q=1$	$q=2$	$q=1$	$q=2$
P1	98	96	46	314
P2	94	94	50	337
P3	92	96	56	407
P4	90	85	33	231
mean	93	93	46	322

3D–object	**recognition [%]**		**time [sec]**	
	$q=1$	$q=2$	$q=1$	$q=2$
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

Table1. Recognition results and computation time on a HP 735 for 2D– and 3D–experiments

6 Conclusions

The semantic network concept is suitable for descriptions of complex objects and scenes. It allows abstraction and specialization. This paper has introduced an alternative statistical approach to scene and object modeling. In contrast to classical pattern recognition theory an object is not represented by a single feature vector, but a set of related features. Different stages of modeling are possible, and the problem of automatic learning was solved by the EM–algorithm. Future work should concentrate on the elaboration of further statistical models for more complex real world objects and the decrease of computation times.

References

1. 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.
2. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* J. Wiley, New York, 1972.
3. J. Hornegger. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten.* Dissertation, Technische Fakultät, Universität Erlangen–Nürnberg, Erlangen, 1996.
4. H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. Ernest: A semantic network system for pattern analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:883–905, 1990.
5. M. Ross Quillian. Semantic Memory. In Marvin Minsky, editor, *Semantic Information Processing*, pages 216–270. MIT Press, Cambridge, 1968.