

A Framework For Statistical 3-D Object Recognition

Dietrich Paulus & Joachim Hornegger & Heinrich Niemann

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

Accepted at

PRIP V 1997

Submitted for inclusion into the proceedings

Rev.: 4.19

(Max: 6 pages / actual 6 pages)

Contents

1	Introduction	1
2	Software-Engineering for Computer Vision	2
3	Statistical Object Recognition	2
4	Object Recognition Using 2-D Points and Lines	4
5	Discussion and Further Work	6
	Bibliography	6
	References	6

A Framework For Statistical 3–D Object Recognition ¹

Dietrich Paulus & Joachim Hornegger & Heinrich Niemann

*Lehrstuhl für Mustererkennung (Informatik 5)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstr. 3, D-91058 Erlangen, Germany*

Abstract

In this contribution we describe an object-oriented software architecture for image segmentation, 3–D pose estimation as well as Bayesian object recognition: models are represented by densities, model generation corresponds to parameter estimation tasks, and the identification applies the Bayesian decision rule. We show results of 3–D object recognition experiments based on the observation of 2–D points or lines.

Keywords: Object-oriented programming, statistical object recognition

1 Introduction

Image processing and object recognition systems are expected to provide optimality with respect to various factors like efficiency, robustness or modularity and maintainability of software. In this paper we introduce a Bayesian framework for 3–D object recognition and sketch its object-oriented implementation. Whereas Bayesian methods provide solutions for many problems in low-level image processing and in pattern recognition, classification in computer vision is still dominated by geometrical, model-based approaches [3].

In Sect. 2 we briefly report on basic aspects of software-engineering for computer vision. In Sect. 3 we present a novel and unconventional probabilistic framework for 3–D vision: statistical methods for object modeling, algorithms for the automatic estimation of model parameters, Bayesian decision, and localization methods. In Sect. 4 we apply this framework to the problem of 3–D object recognition from 2–D views. A discussion and suggestions for further research conclude the paper.

¹This work was funded partially by the *Deutsche Forschungsgemeinschaft* (DFG) under grant number SFB 182. Only the authors are responsible for the contents.

2 Software–Engineering for Computer Vision

By the dissemination of the *Image Understanding Environment* (IUE) [2], data representation is now widely implemented in classes using the C++ programming language. However, little has been published yet on hierarchies of *operations* for image processing: A common implementation platform for image analysis has to provide unified interfaces to both data and algorithms. A C++ class library called *Ἱππος* (HIPPOS) [8] was designed for the representation of data computed from image processing. A uniform data representation object called the *segmentation object* is used to collect information of segmentation processes. In [4] this system is extended to a hierarchy of operators for image processing and analysis. The reason for this approach is that polymorphic interfaces to image processing operations defined by classes provide and enforce much more uniform syntax and semantics than function libraries. When basic rules for efficiency [8] are observed, no difference in execution times for image processing *objects* can be measured compared to *functions*.

The benefit of operator objects is manifold: the interfaces are safer, easier to document, extensible, etc. Unified interfaces to algorithms enforced by this programming style facilitate simple exchange of individual modules, as demonstrated in [5,6]. For statistical object recognition we need the following basic modules: algorithms for feature detection, interchangeable probabilistic models, parameter estimation algorithms and various global and local optimization algorithms which accept as arguments a function to be optimized, for instance, a parametric density function.

3 Statistical Object Recognition

A Bayesian framework for 3–D object recognition requires that the appearance of objects in the image plane is characterized using probability density functions. These densities have to incorporate prior knowledge on objects, rotation and translation, self–occlusion, projection to the image space, the assignments of image and model features as well as the statistical modeling of errors and inaccuracies caused by varying illumination, sensor noise or segmentation errors [5]. We call these densities *model densities*. The structure of these models can vary: it can be a single multivariate Gaussian density, a hidden Markov model or some other type of density. All model densities have to provide similar methods, like evaluating the density for a given set of random variables. We have implemented a class hierarchy for densities including an abstract base class which defines the interface for all model generation, pose estimation, and classification modules.

Let us assume K possible object classes and observations \mathbf{O} of feature vectors \mathbf{o}_k in a segmentation object $\mathbf{O} = \{\mathbf{o}_k \in \mathbb{R}^2 | 1 \leq k \leq m\}$, where the number m varies for different images. Appearance and position of features in the image show a probabilistic behavior. The statistical description of an object belonging to class Ω_κ consists of a model density $p(\mathbf{O}|\mathbf{B}_\kappa, \mathbf{R}, \mathbf{t})$ combined with discrete priors $p(\Omega_\kappa)$, $1 \leq \kappa \leq K$, for the probability of a single object of class Ω_κ to appear in the scene. The priors are estimated by relative frequencies of objects in the training samples. The parameter \mathbf{R} denotes rotation and projection from the model space to the image plane; \mathbf{t} represents translation. The set \mathbf{B}_κ contains the model-specific parameters for the behavior of features as well as the parameters for the assignment of image and model features.

For the explicit definition of $p(\mathbf{O}|\mathbf{B}_\kappa, \mathbf{R}, \mathbf{t})$ we use the observed feature set \mathbf{O} and the corresponding 3-D features in the model space $\mathbf{C}_\kappa = \{\mathbf{c}_{\kappa,1}, \mathbf{c}_{\kappa,2}, \dots, \mathbf{c}_{\kappa,n_\kappa}\}$, where in general $n_\kappa \neq m$ due to segmentation errors and occlusion. Let the parametric density of the model feature \mathbf{c}_{κ,l_k} corresponding to \mathbf{o}_k be given by $p(\mathbf{c}_{\kappa,l_k}|\mathbf{a}_{\kappa,l_k})$, where $\mathbf{a}_{\kappa,l}$ ($l = 1, \dots, n_\kappa$) characterize single model features. For a normally distributed 3-D point, for instance, \mathbf{a}_{κ,l_k} denotes the mean vector and the covariance matrix. A standard density transform results in the density $p(\mathbf{o}_k|\mathbf{a}_{\kappa,l_k}, \mathbf{R}, \mathbf{t})$, which characterizes the statistical behavior of the feature \mathbf{o}_k in the image plane dependent on the object's pose parameters.

The probabilistic modeling of the assignment from image to model features is based on discrete random vectors. An assignment function ζ_κ defines a discrete mapping from an observed feature \mathbf{o}_k to the index $l_k \in \{1, 2, \dots, n_\kappa\}$ of the corresponding model feature \mathbf{c}_{κ,l_k} , i.e., $\zeta_\kappa(\mathbf{o}_k) = l_k$. A set of observed features can thus be associated with the assignment random vector $\boldsymbol{\zeta}_\kappa = (\zeta_\kappa(\mathbf{o}_1), \zeta_\kappa(\mathbf{o}_2), \dots, \zeta_\kappa(\mathbf{o}_m))^T$ which is related to the discrete probability $p(\boldsymbol{\zeta}_\kappa)$, i.e., the matching problem is also modelled statistically. The discrete probability of $p(\boldsymbol{\zeta}_\kappa)$ extends the probability density function for observing the set of features \mathbf{O} . Due to the statistical interpretation of ζ_κ , the non-observable assignment can be eliminated by the following marginalization:

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

If the structure of the model density (i.e., the number of model features and the dependency structure of single assignments) is known, algorithms for the estimation of the parameter set \mathbf{B}_κ exist (Sect. 4). The computation of \mathbf{B}_κ for each object class Ω_κ , $\kappa = 1, 2, \dots, K$ requires $p(\boldsymbol{\zeta}_\kappa)$ and $\{\mathbf{a}_{\kappa,l}\}$. Due to the projection of the 3-D world to the 2-D image plane, the range information is lost. Furthermore, the assignment of image and model features is not a component of the observations. The calculation of \mathbf{B}_κ thus corresponds to an incomplete data estimation problem which can be solved using the Expectation Maximization algorithm (EM algorithm, [1,6]).

The framework introduced so far requires a a minor modification of the standard Bayesian decision rule, since a segmentation object \mathbf{O} is given instead of a single vector, and the unknown pose parameters are part of the probability density. The modified Bayesian decision rule for the statistical classification of objects is: $\lambda = \operatorname{argmax}_{\kappa} p(\Omega_{\kappa}|\mathbf{O}) = \operatorname{argmax}_{\kappa} p(\Omega_{\kappa})p(\mathbf{O}|\mathbf{B}_{\kappa}, \mathbf{R}, \mathbf{t})$. The a posteriori probabilities $p(\Omega_{\kappa}|\mathbf{O})$ cannot be evaluated explicitly. The pose estimation stage has to compute the best orientation and position \mathbf{R}, \mathbf{t} before the class decision is possible. This corresponds to the maximization problem $\{\widehat{\mathbf{R}}, \widehat{\mathbf{t}}\} = \operatorname{argmax}_{\mathbf{R}, \mathbf{t}} p(\mathbf{O}|\mathbf{B}_{\kappa}, \mathbf{R}, \mathbf{t})$ which requires a global optimization of a concave multimodal likelihood function. A class hierarchy of probabilistic optimization routines similar to the operator hierarchy in Sect. 2 or to the hierarchy for model densities allows practically efficient solutions and alternative strategies, such as stochastic relaxation, simulated annealing, and adaptive random search [5]. The abstract base class provides unified interfaces with methods like `minimize` or `maximize`, deferring all implementation details to derived classes.

4 Object Recognition Using 2-D Points and Lines

The framework described in Sect. 3 was used for the recognition of 3-D objects based on 2-D images [5]: we assume that each input image (e.g. Fig. 1) is transformed into a segmentation object of 2-D feature vectors $\mathbf{O} = \{\mathbf{o}_k \in \mathbb{R}^2 | 1 \leq k \leq m\}$. The elements \mathbf{o}_k may be points (e.g. corners or vertices) or lines, which can be detected by several combinations of segmentation operators which all result in the uniform segmentation object.

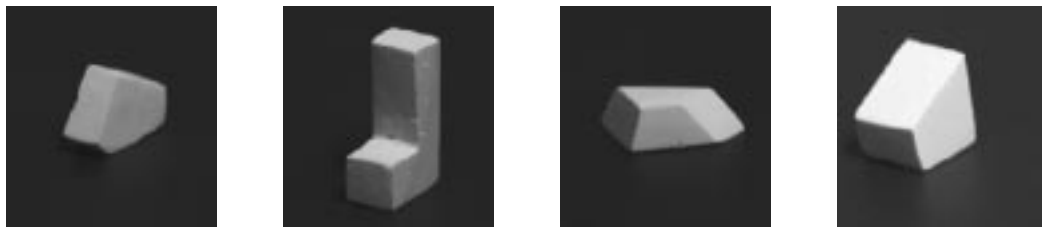


Fig. 1. Simple polyhedral 3-D objects ($\Omega_1, \Omega_2, \Omega_3, \Omega_4$) used in the experiments

For segmented point features, \mathbf{B}_{κ} provides the parameters characterizing the assignments as well as the accuracy and stability of the object points. Closed form iteration formulas can be found for normally distributed point features, which allow the estimation of mean vectors from projections without knowing corresponding features of different views [5].

Although model densities as defined in Sect. 3 tolerate variations of segmentation objects to some extent, stable segmentation is desired and improves

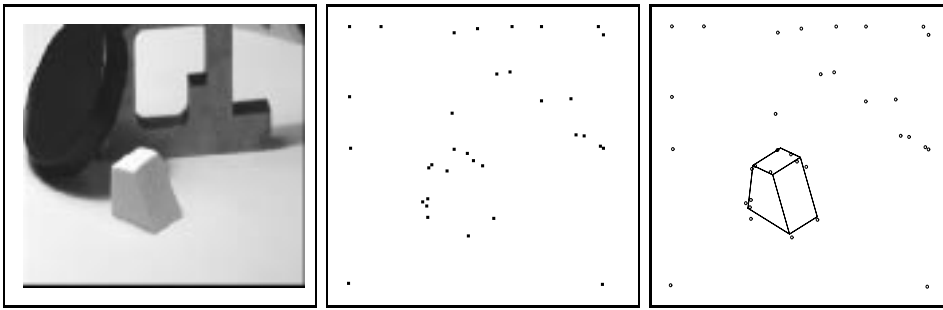


Fig. 2. Experiment for object recognition with heterogeneous background

recognition and localization. From the considerations in Sect. 2, several sequences of operations were evaluated manually. For observed point features, best results were obtained by edge detection, line following and subsequent corner detection using the H -curvature introduced in [4] (Fig. 2 middle).

A training set of 1600 and a test set of 1600 randomly chosen views of the objects in Fig. 1 were evaluated on an HP 9000/735 (99 MHz, 124 MIPS). Fig. 2 (left) shows an example for the localization of one object in a scene of three objects. The segmented points and the estimated position of the object of interest are illustrated in Fig. 2 (middle and right). The recognition rates and run times for object classification are summarized in Tab. 1.

3-D object	recognition rate [%]		run time per image [sec]	
	points	lines	points	lines
Ω_1	47	44	466	1882
Ω_2	78	82	485	2101
Ω_3	58	36	465	1933
Ω_4	89	76	471	1520
average	68	59	472	1859

Tab. 1. 3-D experiments for classification using 1600 images

A comparison of pose estimation techniques based on the geometrical alignment method [7] and the statistical approach on 49 images yielded a correct pose estimation for 45 images using the statistical approach, compared to 38 correct results for the alignment method. The average computation time per image was 70s for the alignment method. The abstract interface for global optimization was used to evaluate different optimization strategies. Using the adaptive random search method, an average computation time of 80s could be reached.

This flexible formalism of model densities can also be extended to use multiple views for pose estimation or classification [5] which remarkably improves recognition rates: in experiments using 400 views, the correct pose parameters increased from 96% to 100%. In average it takes, however, 420s to compute the correct position based on multiple views.

5 Discussion and Further Work

The formalism introduced in this paper can be applied to other problems in computer vision and pattern recognition. Instead of segmented points or straight lines we could also use, for instance, gray-level based object recognition algorithms or embed relational dependencies of features. The method and the software modules can also be used for speech processing applications [5]. Hidden Markov models are derived from (1), if statistically dependent assignments of first order are assumed and the feature transform is omitted [5]. Extended hidden Markov models including feature transforms result from the theoretical framework as well as standard mixture densities [5].

Segmentation algorithms are often evaluated by simple visual inspection. We argue that by a uniform object-oriented interface – as provided by our system – we can easily exchange individual parts in the sequence of operation steps or vary parameters in one module, and then judge the overall system's performance. This opens a wide range of still challenging and still unsolved optimization problems for computer vision. Future research should concentrate on the development of methods which allow the selection the optimal features, the best model density, and the most efficient algorithms for solving a given vision task. An object-oriented framework seems to be a necessary precondition to obtain these techniques.

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] J. Mundy et al. The image understanding environments program. In *Image Understanding Workshop*, pages 185–214, Hawaii, Jan. 1992.
- [3] O. Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.
- [4] M. Harbeck. *Objektorientierte linienbasierte Segmentierung von Bildern*. Shaker Verlag, Aachen, 1996.
- [5] J. Hornegger. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten*. Shaker, Aachen, 1996.
- [6] J. Hornegger, H. Niemann, D. Paulus, and G. Schlottke. Object recognition using hidden Markov models. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, volume 16 of *Machine Intelligence and Pattern Recognition*, pages 37–44, Amsterdam, June 1994. Elsevier.
- [7] D.P. Huttenlocher. Recognition by alignment. In A. K. Jain and P. J. Flynn, editors, *Three-Dimensional Object Recognition Systems*, pages 311–324. Elsevier, Amsterdam, 1993.
- [8] D. Paulus and J. Hornegger. *Pattern Recognition of Images and Speech in C++*. Advanced Studies in Computer Science. Vieweg, Braunschweig, 1997.

