Object recognition using hidden Markov models

J. Hornegger, H. Niemann, D. Paulus and G. Schlottke [a]

[a]Lehrstuhl für Mustererkennung (Informatik 5),
Friedrich–Alexander Universität Erlangen–Nürnberg,
Martensstr. 3, D–91058 Erlangen, Germany

appeared in
Proceedings: Pattern Recognition in Practice IV, p. 37–44
Vlieland, Netherlands, 1994

This contribution describes a statistical approach for learning and classification of two–dimensional objects based on segmented grey–level images. The research concentrates on the application of Hidden Markov Models in the field of computer vision. For that purpose, the theory of Hidden Markov Models is shortly introduced with emphasis on different types of stochastic automata. In the experiments we evaluate several types of Hidden Markov Models with respect to affine invariant geometric features. The implementation uses an object–oriented class hierarchy for different variants of Hidden Markov Models. The paper concludes with a discussion of Hidden Markov Models for 3–D computer vision purposes.

1. INTRODUCTION

For classification purposes, knowledge about the objects is necessary, which can be acquired and represented in various ways. One possibility is the explicit representation of knowledge for a particular problem domain [9]. In some cases, the knowledge base can then be generated automatically in a knowledge–acquisition phase using learning sets of images. Distortions and noise in the input data are inevitable and may cause problems for the algorithms. However, statistical learning algorithms exist which are robust with respect to variations of the input data. Consequently, a statistical approach for learning objects seems natural. In the area of speech analysis, the statistical approach has been very successful; stochastic automata — especially Hidden Markov Models (HMMs) — are an established tool for that purpose.

The following paper is dedicated to the problem of learning 2–D objects by examples and the design of efficient recognition algorithms based on information extracted from training samples. The used technique is based on HMMs combined with affine invariant geometric features. Image segmentation and representation and training of the HMMs are implemented following the object–oriented programming paradigm. The experiments are based on four different object classes, where for each object class 50 images and the corresponding extracted features are used for estimating the model parameters. The contribution concludes with a discussion of the practical results and the consideration of a statistical approach to solve the 3–D object recognition problem from 2–D views.

2. STATISTICAL OBJECT RECOGNITION

Theoretical aspects of statistical pattern recognition and classificatin are well developed [5]. Current research in this field focuses on the investigation of efficient and robust algorithms for practical recognition systems [8]. Most classification algorithms are based on Bayesian decision theory, where the decision relies on the a posteriori probability of the classes. Let us assume, we have classes $\Omega_1, \Omega_2, \ldots, \Omega_n$ and observe a feature vector $\boldsymbol{c}$, then we decide for class Ω_k, if

$$\Omega_k \quad = \quad \arg\max_{\Omega_i} P(\Omega_i \,|\, \boldsymbol{c}). \tag{1}$$

A statistical classification system is expected to provide the capability of learning the statistical properties of classes, for instance the density functions, from training samples. Additionally, Bayesian classifiers should also allow an efficient computation of the a posteriori probabilities $P(\Omega_i \,|\, \boldsymbol{c})$ for each class Ω_i.

3. HIDDEN MARKOV MODELS

In the following section we will briefly introduce the basic concepts of HMMs and refer the interested reader to the literature for more details [11].

3.1. Definitions

Hidden Markov Models are widely used in the field of speech recognition. They are stochastic automata including states, transitions among states, and emission probabilities for elements of a given alphabet. An HMM with N states $\{S_1, \ldots, S_N\}$ can thus be described by a triplet $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, where $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the vector of probabilities for the generation of a sequence of output elements to start at a special state. The state transition matrix $\boldsymbol{A} = (a_{i,j})_{0<i\leq N, 0<j\leq N}$ includes the probabilities $a_{i,j}$ to change from state S_i to state S_j. The third element is either a matrix $\boldsymbol{B} = (b_i(v_l))_{0<i\leq N, 0<l<L}$ including discrete probabilities for a finite output alphabet $\{v_1, v_2, \ldots, v_L\}$ or a vector of density functions for an infinite continuous output alphabet.

Each HMM can generate sequences of output symbols. The name Hidden Markov Model is due to the fact that for an observed sequence of output symbols the underlying state sequence is unknown. Figure 1 shows two examples of HMMs of different topologies.

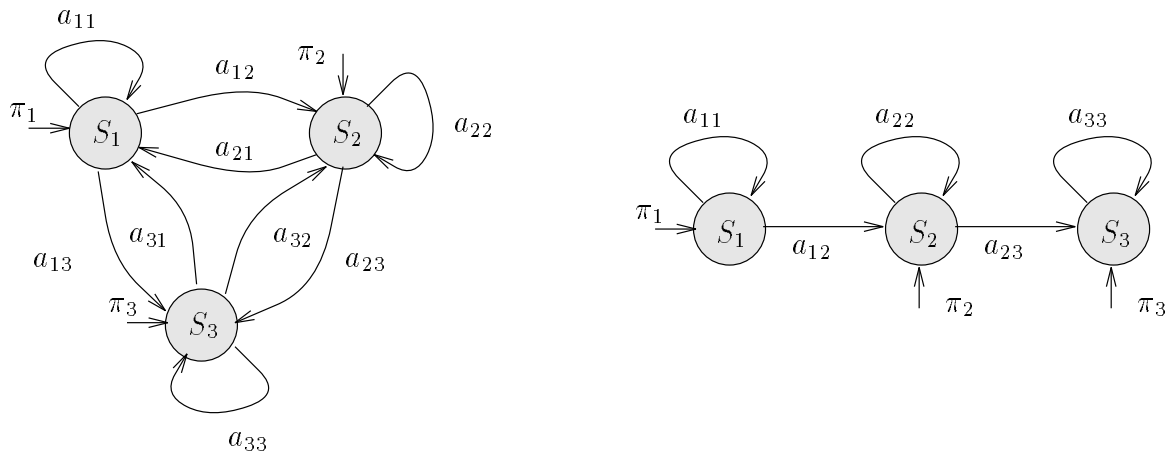

Figure 1. Ergodic and "left right" Hidden Markov Model; each state emits probabilistic output symbols.

3.2. Algorithms

During the training phase, the parameters of an HMM $\boldsymbol{\lambda}$ have to be estimated such that for all observed learning sequences $\boldsymbol{O}_i$ $(1 \leq i \leq L)$ the probability $P(\boldsymbol{O}_i \,|\, \boldsymbol{\lambda})$ that model $\boldsymbol{\lambda}$ generates $\boldsymbol{O}_i$ is maximized. In the recognition stage the decision rule, i.e. the a posteriori probability $P(\boldsymbol{\lambda}_j \,|\, \boldsymbol{O})$ for an observed feature sequence $\boldsymbol{O}$ has to be computed for each HMM $\boldsymbol{\lambda}_j$, in order to find out which HMM most likely created the feature sequence.

The parameter estimation algorithm is inasmuch unsupervised, since due to the nature of HMMs it is not known which state sequence has generated the sequence of output symbols.

The computation of the parameters for the HMM $\boldsymbol{\lambda}$ is done iteratively using the *Expectation Maximization* Algorithm (EM–Algorithm, [2]). For that purpose, the *Kullback–Leibler quantity*

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) \;=\; \sum_{i=1}^{L} \sum_{\boldsymbol{s}} P(\boldsymbol{s} \,|\, \boldsymbol{O}_i, \boldsymbol{\lambda}) \log P(\boldsymbol{s}, \boldsymbol{O}_i \,|\, \hat{\boldsymbol{\lambda}}) \tag{2}$$

is computed for an initial estimation of $\boldsymbol{\lambda}$. Herein $\boldsymbol{s} = s_1 s_2 \ldots s_T$ varies over all possible state sequences which may have produced the output symbols of the i–th observation $\boldsymbol{O}_i = o_1 o_2 \ldots o_T$ (T may be different for every i). $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ is maximized with respect to the parameter set $\hat{\boldsymbol{\lambda}}$. After the maximization step the reestimated model parameters $\boldsymbol{\lambda} := \hat{\boldsymbol{\lambda}}$ are substituted. Both steps have to be repeated until no change in parameters occurs, i. e. $\boldsymbol{\lambda} = \hat{\boldsymbol{\lambda}}$.

$$P(\boldsymbol{s}, \boldsymbol{O} \,|\, \boldsymbol{\lambda}) \;=\; \pi_{s_1} \prod_{t=1}^{T-1} a_{s_t, s_{t+1}} \prod_{t=1}^{T} b_{s_t}(o_t), \tag{3}$$

Since for an arbitrary observation $\boldsymbol{O}$ equation (3) holds, the learning formulas can be computed using numerical or combinatorial optimization techniques. For example, computation of the zero crossings of the first derivatives with respect to the unknown parameters will yield the well known estimation formulas for HMMs with discrete, time independent probabilities [11].

The decision rule for recognition depends on the computation of

$$P(\boldsymbol{\lambda} \,|\, \boldsymbol{O}) = \frac{P(\boldsymbol{\lambda}) P(\boldsymbol{O} \,|\, \boldsymbol{\lambda})}{P(\boldsymbol{O})}, \tag{4}$$

where the complexity of determining $P(\boldsymbol{O} \,|\, \boldsymbol{\lambda})$ is bounded by $O(N^2 T)$ when the *forward–backward* algorithm [8] is used. The optimal state sequence for an observation $\boldsymbol{O}$ can be computed using the *Viterbi* algorithm [11].

4. OBJECT ORIENTED IMPLEMENTATION OF HMMS

With respect to the definition of HMMs given in the previous section, different variants can be distinguished. Types of HMMs are distinguished by the special form of occurring statistical measures $\boldsymbol{\pi}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ and the topology of the stochastic automata. For instance, the transition probabilities $a_{i,j}$ can be time dependent; those HMMs are called *non–stationary*. The output alphabet of an HMM can be discrete or continuous; thus, the measure $\boldsymbol{B}$ represents either discrete probabilities or continuous density functions. For example, the statistical behavior of a state can be modelled by a Gaussian density or a mixture of Gaussian densities. Restrictions on possible transitions induce different topologies. *Left right HMMs* are used in speech recognition algorithms and satisfy the constraint that the state index increases with increasing time.

Object–oriented programming currently seems to be the most promising tool for software management. The similarities in the deduced estimation formulas and classification algorithms suggest the use of polymorphism and inheritance, and for realization of HMM algorithms in class hierarchies.

The algorithm for computing the a posteriori probabilities for an observed feature sequence and the optimal state sequence can be described in terms of the variables $\boldsymbol{\pi}$, $\boldsymbol{A}$, and $\boldsymbol{B}$, independent of the topology or the special form of the statistical measures. Thus, the forward–backward algorithm and the Viterbi algorithm should be implemented in a higher level of the inheritance tree. Dependent on the properties of the output density or the statistical behavior of the transitions the learning formulas have to be computed. These training algorithms have to be implemented in derived, more specialized classes.

A hierarchy of HMM classes has been implemented and tested. An abstract base class HMM provides the interfaces for an abstract specification of training and classification algorithms, like the Viterbi algorithm. Two class subtrees are derived from this class; one is used for the implementation of discrete HMMs which implement $\boldsymbol{B}$ as a matrix; the other subtree describes continuous HMMs and is further subdivided into classes for various densities in $\boldsymbol{B}$. Left right HMMs are special cases of any of those classes. The whole class hierarchy is integrated in an object–oriented environment for image segmentation and analysis (ANIMALS, [10]).

5. AFFINE INVARIANT FEATURES

Applications of HMMs are restricted to those pattern recognition problems, where ordered sequences of features can be constructed. The extracted features of a speech signal are, for instance, a priori time ordered and thus satisfy this central prerequisite. Since objects in an observed scene have several degrees of freedom such as translation or rotation, a sequence of features associated with each object needs to be invariant with respect to this kind of transformations, if the goal is to identify them in varying scenes using HMMs.

In [4] affine invariant features are introduced which are based on simple contours of objects. A contour is called simple, if there are no intersections of the contour with itself. A *local form* of a simple contour is defined as a closed polygon $\boldsymbol{a}, \boldsymbol{p}_k, \ldots, \boldsymbol{p}_l, \boldsymbol{b}, \boldsymbol{a}$, which is part of the polygon approximation of the complete contour determined by the point sequence $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}$. Figure 2 shows a part of a simple contour and a local form and also indicates the possible locations for $\boldsymbol{a}$ and $\boldsymbol{b}$.

Obviously, proportions of areas are affine invariant. For example, let $\boldsymbol{c}$ be the center of gravity of the local form. The quotient κ_1 of the area F_c of the triangle $\boldsymbol{abc}$ and the area F of the local form is obviously an affine invariant feature.

Now the question arises, how the points $\boldsymbol{a}$ and $\boldsymbol{b}$ have to be chosen for a given local form and how many local forms have to be computed for each contour. For each quadruple $(\boldsymbol{p}_{k-1}, \boldsymbol{p}_k, \boldsymbol{p}_l, \boldsymbol{p}_{l+1})$ the points $\boldsymbol{a}$ and $\boldsymbol{b}$ are chosen such that the quotient κ_1 will be minimized. This process produces a large set of local forms. The selection of local forms out of this set is guided by the following criterion: all local forms whose area is greater or equal to half of the area of the complete contour's area are canceled. Furthermore, triangular local forms are not admissible.

The resulting set of local forms provides a set of affine invariant features, which is

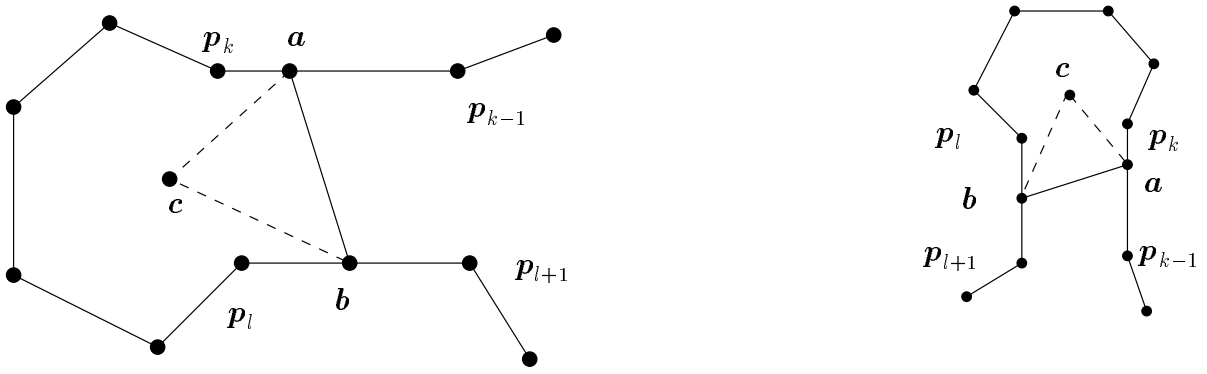

Figure 2. Examples of a local form in different scaling, rotation, and position

naturally ordered by the processing order of the polygon. Thus, we can associate with each contour a sequence of features and HMMs can be used for training and classification purposes.

6. EXPERIMENTAL RESULTS

In the experiments we choose four different objects (see Figure 3). Using a training data set of 50 input images per object, different types of HMMs are trained from the sequences of extracted affine invariant features. We compute both κ_1 for each contour. The classification results using 10 images of each object which are not included in the sample set are shown in Table 1. The used type of HMMs is the continuous version producing normally distributed output in each state. Each column shows the number of correctly classified objects. The last line summarizes the recognition rate related to the number of states. The increase of correct decisions using "left right" models is a remarkable result which is due to the fact that ergodic models have more transitions and thus more parameters which have to be estimated. The higher the number of parameters is, the larger the sample set has to be for good estimates. Conspicuously, in all cases in which the classification result is wrong, the correct object has the second highest a posteriori probability.

Since the features associated with a contour of an object are real numbers, discrete HMMs cannot be used in a direct manner. Discrete feature values can be computed using vector quantization techniques [8].

The same experiment was carried out with the Expectation Maximization Algorithm applied to Gaussian mixture densities and an unordered set of features. The overall recognition rate was approximately 93%. This means that the introduction of an ordering for the features has *decreased* the recognition rate rather than increasing it.

Non–stationary HMMs as introduced in [6] expect feature sequences of equal length for each observed scene. The sequences of features in our experiments have different size –

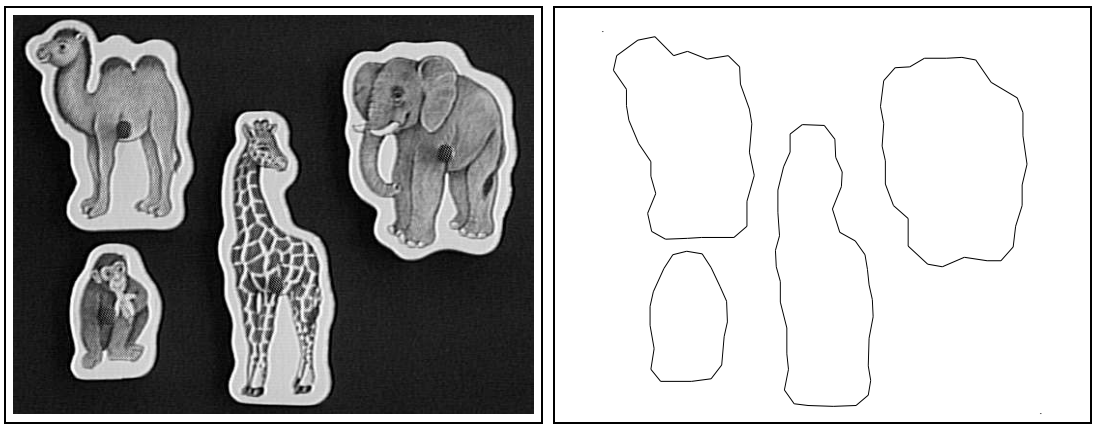

Figure 3. The image shows the original grey–level image (children toys) and the right image the resulting closed polygons of the contour of each object.

object	number of states					object	number of states				
	3	4	5	6	7		3	4	5	6	7
monkey	9	8	9	8	8	monkey	9	8	9	8	8
giraffe	7	7	7	8	8	giraffe	7	7	7	7	7
elephant	8	8	8	5	4	elephant	9	9	9	9	9
camel	5	5	5	3	5	camel	5	5	5	5	6
rate in %	72	72	72	60	62	rate in %	75	72	75	72	75

Table 1
Continuous, ergodic (left) and "left right" (right) HMM with Gaussian output densities

from 1 up to 14 – and thus the mentioned type of HMM was not tested.

All these experiments can easily be carried out using other types of invariant features. Applications to three–dimensional object recognition problems might be for instance the use of geometric 3–D invariants like mean and Gaussian curvatures of surfaces in range images, which are viewpoint independent features.

7. SUMMARY AND CONCLUSIONS

This contribution shows that statistical methods are suitable for object recognition purposes. The experimental evaluation is based on two–dimensional object recognition problems without the computation of the object's location. Real images were used. However, the constraint that only invariant features can be used is profound, since apart from the classification of an object, the computation of its location is further a central problem of computer vision. Actually, this cannot be solved using the proposed approach with HMMs, not even in the discussed two–dimensional object recognition problem. One con-

ceivable extension might be the introduction of parameterized output densities regarding the object's location. Nevertheless, this would cause maximization problems for parameter estimation which do not provide an analytical solution. The computation of the a posteriori probabilities will also be of higher complexity, because the search space is enlarged by the location parameters. Additionally, the introduced method is currently limited to images which include only one object with a homogeneous background.

In order to use HMMs, the inclusion of structural information about the objects as an *ordered* sequence of features was required. However, only few invariant geometric features of the described type can be found in the objects, e.g. only three for the monkey object. This is not sufficient for a stable parameter estimation of the HMMs (e.g. 70 parameters for a 7 state HMM) and explains why the HMM experiments reveal lower recognition rates than expected. Further research for features is required; ideally they should be chosen in such a way that the length of the feature sequence of a given object is fixed. In this case, non–stationary HMMs can be used which are shown to have higher recognition accuracy [6].

For the recognition of three–dimensional objects from 2–D views HMMs are not suitable, because occlusion and the missing depth information lead to the result that there do not exist any geometric invariant features for 3–D objects in 2–D images. Consequently, the use of a view based approach is necessary, i. e. for each possible view of an object an HMM has to be introduced. The relation between the possible number of views of an object and the resulting recognition errors are discussed in [1].

We summarize that HMMs can be naturally implemented in an object–oriented programming environment, and provide high flexibility and programming comfort. Training algorithms can thereby be programmed in an abstract manner for several types of HMMs. Furthermore, we conclude that HMMs in their existing form cannot be used for solving the 3–D object recognition problem from two–dimensional images apart from the view based attempt. Thus, it seems to be indispensable to find a more appropriate statistical framework for building a Bayesian classifier for 3–D object recognition purposes. A first promising approach can be found in [3,7].

ACKNOWLEDGEMENT

Special thanks to E.G. Schukat–Talamazzini who carried out the EM–experiments for mixture density functions.

REFERENCES

1. T. M. Breuel. *Geometric Aspects of Visual Object Recognition.* PhD thesis, MIT, Department of Brain and Cognitive Sciences, Massachusetts, 1992.
2. 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.
3. 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, 1994.

4. Stephan Frydrychowicz. Ein neues Verfahren zur Kontursegmentierung als Grundlage für einen maßstabs- und bewegungsinvarianten Strukturvergleich bei offenen, gekrümmten Kurven. In H. Burkhardt, K.H. Hoehne, and B. Neumann, editors, *Mustererkennung 1989*, Informatik Fachberichte Nr. 219, pages 240–247, Berlin Heidelberg, 1989. Springer.
5. K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Boston, 1990.
6. Y. He and A. Kundu. 2-D Shape Classification Using Hidden Markov Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, 1991.
7. J. Hornegger and H. Niemann. A Bayesian approach to learn and classify 3–D objects from intensity images. pages 557–559.
8. X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition.* Number 7 in Information Technology Series. Edinburgh University Press, Edinburgh, 1990.
9. H. Niemann, H. Brünig, R. Salzbrunn, and S. Schröder. Interpretation of industrial scenes by semantic networks. In *Proc. IAPR Int. Workshop on Machine Vision Applications*, pages 39–42, Tokyo, 1990.
10. D. W. R. Paulus. *Objektorientierte und wissensbasierte Bildverarbeitung.* Vieweg, Braunschweig, 1992.
11. L.R. Rabiner. Mathematical Foundations of Hidden Markov Models. In H. Niemann, M. Lang, and G. Sagerer, editors, *Recent Advances in Speech Understanding and Dialog Systems*, volume 46 of *NATO ASI Series F*, pages 183–205. Springer, Heidelberg, 1988.