2D shape recognition by Hidden Markov Models

Manuele Bicego and Vittorio Murino
Dip. di Informatica, Univ. di Verona
Strada Le Grazie 15, 37134 Verona, Italy.
{bicego|murino}@sci.univr.it

Abstract

In Computer Vision, two-dimensional shape classification is a complex and well studied topic, often basic for three-dimensional object recognition. Object contours are a widely chosen feature for representing objects, useful in many respects for classification problems. In this paper, we address the use of Hidden Markov Models (HMMs) for shape analysis, based on chain code representation of object contours. HMMs represent a widespread approach to the modeling of sequences, and are largely used for many applications, but unfortunately it is poorly considered in literature concerning shape analysis, and, in any case, without reference on noise or occlusion sensitivity. In this paper HMM approach to shape modeling is tested, probing good invariance of this method in term of noise, occlusions, and object scaling.

1 Introduction

Object recognition, shape modeling, and shape classification constitute active research areas in computer vision. Moreover, these issues are receiving a growing attention due to the advent of visual databases and the related necessity to retrieve information not only by using textual queries, but also on the basis of the image content.

Three-dimensional (3-D) object recognition has been faced by a large number of different approaches [1]. Among these, many techniques are based on the analysis of two-dimensional (2-D) aspects (images) of the objects, and a large literature can be found on 2-D shape classification or *planar* object recognition. A basic issue to be solved first consists in the type of representation of the object, i.e., the features to be used to describe it. Object contours are widely chosen features, as they are easily estimated from an image and well represent the semantic information also from a perceptual point of view. Different types of approaches have been proposed in the previous years, like, Fourier descriptors, chain code, curvature-based techniques, invariants, auto-regressive coefficients, Hough-based transforms, associative memories, and others, each one featured by different characteristics like robustness to noise and occlusions, invariance to translation, rotation and scale, computational requirements, and accuracy [1].

In this context, this paper tries to investigate the capabilities of the Hidden Markov Models (HMMs) for shape classification, where shapes are represented by contours expressed using Chain Code [2]. Hidden Markov Models represent a widespread approach to the modeling of sequences as they attempt to capture the underlying structure of a set of symbol strings. HMMs can be viewed as stochastic generalizations of finite-state automata, when both transitions between states and generation of output symbols are governed by probability distributions [3].

The basic theory of HMMs was developed by Baum *et al.* [4, 5] in the late 1960s, but only in the last decade it has been extensively applied in a large number of problems. A non-exhaustive list of such problems consists of speech recognition [3, 6], handwritten character recognition [7], DNA and protein modelling [8], gesture recognition [9] and, more recently, behavior analysis and synthesis [10].

The use of HMM for shape analysis has not been widely addressed. Only a few work have been found to have some similarities with our approach. In the first, He and Kundu [17] utilize HMMs to model shape contours represented by auto-regressive (AR) coefficients. Results are quite interesting and presented in function of the number of HMM states ranging from 2 to 6. Another method [11] proposes the use of circular HMM for shape classification. This particular HMM topology allows to achieve good classification accuracy with respect to scaling and deformations, also presents useful characteristics for the model training and testing. However, in both work, no examples using noise are reported and, although sensitivity to occlusions is analyzed, shapes are constrained to be a closed contour also in these cases. Another work, presented in [12], addresses shape recognition comparing HMMs and a syntactic modeling technique based on stochastic finite-state grammars. No particular original solutions for the HMM design are proposed, the goal of this work was to show the superiority of HMM with respect to the other method.

Although HMMs are largely used for some interesting characteristics like, e.g., the possibility to be trained by a formal algorithm which converge safely, and their implicit generalization capability, there are some drawbacks that are not raised or are disregarded by the literature. For instance, it is not clear how to design the model topology for a given problem. Moreover, the correct training is not always assured because the learning algorithm converges on local minima, so that initial conditions may heavily affect model performances. Other problems arise depending on the tackled application.

In this paper, we will investigate the capability of HMMs in discriminating object classes, showing its performances with respect to noise, scale, occlusions, and, preliminarly, rotation. It is worthwhile noting that our approach does not relies on any specific HMM topology or particular training algorithm, nor object shapes are constrained to be closed, represented using a specific number of symbols, or always start from a fixed point. In our case, HMM are trained using the classic Baum-Welch method without any assumptions on the model topology. Actually when objects are occluded, the resulting boundaries are not necessarily closed, and, in this sense, our algorithm classifies any (closed or open) symbol string. All these features, together with the promising performances achieved, make the proposed method an interesting alternative to the typical shape classification algorithms.

The rest of the paper is organized as follows. In Sect. 2, a formal description of the HMM, and the related training phase are reported. The description of the boundary extraction and representation phases, and experimental results on a small set of objects are presented in Sect. 3. Finally, Section 4 contains conclusions and future perspectives.

2 Hidden Markov Models

An HMM is formally defined by the following elements [3]:

- *A set* $S = \{S_1, S_2, \cdots, S_N\}$ *of (hidden) states.*
- *A state transition probability distribution*, also called transition matrix $A = \{a_{ij}\}$, representing the probability to go from state S_i to state S_j.

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \qquad 1 \leq i,j \leq N \quad (1)$$

with $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1$.

- *A set* $V = \{v_1, v_2, \cdots, v_M\}$ *of observation symbols.*
- *An observation symbol probability distribution*, also called emission matrix $B = \{b_j(k)\}$, indicating the probability of emission of symbol v_k when system state is S_j. For $1 \leq j \leq N, 1 \leq k \leq M$,

$$b_j(k) = P[v_k \text{ at time t } | q_t = S_j] \qquad (2)$$

with $b_i(k) \geq 0$ and $\sum_{j=1}^{M} b_j(k) = 1$.

- *An initial state probability distribution* $\pi = \{\pi_i\}$, representing probabilities of initial states.

$$\pi_i = P[q_1 = S_i] \qquad 1 \leq i \leq N \qquad (3)$$

with $\pi_i \geq 0$ and $\sum_{i=1}^{N} \pi_i = 1$.

For convenience, we denote an HMM as a triplet $\lambda = (A, B, \pi)$, which determines uniquely the model.

An HMM can be classified into one of the following types, in the light of its state transition matrix: *ergodic HMM*, when HMM has full state transition matrix, or *left-right HMM*, when HMM has only partial state transition matrix such that $a_{ij} = 0, \ \forall j < i$; this second type is usually used in modeling sequential signals.

There are three main problems involved with HMM use:

1. Given the HMM $\lambda = (A, B, \pi)$, we want to compute $P(O|\lambda)$, i.e. the probability that an observation sequence $O = O_1, O_2, \cdots, O_T$ (with $O_t \in V$) is generated by the model λ. This usually is solved using the so called *forward-backward procedure* [5]. This method makes use of two inductively computed variables $\alpha_t(i)$ and $\beta_t(i)$, called respectively *forward* and *backward* variables, defined as:
    - $\alpha_t(i) = P(O_1, O_2, \cdots, O_t, q_t = S_i|\lambda)$, that is the probability to have observed partial sequence $O_1, \cdots, O_t$ at time $1, 2, \cdots, t$ and being in state S_i at time t;
    - $\beta_t(i) = P(O_{t+1}, O_{t+2} \cdots, O_T, q_t = S_i|\lambda)$, that is the probability to be in state S_i at time t and to observe partial sequence $O_{t+1}, O_{t+2}, \cdots, O_T$ at time $t+1, \cdots, T$.

    $P(O|\lambda)$ is then computed as $\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$ (for each t).
2. Given the model $\lambda = (A, B, \pi)$, we want to determine the sequence $I = \{i_1, i_2, \cdots, i_T\}$ $(1 \leq i_t \leq N)$ such that $P(O, I|\lambda)$ is maximum with respect to I. In other words, we want to compute the state sequence that most probably generates the observation sequence $O_1, O_2, \cdots, O_T$. This problem is resolved by the *Viterbi Algorithm* [13, 14]. It is an inductive algorithm that at each instant t determines the optimal (i.e., the one leading to the maximum probability) state sequence to obtain $O_1, O_2, \cdots, O_T$. At each instant, it chooses from a set of N probabilities (one for each state), the probability of obtaining $O_1, O_2, \cdots, O_t$ and to be in state S_i.

3. Given a set of L observation string $\{O_t\}_\ell$, $1 \leq t \leq T$, $1 \leq \ell \leq L$, we want to determine $\lambda = (A, B, \pi)$ such that $P(\{O_t\}_\ell|\lambda)$ is maximized: this is the problem of training an HMM. The best-known method to perform this operation is the so-called *Baum-Welch re-estimation* technique [4]. It is an iterative procedure that at each step adjusts model parameters according to $P(O|\lambda)$, computed on their previous values. More precisely, it is based on *Expectation-Maximization* (EM) algorithm [15, 16], and it tries to maximize log-likelihood of the model with respect to the data.

3 Experimental results

3.1 Chain Code representation

Chain Code is a well-known method to represent contours; it specifies the direction of a contour at each edge in the edge list. Directions are coded into one of eight directions, as shown in Fig. 1. In other words, a contour is coded

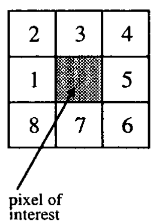


Figure 1. Code rule for assigning chain code to each edge point.

by a generic initial point and a symbol string: each symbol indicates the direction of the following point in the contour. Chain code representation presents some interesting inherent characteristics like discrete invariance to rotation (if code local differences are considered) and translation. Given an image of 2D objects, in our work data are gathered assigning at each object its chain code, calculated on object contours. Edges are extracted using *Canny edge detector* [18], while chain code is calculated as described in [1].

3.2 Results and discussion

In this section, HMM approach for shape modeling is tested, particularly in the case of noise, partial views and scaling. The test image is shown in Fig. 2. Classification was performed as follows: for each object we extract edges, calculate the related chain code and train an HMM on it. This means that HMM parameters are estimated by using

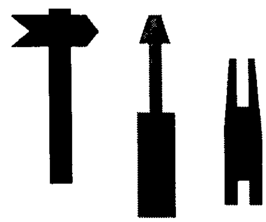
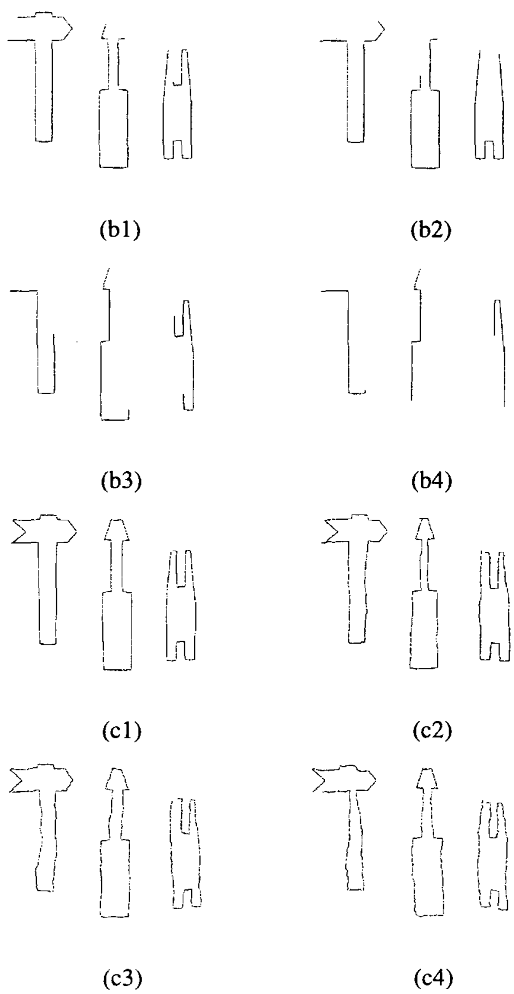


Figure 2. Original image used for experiments.

the object chain code sequence as input to the HMM. At the end of the learning session we have one HMM model, λ_i, for each object. Given a sequence O to be classified, we compute, for each model λ_i, the probability $P(O|\lambda_i)$ of generating the sequence O, using the forward-backward procedure. The sequence O is then classified as belonging to the class whose model shows the highest probability. Each HMM learning started using random initial estimates of A, B and π and ended when likelihood is converged or after 100 training cycles. The Baum-Welch training algorithm converges at local maxima of the likelihood function: the convergence of this technique for actual absolute maxima strongly depends on initial estimates of parameters. To avoid the problem of choosing adequate initial condition, it is customary to perform HMM training by utilizing several learning sessions (in our case, five sessions were utilized), and choosing the one presenting the maximum likelihood.

Classification accuracy of the system was tested by creating three test sets, each one referring to one particular aspect (occlusion, noise and scale). The first set is obtained considering, for each object, fragments of their chain code of variable length, expressed as percentage rate of the whole length. It varies from 40 to 95 percent (i.e. occlusion decreases from 60% to 5%), and the point where fragment starts was randomly chosen. This experiment aims at quantifying robustness of HMM to object occlusions. It is worth noting that the random choice of the initial point is important for assessing the invariance from the specific object part occluded. Examples of occluded object are shown in Fig. 3(b1)–(b4), at varying occlusion levels, that are respectively 60% (of object occlusion), 45%, 25% and 10%. For each occlusion level 150 sequences are generated, and classification accuracy is computed as percentage rate given by the number of correctly classified objects versus the total number of objects. In Table 1(a) these rates are reported, in function of the occlusion level. One can notice that the accuracy is very satisfactory, even for high occlusion factors: up to 20 percent occlusion, object are correctly classified



(b1) (b2) (b3) (b4) (c1) (c2) (c3) (c4)

Figure 3. Images used for experiments: (b1-b4) examples of occluded images, at increasing degree of occlusion; (c1-c4) examples of noisy images, at increasing level of noise.

Occlusion level (%)	Classification Accuracy (%)
5	100.00
10	100.00
15	100.00
20	100.00
25	99.33
30	98.00
35	98.67
40	97.33
45	94.67
50	90.00
55	89.33
60	85.33

(a)

Noise level (%)	Classification Accuracy (%)
1	100.00
4	98.00
7	90.00
10	90.67
13	92.67
16	84.67
19	84.67

(b)

Scale factor	Classification Accuracy (%)
2	100.00
3	100.00
4	100.00

(c)

Table 1. Classification accuracy on: (b) occluded set, varying occlusion level; (a) noised set, varying noise level; (c) scaled set, varying scale factor.

in every case. Moreover, HMMs are able to perform very good classification (over 90%) even seeing less than half of object. Clearly classification accuracy decreases when occlusion level grows, remaining, nevertheless, on very good levels.

The second set is obtained by adding synthetic noise to each chain code, using the following procedure: for each object, each code is changed with fixed probability P, i.e. if cc_i is the i-th symbol of the original code, the change $(((cc_i - 1) \pm 1) \bmod 8) + 1$ is carried out, with probability P. Probability ranges from 1% to 19%, and, for each value, 150 sequences are generated. In Fig. 3(c1)–(c4) examples of noisy images are shown, for noise level equal to 1%, 7%, 13% and 19%, respectively. Results of the tests are shown in Table 1(b): one can notice that classification accuracy remains high, even when noise level increases. Up to 13% of noise level, the accuracy remains over 90%: a good result if compared with degradation of objects shown in Fig. 3(c3).

The third set is obtained by scaling objects by factor 2, 3, and 4. Classification accuracies are shown in Table 1(c). The algorithm works perfectly on scaled objects, giving 100% accuracy.

In our work we have tried to exploit also invariance of

HMM for object rotation. It is known that using differential chain code, calculated by local differences of code, we can obtain invariance over rotations of angles that are multiple of 45°. To obtain invariance for other rotations our approach was to learn each HMM on all 45° views of each object; we then test our system on 30° views, obtaining an accuracy of 86.7%. This preliminary result shows that when choosing suitable contour representation, HMMs can accurately classify also rotated objects.

4 Conclusions

In this paper HMM is used for 2D shape classification, where shape is modeled using chain code. One HMM was trained for each object chain code, with states varying from 3 to 12. The method was tested using object partially occluded, noised or scaled, showing good performances on examples proposed.

References

[1] Jain, R., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill (1995).

[2] Freeman, H.: Computer Processing on line Drawing Images. Computer Surveys **6** (1974) 57–98.

[3] Rabiner, L.R.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of IEEE **77(2)** (1989) 257–286.

[4] Baum, L.E., Petrie, T.E, Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Math. Statistics. **41(1)** (1970) 164–171.

[5] Baum, L.E.: An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequality **3** (1970) 1–8.

[6] Ljolje, A., Levinson, S.E.: Development of an acoustic-phonetic Hidden Markov Model for continuous speech recognition. IEEE Tr. Sign. Proc. **39** (1991) 29–39.

[7] Hu, J., Brown, M.K., Turin, W.: HMM based on-line handwriting recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, **18(10)** (1996) 1039–1045.

[8] Hughey, R., Krogh, A.: Hidden Markov Model for sequence analysis: extension and analysis of the basic method. Comp. Appl. in the Biosciences **12** (1996) 95–107.

[9] Eickeler, S., Kosmala, A., Rigoll, G.: Hidden Markov Model based online gesture recognition. Proc. Int. Conf. on Pattern Recognition (ICPR) (1998) 1755–1757.

[10] Jebara, T., Pentland, A.: Action Reaction Learning: Automatic Visual Analysis and Synthesis of interactive behavior. In 1st Intl. Conf. on Computer Vision Systems (ICVS'99) (1999).

[11] Arica, N, Yarman Vural, F.T.: A shape descriptor based on circular Hidden Markov Model. In 15th Intl. Conf. on Pattern Recognition (ICPR00) (2000).

[12] Fred, A.L.N., Marques, J.S., Jorge, P.M.: Hidden Markov Model vs Syntactic Modeling in Object Recognition. In Intl. Conf. on Image Processing (ICIP97) (1997).

[13] Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans. Informat. Theory **IT-13** (1967) 260–269.

[14] Forney, G.D.: The Viterbi algorithm. Proc. of IEEE **61** (1973) 268–278.

[15] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum–likelihood from incomplete data via the EM algorithm. J. Royal Statist. Soc. Ser. B. **39** (1977).

[16] Wu, C.F.J.: On the convergence properties of the EM algorithm. The Annals of Statistics **11(1)** (1983) 95–103.

[17] He, Y., Kundu, A.: 2-D shape classification using Hidden Markov Model. IEEE Trans. Pattern Analysis Machine Intelligence. **13(11)** (1991) 1172–1184.

[18] Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Analysis Machine Intelligence **8(6)** (1986) 679–698.