

# EM Training of Hidden Markov Models for Shape Recognition Using Cyclic Strings

Vicente Palazón-González, Andrés Marzal, and Juan M. Vilar\*

Dept. Llenguatges i Sistemes Informàtics and Institute of New Imaging Technologies  
Universitat Jaume I de Castelló. Spain  
{palazon, amarzal, jvilar}@lsi.uji.es

**Abstract.** Shape descriptions and the corresponding matching techniques must be robust to noise and invariant to transformations for their use in recognition tasks. Most transformations are relatively easy to handle when contours are represented by strings. However, starting point invariance is difficult to achieve. One interesting possibility is the use of cyclic strings, which are strings with no starting and final points. Here we present the use of Hidden Markov Models for modelling cyclic strings and their training using Expectation Maximization. Experimental results show that our proposal outperforms other methods in the literature.

**Keywords:** hidden markov models, cyclic strings, shape recognition.

## 1 Introduction

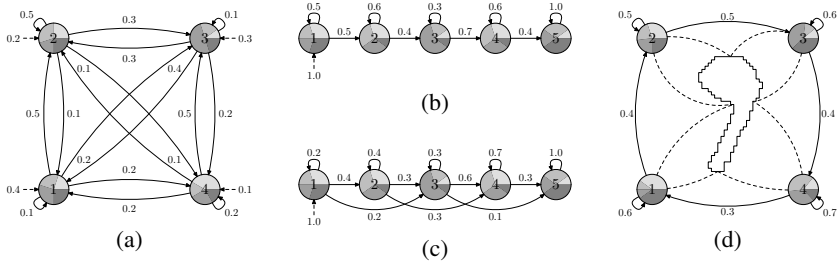
In a shape classifier, shapes can be represented by their contours or by their regions. Contour based descriptors are widely used as they preserve local information, which is important in the classification of complex shapes.

Dynamic Time Warping (DTW) is being increasingly applied for shape matching [1]. A DTW-based dissimilarity measure is a natural option for optimally aligning contours, since it is able to align parts as well as points and it is robust to deformations. Hidden Markov Models (HMMs) [2] are also used for shape modelling and classification [3–7]. HMMs have some of the properties of DTW matching and they also provide a probabilistic framework for training and classification.

Shape descriptors, combined with shape matching techniques, must be invariant to many distortions, including scale, rotation, noise, etc. Most of these distortions are relatively easy to deal with. However, invariance to the starting point is difficult to achieve no matter the representation. In the case of HMMs there are several approaches for dealing with this invariance [3, 5–7, 4], but all of them have drawbacks. The best solution to this problem is to consider every possible starting symbol of the string that represents the contour, that is, to use cyclic strings. HMMs can only generate ordinary strings and not cyclic strings. To overcome this problem, in this paper, we present the use of HMMs for modelling cyclic strings and their training using Expectation Maximization. Preliminary work on this problem appears in [8].

---

\* Work partially supported by the Spanish Government (TIN2010-18958) and the Generalitat Valenciana (*Prometeo*/2010/028).



**Fig. 1.** Examples of HMMs. An HMM state can emit any of four symbols according to the probability distribution represented by a pie chart. Probability distribution for initial states is represented by dotted arrows. (a) Ergodic topology with four states. (b) Linear left-to-right topology. (c) Bakis left-to-right topology. (d) Circular topology as proposed in [4]. Contours are segmented by associating a state to each segment. Ideally, each state is responsible for a single segment.

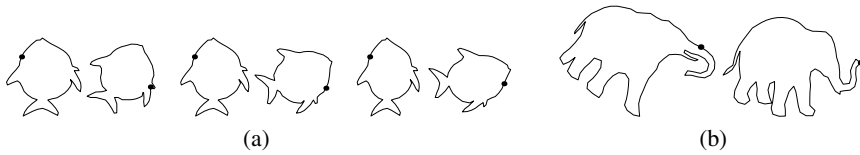
## 2 Defining the Problem of Cyclic Strings with HMMs

### 2.1 Common Approaches

One crucial aspect when applying HMMs is the definition of an adequate topology. Many works use ergodic topologies [5–7], which have some problems. The main is that it is possible to visit a state more than once without using self transitions (Fig. 1a). Ergodic models do not impose restrictions in the order of the strings of observations. When the string of observations is temporal or an order exists (as in shape contours), these topologies do not fully exploit the sequential or temporal information of the data and many states are used to explain multiple observations from different parts along the contour. This makes training and recognition a complex problem.

From the previous observations, left-to-right topologies seem more suitable. These topologies do not allow to visit states that are to the left of the current one (Figs. 1b and 1c). In left-to-right models there is an initial state and a final state. This way, the sequence of states is forced to begin in the initial state and it never revisits a state once it leaves it. When a string of symbols is segmented, all the symbols of a segment are emitted by the same state, and consecutive segments are associated to consecutive states. Although these topologies usually have more states, their number of transitions is low, and consequently the overall complexity of the algorithms is reduced.

In [4], a circular topology is proposed to model contours (Fig. 1d), which can be seen as a modification of the left-to-right topology, where the last emitting state is connected to the first. This topology eliminates the need for a starting point: the contour can be segmented by associating consecutive states to consecutive segments in the cyclic strings, but there is no assumption about which is the first or last segment (Fig. 1d); therefore, there is an analogy with left-to-right topologies. However, there is a problem that breaks this analogy: like in the case of ergodic models all the states can be reached from any state and we can finish in any of them. Therefore, the optimal path can contain non-consecutive repeated states and one state can be responsible of the emission of several non-consecutive segments of the contour. Besides, it is possible to have an optimal path that does not visit all the states at least once.



**Fig. 2.** (a) Original shapes and their canonical versions using Fourier descriptors [9]. The same shapes compressed in the horizontal axis have different rotation and starting point. (b) Canonical version of an elephant with its trunk down and with its trunk raised using the method of least second moment of inertia [3].

Another approach is the election of a reference rotation and from it a starting point [3, 9]. The basic idea is that, after normalization, all shapes have a canonical version with a “standard” rotation and starting point, and thus, they can be compared as if their representations were linear. But invariance is only achieved for different rotations and starting points of the same shape. Different shapes (even similar ones) may differ substantially in their canonical orientation and starting point. Fig. 2a shows three perceptually similar figures whose canonical versions are significantly different in terms of orientation and starting point. This problem frequently appears in shapes whose basic ellipse is almost a circle. Besides, shapes of the same class with little differences can substantially alter the selection of the starting point. Fig. 2b shows two elephants, one with its trunk down and the other with its trunk raised, this fact and other little differences modify the canonical rotation of the method of least second moment of inertia, and with it, the selection of the starting point.

## 2.2 Cyclic Strings

The most suitable solution for obtaining the invariance to the starting point is to use every possible starting point of the strings, i.e., using cyclic strings. A cyclic string can be seen as the set of strings obtained by cyclically shifting a conventional sequence. Let  $x = x_1 \dots x_m$  be a string from an alphabet  $\Sigma$ . The cyclic shift  $\rho(x)$  of a string  $x$  is defined as  $\rho(x_1 \dots x_m) = x_2 \dots x_m x_1$ . Let  $\rho^k$  denote the composition of  $k$  cyclic shifts and let  $\rho^0$  denote the identity. Two strings  $x$  and  $x'$  are cyclically equivalent if  $x = \rho^k(x')$ , for some  $k$ . The equivalence class of  $x$  is  $[x] = \{\rho^k(x) : 0 \leq k < m\}$  and it is called a *cyclic string*.

To achieve starting point invariance using cyclic strings we model the generation process as follows. An HMM generated a string that later suffered an unknown cyclic shift. That is, a model,  $\lambda$ , has generated a string,  $x = x_1 x_2 \dots x_m$ , that has suffered the transformation,  $\rho^{k'}(x)$ , for an unknown  $k'$ . We treat  $x$  as a cyclic string,  $[x]$ , and we assume that all the cyclic shifts are equiprobable. Thus, the probability of  $[x]$  given a model  $\lambda$  is

$$P([x]|\lambda) = \sum_{k=0}^{m-1} P(x|\lambda, k)P(k|\lambda) = \frac{1}{m} \sum_{k=0}^{m-1} P(\rho^k(x)|\lambda), \quad (1)$$

that is, we must compute the probability for every possible cyclic shift and add them.

Similarly, finding the best cyclic shift and sequence of states amounts to compute

$$\hat{P}([x]|\lambda) = \frac{1}{m} \max_{0 \leq k \leq m-1} \hat{P}(\rho^k(x)|\lambda) \propto \max_{0 \leq k \leq m-1} \hat{P}(\rho^k(x)|\lambda). \quad (2)$$

where  $\hat{P}$  is the Viterbi score for a string.

Initially, we adopt  $\hat{P}$  as an estimation of the real probability because it is a very good approximation<sup>1</sup>.

### 3 Cyclic Training

To train HMMs cyclic strings we have to estimate the Markov model parameters that maximize the probability of the observed cyclic strings. That is, our objective is to maximize:

$$P(X|\lambda) = \prod_{l=1}^L P([x]^{(l)}|\lambda) = \prod_{l=1}^L \frac{1}{m^{(l)}} \sum_{k=0}^{m^{(l)}-1} P(\rho^k(x^{(l)}|\lambda), \quad (3)$$

where  $X$  is a set of cyclic strings,  $X = \{[x]^{(1)}, [x]^{(2)}, \dots, [x]^{(L)}\}$ .

We use an iterative procedure. First, we set some initial values for  $\lambda$ . Then, we obtain new values of these parameters in each iteration, using increasing transformations, applying the Baum-Eagon inequality [10, 11]. It is guaranteed that the new estimated values increase the value of the objective function and, therefore, its convergence.

Let  $\Sigma = \{v_1, v_2, \dots, v_w\}$ , be an alphabet (the set of observable events is discrete and finite). Let  $A = \{a_{ij}\}$  be the matrix of transition probabilities between states ( $1 \leq i, j \leq n$ , where  $n$  is the number of states) and let  $B = \{b_{ij}\}$  be the matrix of observation probabilities ( $1 \leq i \leq n$  and  $1 \leq j \leq w$ ) [2]. As we know that  $\sum_{j=0}^n a_{ij} = 1$  for  $0 \leq i \leq n$  and that (3) is a polynomial with respect to  $A$ , the new estimation,  $\bar{a}_{ij}$ , can be obtained with the Baum-Eagon inequality [10, 11]. Applying logarithms and [10] to (3), we conclude that:

$$\bar{a}_{ij} = \frac{\sum_{l=1}^L \sum_{k=0}^{m^{(l)}-1} \frac{1}{\sum_{k=0}^{m^{(l)}-1} P(\rho^k(x^{(l)}|\lambda)} \sum_{t=0}^{m^{(l)}-1} \alpha_t^{l_k}(i) a_{ij} b_j(\rho^k(x_{t+1}^{(l)})) \beta_{t+1}^{l_k}(j)}{\sum_{l=1}^L \sum_{k=0}^{m^{(l)}-1} \frac{1}{\sum_{k=0}^{m^{(l)}-1} P(\rho^k(x^{(l)}|\lambda)} \sum_{t=0}^{m^{(l)}-1} \alpha_t^{l_k}(i) \beta_{t+1}^{l_k}(j)}, \quad (4)$$

where  $\alpha_t^{l_k}(i)$  and  $\beta_t^{l_k}(j)$  are the forward and backward probabilities for the shifted string,  $\rho^k(x^{(l)})$ , and are defined similarly to those in [2].

Analogously, with  $b_i(v_j)$  (the probability of observing the symbol  $v_j$  being in state  $i$ ) and knowing that  $\sum_{j=0}^w b_i(v_j) = 1$  for  $1 \leq i \leq n$  and that (3) is a polynomial with respect to  $B$  (the matrix of observation probabilities [2]), we arrive at

$$\bar{b}_i(v_j) = \frac{\sum_{l=1}^L \sum_{k=0}^{m^{(l)}-1} \frac{1}{\sum_{k=0}^{m^{(l)}-1} P(\rho^k(x^{(l)}|\lambda)} \sum_{t=1}^{m^{(l)}-1} \alpha_t^{l_k}(i) \beta_t^{l_k}(i)}{\sum_{l=1}^L \sum_{k=0}^{m^{(l)}-1} \frac{1}{\sum_{k=0}^{m^{(l)}-1} P(\rho^k(x^{(l)}|\lambda)} \sum_{t=1}^{m^{(l)}-1} \alpha_t^{l_k}(i) \beta_t^{l_k}(i)}. \quad (5)$$

<sup>1</sup> And it offers certain implementation advantages and reduction of computational time.

A similar reasoning can be used for continuous models.

This cyclic Baum-Welch (CBW) algorithm is an application of Expectation Maximization, and then, it needs a good initialization for  $\lambda$ . In the next section we propose a heuristic to solve this.

## 4 A Heuristic for Selecting the Starting Point

We will see in the experiments that the labelling of the training samples gives us a heuristic for obtaining a starting point that improves those in the literature (Section 2).

We need to perform a preprocessing. For it we use cyclic DTW (CDTW) [1] that, apart from returning the cost (distance) of the cyclic alignment, can also return the corresponding cyclic shift of one of the strings for the alignment with the other string. Starting from a set of training samples, our aim is to choose an appropriate starting point for them. We select a representative (the centroid of the class using CDTW) and an arbitrary starting point for it. With the representative of each class and its starting point, we compute the CDTW for each one of the other members of the class and the representative, obtaining the cyclic shift of the alignment that defines a good starting point for each of them. Once we have an appropriate starting point for the training samples, we can train the model of each class as if the cyclic strings were ordinary strings.

In a similar way, to classify a new sample, we begin by finding adequate starting points for it (one for each class). These starting points are computed by CDTW with the representative of each class. Thus, with this starting point for each class we can compute probabilities (or Viterbi scores) in a conventional way.

Although, as we will see in the next section, this solution has worse results than the CBW algorithm, both training and recognition are much faster. Moreover this training can be used as a good initialization for the CBW algorithm.

## 5 Experiments

In order to assess the behaviour of the presented methods, we performed comparative experiments on a shape recognition task on publicly available databases: MPEG7 CE-Shape-1 corpus part B (MPEG7B) (1440 samples in 70 classes) [12], Silhouette corpus (1070 samples in 41 classes) [13], He-Kundu corpus (8 samples) [3] and Subset 1 corpus (7 classes of the MPEG7B corpus, 20 samples per class) [7].

The outer contours of the images were extracted as sequences of points. A random starting point in each sequence was also selected and 128 landmark points were sampled uniformly. As it is customary in the literature of HMMs and shape recognition we used the curvature shape descriptor. The evaluation was done with classification rates for different number of states (we train an HMM for each class): 10 to 120 in steps of 10. We only use a gaussian per state. The experiments of Section 5.1 were performed using cross validation. The ones of Section 5.2 were performed with a leaving-one-out approach for comparing with other results in the bibliography. In all of them, for classification, we use the Viterbi scores.

## 5.1 Invariance to the Starting Point, Left-to-right Topologies and Cyclic Approach

In Section 2 several solutions to the starting point invariance problem are commented. We compare our heuristic (Section 4) with the circular topology [4], the election of the starting point using Fourier descriptors [3], and the ergodic topology [5–7]. In Fig. 3a the results of the comparison are shown, for MPEG7B and Silhouette corpora. The election of the starting point and the circular topology (especially the latter) happen to be the most competitive with respect to our heuristic while the ergodic topology obtains the worst results<sup>2</sup>.

In Section 2 we mentioned that left-to-right topologies are the most suitable for modelling strings. However, within these topologies, the linear topology seems to be the best for this purpose, because having more transitions increases the complexity of the model. Here we empirically prove this affirmation with a comparison between three left-to-right topologies: linear, Bakis and the one with four transitions per state. The last one is similar to Bakis but with another transition to the state three places from the current. The method used for training and classifying is our heuristic. The results are shown in Fig. 3b. As we can see the linear topology outperforms the others.

We compare our cyclic approach, the CBW algorithm (Section 3) with our heuristic and the circular topology [4]. Cyclic training is initialized using the heuristic in a linear left-to-right topology. Comparative results are in Fig. 3c. We can observe that cases where CBW algorithm wins predominate. The best results that we obtain using CBW are 93.93% and 93.84% for the MPEG7B and Silhouette corpora.

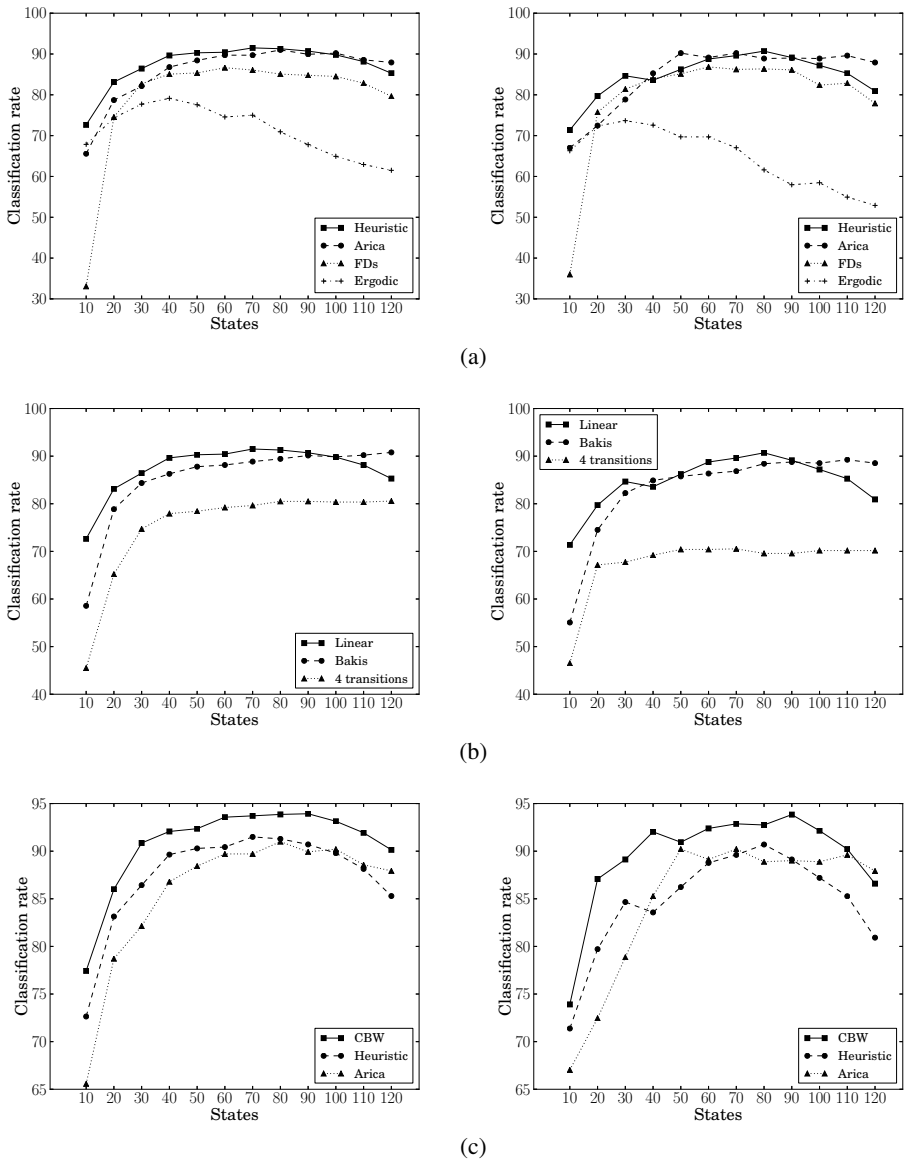
## 5.2 More about the Ergodic Topology

In Section 5.1 we experimentally saw that the ergodic topology does not offer good results. However, in the literature there are works [3, 5–7] where this topology is used.

More specifically, in [5] experiments are performed with this topology. For training, the authors choose a number of states with BIC (*Bayesian Inference Criterion*) over a clustering of curvatures. The obtained results are good enough but their corpora have few samples and classes. They use a subset of the MPEG7B corpus of 6 classes with 10 samples per class (a subset of subset 1). They also use He-Kundu corpus for performing an experiment of invariance to the starting point achieving a classification rate of 100%. This way, they conclude that HMMs with an ergodic topology are enough for obtaining this invariance. In our opinion, this experiment is not enough for claiming that affirmation. For this corpus we also achieve a 100% with the CBW algorithm. In [6], a work of the same authors, another subset of MPEG7B is used (a subset of subset 1, with 12 samples per class). We call this corpus subset 2, as it is done in [7]. In this case, they use a canonical method for the election of the starting point. Instead of using BIC for obtaining the number of states, they use a fixed number of states. In [7], the authors, parting from the work of [5, 6], try to improve their results with a training based on GPD (*Generalized probabilistic descent method*). They also use the subset 2 and create a new one, subset 1. With subset 2 they obtain a classification rate of 97.63% ([6] obtains a

---

<sup>2</sup> We will talk more about this topology in Section 5.2.



**Fig. 3.** Classification rates (with corpora MPEG7B (left) and Silhouette (right)) for the comparison between: (a) The circular topology (Arica), the election of the starting point with Fourier descriptors (FDs), the ergodic topology (Ergodic) and our heuristic (Heuristic); (b) Different left-to-right topologies. Linear topology, Bakis topology and topology of four transitions. Our heuristic is used for training and classifying; and (c) cyclic Baum-Welch (CBW), our heuristic (Heuristic) and circular topology (Arica).

98.8%). With subset 1 they obtain a 96.43%. With subset 1 and the CBW algorithm, we achieve a 99.28%, that even outperforms the classification rate of [6] with subset 2. None of the previous works show results with the entire MPEG7B corpus.

## 6 Discussion

In this work, we have argued and empirically proved that other proposals in the literature for obtaining the invariance to the starting point do not offer a suitable solution. We have formalized training and recognition for cyclic strings with Hidden Markov Models, formulating the corresponding Baum-Welch or Expectation Maximization algorithm. We have shown that this cyclic treatment is the current best solution for obtaining the starting point invariance.

## References

1. Palazón-González, V., Marzal, A.: On the dynamic time warping of cyclic sequences for shape retrieval. *Image and Vision Computing* 30(12), 978–990 (2012)
2. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2) (1989)
3. He, Y., Kundu, A.: 2-D shape classification using hidden Markov model. *IEEE Trans. Pattern Anal. Mach. Intell.* 13(11), 1172–1184 (1991)
4. Arica, N., Yarman-Vural, F.: A shape descriptor based on circular hidden Markov model. In: *ICPR*, vol. I, pp. 924–927 (2000)
5. Bicego, M., Murino, V.: Investigating hidden Markov models' capabilities in 2D shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(2), 281–286 (2004)
6. Bicego, Murino, Figueiredo: Similarity-based classification of sequences using hidden Markov models. *Pattern Recognition* 37, 2281–2291 (2004)
7. Thakoor, N., Gao, J., Jung, S.: Hidden Markov model-based weighted likelihood discriminant for 2-D shape classification. *IEEE Trans. Image Processing* 16(11), 2707–2719 (2007)
8. Palazón, V., Marzal, A., Vilar, J.M.: Cyclic linear hidden Markov models for shape classification. In: Mery, D., Rueda, L. (eds.) *PSIVT 2007*. LNCS, vol. 4872, pp. 152–165. Springer, Heidelberg (2007)
9. Bartolini, I., Ciaccia, P., Patella, M.: WARP: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(1), 142–147 (2005)
10. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model of ecology. *Bull. Amer. Math. Soc.* 73, 360–363 (1967)
11. Levinson, S.E., Rabiner, L.R., Sondhi, M.M.: An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal* 62(4), 1035–1074 (1983)
12. Latecki, L., Lakämper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: *CVPR*, June 13–15, pp. 424–429. IEEE, Los Alamitos (2000)
13. Sharvit, D., Chan, J., Tek, H., Kimia, B.B.: Symmetry-based indexing of image databases. In: *Workshop on Content-Based Access of Image and Video Libraries*, pp. 56–62 (1998)