Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 1380–1390
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Detection of a polymorphic Mesoamerican symbol using a rule-based approach

Yann Frauel[a], Octavio Quesada[b], Ernesto Bribiesca[a,*]

[a]*Department of Computer Science, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apdo. 20-726, D.F., 01000 Mexico*
[b]*Instituto de Investigaciones Filológicas, Universidad Nacional Autónoma de México, Mexico*

Received 14 December 2004; received in revised form 17 November 2005; accepted 17 November 2005

Abstract

This paper describes a technique to recognize a Mesoamerican symbol whose shape is extremely variable. We extract symbols from a drawing and we encode them as discrete curves. We perform the recognition using a set of rules that define the correct symbol. One of the rules is the presence of a single symmetry axis. We describe a comparison metric between curves in order to search for symmetries. The other rules used for the recognition concern the morphology of the symbol. The proposed technique proves to be fast and efficient. We present recognition results obtained on various pre-hispanic images. The rule-based approach proposed and implemented here, appears suitable to detect polymorphic signs, a common feature of Mesoamerican symbols. To the best of our knowledge, this is the first study of pattern recognition into the field of Mesoamerican iconography.
© 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Rule-based recognition; Polymorphic symbol; Mesoamerican symbol; Symmetry axes; Curve comparison metrics; Morphology analysis

1. Introduction

Most of the high cultures developed in Mesoamerica (Olmec, Maya, Teotihuacan, Aztec, etc.), characteristically used plastic shapes and media to represent religious thinking. Those images are then, direct sources of data on the subject and have been studied by iconography and other disciplines, in pursuing that fundamental piece of knowledge. The vastness and complexity of such a universe, however, have naturally limited research to specific sites, cultures, regions and epochs. Even so, few studies have conceived the Mesoamerican iconographic complex as a whole, focusing on the common elements found in different cultures, times and regions [1–4]. With the same approach and comparative iconographic analyzes, four main signs have been recently identified in a large sample of Mesoamerican monuments [5,6]. Based on their high incidence in different cultures, regions and epochs in Mesoamerica, their continuous presence in principal monuments, and the central positions they often occupied in images, they have been identified as symbols.

In this work, the polymorphic shape of one of those symbols was studied, defined and compared by pattern recognition means. Thus, this work deals with shape of symbols. Shape is an intrinsic property of objects. Whereas color, motion, and intensity are relatively simply quantified by a few well-understood parameters, shape is much more subtle [7]. Natural or painted shapes are incredibly complex. It is not clear what aspects of shapes are important for applications such as recognition. Several authors have used different techniques for shape analysis and recognition, such as sequential extraction of shape features [8], medial axis transforms, mirroring axes [9]. Pavlidis [10] reviewed these and other methods.

Ballard and Brown [7] proposed the template matching, which is an operation for finding out how well a template

* Corresponding author. Fax: +5255 5622 3620.
E-mail addresses: yann@leibniz.iimas.unam.mx (Y. Frauel), oquesada@ifc.unam.mx (O. Quesada), ernesto@leibniz.iimas.unam.mx (E. Bribiesca).

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.11.019

(a shape) matches a window of a given binary image. This binary image may be considered as another shape. Strachan et al. [11] presented a method to compare the shapes of two fishes based on invariant moments, optimization of the mismatch, and shape descriptors. Another measure of shape similarity was presented in Ref. [12] by means of the *shape numbers*, which are related to the resolution of the digitalization scheme. These methods find the best matching between shapes.

Recently, other methods related to shape similarity have been proposed. Gunsel and Tekalp [13] propose a new shape-similarity metric in two eigenshape space for object/image retrieval from a visual database via query-by-example. Carlin [14] presents five different new performance measures for shape similarity retrieval. Mokhtarian and Abbasi [15] propose a method for shape-similarity retrieval under affine transforms based on the maxima of curvature scale space image. Mahmoudi et al. [16] present a method for image retrieval based on shape similarity by edge orientation autocorrelogram. Torsello and Hancock [17] propose a skeletal measure of 2D shape similarity.

Due to the complexity to detect Mesoamerican symbols, which are polymorphic, we propose a rule-based approach. Several authors have used this type of approach, such as: Ding and Young [18] have implemented a rule-based system to complete shape from imperfect contour. Another example of this type of systems is presented by Ahmed and Ward [19], who developed a novel rule-based system for thinning of characters. In this paper, we define a few rules that appear to be common to the many forms of the studied symbol. These rules are general enough to characterize the symbol in spite of its variability but they are specific enough to discriminate between correct and incorrect symbols in most of the cases. In Section 2, we will present an overview of the problem we want to solve and the approach we developed. In Section 3, we will describe the preliminary stages of extracting shapes from an image and normalizing them. In Sections 4 and 5, we will explain in more details the implementation of the proposed rules, concerning the number of symmetry axes and the morphology of the shape, respectively. In Section 6, we will give some recognition results. We will present our conclusions in Section 7.

2. Overview

2.1. Presentation of the problem and proposed solution

Iconographists have constituted large databases of pictures and drawings of Mesoamerican artifacts and monuments. We would like to be able to perform an unsupervised inspection of this database and automatically locate in each image some particular symbols. The symbol we study in this paper is the most basic of four newly identified transcultural Mesoamerican signs [5,6]. We refer to it as the symbol *One*. Fig. 1 shows a few examples of occurrences of this symbol.

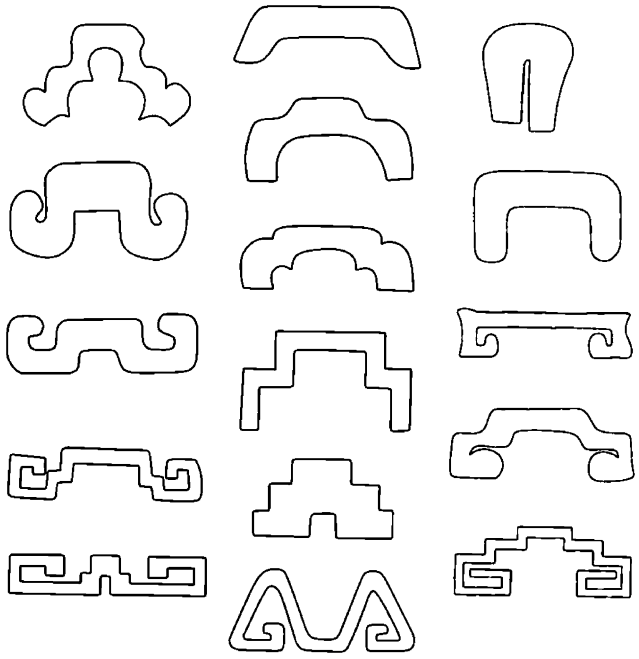

Fig. 1. Examples of occurrences of the symbol One.

It can be seen that there is a great variability of shape between the various instances of the class One. This variability makes it almost impossible to use a template matching approach. This is all the more true as we do not know in advance all the possible forms of the symbol. So we have to rely on a more conceptual representation of the shape. We therefore looked for rules that characterize prominent features of the symbol and are verified by the various instances of the class. We found out that the symbol One can be mostly defined as a symmetric arch. In order to comply with the symmetry property, we define as a first rule that a One has to possess one and only one symmetry axis. Second, we test the arch property by defining several morphological rules that basically check that the central part of the symbol is an arch with suitable proportions.

In addition to the problem of shape variability, we also have to deal with the problem of speed. Since we need to test a large number of images and every image contains many symbols, we are interested in designing a fast recognition algorithm. With this idea in mind, we chose to use a boundary representation of our symbols. This approach is faster than a region-based approach because we only need to perform operations on the boundary points rather than on all of the inside points of the symbol. In Section 3, we describe how we extract elemental symbols from a composite image and how we represent them using their boundaries.

2.2. Vocabulary convention

In the following we will use a number of vocables to refer to various aspects of the symbols. For the sake of

clarity, we define here the precise meaning that we assign to each of these vocables. A *point M* is a pixel of a digital image. A *discrete curve* is a series of N ordered points $(M_1, M_2, \ldots, M_N)$. Two consecutive points are not necessarily adjacent; if they are not, they are assumed to be virtually linked by a straight segment. Since we only deal with discrete curves in this paper, we also refer to them as *curves*. A curve that has different starting and ending points ($M_1 \neq M_N$) is an *open curve*. A curve that has the same starting and ending point ($M_1 = M_N$) is a *closed curve* or a *loop*. The object whose shape is defined by a loop is called a *symbol*. A symbol is said to be a *wire symbol* if its thickness is zero. Otherwise it is called a *thick symbol*.

3. Preparation

3.1. Shape extraction

Segmenting an image to extract symbols is by itself a difficult problem that exceeds the limits of this paper. In order to simplify the problem, we assume that our input image is a binary (black and white) line drawing. We also assume that the image is clean: contours are not erroneously interrupted and there are no extraneous connections due to noise. Such an image can be a hand drawing or the result of a filtered edge detection on a more complex image. The proposed technique works as well on thick symbols as on wire symbols. For filled up thick symbols, it is necessary to perform an edge detection first.

In an image, lines are typically several pixels thick. In order to easily follow the lines, we first reduce their thickness to a single pixel thanks to a thinning algorithm. We first apply the Zhang–Suen algorithm [20,21] that iteratively deletes boundary pixels following a set of rules until only the skeleton of the shape subsists. However, the final lines resulting from this algorithm can still be more than one pixel thick. In order to strictly reduce the thickness to one pixel, we apply our own final round of deletions: we sequentially scan every pixel of the image and we delete it if its 3×3 neighborhood corresponds to one of the four cases given in Fig. 2.

The next step consists in extracting symbols from the image. Of course there are many ways to group lines of an image in order to form various symbols. We propose to

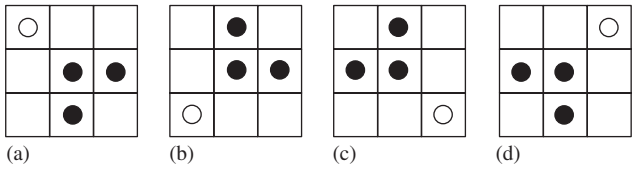

Fig. 2. Final thinning configurations. A dark circle symbolizes an object pixel, a white circle symbolizes a background pixel. Other pixels are indifferent. If one of these configurations is encountered, the central pixel is deleted.

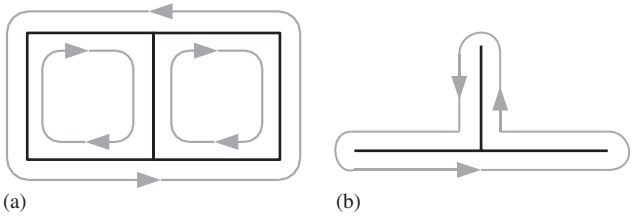

Fig. 3. Finding the minimally closed loops by following the rightmost connections. (a) Inside and outside loops are found. (b) Wire symbols are found as zero thickness loops.

extract only elemental symbols obtained by looking for the minimally closed loops contained in the image. In order to find these loops, we first need to inventory all the curves present in the image. This is done by first searching the image for all the nodes. A *node* is either the extremity of a line or the intersection of several lines. In more precise terms, it is a point with connectivity different from two [20,21]. Once we have found all the nodes, we look for all the lines starting from each node. We follow every line until reaching another node. We call *branch* a line that connect two nodes. Branches are curves in the sense of Section 2.2. Each branch is registered twice: once for each travel direction. As soon as a branch is registered, we erase it. So after taking into account all the possible starting nodes, we are left with only the lines that do not have extremities nor intersections. These lines are therefore isolated closed loops. We register each of these loops as a branch. After this process, we know all the branches contained in the image.

Now we take the first branch and we look for all the possible continuation branches (branches that start where the first branch ends). Among all the possible continuations we select the rightmost one, that is, the one that forms the smallest oriented angle with the first branch. Now from this continuation branch again we look for the rightmost continuation and so on and so forth until we reach our starting point. We delete all the branches that were used in the circuit. We repeat this process until there is no branch left. At this point, we have found all the minimally closed loops. These loops constitute the elemental shapes that we are going to test. Note that, since every branch is registered in both travel directions, we actually find inside and outside symbols (see Fig. 3(a)). Also, wire symbols are found as loops with zero thickness (Fig. 3(b)). We discard loops that are less than 15 pixels long since they are not significant.

3.2. Normalization

For easier comparison of the symbols and standardization of the rules, we first need to normalize the symbols in position, size and orientation. The normalization in position is done with respect to the centroid of the loop that defines the symbol. That is, we subtract the coordinates of the centroid from the coordinates of every point of the loop. As for the size normalization, we scale the symbol in such a way

that its longest axis has a length of about 100 pixels. We preserve the aspect ratio of the symbol while scaling. The longest axis is found by calculating the principal axes of the boundary loop, that is by finding the eigenvalues of the covariance matrix of the point coordinates:

$$C = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} x_i^2 & x_i y_i \\ x_i y_i & y_i^2 \end{bmatrix}, \quad (1)$$

where (x_i, y_i) are the coordinates of one point of the loop and N is the number of points. Note that the means of x_i and y_i are 0 because of the position normalization. If V_M is the largest eigenvalue of C, we scale the symbol by a factor $100/\sqrt{V_M}$.

The scaling operation on a discrete loop results in separating the points by a distance that depends on the original size of the symbol. In order to standardize the distance between points and also to reduce computation times, we perform a subsampling of the loop. That is, we discard some intermediate points in a way that the distance between consecutive remaining points is greater or equal to 15 pixels. This procedure also results in a lowpass filtering of the loop and in a removal of the local details that are not significant at the level of the shape description of the symbol.

The last normalization we perform is a normalization in orientation. Since we are interested in symmetric symbols, we rotate the loop in such a way that its best symmetry axis—if it has one—is vertical. The problem is now to find the symmetry axes. Ideally, if a single symmetry axis exists, it should coincide with a principal axis. However, this coincidence does not always occur in the case of poorly drawn symbols. Moreover, a symbol can possess multiple and non-orthogonal symmetry axes. These additional axes therefore do not correspond to any principal axis, yet they should also be detected. For these reasons, the principal axes cannot be used to find the symmetry axes. The solution we propose is described in the following section.

4. Number of symmetry axes

The idea we use in order to determine whether a symbol is symmetric with respect to a particular axis, is to measure the similarity between the original loop and its mirror image with respect to the considered axis. We therefore need to define a way to compare two curves.

4.1. Comparison metrics for two closed curves

Let us first assume that we want to compare two open curves A and B (curves that have defined starting and ending points). Several metrics have been proposed and one of the most known and easy to implement is the Hausdorff distance [22]. However, it is known that this metric can be misleading as the distance between two very different curves can be low in some cases. This problem is due to the fact that the Hausdorff distance does not take into account the succession order of the points. A proper comparison must consider that both curves are walked in one direction, without the possibility to go back. A possible measure of the distance between the two curves is the mean distance between the current point on A and the current point on B when performing these walks. Imagine that a man is walking on A and his dog on B. The distance between the curves is the average size of the leash. This idea is very similar to the Fréchet distance [23] except that we take into account the average distance rather than the maximum distance. The walks have to be optimized in order to minimize the mean distance. We use the mean rather than the total sum in order to be able to compare results obtained for curves with different number of points. We call this proposed metric *morphing distance* since it is related to the amount of energy necessary to transform one curve into the other. The mathematical definition of the morphing distance is as follows:

Definition 1 (*morphing distance for discrete curves*). Let $A = (A_1, A_2, \ldots, A_N)$ and $B = (B_1, B_2, \ldots, B_N)$ be two open discrete curves with N points and let $\alpha : [1, N'] \to [1, N]$ and $\beta : [1, N'] \to [1, N]$ be two continuous and non-decreasing reparametrizations. Then the *morphing distance* is defined as

$$\delta_M(A, B) := \inf_{\alpha,\beta} \frac{1}{N'}\sum_{i=1}^{N'} \|A_{\alpha(i)} - B_{\beta(i)}\|, \quad (2)$$

where $\|\cdot\|$ stands for the Euclidean distance and N' is the number of steps required to go from the starting point to the ending point on both curves for a particular reparametrization.

It is possible to construct a bidimensional graph that represents all the possible associations between a point of A and a point of B. The columns of this graph correspond to the points of A and its rows correspond to the points of B. The intersection between a row and a column corresponds to a pair of points (A_i, B_j). A simultaneous progression on curves A and B can be represented as a path on this graph (Fig. 4) [24,25]. Now each intersection or node of the graph can be weighted by the Euclidean distance between the corresponding points A_i and B_j. Finding the morphing distance between A and B is therefore equivalent to finding the optimal path between (A_1, B_1) and (A_N, B_N), i.e. the path with the minimum cumulative weight. An efficient way to compute this morphing distance is to use dynamic programming. The idea is that the progression on A and B is in only one direction. Going back is forbidden. So in order to arrive at node (i, j) of the graph, we have to come from one of the left or above nodes $(i-1, j-1)$, $(i-1, j)$ or $(i, j-1)$. If we know the cumulative weights of the optimal paths that lead to each of these three nodes, we can select the node with the minimum cumulative weight. The optimal path to go to node (i, j) is the one that comes from this best

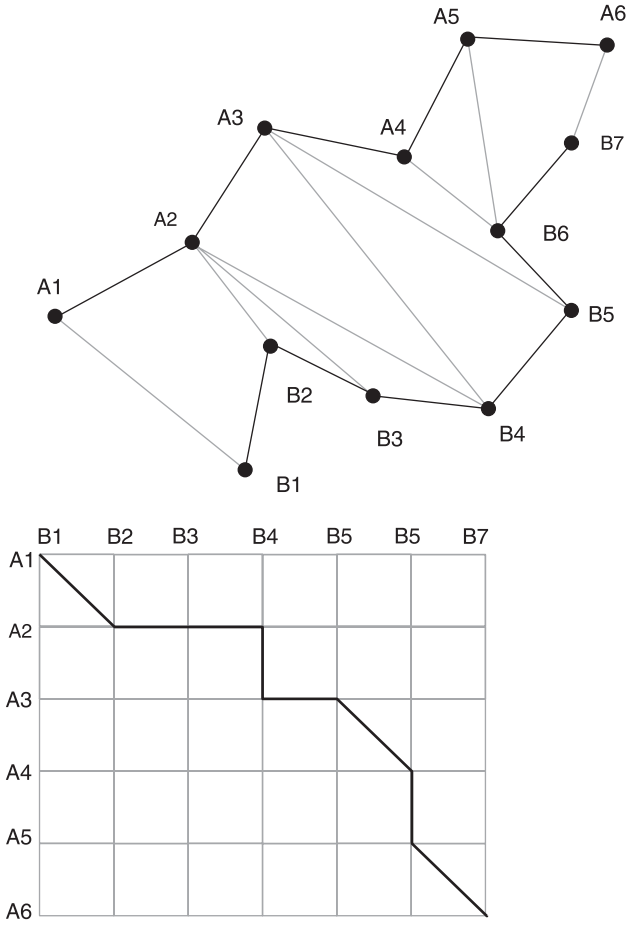

Fig. 4. Graph representation of a simultaneous progression on two discrete curves.

neighbor. The sum of the weight of node (i, j) and of the cumulative weight of its best neighbor is the cumulative weight of the optimal path that leads to node (i, j). We can therefore recursively compute the cumulative sum of every node, starting from low indices nodes. This idea is similar to what was already proposed to compute the Fréchet distance between two curves [26]. Computing the morphing distance between A and B is equivalent to compute the cumulative sum of node (N, N), which requires to compute the cumulative sum for every node of the graph. This can be done in $O(N^2)$ operations.

Now the problem is that the curves we study are not open; they are loops. This means that there is no definite starting or ending point. Actually, a loop can be considered as an open curve by choosing an arbitrary starting point and walking the loop until going back to the chosen starting point. However, each starting point provides a different "open" curve. Now, in order to compare two loops, it is possible without loss of generality to choose an arbitrary starting point on the first loop. However, we have to consider every point of the second loop as a possible starting point. We need to compute the morphing distance between the first curve and each curve resulting from a choice of starting point on the second loop. The morphing distance between the two loops is the smallest of the computed distances, that corresponds to the optimal starting point. Since there are N possible starting points, the total number of operations to compare two loops should be $O(N^3)$. However, as shown in Ref. [24], it is actually possible to do the full comparison in $O(N^2 \log_2 N)$ operations. Using the optimization described in Ref. [24], we are able to compute the morphing distance between two 500 points loops in about 160 ms on a Pentium 4, 3.2 GHz computer.

4.2. Finding the symmetry axes

By definition, a symbol that possesses an axial symmetry is identical to its mirror image. However, if the mirror image is obtained by flipping the symbol with respect to an axis that is different from the symmetry axis, then both symbols are not directly superposable. One of them has to be rotated in order to be perfectly superimposed with the other. The necessary rotation angle is the double of the angle between the flipping axis and the actual symmetry axis.

The procedure to find the number of symmetry axes of a symbol consists in generating the mirror image of its boundary loop with respect to an arbitrary axis—for example the vertical axis—and to compare rotated versions of this mirror image with the original loop. We use all the rotations between 0 and 2π rad with a step of 0.2 rad. Every time the morphing distance between the two curves is less than a threshold (we use 15 pixels), we add one to our count of symmetry axes. We discard the loop if it has a number of symmetry axes different from one. Otherwise, we rotate it so that its symmetry axis becomes vertical, as mentioned in Section 3.2.

5. Morphology analysis

The presence of a unique symmetry axis is a necessary but insufficient property to recognize a One. The other prominent characteristics of a One is its arch shape. Since the symbols we study are usually not perfectly symmetric, we separately analyze the shape of both halves of each loop. In order to help this analysis, we first modify the coding of the symbols as described below.

5.1. Re-coding the symbols

We need to split the boundary loop of a symbol in two halves separated by the symmetry axis in order to analyze each half separately. We first walk the loop to find where it crosses the symmetry axis. There should be exactly two crossings. If more are found, then it is a sign that the symbol is not a One and it is dismissed. Otherwise, the loop is split in two open curves starting and ending at the crossing points. We refer to these two curves as *halves*.

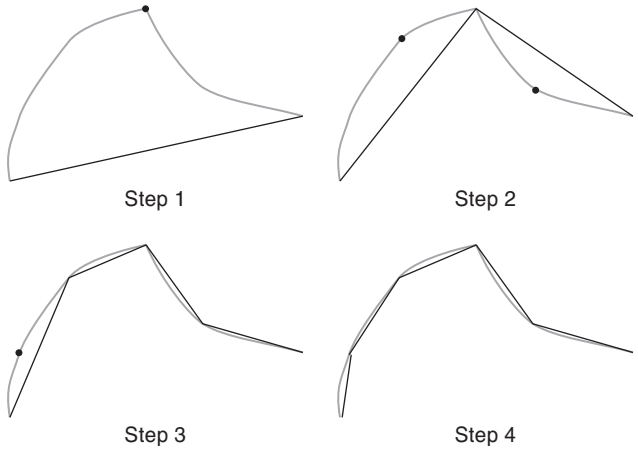

Fig. 5. Vectorization of a curve. The approximation starts with one segment and the farthest points are recursively added.

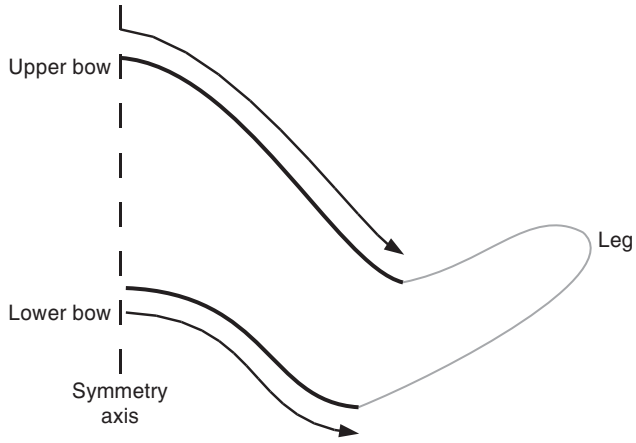

Fig. 6. Extraction of the upper and lower bows from the boundary curve of a half-symbol.

In order to smooth the high frequency spatial irregularities of the curves, we perform a *vectorization* of each half. This vectorization consists in describing the curve by the smallest possible number of straight segments. We construct this description recursively. We start with a single segment that directly joins the first and the last point of the curve, respectively, M_1 and M_2. We then look for the point of the curve M_3 that is farthest away from the segment and we include it in the vectorized form of the curve. That is, the single segment M_1M_2 is replaced with the pair of segments M_1M_3 and M_3M_2. This two-segment description is a better approximation of the curve than the single-segment description. We repeat the process of splitting every segment in two until the distance between the segment and the farthest point of the curve is less than 8 pixels (Fig. 5). At the end, we are left with the vectorized curve.

5.2. Testing the shape

In order to qualify for being a One, a symbol has to contain an arch in its central part, possibly prolongated by legs. This means that each half-symbol has to contain a semi-arch starting at the symmetry axis. This is only the case if its boundary curve presents upper and lower bows near the symmetry axis (see Fig. 6). In order to avoid confusions, we use the term *arch* for the thick symbol and *bow* for the boundary, although both words essentially refer to similar shapes. From now on, we consider the boundary curve that defines a half-symbol. We obviously discard the curve if it is just a straight flat segment. Otherwise, when we walk either its upper or lower part starting from the symmetry axis, we eventually reach a non-straight section. We then keep following the curve while it continues in the same general direction (one of North-West, North-East, South-West or South-East). The traveled part of the upper (resp. lower) boundary, from start to end, defines the upper (resp. lower) bow. The non-traveled part of the boundary is called the *leg* (see Fig. 6). Note that this analysis can equally be performed on wire curves. We now verify several properties of the extracted bows:

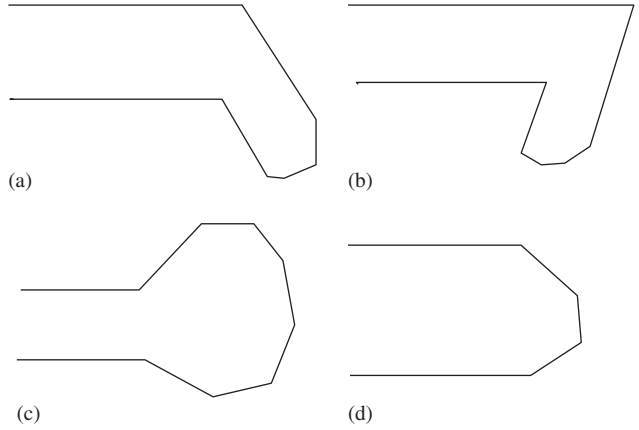

Fig. 7. Checking the compatibility of the bending directions of the upper and lower boundary bows. Cases (a) and (b) are accepted while (c) and (d) are rejected.

(1) Identical bending directions: the central part of the curve is a semi-arch only if the upper and lower boundary bows are bent in the same direction (NW, NE, SW or SE). If this condition is not fulfilled, we discard the curve (Fig. 7). Otherwise, we call outer bow (resp. inner bow) the bow that bends toward (resp. away from) the other bow.
(2) Height of the bows: a bent curve is a bow only if its height is sufficient; otherwise it is almost flat. We define the height of a bow h_O or h_I as the height difference between its starting point (on the symmetry axis) and its ending point (Fig. 8). We compare this height to the total height of the half-symbol h_T. We fix a 13% threshold, namely the curve is rejected if $h_O < 0.13h_T$ or $h_I < 0.13h_T$.
(3) Compactness: Ones are little compact. More precisely: their length is significantly larger than their thickness.

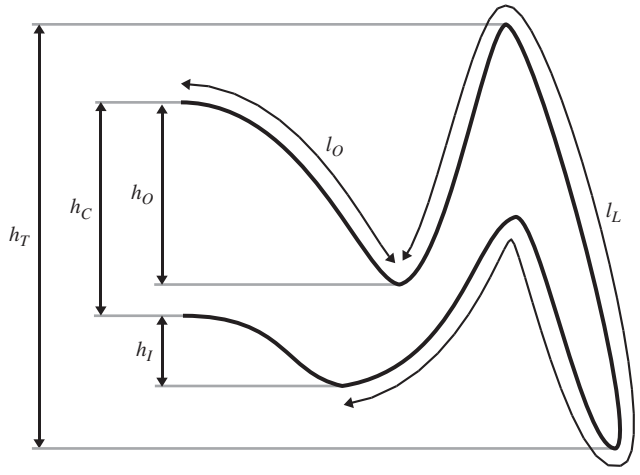

Fig. 8. Definition of the parameters used in the morphological analysis.

We approximate the thickness of a symbol by its thickness on the symmetry axis, which is equal to its central height h_C (see Fig. 8). The length of the half-symbol is approximated as the length of the outer bow l_O plus half the length of the leg l_L (Fig. 8). The symbol is rejected if $l_O + l_L/2 < 1.6h_C$.

(4) Arch integrity: all the above rules guarantee the presence of a well-shaped semi-arch in the half-symbol we are studying. The part of the curve that is not included in the outer or inner bows constitutes the leg of the symbol. This leg is supposed to be a continuation of the arch and, in particular, to lie out of the arch. In occasions, the leg curve can actually enter the arch area and form an undesirable cavity inside the arch. In this case, the symbol can no longer be considered as a One. In order to test this condition, we consider that the semi-arch is closed by the segment S that joins the ends of the outer and inner bows (Fig. 9). We then check if any point of the leg curve lies inside the closed semi-arch. In practice, we tolerate a small cavity, that is we accept a point that is inside the closed semi-arch at a distance no greater than 25 pixels. This tolerance has been found to be necessary in order to accept certain particular instances of Ones. However, any symbol with a cavity deeper than 25 pixels is rejected. Fig. 9 presents examples of accepted and rejected curves.

At this stage, all the symbols that have not been rejected are considered as Ones. We have found that the application of these few rules provides a good recognition rate. In the following section, we present some results obtained on various images.

6. Results

We now apply the proposed technique on several test images. The first attempt is done on the image of Fig. 1 that

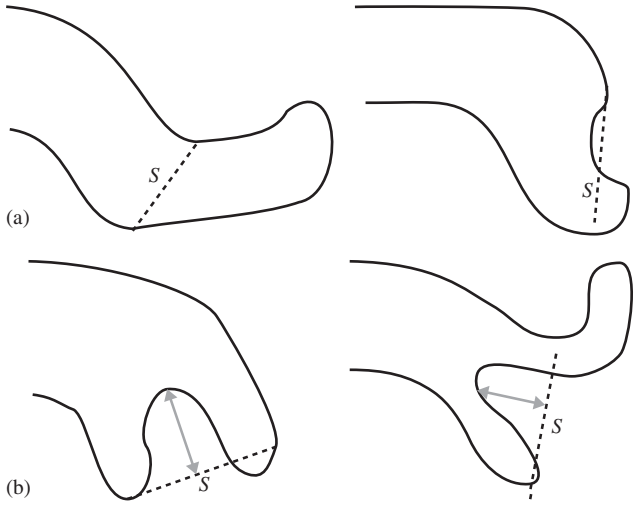

Fig. 9. Testing the integrity of the semi-arch. S is the segment closing the arch. The arrows denote cavities deeper than 25 pixels. (a) Accepted curves. (b) Rejected curves.

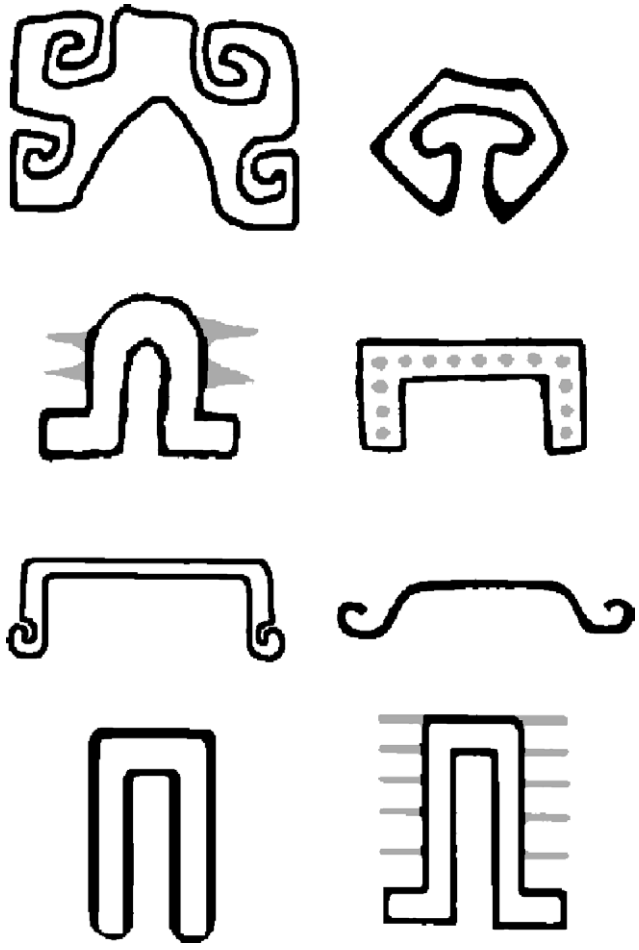

Fig. 10. Recognition of the Ones. Black parts are the symbols recognized as Ones. Parts in light gray are rejected.

contains many different examples of Ones. All of them are correctly recognized. Fig. 10 shows other instances of Ones with additional ornaments. We drew in black the parts of

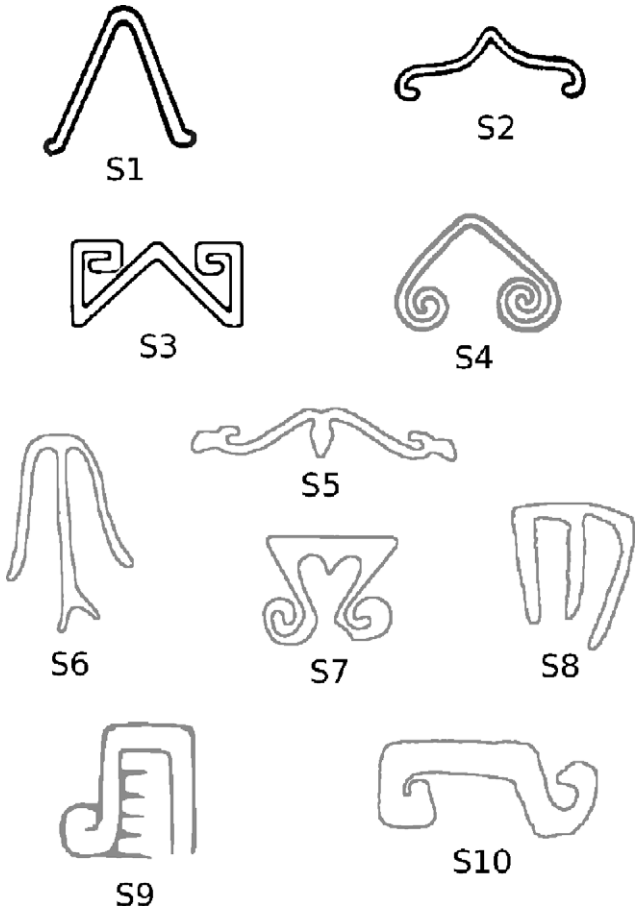

Fig. 11. Recognition of the Ones (continuation). S4 is wrongly rejected.

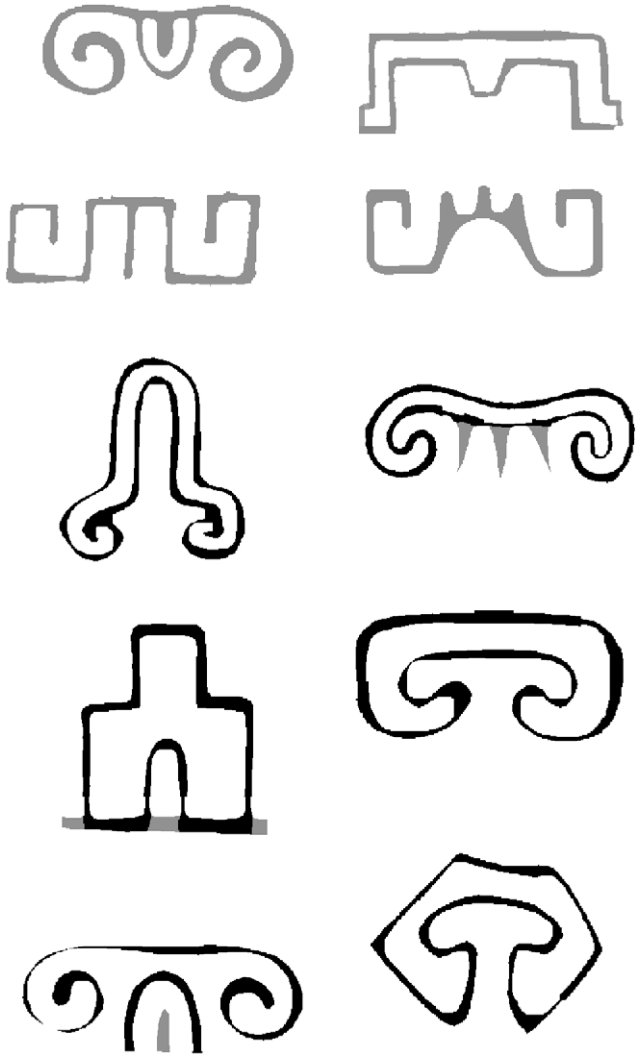

Fig. 12. Recognition of the Ones (continuation).

the image that are recognized as Ones and in light gray the rejected parts. It can be seen that the Ones are correctly extracted and recognized. In Fig. 11, symbols S1 through S3 are correctly recognized as Ones. S4 is incorrectly rejected because it is poorly drawn and its spiral legs are not symmetric. Symbols S5 through S10 might be considered as exotic variations of Ones but this would suppose a substantial extension of the definition of Ones. Since our algorithm considers a narrower definition, it correctly rejects these symbols. Similarly, in Fig. 12, only the symbols that correspond to our definition are classified as Ones. In Fig. 13 are shown examples of symbols that are not Ones. They are all rejected except one of them. The latter is accepted in spite of its poor symmetry because we have been quite tolerant in the symmetry criterion. Namely, we chose a relaxed threshold in order to accept Ones that are sometimes poorly drawn. This is a choice we made: we prefer to recognize most of the Ones at the cost of accepting some non-Ones, rather than wrongly reject many Ones. Fig. 14 presents a collection of more complex drawings. The algorithm correctly recognizes the Ones that are presented in the image, plus a few symbols that are limit cases and might or might not be considered as Ones, such as the two heart-shaped symbols in the third group.

Finally, Figs. 15–17 contain complete drawings extracted from archaeological objects. Fig. 15 was extracted from a flat stamp found in Texcoco, Mexico. The whole drawing is almost entirely composed of Ones and all of them are correctly recognized. Fig. 16 is a stone tablet with incised glyphic inscription from Guerrero, Mexico. The three Ones are well extracted and recognized. Lastly, Fig. 17(b) is the anterior face of the Stela 2, from Xochicalco, Mexico, seen on Fig. 17(a). This image is the most complex we have tested. As mentioned previously, because of our loose criteria, a few symbols are wrongly recognized as Ones. However, the most important is that all the Ones are recognized, except those that are occluded in the upper corners. This result can be considered very good given the difficulty of the task. The processing time for the extraction and recognition of all the symbols ranges from 1 s for Fig. 15 to 13 s for Fig. 17(b) on a Pentium 4, 3.2 GHz computer.

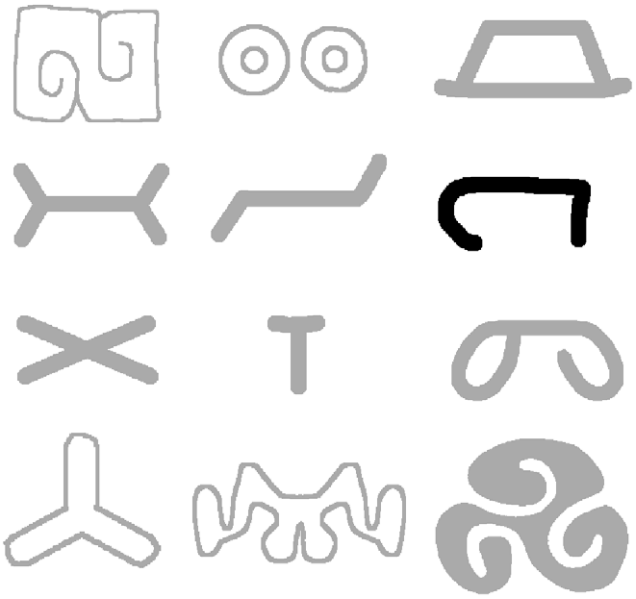

Fig. 13. Recognition of the Ones (continuation). One symbol is incorrectly recognized as a One.

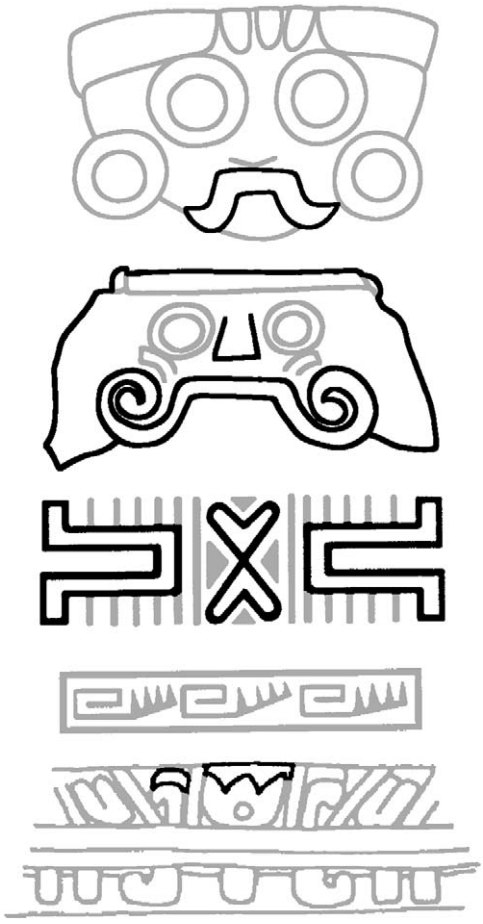

Fig. 14. Recognition of the Ones (continuation).

Fig. 15. Recognition of the Ones (continuation). Flat stamp from Texcoco, Mexico.

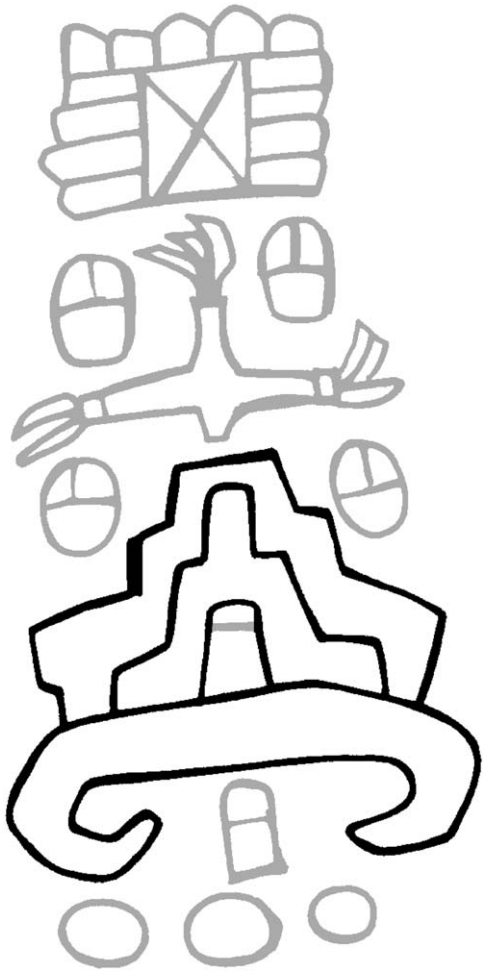

Fig. 16. Recognition of the Ones (continuation). Incised stone tablet from Guerrero, Mexico.

7. Conclusions

In this paper, we have presented a technique to recognize a Mesoamerican symbol that has a very variable shape. In order to deal with this variability, we have proposed an approach based on a series of rules that are common to all the instances of the symbol. The rules we have used are one of

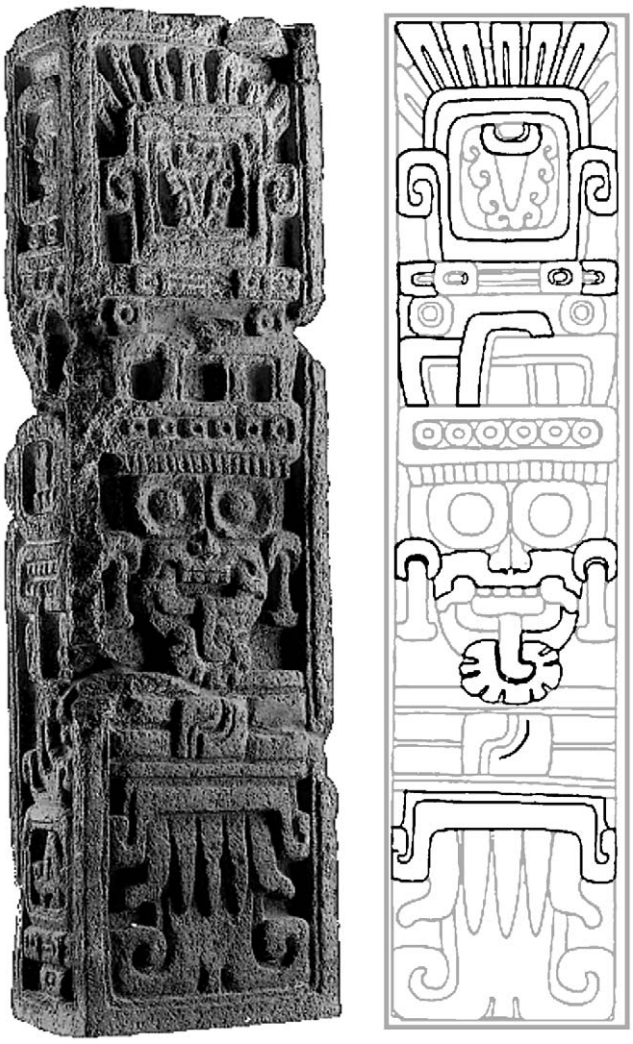

Fig. 17. Recognition of the Ones (continuation). (a) Stela 2 of Xochicalco, Mexico. (b) Anterior face of Stela 2.

symmetry and a few rules of morphology. The number of rules is actually very small and simple enough so that the processing time of a complete image—including extraction of the symbols—is quite small (usually a few seconds). In addition, the recognition results are very satisfactory given the difficulty of the problem. Actually, for now we have only included simple rules that seem sufficient to recognize most instances of the studied symbol. If needed, it is easy to use additional rules in order to make the recognition process even more accurate. Such rules could for example concern the length and shape of the legs of the symbol. Also, the various numerical parameters—such as thresholds—that we have used in the paper are the result of a heuristic study. They have been tuned to be widely tolerant. That means that we rather accept some false positives than reject correct symbols. A next step could be to train the system with a large set of examples in order to statistically determine the optimal parameters. Finally, the proposed algorithm is only able to recognize complete and unoccluded symbols. Dealing with interruptions or occlusions is a much more complex problem.

Acknowledgements

The authors' gratitude is extended to Professor Rubén Bonifaz Nuño for his continuous support and encouragement in this work. In addition, the authors wish to thank Dr. Hermilo Sanchez Cruz and Lic. Yazmin Ocadiz Martinez for their useful scientific discussions. This study was supported in part by CONACYT, Mexico, Grant C-811-2000.

References

[1] M. Covarrubias, Indian Art of Mexico and Central America, Alfred A. Knopf, New York, 1957.
[2] R. Bonifaz Nuño, Imagen de Tláloc: Hipótesis iconográfica y textual, Universidad Nacional Autónoma de México, Mexico City, 1986.
[3] R. Bonifaz Nuño, Hombres y serpientes. Iconografía Olmeca, Universidad Nacional Autónoma de México, Mexico City, 1989.
[4] R. Bonifaz Nuño, Olmecas: esencia y fundación. Hipótesis iconográfica y textual, El Colegio Nacional, Mexico City, 1992.
[5] R. Bonifaz Nuño, Cosmogonía antigua Mexicana. Hipótesis iconográfica y textual, Universidad Nacional Autónoma de México, Mexico City, 1995.
[6] O. Quesada, Tres signos. Escritura antigua de Mesoamerica. Universidad Nacional Autonoma de Mexico, Mexico City, 2006, to appear.
[7] D.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ 07632, 1982.
[8] A.K. Agrawala, A.V. Kulkarni, A sequential approach to the extraction of shape features, Comput. Graphics Image Process. 6 (1977) 538–557.
[9] H. Wechsler, A structural approach to shape analysis using mirroring axes, Comput. Graphics Image Process. 9 (3) (1979) 246–266.
[10] T. Pavlidis, Algorithms for shape analysis of contours and waveforms, IEEE Transactions on Pattern Analysis Machine Intelligence PAMI-2 (1980) 301–312.
[11] N.J.C. Strachan, P. Nesvadba, A.R. Allen, Fish species recognition by shape analysis of images, Pattern Recognition 23 (1990) 539–544.
[12] E. Bribiesca, A. Guzmán, How to describe pure form and how to measure differences in shapes using shape numbers, Pattern Recognition 12 (1980) 101–112.
[13] B. Gunsel, A.M. Tekalp, Shape similarity matching for query-by-example, Pattern Recognition 31 (7) (1998) 931–944.
[14] M. Carlin, Measuring the performance of shape similarity retrieval methods, Comput. Vision Image Understanding 84 (2001) 44–61.
[15] F. Mokhtarian, S. Abbasi, Shape similarity retrieval under affine transforms, Pattern Recognition 35 (2002) 31–41.
[16] F. Mahmoudi, J. Shanbehzadeh, A.-M. Eftekhari-Moghadam, H. Soltanian-Zadeh, Image retrieval based on shape similarity by edge orientation autocorrelogram, Pattern Recognition 36 (2003) 1725–1736.
[17] A. Torsello, E.R. Hancock, A skeletal measure for 2D shape similarity, Comput. Vision Image Understanding 95 (2004) 1–29.
[18] Y. Ding, T.Y. Young, Complete shape from imperfect contour: a rule-based approach, Comput. Vision Image Understanding 70 (2) (1998) 197–211.
[19] M. Ahmed, R. Ward, A rotation invariant rule-based thinning algorithm for character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1672–1678.
[20] Y.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (1985) 236–239.

[21] J.R. Parker, Algorithms for Image Processing and Computer Vision, Wiley, New York, 1994.
[22] E. Belogay, C. Cabrelli, U. Molter, R. Shonkwiler, Calculating the Hausdorff distance between curves, Inform. Process. Lett. 64 (1997) 17–22.
[23] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, Int. J. Comput. Geom. Appl. 5 (1995) 75–91.
[24] H. Fuchs, Z.M. Kedem, S.P. Uselton, Optimal surface reconstruction from planar contours, Commun. ACM 20 (1977) 693–702.
[25] T. Sederberg, E. Greenwood, A physically based approach to 2D shape blending, Comput. Graphics 26 (1992) 25–34.
[26] T. Eiter, H. Mannila, Computing discrete Fréchet distance, Technische Universitat Wien Technical Report CD-TR 94/64, 1994.

About the Author—YANN FRAUEL obtained the M.Sc. degree in Optics in 1995 and the Ph.D. degree in Photonics in 1999, both from the Institut d'Optique—Université Paris XI (France). He carried out two postdoctoral stays, respectively, at the University of Cambridge (UK) and at the University of Connecticut (USA). Since 2002, he has been an Associate Researcher with the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas of the Universidad Nacional Autónoma de México (Mexico). His current research interests are in the domains of 3D acquisition techniques and pattern recognition.

About the Author—OCTAVIO QUESADA obtained the B.Sc. degree in Basic Biomedical Research in 1986, and the Ph.D. in Neurochemistry in 1990, from the Universidad Nacional Autónoma de México (UNAM). He has worked as full time researcher at the Neurosciences Department of the Instituto de Fisiología Celular, UNAM, for 10 years. Since April of 2000, he is ascribed to the Instituto de Investigaciones Filológicas of the same university, were he develops studies on Mesoamerican iconography.

About the Author—ERNESTO BRIBIESCA received the B.Sc. degree in electronics engineering from the Instituto Politécnico Nacional in 1976. He received the Ph.D. degree in Mathematics from the Universidad Autónoma Metropolitana (UAM) in 1996, he was researcher at the IBM Latin American Scientific Center, and at the Dirección General de Estudios del Territorio Nacional (DETENAL). He is associate editor of the Pattern Recognition journal. He has twice been chosen Honorable Mention winner of the Annual Pattern Recognition Society Award. Currently, he is Professor at the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) at the Universidad Nacional Autónoma de México (UNAM), where he teaches graduate courses in Pattern Recognition.