Proceedings of the IASTED International Conference
Signal and Image Processing
October 27-31, 1998, Las Vegas, Nevada - USA

Object Recognition Using Signatures

J R Parker
Laboratory for Computer Vision
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA

Zhang Zhou
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA

Abstract:

A method is described for quickly detecting the similarity between planar shapes. By normalizing the distance from the boundary to the geometric center of an object, a 2D shape can be represented as a 2π periodic function which is invariant to both translation and scaling. In this *signature space*, a rotation of the original shape is a translation; thus, the similarity measurement is the minimum value over the translation from 0 to 2π. Experimental results are included for the comparison of a group of 24 shapes.

Introduction

The problem of characterizing the shape of an object is key to pattern recognition. Shape is not a single property, but a large family of properties, and often many such properties are applied to a recognition task. Most often, we use shape information to distinguish one of a set of possible shapes (classification), and it is this problem that will be addressed. Most shape analysis and recognition methods are based on regions, local properties of the skeleton or the boundary (contour) of the object [FU82].

Kupeev and Wolfson [KW94] give an algorithm for the detection of similarity among a certain class of planar shapes. These shapes are represented by closed curves without self-intersections. Such shapes could, for example, describe the outline of a 2D object or the contour of the visible part of a 3D object. Our work suggests the use of the same set of 2D contours for recognition.

Traditionally, recognition methods can be divided into three classes. The first class uses global features of the shape, e.g. perimeter and area. Statistical pattern recognition techniques are used in this case for matching [PARK94]. The second class uses local features, generally in terms of line and curve segments defining the boundary. These features are organized in a highly structured manner, such as a sequence of strings with a grammar, and matching is performed by parsing. This is referred to as syntactic recognition [FU82]. The third type also uses local and relational features which are organized in a graph [KUPE94], where graph searching algorithms are used for matching. The graph homomorphism algorithm is NP-complete, rendering it intractable for large graphs.

A similarity measure is a number that indicates how near two shapes are to being the same. It should have a known maximum, which indicates a perfect match, and minimum. An ideal similarity measure would be scale, translation, and rotation independent, and would be computed by a digitally continuous function; that is, a single pixel change in the target shape should yield a small, consistent change in the similarity measure.

Signature of a Contour

A planar shape has three basic parameters: position, orientation and scale. Two otherwise identical shapes can have different parameters, and so these cannot be used for recognition purposes; indeed, we wish to transform the objects so that they are independent of these three parameters.

A *signature* is a functional representation of a contour, which may be generated in various ways [GONZ92; HOLT93; OROU86; TCHO92]. Here we will use the distance-versus-angle signature which plots the distance from the geometric center to the boundary as a

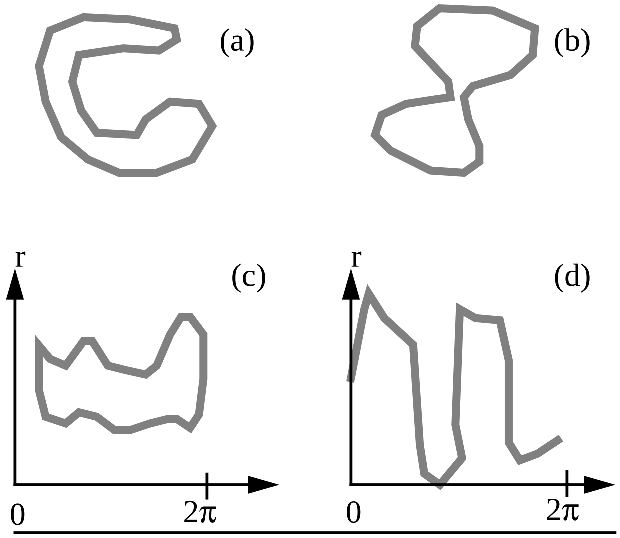


Figure 1 - Two shapes (top) and their distance-VS-angle signatures (bottom).

function of angle. Figure 1 illustrates this using two simple shapes and their signatures. This signature cannot always be described as a 1D function, as seen in Figure 1a, in where one angle θ may correspond with several different r values. Therefore, a 2D function $s(\theta, r)$ represents the signature, where s can only be the value of 0 or 1. Hence, $s(\theta, r)$ can be considered as a bi-level image.

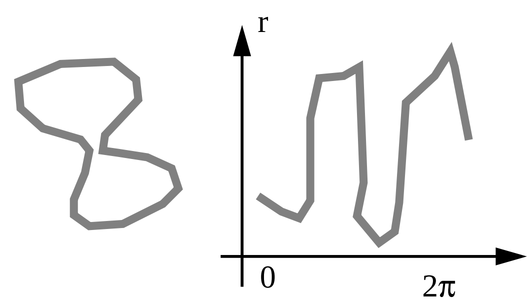
The distance-versus-angle signature is the result of a transformation from Cartesian to the polar coordinate system. Using the geometric center of the shape as the origin guarantees that such a transformation is invariant to translation. In addition, we normalize with respect to the distance, so that the transformation is also scale invariant. However, it is still dependent on the orientation of the original shape.

Gonzalez and Woods [GONZ92] suggest two ways to select the starting point for generating a signature invariant to rotation. One way is to choose the point farthest from the centroid. The other is to use the point on the principal eigen-axis farthest from the centroid, which requires more computation. In our case, however, neither of above methods work well. In general for either method, when two shapes are not very similar the distance between their signatures rarely reaches the minimum value, even though they are created from either of the above starting points. What is more, there is no simple trick to finding such an extreme value. Therefore, we need to rotate one of the shapes from 0 to 2π in order to find the angle for which the signature is a minimum. Fortunately, a rotation in the Cartesian coordinate system is a translation along the θ direction in the polar coordinate system, so a rotation matrix need not be computed. Since all signatures are the 2π periodic functions, the comparisons are made within one period.

The distance-versus-angle signature is also sensitive to the mirror operation (reflection) of a shape. This operation is not equivalent to a any rotation. Kupeev and Wolfson [KUPE94] did not examine such a situation although their attribute graph is also sensitive to it. There are two kinds of mirror operations, horizontal and vertical. The vertical mirror operation is equivalent to a horizontal one plus a rotation of π. Therefore, we only need to consider horizontal reflection. Obviously, the signature of a horizontally reflected shape is the reflection of its original signature because reflection can be carried out by substituting θ with $-\theta$ (Figure 2). And so, the comparisons have to go through one period of the reflected signature as well. In fact, there is also a displacement along the θ direction in addition to changing the sign of θ. Since the signature is a periodic function and the entire period is considered, this displacement can be ignored.

The basic signature calculation can be carried out in parallel, with each point on the boundary being an independent result. In practice points would be grouped to provide sufficient computation in each process to make up for overhead. In addition, each translation of the signature (IE rotation of the object) can also be computed in parallel, giving a large potential real time advantage.

Figure 2 - Reflection of the right shape from Figure 1, and its signature.



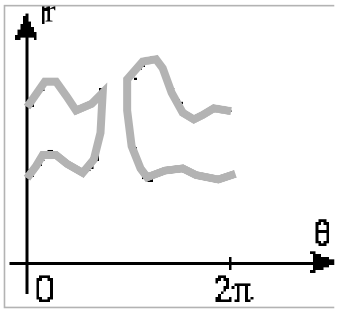


Figure 3 - A form of closed curve signature, rotated so that is has been cut.

Distance between Signatures

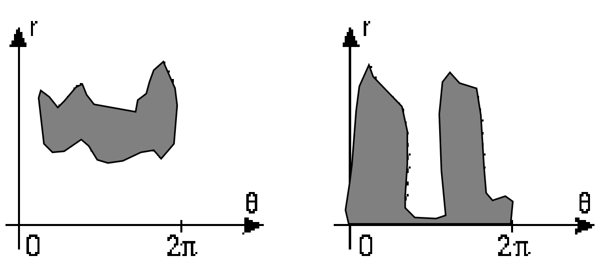
As mentioned above, a signature is easier to handle than its original shape, and the comparison of two shapes can be executed using only the signatures. In our case, we compare two shapes by measuring the difference of their signatures, and to do this we first define the *area* of a signature.

Since our shapes are closed curves without holes, their signatures can only belong to either of the following two cases: the signature is a closed curve as, in Figure 4a, or the signature is an open curve, as in Figure 4b. The signature in Figure 3 is also a closed curve case. Here the curve means a sequence of points rather than a continuous line.

For the closed curve, we define the area of the signature as the area of that closed region; for the unclosed curve, the area is the area of the region between the curve and the θ axis from 0 to 2π as Figure 4 shows.

If we know the sequence of the points, we can then use an approach such as the chain code area (from [PARK94]) to find the area. It is first necessary to determine the sequence of the points in the signature, because the distances between two consecutive points are not uniform as they are in the original. Since the points in the original shape are connected, so we can first use the chain code [FREE61; PARK94] to find the sequence of points on the original shape. Then, since there is a one to one correspondence between a shape and its signature, we can obtain the same sequence of points in a signature. There are also two directions of point sequences, one giving a positive area and one giving a negative area. To compare two shapes, they must both be scanned in the same direction, or we will be comparing one shape with a reflection of the other. Now that we know the sequence of the points in signature, and since the signature is also a bilevel image s(θ, r), we can fill that area with a known value, say S = 1.

Figure 4 - Closed curve area (left) and unclosed curve area (right).

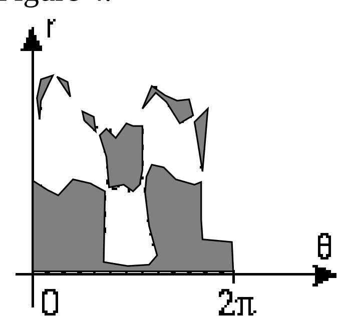


Next, we can define the *distance* between two signatures. The distance is the area after performing an XOR operation on the two signature areas. We can say the distance is the number of points which only belong to one of the signature's area (see Figure 5). Finally, the *distance* between two shapes is the minimum the distance between one signature and the other signature which shifts from 0 to 2π.

Algorithm

From the above discussion, we can see that the algorithm should include four parts: input image, create signature, fill the area, and perform XOR operation. The original signature s(θ, r) is in the range of [0,2π) x [0,1]. Since s(θ, r) will be treated as a image, we should expand its original range to a proper size. Therefore we need two values **Q** and **R** (scaled size) to expand θ and r respectively. Then the size of the signature changes to 2π **Q*R**. We also need another constant **N**, the total number of times we shift a signature in one period: **N** should be called the *sampling density.* So the value 2π /**N**

Figure 5 - XOR of the two signatures in Figure 4.



is the step for each shifting. The algorithm proceeds as follows:

- Input 2 shapes (contours)
- For each shape

  [1] Find the centroid and the starting point for the chain code

  [2] Use the chain code to find the sequence of the points

  [3] Create the signature and fill the area $S_i(\theta,r)$

- Find the minimum value $d = \min_i(d_i)$, where $d_i = \min(S_1(\theta,r) \text{ XOR } S_2(\theta+2\pi i/\mathbf{N}, r), S_1(-\theta, r) \text{ XOR } S_2(\theta+2\pi i/\mathbf{N}, r)$
- Output d and related signatures (optional).

Suppose the input shapes are within the image of size m x n; then the number of points on the contour is less than mn. Step 2.(a) and 2.(b) require O(nm) time units. The computation steps for step 2.(c) are O(nmQR). Step 3 needs 2N times of XOR operations. Each operation costs O(QR) time. Therefore the complexity of the algorithm is polynomial, unlike the structure recognition or relational graph methods which use NP-complete algorithms.

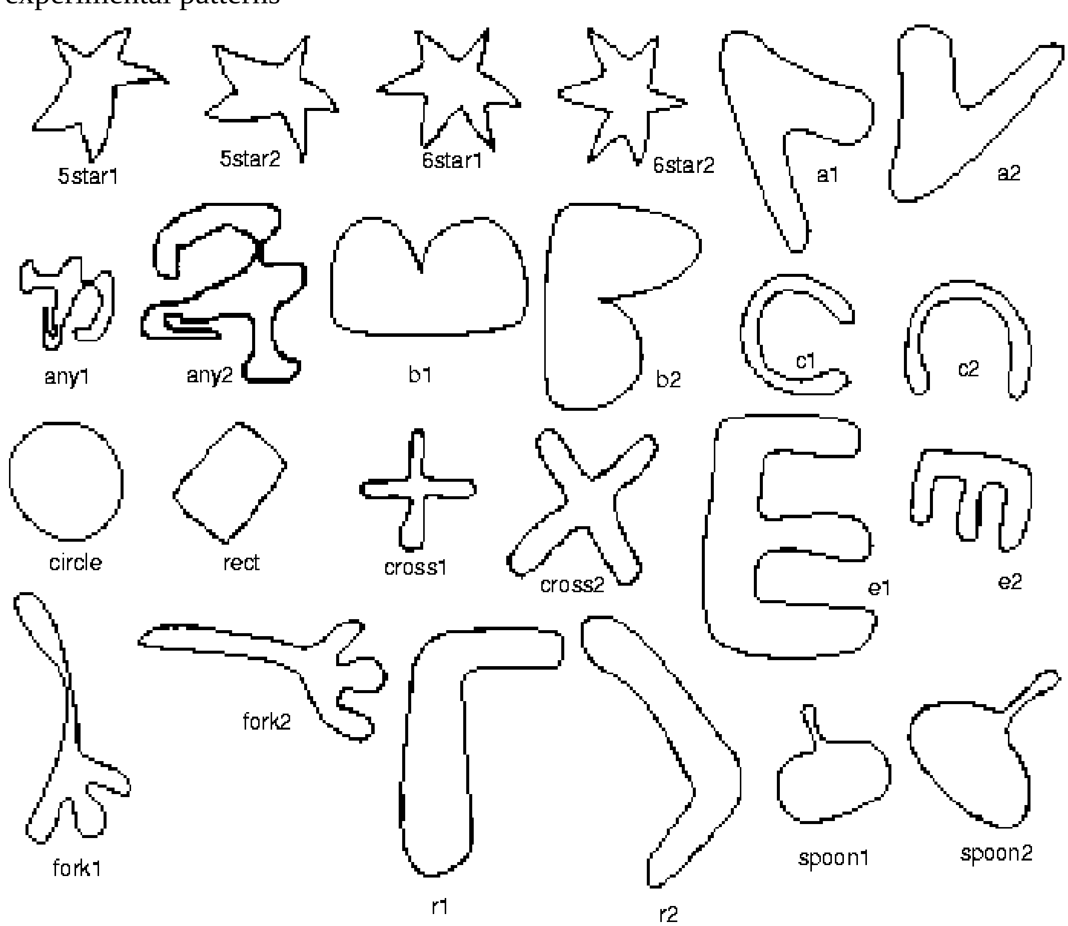
Experimental Results

The above algorithm has been implemented, and was tested using the same group of 24 shapes as in [KUPE94]. There are 12 pairs of perceptually similar shapes in the database (see Figure 6). For each shape all the 24 distances (including to itself) have been evaluated. The experiment was conducted with different values of **N** (sampling density). The values used were **N**=12, **N**=24, and **N**=32. Other parameters are **Q**=40 and **R**=255; the size of the input images is less than 300 x 300. Table 1 gives the results of **N**=24. Table 2 gives the results of **N**=12, and Table 3 is **N**=32.

The values in Table 1 are distances between shapes: the smaller the value is, the closer the two shapes are. It is a trivial observation that a shape is closest to itself, having a distance of 0. Most values in the table seem reasonable in the way that they indicate similarities between objects.

For the different values of **N**, we have different distance values in the tables. For **N**=12 (see Table 2), most of the values are also

Figure 6 - The 24 experimental patterns



plausible, but the results seem too coarse. The results of **N**=32 are very close to those for **N**=24, indicating that using **N**=24 is enough for this group of shapes. The computational time for calculating those 24 x 24 tables in the cases of **N**=12, **N**=24 and **N**=32 are 493 seconds, 689 seconds (1.2 seconds each) and 841 seconds respectively on an SGI Indigo2. The code has not been optimized. The only time available for the Kupeev and Wolfson scheme is 878 seconds for the N=32 table on a Sun Sparc II, in which one of the patterns is misclassified.

Summary

We have presented a simple algorithm for the recognition of planar shapes. The algorithm uses signatures for comparisons, the measurement used is area. Since there is the only one feature, matching process is fast, certainly faster than the scheme it was compared against. While the application here was shapes with no holes, holes themselves have a shape, and signature matching can be extended to deal with such objects. We are experimenting with this at the moment.

It is also possible to combine our method with other shape matching algorithms, which is more robust and accurate than using only one method. The specific application of the multiple algorithm shape matching system is the classification of respirogram curves (essentially a simple data graph) for waste water treatment applications[PARK97]. The fact that signature matching can largely be parallelized lends speed to the method is real applications.

References

[FU82]Fu, K. S. Syntactic Pattern Recognition and Applications, Prentice-Hall, 1982.

[FREE61] Freeman, H and Shapira, R., On The Encoding Of Arbitrary Geometric Configurations, IEEE Transactions on Electronic Computers, Vol. EC-10, 1961. Pp. 260-268.

[GONZ92] Gonzales, R. C., and Woods, R. E., Digital Image Processing, Addison Wesley, 1992.

[HOLT93] Holt, R. J. and Netravali, A. N., Using Line Correspondences In Invariant Signatures For Curve Recognition, Image and Vision Computing, Vol. 11, No. 7, Sept. 1993. Pp. 440-446.

[KUPE94] Kupeev, K. Y. and Wolfson, H. J., On Shape Similarity, ICPR, October, 1994. Pp. 227-231.

[OROU86] O'Rourke, J., The Signature Of A Plane Curve, SIAM J. Comput, Vol. 15, No. 1, Feb 1986. Pp. 34-51]

[PARK94] Parker, J. R., Practical Computer Vision Using C, John Wiley & Sons, 1994.

[PARK97] Parker, J. R. , Histogram Methods For Scientific Curve Classification, SPIE Vision Geometry VI, San Diego, July 28-29, 1997.

[TCHO92] Tchoukanov, I., Safee-Rad, R., Smith, K., and Benhabib, B., The Angle Of Sight Signature For Two Dimensional Shape Analysis Of Manufactured Objects, Pattern Recognition, Vol. 25, No. 11, 1992. Pp. 1289-1305.

Tables 1-3: Portions of the complete distance tables for N=24,12, and 32.

Table 1: N = 24

	5star1	5star2	6star1	6star2	a1
5star1	0	1955	9083	9559	15028
5star2	1359	0	9188	9326	15191
6star1	8953	9026	0	4608	14299
6star2	10067	9726	5864	0	14265
a1	15028	15191	14209	14265	0

Table 2: N = 12

	5star1	5star2	6star1	6star2	a1
5star1	00000	1955	9083	9559	15356
5star2	08853	0	9188	10370	15647
6star1	10085	9188	0	6086	14855
6star2	10431	9726	5864	0	14291
a1	15356	15733	14209	14775	0

Table 3: N = 32

	5star1	5star2	6star1	6star2	a1
5star1	0	2723	8993	9503	14596
5star2	1781	0	9188	9326	14617
6star1	8869	9188	0	4608	13971
6star2	9559	9108	3292	0	14265
a1	14596	14617	13971	14185	0