

Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments

Brian Scassellati

MIT, Department of Computer Science and Department of Brain and Cognitive Science
545 Technology Square, Cambridge MA 02139

Sophoclis Alexopoulos

UCSD, Department of Electrical and Computer Engineering
9450 Gilman Drive, La Jolla, CA 92093

Myron Flickner

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099

ABSTRACT

In content based image retrieval, systems allow users to ask for objects similar in shape to a query object. However, there is no clear understanding of how computational shape similarity corresponds to human shape similarity. In this paper several shape similarity measures were evaluated on planar, connected, non-occluded binary shapes. Shape similarity using algebraic moments, spline curve distances, cumulative turning angle, sign of curvature and Hausdorff-distance were compared to human similarity judgments on twenty test shapes against a large image database.

1 INTRODUCTION

One of the most challenging aspects to content based image retrieval is retrieval by shape. No mathematically rigorous definition of shape similarity accounts for the various semantic qualities that humans assign to shapes.¹ Without textual tags or other semantic descriptors, we can only approximate shape similarity from the 2D image. There have been a variety of proposed methods for determining similarity between planar shapes, including moment-based matching algorithms,^{2,3} Hausdorff-distance based metrics,^{4,5} and schemes for matching turning angle around the perimeter of a shape.⁶ The goal of any of these algorithms is to approximate the perceptual judgments of the user. Algorithms which more closely approximate the perceptual judgments of the user are considered to be more successful. This paper will explore the relative effectiveness of seven shape matching algorithms on selecting shapes similar to a given query object from a large database of images. Each algorithm will evaluate the similarity of the query shape to each member of the database and produce a ranked listing of similarity. The algorithms will be judged in relation to the results that human subjects gave for a similar shape matching experiment.

2 QUERIES AND DATABASES

The database used in all of our experiments was composed of over 1400 images taken from the QBIC (Query by Image Content) system developed at the Almaden Research Center.⁷ QBIC contains images taken from a variety of sources including Canadian trademarks, aircraft silhouettes, and commercial clip-art images. Binary shape masks were generated from the database images through a combination of automated and manual methods. Because of the automated outlining procedures, some of the shape masks contain multiple (non-connected) objects and some contain objects with holes. For our experiments, we will be concerned only with planar, connected, non-occluded shapes. In the case that the mask contains multiple objects, we will consider only the object with the largest surface area. Only one of our computational methods will account for objects with holes, although none of the query objects will contain holes.

Twenty original query shapes were drawn by the authors. All of the query shapes were drawn to be perceptually close to at least one object in the database. The shapes varied in complexity, number of angles, perimeter, and other standard measures. Some of the shapes (rectangles and ellipsoids) were constructed using basic drawing tools while others were drawn freehand. None of the query shapes were taken directly from the database. Each of the query objects was assigned a number and a short designation. The query shapes and their designations can be seen in Figure 1.

3 HUMAN PSYCHOPHYSICAL RESULTS

A simple testing paradigm was designed to allow subjects to make similarity matchings between the query shapes and the database images. Subjects were instructed that they would be shown one image and then asked to look through a photo album to find similar pictures. No definition of “similar” was offered, but subjects were told that the relative size and orientation of the pictures in the photo album were unimportant. An X-windows interface allowed subjects to view both the query object and the database while selecting objects. For each subject, the order of query objects and the order of objects in the database was randomly permuted to prevent ordering bias. Subjects were told to choose up to ten matching objects for each query object and that selection order is unimportant. Subjects could not select the same object twice for a given query object. (See Figure 2 for a sample session.)

A total of 40 volunteer subjects from the Massachusetts Institute of Technology and the Almaden Research Center took part in the experiment. Due to the size of the database, completing all twenty query objects took approximately 100 minutes. Because of the length of time required for a full testing, partial results from incomplete tests were accepted for completed query objects. On average, subjects completed 70% of the query shapes.

4 SHAPE MATCHING METHODS

Seven computational shape matching methods were selected for comparison testing against the human data results. Each of the seven techniques produces a numeric value for the similarity between the features of any two binary shape masks. The seven methods used in our study were:

1. Algebraic Moments
2. Parametric curve distance
3. Parametric curve and first derivative distance
4. Parametric curve, first and second derivative distance
5. Turning angle
6. Sign of curvature
7. Modified Hausdorff distance

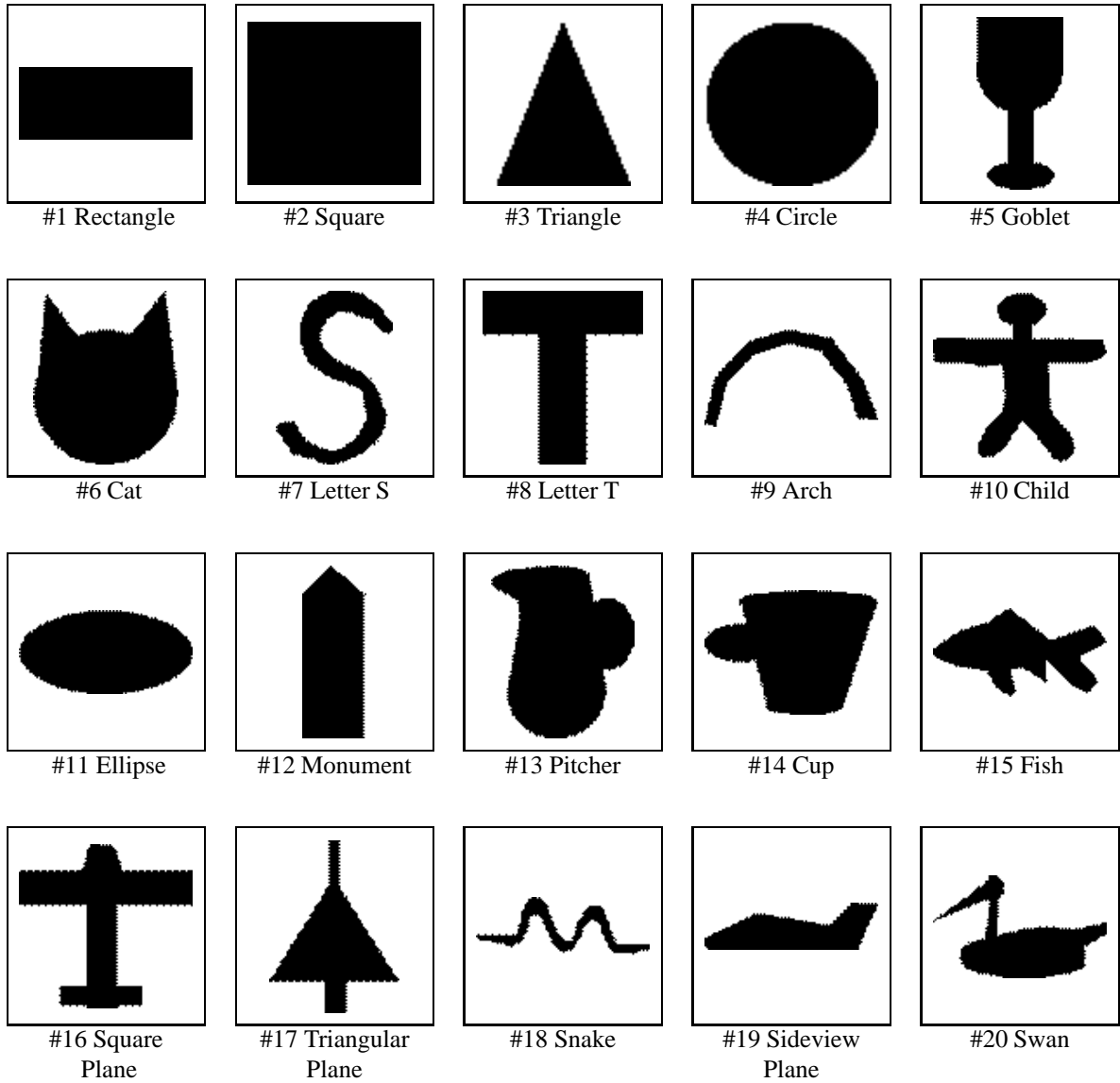


Figure 1: The twenty query shapes and their designations.

Each method was used to rank the similarity between the query shapes and the objects in the database. From the ordered listing of similarity results, the top ten similar objects for each query shape using each method were selected.

In all cases, features of the binary shape masks were pre-computed during database population time to allow for faster query times. For all boundary methods (all except the algebraic moment method), boundary lists were extracted from the shape mask. The boundary lists were low-pass filtered with $\rho = b/k$ where b is the number of points in the boundary list and k is the number of spline control points. The boundary lists were then fitted by least-squares approximation with cubic Q-splines at resolutions of $k=8, 16, 32, 64,$ and 128 control points as described in Flickner.⁸ Not all methods which use Q-splines make use of all resolutions. The control points were ordered to appear in a clockwise order starting from the end of the major axis which intersects the boundary closer to the centroid. This ordering gave visually good alignments within one or two control points.

Each algorithm required a number of tuning parameters. To set the parameters, a number of sample queries, unrelated to the twenty test queries presented here, were performed on the QBIC database. Parameters were chosen to maximize the perceptual similarity to the queries as judged by the authors. In some cases, limits were also set on the parameters to ensure reasonable running times. While these parameter settings are certainly not maximal under any set of conditions, they represent good approximations. Slight variations in the parameters affect the results of the queries in a very limited form and do not detract from the comparative results presented in the following sections.

4.1 Algebraic moments

The second order covariance matrix is computed using just the four connected boundary pixels. The major axis orientation is the direction of the largest eigenvector of the covariance matrix. The algebraic moment invariants are computed from the first m central moments and are given as the eigenvalues of predefined matrices, $M_{[j,k]}$, whose elements are scaled factors of the central moments.³ Using the notation of Taubin and Cooper³ we use moments up to degree 8 and the eigenvalues of the matrices $M_{[2,2]}, M_{[2,3]} \times M_{[3,2]}, M_{[3,3]}, M_{[3,4]} \times M_{[4,3]}, M_{[4,4]}, M_{[4,5]} \times M_{[5,4]}$ for a total of 18 features invariant to similarity transformations.

4.2 Parametric curve distance

For boundary representation we use a multi-resolution spline representation which is formed by least squares cubic polynomial spline fits to the digital boundary with 8, 16, 32, 64 and 128 control points (knots).

Given a function, f , a basis, $\{Q_i\}_{i=0}^{k-1}$, and an inner product, $\langle \cdot, \cdot \rangle$, its least-squares approximation can be written in matrix form as $Hc = P$, where P is a vector with coordinates $P_i = \langle f, Q_i \rangle$, c is a vector of control points with coordinates c_i , and H is the $k \times k$ Gram matrix with coefficients $H_{ij} = \langle Q_i, Q_j \rangle$. The least-squares solution is given by $g = \sum_{i=0}^{k-1} c_i Q_i$ or in vector form as $g = c^t Q$.

For 2-dimensional shapes, one system for each coordinate is solved since there is no interaction between the coordinates. The results of a least squares fit as outlined above is a curve, $f(t) = [x(t), y(t)]$, that can be written in vector form as $f(t) = [c_x^t Q(t), c_y^t Q(t)]$. By using the orthonormal Q-spline basis set, Q , derived in earlier work,⁸ the Gram matrix H is diagonal and the control points can be extracted via a simple inner product (convolution) computation. Clearly, the curve is uniquely defined by the control points since the same basis set is used.

A distance between two curves can be defined as $f(w^0(f_1 - f_2)^2 + w^1(\dot{f}_1 - \dot{f}_2)^2 + w^2(\ddot{f}_1 - \ddot{f}_2)^2)$ or in vector form with cubic splines

$$\int (w_k^0(d^t Q Q^t d) + w_k^1(d^t \dot{Q} \dot{Q}^t d) + w_k^2(d^t \ddot{Q} \ddot{Q}^t d)) = w_k^0 D_k^0 + w_k^1 D_k^1 + w_k^2 D_k^2 \quad (1)$$

where $\mathbf{d} = \mathbf{c}_1 - \mathbf{c}_2$. Note that the fact that Q-splines are orthonormal, $\int \mathbf{Q}\mathbf{Q}^t = \mathbf{H} = \mathbf{I}$ allows computation of the first term in (1) using the L_2 distance between the control points. The second and third terms can be computed by pre-computing the pseudo-Gram matrices ($\int \dot{\mathbf{Q}}\dot{\mathbf{Q}}^t, \int \ddot{\mathbf{Q}}\ddot{\mathbf{Q}}^t$) and evaluating the resulting quadratic forms.

Before using the distance formula above, it is prudent to optimally align the shapes. Since the ordering of control points is known, we can use least squares to solve for the parameters (s, θ, t_x, t_y) of the similarity transformation needed to align two shapes. Given two $k \times 2$ vectors, $(\mathbf{c}, \tilde{\mathbf{c}})$, of control points, the parameters of the similarity transformation that aligns with minimum L_2 error can be obtained⁹ by the least squares solution of

$$\begin{pmatrix} c_{1,x} \\ c_{1,y} \\ c_{2,x} \\ c_{2,y} \\ \vdots \\ c_{k,x} \\ c_{k,y} \end{pmatrix} = \begin{pmatrix} \tilde{c}_{1,x} & \tilde{c}_{1,y} & 1 & 0 \\ -\tilde{c}_{1,y} & \tilde{c}_{1,x} & 0 & 1 \\ \tilde{c}_{2,x} & \tilde{c}_{2,y} & 1 & 0 \\ -\tilde{c}_{2,y} & \tilde{c}_{2,x} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{c}_{k,x} & \tilde{c}_{k,y} & 1 & 0 \\ -\tilde{c}_{k,y} & \tilde{c}_{k,x} & 0 & 1 \end{pmatrix} \begin{pmatrix} s \cos \theta \\ s \sin \theta \\ t_x \\ t_y \end{pmatrix}. \quad (2)$$

It can be shown the error obtained from the least squares solution is equal to the minimal L_2 distance between the control points. However, note that if standard least squares is used, the above distance does not form a metric since it is not symmetric i.e. $D(\mathbf{c}_1, \mathbf{c}_2) \neq D(\mathbf{c}_2, \mathbf{c}_1)$. However, symmetry can be obtained using combinations (average, max, min) of both distances. Alternatively, total least squares could be used. Although the ordering of the control points is known, experimental results showed that due to starting point variations, the correspondence may be shifted by a small amount. This can easily be fixed by a circular shift of one of the control points before least squares fitting. The best alignment is then chosen to be the minimum of the fits over a range, r , of shifts.

Using the above approach, three methods were evaluated using two resolutions, $k = 32$ and $k = 64$, and a total distance of

$$D_t = w_{32}^0 D_{32}^0 + w_{32}^1 D_{32}^1 + w_{32}^2 D_{32}^2 + w_{64}^0 D_{64}^0 + w_{64}^1 D_{64}^1 + w_{64}^2 D_{64}^2.$$

The L_2 distance was computed independently at each resolution using singular value decomposition¹⁰ on (2) over a circular shift of $r = \pm 2$. The distance was not made symmetric.

In the first method, no derivative information is used and the resolutions are equally weighted, i.e. $(w_{32}^0 = w_{64}^0 = 1)$. All other weights are 0.

The second method uses curve and first derivative information only. The L_2 distance terms are weighted as before $(w_{32}^0 = w_{64}^0 = 1)$ and the first derivatives are weighted such that the weighted L_2 distances and the weighted first derivative distance are of roughly the same magnitude $(w_{32}^1 = w_{64}^1 = .150)$.

The third method uses curve, first and second derivative information. The L_2 distance terms are weighted as before $(w_{32}^0 = w_{64}^0 = 1)$, as are the first derivatives terms $(w_{32}^1 = w_{64}^1 = .150)$. The second derivative terms are weighted to give the distances for first and second derivatives roughly the same magnitude $(w_{32}^2 = w_{64}^2 = .0005)$.

4.3 Turning angle

To describe a shape, you might think about starting at a point on the perimeter and describing what directions you would move to travel around the perimeter. Turning angle techniques are based on this type of approach. For a shape a , the turning function $\Theta_a(s)$ measures the angle of counterclockwise tangent as a function of the arc-length s measured from some reference point on the boundary.⁶ The turning function measures the turning that takes place as you move along the perimeter. Mathematically, if $\kappa(s)$ is the curvature function of a curve then $\Theta(s) = \int \kappa(s)$. In our experiments, turning functions are sampled from spline fitted boundaries at a rate four times the number of control points.

Our turning angle method, similar to the work of McConnell,¹¹ uses dynamic programming to try to account for warpings that may exist between the query object and database object that result in stretching and compression of the horizontal scale of the turning angle values. It computes the global best match between Θ_1 and Θ_2 in the sense that it pairs up each element of Θ_1 with an element of Θ_2 to give the minimum sum of absolute differences. In the pairing, multiple elements of Θ_1 can match a single element of Θ_2 (and vice versa), but the matching must proceed monotonically through both sets. Thus it computes two sequences i_1, i_2, \dots, i_k and j_1, j_2, \dots, j_k such that either $i_{t+1} = i_t$ or $i_{t+1} = i_t + 1$ (similarly for j), by minimizing the distance between matched turning angle points:

$$D = \sum_{t=1, \dots, k} |i_t - j_t| \quad (3)$$

For example, the pairing sequence may begin as (1,1), (1,2), (1,3), (2,4), (3,4), etc. D is then used as our measure of similarity.

To speed up the dynamic programming, we allow the two sequences i and j to differ by some maximum amount B . Therefore, out of the full $4k \times 4k$ dynamic programming array, only the central diagonal band of width $2 * B + 1$ is considered. For our experiments, only one resolution ($k = 16$) was used with $B = 8$ for a band width of 17 in a dynamic programming matrix of size 64×64 .

4.4 Sign of curvature

In the curvature sign method, the differences between the sign of the curvature of two shapes determine the similarity. To compute this value, we first perform a Q-spline fit to the boundary of the object, as described above. The curvature of each spline is then sampled at a rate of four times the number of control points. Since the splines are cubic, sampling at four times the number of control points ensures that we will have a sample point for each inflection of curvature around the spline. These curvature points are then thresholded to remove some of the noise associated with the original spline fits. Any points with values greater than the threshold (7 in our experiments) are considered to be points of positive curvature (+1), any with values below the negative of the threshold are considered to be points of negative curvature (-1), and all points in between are considered to have no curvature (0). At this point, we have a listing of +1 / -1 / 0 values for each of the $4 \times k$ sample points. These sign of curvature values are then smoothed with a low-pass filter ($\rho = 4$) to allow for some elasticity in the positions of the sign of curvature values around the perimeter of the shape. All of this processing is done at database population time. The curvature sign points are compressed and saved as a feature set for query time.

At query time, the filtered curvature sign points (A_i and B_i) for two splines with k control points are used to compute the distance D , which is the difference in absolute values of each of the points:

$$D = \sum_{i=0}^k |A_i - B_i| \quad (4)$$

As before, we need to allow for a slight circular rotation s of the control points to solve fitting problems:

$$D = \min_{s \in [-r, r]} \sum_{i=0}^k |A_i - B_{\text{mod}(i+s, k)}| \quad (5)$$

where s is the shift of control points between r and $-r$. For our comparison experiments, the rotation of the control points was limited by $r = 2$. This distance is computed for all resolutions under consideration, and each resolution may have an associated weight attached to that term. For our sign of curvature experiments, we considered two lower resolutions ($k = 8$ and $k = 16$) than the other experiments, since only the broad outlines and the points of high inflection are needed to determine the shape. Because the distance D is not normalized with respect to the number of control points, we weight the $k=8$ case to be four times that of the $k=16$ case.

4.5 Modified Hausdorff distance

Our Hausdorff distance algorithm is different from other Hausdorff similarity measures.^{4,5} Our algorithm is based not on the distances between all points in a rasterized mask, but on distances between the rasterized Q-spline control points. This allows for the flexibility of the Hausdorff approach without the computational expense. At query time, the spline control points for the first shape undergo a least-squares fit to the control points of the second shape. These transformed control points are then rasterized into a square grid of size g . The Hausdorff distance D_i for each of the k control points is computed:

$$D_i = \min_{j \in [0, k-1]} \|c_1^i - c_2^j\| \quad (6)$$

where $\|a - b\|$ gives the Euclidean distance between point a and point b , and c_1 and c_2 are the lists of control points for the first and second shapes, respectively. The list of distances for the set of control points ($D_i : i \in [0, k - 1]$) is sorted in increasing order. The Hausdorff distance is the $(r * k)$ element of that list, where r is a threshold rank to discredit outliers. For our experiments, we examined a resolution of $k = 32$ control points, a grid of size $g = 40$, and a threshold rank of $r = .9$.

5 PERFORMANCE MEASURES

Performance measures for text based information retrieval have been studied^{12,13} and some of the methods can be adapted to image content based retrieval.

In order to obtain some ranking of the importance of the objects selected by our human subjects, a simple weighting scheme was devised. Initially, all objects start with zero weight. Each time a human selects a given object, the weight of that object for that particular test shape is incremented by one. Thus, the weight of an object (w_i) is equal to the number of humans who selected it. In this way, shapes that every subject picked have a greater weighting on our results than did those objects which were chosen only once.

All of the above shape methods were computed for the twenty test shapes against an image database of 1415 objects, consisting of Canadian trademarks, solid airplanes, and a variety of clip-art objects. The query methods all produce a sorted list of images in decreasing similarity order ($q_i : i \in [1, 1415]$), but only the top twenty will be considered. We then computed a number of performance measures based on those 20 results. The total number of hits (h) is simply the number of objects in the top twenty which had non-zero weights. The weighted number of hits (h_{wt}) was defined as the sum of the weights of the twenty objects that were recorded.

$$h_{wt} = \sum_{i=1}^{20} weight[q_i] \quad (7)$$

where $weight[q_i]$ is the weight of the i^{th} object returned by the query. Recall that the weight is equal to the number of human subjects who selected that object as a match for the query object. We also computed the average position of the hits (A), which is simply the sum of the positions of objects that had non-zero weight divided by the number of hits:

$$A = \sum_{i=1}^{20} \frac{x_i}{h} \quad \text{where } x_i = \begin{cases} i & \text{if } weight[q_i] \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

A weighted average position (A_{wt}) was also determined for each query. The weighted average position simply accounts for the fact that shapes with higher weights should be clustered closer to the top of our list.

$$A_{wt} = \sum_{i=1}^{20} \frac{i \times weight[q_i]}{h_{wt}} \quad (9)$$

Note that it is not proper to compare either the average position (A) or the weighted average position (A_{wt}) unless both queries have the same number of hits. A query that has a very low average position because it hits only once is not as good as a query which recovers seven hits in the top ten. The last statistic generated for each query is the percentage of weighted hits achieved (P_{wt}), which is simply the weighted hits gained by the query divided by the total number of selections made by humans. This gives the percentage of the “weight points” that the query achieved out of the total number of points available:

$$P_{wt} = \frac{h_{wt}}{\sum_{i=1}^{1415} weight[q_i]} \quad (10)$$

We will not expect P_{wt} to approach 100% for any of our computational methods because of the high degree of variance in the human data.

6 RESULTS AND DISCUSSION

We determined the “winner” for each query shape to be the method which produced the largest weighted hits. In the case of a tie, the method with the lowest weighted average position was declared the winner. Winners for query shapes #1 (the rectangle), #2 (the square), and #4 (the circle) were computed, but because so many database images matched these query shapes, we will discuss their results separately. Of the seventeen remaining test shapes, turning angle won 7, algebraic moments won 3, modified Hausdorff won 3, sign of curvature won 1, and each of the spline control point matching methods won 1. These results are summarized in Figure 3. (To eliminate ambiguity, we will label the parametric curve method as D^0 for parametric curve with no derivative terms, D^{01} for parametric curve with only first derivative terms, and D^{012} for parametric curve with first and second derivative terms.)

Method	# Wins	Wins on Objects
Moments	3	arch, square plane, cat
D^0	1	swan
D^{01}	1	cup
D^{012}	1	pitcher
Turning Angle	7	child, ellipse, goblet, triangular plane, triangle, letter S, letter T
Sign of Curvature	1	fish
Hausdorff	3	snake, monument, sideview plane

Figure 3. Wins attributed to each computational method

However, the number of wins for a method is not a singular measure of success since none of the methods performed well on all twenty of the test shapes, and in many cases the “winner” was followed by a close second. For a better picture, it is necessary to examine the graphs of the percentage of weighted hits achieved for each query shape (see Figure 5). The winner in each graph will have the highest percentage of weighted hits. By examining Figure 5, it is evident that there is no clear winner for many of the test queries. For example, the turning angle method won for the letter T, but the race was almost too close to call.

For the circle, rectangle, and square test shapes, we relied on a visual inspection of the query results to verify that each method was giving only objects that were visually well matched. The results are summarized in Figure 4.

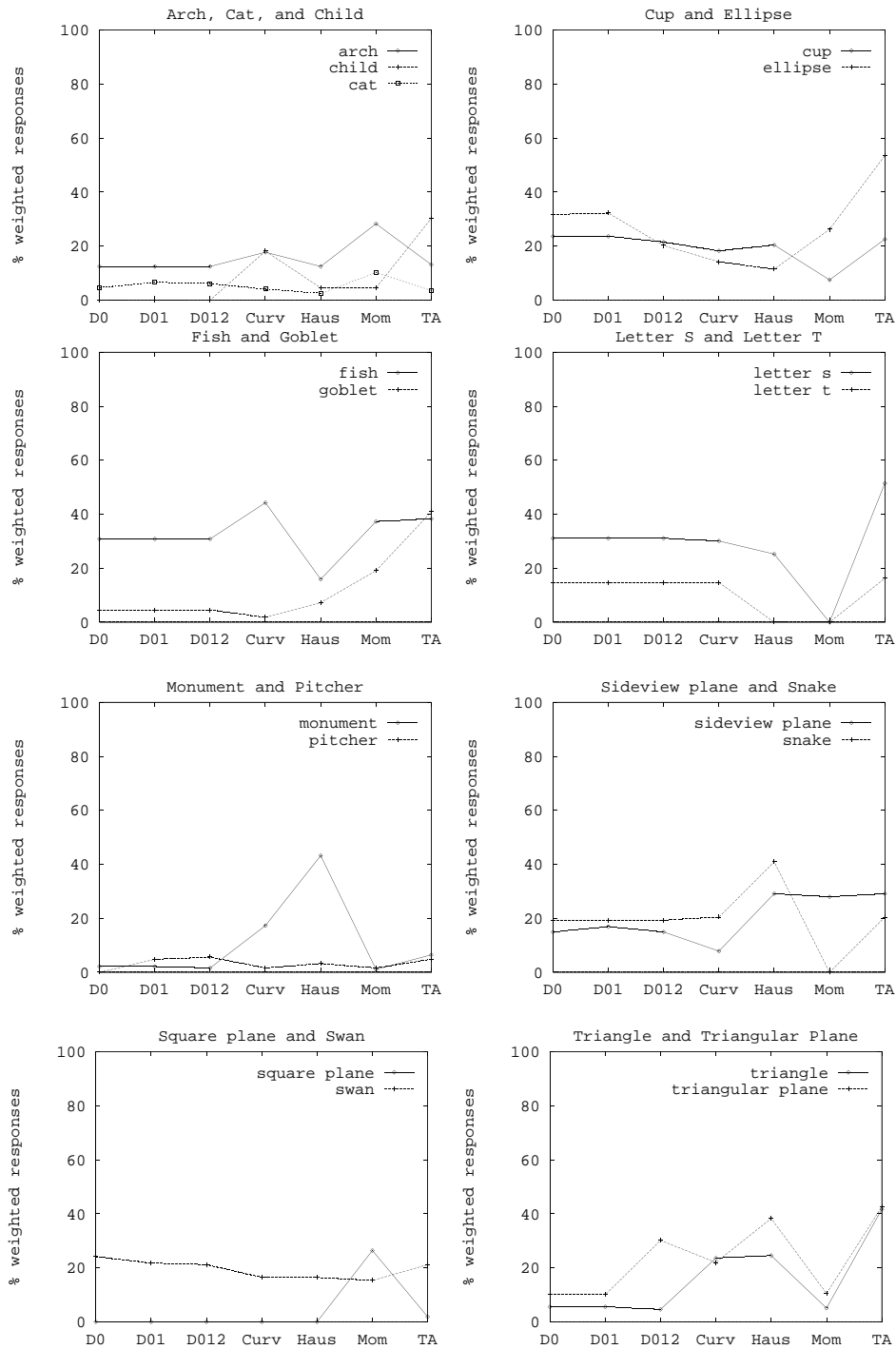


Figure 5: Percentages of weighted human responses achieved vs computational methods for each of the 20 query objects. The seven computational methods are parametric curve distance (D0), parametric curve distance with first derivatives (D01), parametric curve distance with first and second derivatives (D012), sign of curvature (Curv), modified Hausdorff distance (Haus), algebraic moments (Mom), and turning angle (TA).

Method	Rectangle	Circle	Square
Moments	+	-	0
D^0	0	-	-
D^{01}	0	-	--
D^{012}	+	-	--
Turning Angle	0	0	0
Sign of Curvature	0	0	0
Hausdorff	--	0	-

Figure 4. Performance of each computational method on query objects #1 (Rectangle), #4 (Circle), and #2 (Square). A plus sign (+) indicates that all matches were excellent, a zero (0) indicates that all matches were adequate, a single minus sign (-) indicates that one or two results did not match, and a double minus sign (--) indicates that between two and five shapes did not match.

From these visual inspections and the results of the other 17 query objects, it is evident that turning angle, sign of curvature, and algebraic moments most closely matched human performance on the similarity matching task.

An evaluation of the cases where there were clear winners and losers will give us some insight to the relative merits of each technique. First, it is interesting to note that the first and second derivative information applied to the parametric curve distance method has very little effect. The results for D^0 , D^{01} , and D^{012} are virtually identical on all of the query objects except for the triangular plane (in which case D^{012} is superior) and the ellipse (in which case D^{012} falls below the others). Additional information on the first and second derivatives does not seem to help for these query shapes. When one of these three methods won for a shape, it was only by a slight margin over the other two. All of the parametric curve distance techniques seemed to fair poorly on our test queries.

The moment based method performed either very well or very poorly on each of the query objects. Moments was the only technique to match the square plane, and won convincingly on the arch. However, for the snake, triangular plane, the letter T, and the letter S, moments matched virtually none of the human responses. As could be expected, moments tended to work very well on objects where the distribution of the pixels, and not the outline of the shape, were important (as in the arch or the square plane). On objects where the configuration of the outline was important, as in the difference between the letter S and the snake, moments performed poorly.

The sign of curvature technique worked very well on shapes that were defined by protrusions from a central mass. For example, on the child shape, there are 5 protrusions from the central body. In the fish shape, there are three fins and a tail. Sign of curvature performed very well on shapes of this type, but was not accurate when there were no protrusions or when there was no central body. Sign of curvature failed on shapes like the letter T and the square plane, where most of the mass lies outside the center. Sign of curvature also performed very poorly on the ellipse, which may have been the result of a bias in the thresholding.

The modified Hausdorff distance method performed very well on shapes that were very close to the database images. On the monument and the snake, Hausdorff was clearly the winner. On both these shapes, the query object was very similar to only a few objects in the database. Hausdorff was very accurate in finding these exact images. However, for query shapes that were not so exact, Hausdorff performed quite poorly. On queries where some greater amount of flexibility was required, such as the fish or the ellipse, Hausdorff was unable to generalize to other shapes. This may have been caused by setting the threshold rank r too high. However, the modified Hausdorff method failed to match even simple squares and rectangles at this level because the thresholding rank allowed for too many shapes to be considered "similar."

The turning angle method seemed to provide the best overall results. Turning angle was clearly the winner on five of the seven wins attributed to it, and performed better than average on almost all other queries. Turning angle excelled in shapes that required the flexibility of parts of the contour which dynamic programming provides.

7 CONCLUSION

Retrieval by shape is one of the most challenging aspects of content based image retrieval. The available method for evaluating a shape matching technique is comparison against human perceptual judgments. In this paper, we have presented the results of a few computational shape matching algorithms against the perceptual judgments of 40 subjects on a large database of images. There are many other algorithm choices and many other parameter choices for these algorithms that should be evaluated. However, such research is impossible without a standardized database of shapes and the results of psychophysical comparison experiments on that database. This paper has demonstrated the feasibility of such an undertaking in a reasonable period of time. Future efforts should be directed towards development of a more complete shape database and more accurate psychophysical results.

8 ACKNOWLEDGMENTS

This research was performed during internship with the Almaden Research Center, IBM Research Division, San Jose, CA. The authors wish to thank Wayne Niblack, Harpreet Sawhney, Jim Hafner, Chitra Dorai and Byron Dom for their helpful comments and discussions.

9 REFERENCES

- [1] David Mumford. Mathematical theories of shape: Do they model perception? In *Geometric Methods in Computer Vision*, volume 1570, pages 2–10. SPIE, SPIE, 1991. Volume 1570.
- [2] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, pages 179–187, February 1962.
- [3] Gabriel Taubin and David B. Cooper. Recognition and positioning of rigid objects using algebraic moment invariants. In *Geometric Methods in Computer Vision*, volume 1570, pages 175–186. SPIE, SPIE, 1991. Volume 1570.
- [4] D. Huttenlocher, W. Rucklidge, and G. Klanderman. Comparing images using the Hausdorff distance under translation. *IEEE*, pages 654–656, June 1992.
- [5] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. *IEEE*, pages 705–706, 1993.
- [6] Esther M. Arkin, L. Paul Chew, Daniel P. Huttenlocher, Klara Kedem, and Joseph S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on PAMI*, 13(3):209–216, March 1991.
- [7] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture, and shape. In *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science & Technology, Conference 1908, Storage and Retrieval for Image and Video Databases*, February 1993.
- [8] Myron Flickner, James Hafner, Eduardo Rodriguez, and Jorge Sanz. Periodic quasi-orthogonal spline bases and applications to least-squares curve fitting of digital images. Technical Report RJ 9420, IBM Almaden Research Center, July 1992.
- [9] H. S. Sawhney and A. R. Hanson. Trackability as a cue for potential obstacle identification and 3d description. *IJCV*, pages 237–265, 1993.
- [10] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

- [11] Ross McConnell, Ronald Kwok, John C. Curlander, W. Kober, and Shirley S. Pang. $\psi - s$ correlation and dynamic time warping: Two methods for tracking ice floes in SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 29(6):1004–1012, November 1991.
- [12] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- [13] Rosanne Price, Tat-Seng Chua, and Suliman Al-Hawamdeh. Applying relevance feedback to a photo archival system. *Journal of Information Science*, 18:203–215, 1992.