IJSEAT International Journal of Science Engineering and Advance Technology

A Comparative Study on Shape Reorganization

S.Suresh[1], Dr. R Siva Ram Prasad
[1]Research Scholar , [2], Research Director,
Dept. of Computer Science and Engineering Acharya Nagarjuna University, A.P. -India

Abstract- This paper proposes a new mechanism for identifying two-dimensional shapes called the SKS algorithm and compares it with three other state-of-art methods in detail. These include the Hu Moments, CSS matching and Shape context. The algorithm uses the philosophy of evidence accumulation equal to generalized Hough transform and is highly parallel in nature. The performance of each algorithm is evaluated under affine transforms - translation, rotation in the plane, scale (zoom) and also under partial occlusion.

1 Introduction

This paper introduces a new approach to two dimensional shape recognition based on the evidence accumulation philosophy of the generalized Hough Transform[10] and compares the performance with other state-of-the art methods. The shapes considered will be restricted to simple shapes without holes. We will also ignore the interior of regions, and all related interior information such as texture, color, and shading. We will evaluate performance of algorithms under scale (zoom), rotation in the plane and translation. We will also evaluate the performance of the algorithms under partial occlusion, but will not attempt to simultaneously recognize the occluding and occluded shape. Basically, we are recognizing partially occluded silhouettes. A survey of the current shape recognition algorithms can be found in [22]. In this paper, we survey four different algorithms in detail. In section 2, we survey region descriptors such as area, thinness, etc, as well as the Hu invariant moments. We show that a variety of statistical methods can be utilized to make use of such features. In section 3, we look at the correspondence-based Shape Context algorithm. In section 4, we analyze the contour-based CSS shape matching algorithm, which is also a MPEG-7 Standard. In section 5, we introduce a new shape matching algorithm called the SKS algorithm, which is invariant to translation, rotation, scale and robust against partial occlusions. In section 6, we experimentally compare the performance of these algorithms followed by the discussion of the results and the conclusion. One collection of methods not included in the comparisons is the work measuring the distance between two shapes as the length of a geodesic in shape space [13]. These are omitted because at the current state of the art, all these methods require non-occluded boundaries. This paper is intended to be slightly more tutorial in nature than is common in journal papers. We assume a reasonable background in vector calculus, but will provide a bit more explanation of image recognition topics than a less pedagogical paper might.

1.1 Notation

One may assume a boundary to be either continuous, or parameterized (typically) by arc length, or discrete, and parameterized by an index, say i. If the continuous representation is used, in order to perform computation, one must eventually use discretization. Therefore, we will usually go directly to the discrete form, unless the representation used by the original author demands a continuous derivation. Our objective is to compare two boundaries, iC and jC. We use the superscript on the left to denote which boundary is being referred to. In the discrete form, a boundary, say boundary i, is an ordered set of points in the plane, $iC = \{{}^iC_1, {}^iC_2, . . . , {}^iCN\}$, where ${}^iC_k = [{}^ix_k, {}^iy_k]T$, using the usual 2-vector notation for points in the plane. For convenience, we assume each boundary has N points in its perimeter unless otherwise explicitly mentioned. Boundaries with different length perimeters can be easily normalized. When we use the continuous form, we will denote the ith curve as iC(s), using s to denote arc length.

2 Shape features

In this section, we consider an approach to shape recognition which has been prevalent for many years, the “statistical pattern recognition” approach. We make a set of measurements which independently characterize some aspect of the shape. Hopefully, we have a large collection of examples, so we may then characterize the shape statistically. Suppose, for example, the mission of the project is to distinguish between sharks and sting rays, as illustrated in Figure 1. easurements may include properties of a region such as area, perimeter, aspect ratio, eigenvalues, convex discrepancy, and various central moments. Though the computation of such features can be challenging, we do not discuss the actual computational process here, but rather refer the reader to texts on computational geometry[17],[18]. Before we can consider the use of simple geometric features, we must discuss briefly how such features might be used, which is, in turn, a pattern recognition problem.

2.1 Pattern Recognition

In this section, we restrict the problem to "to which shape in the data base is the observed shape most similar?" Thus, we are not classifying the shape as a shark or a sting ray, and we do not assume that we have any statistical information about the properties of the typical shark or sting ray. This restriction eliminates most of the "statistical" part of statistical pattern recognition, and leaves us with the problem of finding, not the class to which an observation belongs, but which prototype it most resembles. We expound on this in the next subsection.

2.1.1 Pattern Classification

In a traditional pattern classification problem, we have several classes which are characterized by their statistics. That is, we assume we have, in the past, 3

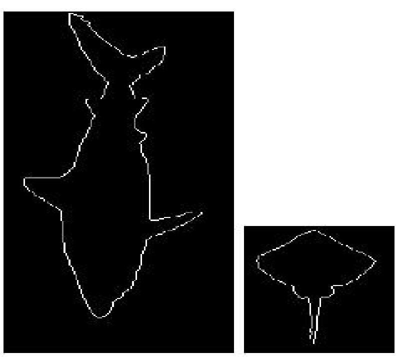


Figure 1: Shark versus Sting Rays

observed more than one example from each class, and are therefore able to characterize each class by some properties of the set of past observations (called a training set). A training set may be characterized by a parametric model (often a multivariate normal), in which the entire set is equivalently represented by small set of parameters such as a mean vector and covariance matrix. In figure 2, we illustrate a situation in which two measurements have been made, of length and width of a region, and examples of pictures of sharks and sting ray have been measured. Each circle represents one example shark and each square, a sting ray. We first note they are all different, presumably because either the examples actually vary, or the imaging system has noise or other corruption.

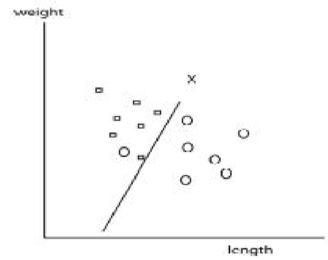


Figure 2: Classification Example

Second, we observe it is impossible to draw a simple straight line which separates the two classes. One good line has been drawn, but a decision based on this line will not be perfect. Finally, the figure illustrates an unknown, denoted by "x". Should we classify it as a shark or a sting ray? The correct decision depends on the underlying distribution of the data from which the samples were drawn, and is beyond the scope of this paper. But, if we have a good statistical model, we can develop a useful algorithm.

In some cases, a parametric representation is not appropriate, and it is necessary to choose a nonparametric model, and use density estimation. In the nonparametric case, one of the more popular representations is the K-Nearest neighbor (K-NN) form. In that case, it turns out that the best decision is to simply assign the unknown observation to the class in which the K nearest neighbors of the observation belong. The extreme case of K-NN, of course, occurs when the unknown observation is assigned to the class of its nearest neighbor, and the K-NN algorithm simplifies to the 1-NN algorithm. One must, of course, define what "nearest" means. For clarity of explanation,we will do an example using the Hu "invariant moments." The Hu moments [11] are seven numbers which can be shown to represent the shape of a region in the 2-D image, and which are invariant to rotation, translation, and scale (zoom) changes. A simple moment of a region is

$$m_{pq} = \sum x^p y^q f(x, y)$$

where the summation is taken over all points (x, y) in the region, and f(x, y) is the brightness at a particular point. If we assume the region is uniform in gray value and that gray value is rbitrarily set to one inside the region and zero outside, the area of the region is then m00 and the center of gravity is

$$m_x = \frac{m_{10}}{m_{00}} \quad m_y = \frac{m_{01}}{m_{00}}$$

Invariance to translation is achieved by referencing all points to the CG, producing the "central moments."

$$\mu_{pq} = \sum (x - m_x)^p (y - m_y)^q f(x, y)$$

By taking into account rotation and zoom, we can continue this sort of derivation and can define the normalized central moments by $\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma}$, where $\gamma = \frac{p+q}{2} + 1$. Hu's original paper[11] is the best reference for details. Finally, the first three invariant moments turn out to be

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3)$$

The classification process is as follows: First, for each element of the data base, say, element i, calculate the feature vector ${}^iH = [{}^ih_1, {}^ih_2, \ldots, {}^ih_7]^T$. Make the same measurements on the unknown region $O = [O_1 O_2, \ldots, O_7]^T$. Then, decide that the observation O belongs to class i iff $\|O - {}^iH\| < \|O - {}^jH\| \; \forall j \neq i$. Here $\| \cdot \|$ denotes some norm. Usually, this is the 2-(Euclidean) norm. This philosophy is usually referred to as a minimum distance classifier, but in the case of singleton training sets, it is more correct to call it a 1-NN classifier. We thus may use shape measures, such as these to match the shape of one region to another. Surprisingly, a simple classifier based on the Hu moments does not work as well as one might expect. This will be discussed in section

3 Shape Context

The Shape Context [5] [6] descriptor for matching 2D shapes was introduced by Belongie et al. in [4]. Given a contour iC, assumed sampled, $iC = \{{}^iC_1, {}^iC_2, \ldots {}^iC_N\}$, let $P \in {}^iC$. For any element of ${}^iC$, we can compute the vector from that point to P, and thus can construct an ordered set of such vectors, ${}^i\Delta_P = \{P - {}^iC_1, P - {}^iC_2, \ldots, P - {}^iC_{Ni}\}$. The elements of ${}^i\Delta_P$ are converted to log-polar coordinates and coarsely1 quantized, and then a two-dimensional histogram constructed, defining the shape context of point P:

$$h(P, \theta, r) = \sum_j \delta(\theta - \theta_j, r - r_j) \qquad j = 1, \ldots, N,$$

(4)

where the δ denotes the Kronecker delta,

$$\delta(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{otherwise} \end{cases}$$

The histogram is simply a count of how many times the particular angle, log-distance pair occurs on this contour. Since the histogram is of a fixed size, (in this case, 60 bins) we may index it with a single index and denote it by h(P, k). We refer to h(P, k) as the shape context of point P in curve iC. Clearly, the choice of a reasonable resolution for the histogram depends on N.

The shape context for a particular contour is the collection of the shape contexts of all the points in that contour. In the absence of occlusion, the shape context may be made invariant to similarity transforms [4]. Invariance to translation is automatic, since the Shape Context only uses relative distances and angles. Invariance to scale is accomplished by dividing all distances ${}^i\Delta_P$ by the median of all such distances in the shape. Since a log-polar system is used, the calculation used is actually a subtraction of logs.

To use the shape context in matching, we will use it to find a measure which will characterize how well ${}^iC$ matches ${}^jC$. Let P denote a point on curve ${}^iC$ and let $Q \in {}^jC$. Denote the corresponding histograms as h(P, k) and h(Q, k). For any particular P and Q, we may match their shape contexts by matching individual points along the two curves. Since P and Q denote points along specific curves, they may be thought of as indices. Since P indexes curve ${}^iC$, and Q indexes ${}^jC$, we can construct a matrix of matching costs. Specifically, if we believe point P in curve ${}^iC$ is the same as point Q in curve ${}^jC$, we may define a cost of assigning P to correspond to Q, This "assignment cost" is

$$\gamma_{PQ} = \frac{1}{2} \sum_{k=i}^{K} \frac{(h(P,k) - h(Q,k))^2}{h(P,k) + h(Q,k)} \quad (5)$$

All these assignment costs may be calculated and entered into a matrix

$$\Gamma_{ij} = [\gamma_{PQ}] \quad P \in {}^iC, Q \in {}^jC \quad (6)$$

The matrix Γ is a representation of the cost of matching shape i to shape j. if $N_i = N_j$, Γ_{ij} is square, if not, Γ ij may be made square by adding "dummy" nodes with constant match cost. To extract a scalar measure of this cost, we must find the mapping f : ^iC ^jC which is a bijection (1:1 and onto), and which minimizes the total cost. That is, to the P^{th} element of ^iC we must find exactly one element of ^jC to assign it to. This is referred to as the "linear assignment problem" (LAP). This problem may be stated in terms of the matrix Γ ij , by observing that, for each row, we must choose one and only one column for the match, and that column must, in turn match no other rows. The linear assignment problem is one of a large set of linear programming problems, the worst of which is exponential in complexity, which may be solved using the simplex algorithm. However, the special case of the linear assignment problem may be solved in $O(n^3)$, using several strategies, the most well-known of which is the "Hungarian algorithm" [15]. We used the implementation by Jonker and Volgenant [12]. Assuming that unknown curve ^iC is being matched to a data base of models, B, The solution to the LAP produces a number, $\Gamma_{ij} = \min_j(\Gamma_{ij})$, ^iC € B , which is the best assignment possible for these two curves. Thus the index of the best match m = $\mathrm{argmin}_j(\Gamma_{ij})$.

4 Curvature Scale Space Matching

Curvature Scale Space(CSS) is a shape representation method introduced by Mokhtarian and Macworth[16]. CSS has also been adopted in the MPEG-7 standard as a contour shape descriptor[21]. The CSS representation is a multi-scale organization of curvature zero crossing points of a planar curve. The authors define the representation in the continuous domain but sample it later. Consider a curve parametrized by arc length s, s €[0, 1],

$$C(s) = [x(s), y(s)]^T \quad (7)$$

The curvature of such a curve is given by:

$$\kappa(s) = \frac{(\dot{x}(s)\ddot{y}(s) - \ddot{x}(s)\dot{y}(s))}{(\dot{x}(s)^2 + \dot{y}(s)^2)^{3/2}} \quad (8)$$

The above contour can be successively blurred by convolving it with a 1-d Gaussian kernel of width σ, where the scale(σ) is increased at each level of blurring. Let this blurred version of the contour be given by C: $\Re \times \Re$ $\Re \times \Re$

$$C(s,\sigma) = (x(s) * G(s,\sigma), y(s) * G(s,\sigma)) \quad (9)$$

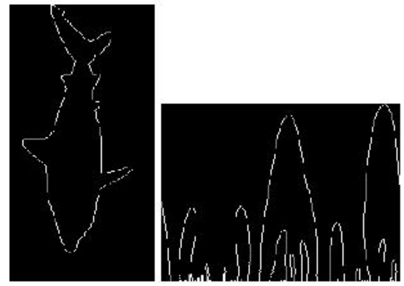


Figure 3: A fish contour and the corresponding CSS image

The first and second derivatives of the curve at each scale σ can be estimated as follows:

$$x_s(s,\sigma) = x(s) * G_s(s,\sigma) \quad (10)$$

$$y_s(s,\sigma) = y(s) * G_s(s,\sigma) \quad (11)$$

$$x_{ss}(s,\sigma) = x(s) * G_{ss}(s,\sigma) \quad (12)$$

$$y_{ss}(s,\sigma) = x(s) * G_{ss}(s,\sigma) \quad (13)$$

where the subscript denotes partial differentiation. The curvature at each scale σ can be estimated as:

$$\kappa(s,\sigma) = \frac{(x_s(s,\sigma)y_{ss}(s,\sigma) - x_{ss}(s,\sigma)y_s(s,\sigma)}{(x_s(s,\sigma)^2 + y_s(s,\sigma)^2)^{3/2}}$$

(14)

The curvature zero-crossings are the points on the contour where the curvature changes sign. The CSS representation computes the points of curvature zero-crossings at each scale and represents them in the (s, σ) plane. This is shown in figure 3 with arc length(s) is along the horizontal axis and the scale(σ) is along the vertical axis. This is called the CSS image. The set of maxima of the CSS image is used as the shape signature for the given contour. Maxima with low scale values in the CSS image are related to the noise in the curve and are ignored. Following Mokhtarian[2], we assume anything below $\frac{\sigma_{max}}{6}$ is considered to be noise and is not used in the matching process, where _max is the highest maximum in the CSS image. CSS descriptors are translation-invariant because of the use of curvature. Scale invariance is obtained by re-sampling the curve to a fixed number2 of boundary points. Rotation and starting point changes cause circular shifts in the CSS image and are compensated for during the matching process.

4.1 CSS Matching

The CSS matching algorithm is described in detail in [16] [2]. Every object in the database is represented by the set {(s, σ)} of pairs of the maxima in the CSS image. The matching process involves finding correspondences between the maxima of the two contours. We will refer to the two images as the target and the model.

Let $M_{target} = \{(t_1, \gamma_1), (t_2, \gamma_2), ..., (t_L, \gamma_L)\}$ be the maxima of the target, parametrized by arclength t and arranged in descending order of scale γ. Let M model = $\{(s_1, \sigma_1), (s2, \sigma 2), ..., (s_N, \sigma_N)\}$ be the maxima of the model, parametrized by arclength s and arranged in descending order of scale . The matching algorithm is as follows:

1. The highest maximum (with respect to the scale) in the target and the model CSS image are used to find the CSS shift parameter.

$\alpha = s_1 - t_1$ (15)

This will compensate for any changes in starting point and orientation of the contour. Two lists are initialized. One with the maximum pair (t_1, Υ_1) from the target and another with the maximum pair (s_1, σ_1) from the model. The cost for the match is initialized as:

$MC = |\sigma 1 - \Upsilon 1|$ (16)

2. Now, for the next highest maximum $(t_2, \Upsilon 2)$ in the target, we apply the CSS shift parameter(σ) calculated earlier. We find the closest maximum in the model to this shifted target maximum which is not present in the model list.

$$(s_m, \sigma_m) = \arg\min_{i=1}^{N} \|(s_i, \sigma_i) - (t_2 + \alpha, \gamma_2)\|, (s_i, \sigma_i) \in M_{model} \quad (17)$$

The two maxima are added to their respective lists. The cost of the match is updated as follows:

$$MC = \begin{cases} MC + \|(s_m, \sigma_m) - (t_2 + \alpha, \gamma_2)\| & \text{if } \|t_2 + \alpha - s_m\| < T \\ MC + \|\gamma_2\| & \text{if } \|t_2 - s_m\| > T \\ MC + \|\gamma_2\| & \text{No more maxima in model} \end{cases}$$

where T is a user defined threshold.

3. Repeat 2 for all the elements in the target.

4. Calculate the CSS shift parameter using the second highest maximum (t2, 2) in the target and the model (s2, _2) and also for maxima in the model which are close to the highest maximum of the target(within 80% of the maximum scale value). Repeat 1-3 for using these CSS shift parameters and calculate the match cost.

5. Repeat 1-4 by interchanging the place of the target and the model.

6. The lowest cost from all these matches is taken as the best match value.

A modification of this method described in [2] involves using global parameters of a shape like eccentricity and circularity to eliminate certain shapes before the matching process and also overcomes some of the problems of the original method like shallow concavities.

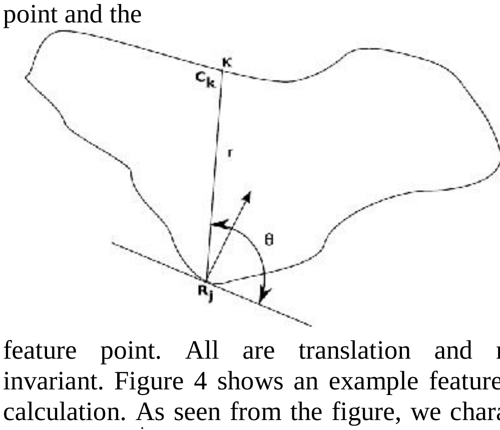
5 SKS Algorithm

The SKS algorithm is a robust 2-d shape recognition algorithm which uses the philosophy of evidence accumulation, similar to the Hough transform. Such an approach uses considerable memory but performs most computations using simple arithmetic and look up. The algorithm is invariant to translation, rotation, scale(zoom) and robust against partial occlusion. The algorithm is also highly parallel in nature which favors neural implementations.

5.1 Model Building

Consider a digital contour $iC = \{^iC_1, {}_iC_2, ...^iC_N\}$ with N points. The first step in the model building process is to pick a set of J, $J \leq N$ reference points on the contour ^iR_j , ^iR_j € ^iC, j = 1 . . . J. We use points of high curvature as the reference points, although other choices, such as inflection points[20], are possible. Curvature is determined using Digital Straight Segments(DSS)[9],[14] which has been shown[7],[8] to be more reliable and accurate when compared to other techniques. In the absence of occlusion, only one reference point is required. At each reference point(^iR_j), we use the Frenet Frame (the tangent and the normal at that point) to establish a rotationally invariant reference coordinate system with respect to all the other points on the contour. We now build the model with respect to each of these reference points.

Figure 4: Feature Vectors, k, curvature at the point; θ, , angle between the vector from the reference point to the feature point and the tangent frame of the reference point; r, the distance between the reference point and the



feature point. All are translation and rotation invariant. Figure 4 shows an example feature vector calculation. As seen from the figure, we characterize each point ^iC_k with respect to the frame at the reference point ^iR_j . Example features of point ^iC_k, as shown in figure 4, could include the distance to the reference point, the polar coordinates(r, θ), θ€ [0, 2π) of the point with respect to the reference (^iR_j), and the curvature(k) at that point. Translation invariance is achieved automatically by picking the reference points on the contour since the coordinate system origin is moved to the reference point. The feature vectors are normalized for scale using the procedure described in section 5.3. Furthermore, all the three feature vectors are invariant to rotation.

Let $v_{jk} = (r_{jk}, \theta_{jk}, K_{jk})$ be the feature vector at an arbitrary point ${}^{i}C_k$ with respect to the reference point ${}^{i}R_j$. The model for curve i with respect to the reference point ${}^{i}R_j$ is given by:

$$ {}^{i}M_j(\mathbf{v}) = \max_{k=1}^{N} \exp\left(\frac{-||\mathbf{v} - \mathbf{v}_{jk}||^2}{2\sigma^2}\right) \qquad (18)$$

where _ will be determined later.

The model function(${}^{i}M_j(v)$) can be viewed as a function which estimates the presence of a feature(v) in the model. The model function can be pre computed and stored as a look up table which considerably speeds up the matching process.

5.2 Matching

The matching process uses evidence accumulation similar to the generalized Hough transform[3] to determine the similarity between two shapes. Again, consider a digital contour ${}^{1}C = \{{}^{1}C_1, {}^{1}C_2, ..., {}^{1}C_L\}$ with L points, which we wish to match against the data base. We again pick a set of K, $K \le L$ reference points on the contour$\{{}^{1}R_k\}$,${}^{1}R_k$ € ${}^{1}C$, k = 1 . . .K and calculate a feature vector vkl at each point on the contour with respect to each reference point as shown in figure 4. Consider matching the curve ${}^{1}C$ using the k^{th} reference point 1Rk to the model(${}^{2}M_j(v)$) of the curve 2C model built using the j^{th} reference point ${}^{2}R_j$. As match quality, we compute

$$ {}^{12}A_{kj} = \frac{1}{L}\sum_{l=1}^{L} {}^{2}M_j(\mathbf{v}_{kl}) \qquad (19)$$

To determine the best match between a particular model and the target, we consider all pairs of reference points,

$$ {}^{12}A = \max_{k,j} A_{kj} \qquad (20)$$

The matching process is just a simple summation over all the points in the contour and is considerably sped up by storing ${}^{2}M_j(v)$ as a look-up table. The matching process is also robust to partial occlusion as we calculate the maximum of the match between all the reference points. Thus, the absence of some reference points due to occlusion will not affect the match. Sensitivity to partial occlusion is explored experimentally in Section 6. The process of finding the data base entry which best matches a particular contour is then simply finding the match pair which produces the highest A.

5.3 Scale Estimation

Scale is a global characteristic of a shape. In order to maintain robustness to partial occlusion, we must estimate scale using local features. Consider the digital contour iC = $\{{}^{i}C_1, {}^{i}C_2, ..., {}^{i}C_N\}$ with N points. As shown in section 5.1, we use a set of reference points(${}^{i}R_j$), j = 1, ..., J on the contour and the Frenet frames at those points as a rotationally invariant reference coordinate system. The polar coordinates - (r, θ) of a point ${}^{i}C_i$ on the contour are calculated with respect to the reference point ${}^{i}R_j$. The "distance map" with respect to a reference point, ${}_{i}R_j$, is the distance of the furthest point at a particular polar angle ${}^{i}D_j$: $\Re \rightarrow \Re$

$$ {}^{i}D_j(\theta) = \max_{(r,\theta)\in {}^{i}C} r \qquad (21)$$

Let ${}^{2}D_j(\theta)$ and ${}^{1}D_k(\theta)$ be the distance maps of the j^{th} reference point on model(2C) and the k^{th} reference point on the target shape(1C) respectively. The scale at each θ can then be estimated as:

$$ s_{kj}(\theta) = {}^{1}D_k(\theta)/{}^{2}D_j(\theta) \qquad (22)$$

The best estimate for the scale is then taken as the median of all the scales(s_{kj}).

6 Experimental Results

This section compares the performance of the four algorithms - the Hu Moments, Shape Context, CSS matching and the SKS algorithm to similarity transforms (translation, rotation and zoom) and robustness to partial occlusion. Furthermore, we also show the applicability of the algorithms to similarity-based shape retrieval

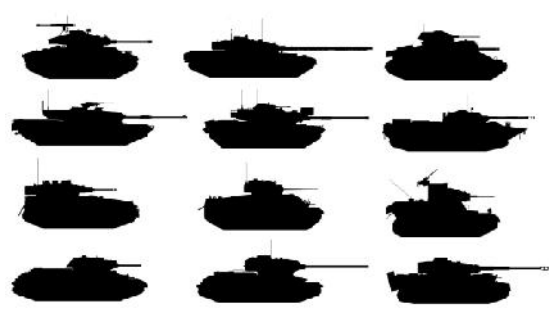


Figure 5: Database of 12 tanks

6.1 Datasets

For the similarity transform tests, we use a collection of 12 similar tank contours as shown in Figure 5. Each tank is further rotated through angles of 15°, 30°, 45°, 60° , 75° and 90°. We also generated tanks scaled to 0.25, 0.5, 0.75, 1.25 and 1.5 times the original. The tanks look very similar and are mostly made up of straight lines and thus, have zero curvature at most points which makes them a slightly harder dataset to work with than the SQUID. For the occlusion and the shape retrieval tests, we used the dataset of 1100 marine animals from the standard SQUID database[1].

6.2 Invariance to Similarity Transforms

In this experiment, we built models using all the tanks (rotated and scaled) and matched each tank contour with every model. The number of correct matches in the top 12 retrieved shapes was determined. Since there are 12 tanks and each tank has 6 rotated and scaled versions of itself, the total number of correct matches is 1728. Table 1 shows the retrieval results of the four algorithms. The results show that the performance of both SKS and shape context are similar with both getting around 99% classification accuracy. The performance of CSS is surprisingly poor, however, the results are consistent with other re-implementations of the algorithm[19].

Algorithm	SKS	Shape Context	Hu Moments	CSS
Correct Retrievals(%)	98.26	99.13	77.60	75.11

Table 1: Invariance to Similarity Transforms

6.3 Robustness to Occlusion

In this experiment, we randomly picked 31 fishes from the SQUID database. These are shown in figure 6. We then partially occluded each fish by retaining 10-90% of the points. To determine a partial occlusion of, say _ percent, the following algorithm was used: Consider a digital contour with points C = $\{C_1, C_2, ..., C_N\}$. Starting at point C_1, K sequential points are chosen such that K/N = _/100. Those K points are removed from the boundary. This produces an occluded boundary with _ percent occlusion. Since it can be reasonably anticipated that some areas of the boundary will be more sensitive to occlusion than others, the starting point was moved from 1 to 2 , then 3, etc. and more occluded boundaries were generated. An example of this is shown in figure 7. The performance reported is the average of all the occlusions of that particular boundary. At each occlusion level, the occluded fish generated using all possible starting points were matched with the unoccluded original set of 31 fish and classified. The results of the occlusion experiment for the four algorithms are shown in figure 8. As it can be seen, the SKS algorithm significantly outperforms the others. Even at 60% occlusion, the classification is essentially perfect.

6.4 Application to Content-based Image Retrieval

In the previous sections, we have shown the robustness of the SKS algorithm to similarity transforms and partial occlusion. This robustness should help in the retrieval of shapes with similar structure to given input shape. Since, there is no universally accepted formulation for measuring shape similarity, we look at some of the top matches for a given query shape. For a given query shape, the top 5 matches based on the match measure, Eq. 20, were retrieved. Figures 9 and 10 shows the top retrievals for the some of the shapes from the Squid database using the four algorithms. In all the examples, it can be seen that the top matches are very similar to the query shape demonstrating the ability of the SKS algorithm in retrieving similar shapes.

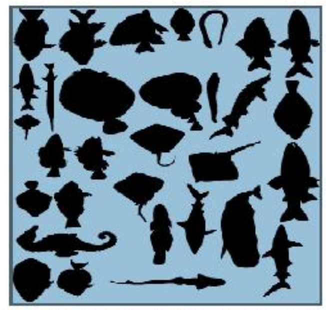
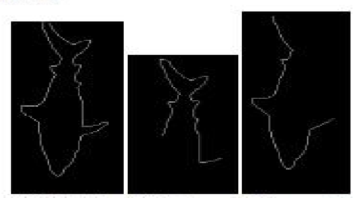


Figure 6. The 31 randomly picked silhouettes from the SQUID database. Used with permission.



(a) Original Contour (b) Contour at 40% occlusion (c) Same contour at 40% occlusion but with a different region occluded

Figure 7: A fish contour occluded by 40% with different occluded regions

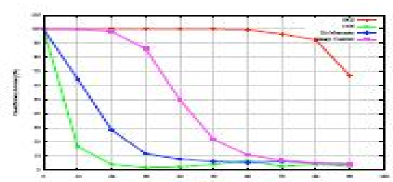


Figure 8: Classification accuracy at various levels of occlusion.

7 Discussion

In the search for a general theory for 2-D shape recognition, there is a tendency among recent authors to seek algorithms which do not demand recognition of specific feature points. For example, in arguing against the use of key points, Belongie et al. [6] say "Not all objects have distinguished key points (think of a circle for instance), and using key points alone sacrifices the shape information available in smooth portions of object contours”. Srivastava[13] agrees, saying “Among the papers that explicitly study shapes, a major limitation in many of them is the use of landmarks to define shapes. Shapes are often encoded by a coarse sampling of the objects boundaries, and the outcome and accuracy of the ensuing shape analysis is heavily dependent on the choices made. In addition, it is usually difficult to automate the selection of these landmarks. A more fundamental approach is to represent the continuous boundaries as curves, and then study their shapes.” Both these highly respected authors (and others) are arguing for a gestalt approach, seeking a representation describing the entire shape which is more than (or more robust than) the sum of its parts. This is a highly admirable goal, and the reference to the circle (or any ellipse for that matter) is particularly relevant. But the strategy breaks when large occlusions occur. If we think of older literature in Computer Vision, we observe the opposite philosophy being supported, based on cognitive psychology. The point was made that humans make substantial use of key points, and figure 11, or similar, was used to make this point. The figure consists of only straight lines, connecting key points–points of high curvature. We propose that both types of operations occur, identification of salient points3 and gestalt shape recognition. In the SKS algorithm, only a single salient point correspondence is required. Although this is still a problem of complexity n2 (where n is the number of salient points, NOT the number of boundary points), it is still less than the complexity of the correspondence problem required for the shape context algorithm. It seems clear, when we humans do “thinking about seeing”, that when boundaries are partially occluded, we move into a “search mode”, presumably using higher cognitive centers, which look for matching saliencies and matching segments. In SKS, the salient points are identified first, but that is a computational convenience, which is not necessarily required for a biological implementation. In conclusion, we propose that a combination of correspondence of salient points with gestalt mechanisms, such as that implemented in the SKS algorithm is a good model for sophisticated shape recognition. Future research demands a study of two things: First, how can the SKS strategy be implemented in a biologically reasonable way, and second, how does the human make the transition from objects represented in the 3D world to a 2D abstraction which allows for fast, yet general algorithms like SKS?

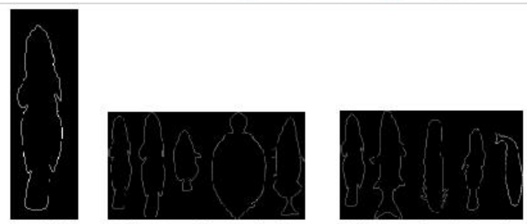
(a) kk732 - Query shape. (b) Top 5 matches in random order for the shape kk732 using CSS.

(c) Top 5 matches in random order for the shape kk732 using Hu Moments.

(d) Top 5 matches in random order for the shape kk732 using Shape Context.

(e) Top 5 matches in random order for the shape kk732 using the SKS algorithm.

Figure 9: Example Queries



(a) kk942 - Query shape. (b) Top 5 matches for the shape kk942 using CSS. (c) Top 5 matches for the shape kk942 using Hu Moments.

(d) Top 5 matches for the shape kk942 using Shape Context. (e) Top 5 matches i for the shape kk942 using the SKS algorithm.

Figure 10: Example Queries

Figure 11: A line drawing, without color or texture information, and consisting of only straight lines connecting salient points.

References

[1]Squid database,“ http:// www.ee.surrey.ac.uk /research/ vssp/imagedb/”.

[2] Sadegh Abbasi, Farzin Mokhtarian, and Josef Kittler. Curvature scale space image in shape similarity retrieval. Multimedia Syst., 7(6):467–476, 1999.

[3] D.H Ballard. Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111–122, 1981.

[4] S. Belongie and J. Malik. Matching with shape context, 2000.
[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts, 2001.
[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE PAMI, 24(4), april 2002.
[7] David Coeurjolly, Serge Miguet, and Laure Tougne. Discrete curvature based on osculating circle estimation. Lecture Notes in Computer Science, 2059:303–312, 2001.
[8] David Coeurjolly and Stina Svensson. Estimation of curvature along curves with application to fibres in 3d images of paper. Lecture Notes in Computer Science, 2749:247–254, 2003.
[9] J.-P. Debled-Rennesson, I. Reveilles. A linear algorithm for segmentation of digital curves. International Journal of Pattern Recognition and Artificial Intelligence, 9(4):635–662, 1995.
[10] P. V. C. Hough. Method and means for recognizing complex patterns. U.S. Patent 3069654, 1962.
[11] M K Hu. Visual pattern recognition by moment invariants. IRE Trans. Information Theory, 1962.
[12] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing, 38, 1987.
[13] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. IEEE Trans. Pattern Anal. Mach. Intelligence, 26(3), 2003.
[14] V.A. Kovalevsky. New definition and fast recognition of digital straight segments and arcs. Pattern Recognition, 1990. Proceedings., 10th International Conference on, 2:31–34, 16-21 Jun 1990.
[15] Harold W. Kuhn. The hungarian method for the assignment problem. Naval Research Logistic Quarterly, 2, 1955.
[16] F Mokhtarian, S Abbasi, and J Kittler. Efficient and robust retrieval by shape through curvature scale space. In Proceedings of the First International Workshop on Image Databases and Multi-Media Search, pages 35–42, Aug 1996.
[17] Joseph O'Rourke. Handbook of Discrete and Computational Geometry.CRC Press, July 1997.
[18] Franco P. Preparata and Michael I. Shamos. Computational Geometry, An Introduction. Springer, 1985.
[19] Remco C. Veltkamp and Longin Jan Latecki. Properties and performance of shape similarity measures. In Data Science and Classification, Studies in Classification, Data Analysis, and Knowledge Organization.Springer Berlin Heidelberg, 2006.
[20] Daniel M. Wuescher and Kim L. Boyer. Robust contour decomposition using a constant curvature criterion. IEEE Transactions and Pattern Analysis and Machine Intelligence, 13(1), January 1991.
[21] D. Zhang and G. Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. Journal of Visual Communication and Image Representation, 14(1):39–57, March 2002.
[22] Dengsheng Zhang and Goujun Lu. Review of shape representation and description techniques. Pattern Recognition, 37(1):1–19, January 2004.