Lecture 4, Part 2: The SKS Algorithm

Wesley Snyder, Ph.D.

UWA, CSSE

NCSU, ECE

The SKS Algorithm

- 2-d shape recognition algorithm which uses the philosophy of evidence accumulation.
- Invariant to translation, rotation, scale(zoom) and robust against partial occlusion.
- Highly parallel in nature which favours neural implementations.

Model Building

A digital contour ${}^iC = \{{}^iC_1, {}^iC_2, ...{}^iC_N\}$. Pick a set of reference points on the contour $\{{}^iR_j\}, {}^iR_j \in {}^iC, j = 1 \ldots J$. (could use just one reference point if no occlusion)

We use points of high curvature – Others possible, such as inflection points[**?**]. Curvature is determined using Digital Straight Segments(DSS)[**?**],[**?**] which has been shown[**?**],[**?**] to be more reliable and accurate.

Model Building

At $({}^{i}R_j)$, use the Frenet Frame to establish a rotationally invariant reference coordinate system with respect to all the other points on the contour. Build the model with repect to each of these reference points.

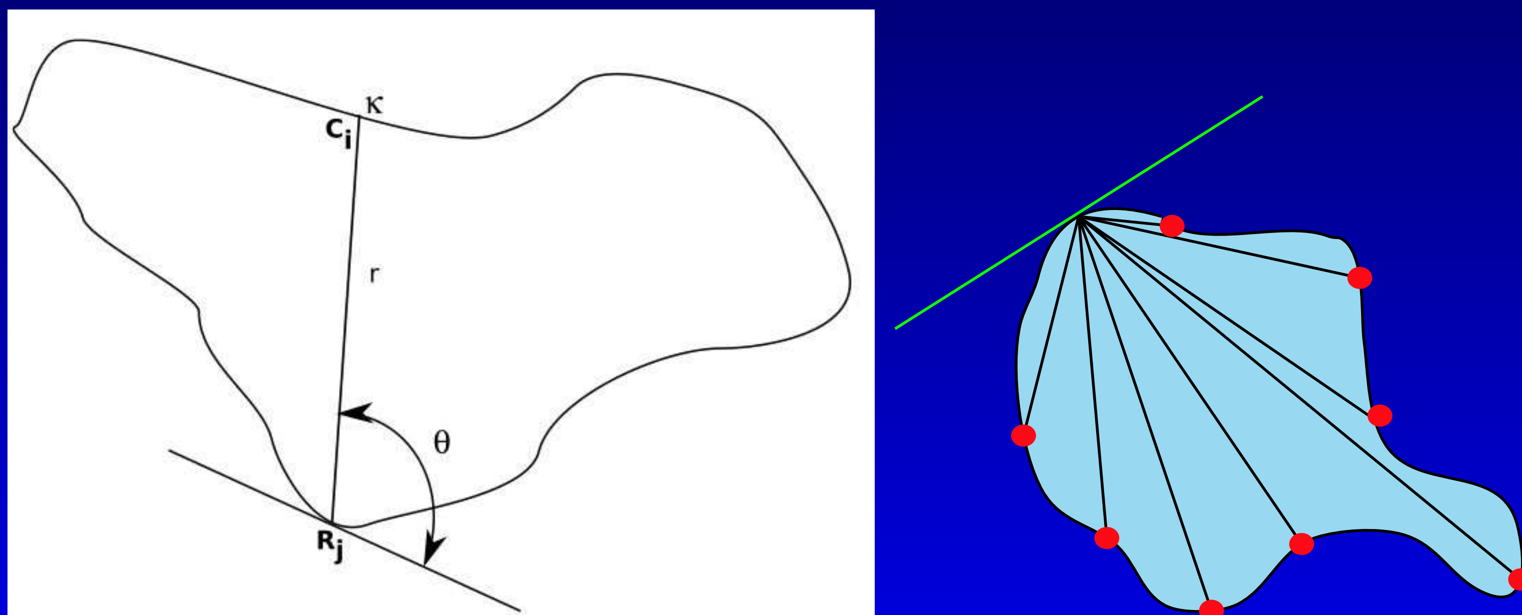


Figure 1: Feature Vectors

The model for a shape

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

$$ {}^iM_j(\mathbf{v}) = \max_{k=1}^{N} \exp\left(\frac{||\mathbf{v} - \mathbf{v_{jk}}||^2}{2\sigma^2}\right) \tag{1} $$

where σ will be determined later.
$({}^iM_j(\mathbf{v}))$: a function which estimates the presence of a feature($\mathbf{v}$) in the model. Can be precomputed.

Matching

Same philosophy as Hough transform – evidence accumulation. To match shape ${}^1C$ which has L points, against shape ${}^2C$,

1. Choose a particular reference point ${}^1R_k \in {}^1C$, $k = 1, \ldots, K$.
2. Compute the feature vectors to every other point in ${}^1C$, and denote those vectors by $\mathbf{v}_{kl}$.

Matching

We have a collection of models, each referencing a curve and a reference point. Consider matching curve ${}^1C$ using reference point ${}^1R_k$ to curve ${}^2C$ using reference point ${}^2R_j$, and compute

$$ {}^{12}A_{kj} = \frac{1}{L} \sum_{l=1}^{L} {}^2M_j(V_{kl}) \tag{2} $$

The best match between shapes 1 and 2 is then

$$ {}^{12}A = \max_{k,j} A_{kj} \tag{3} $$

Computational Complexity

The computational complexity of the algorithm as described above would seem high since it requires a search over all paris of reference points, but this is mitigated by the facts that

- there are very few reference points on any particular contour
- the search may be ordered by considering only pairs of reference points which are very similar

Test set with lots of straight lines

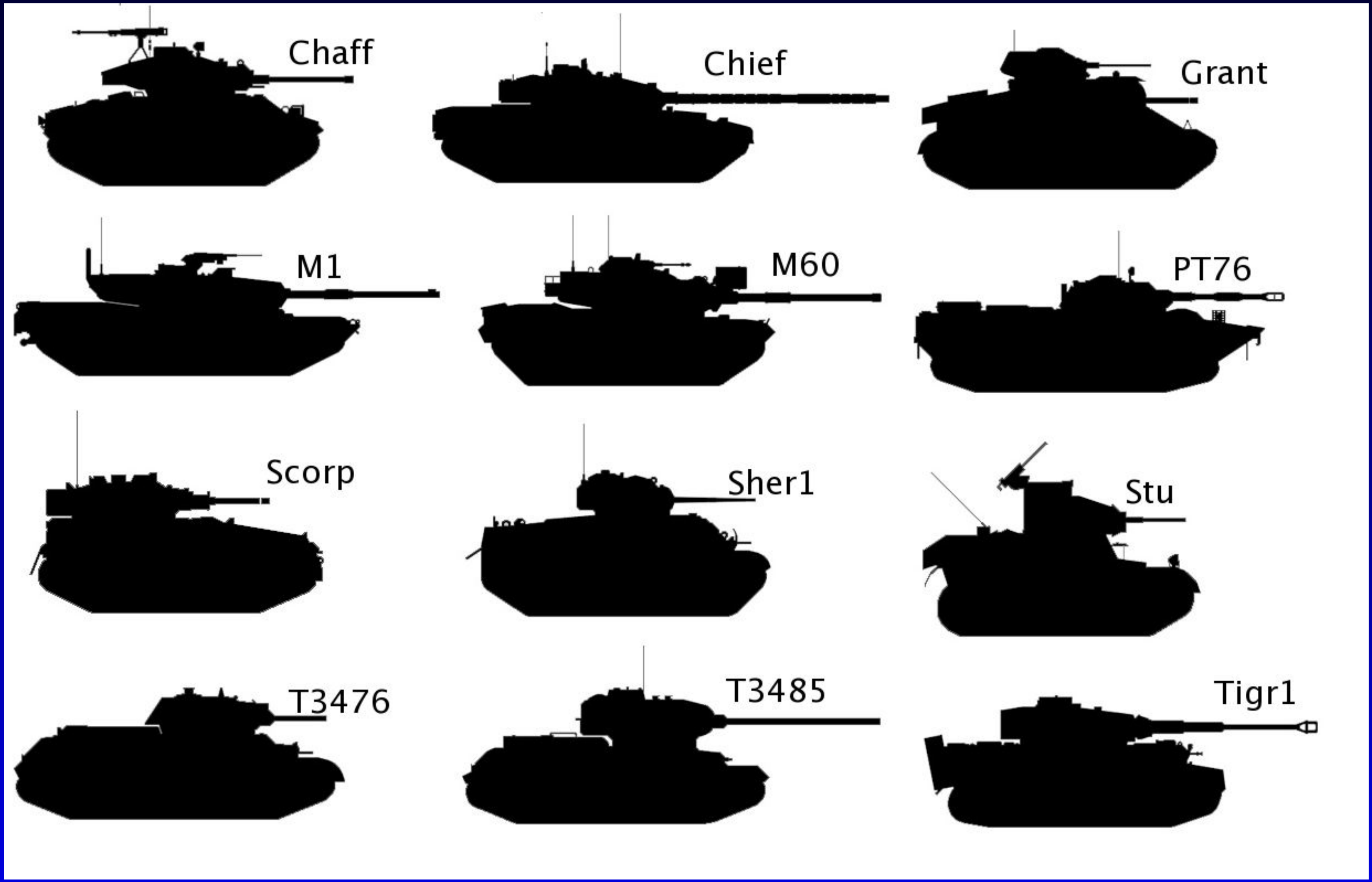


Figure 2: Database of 12 tanks

Test Set with lots of examples

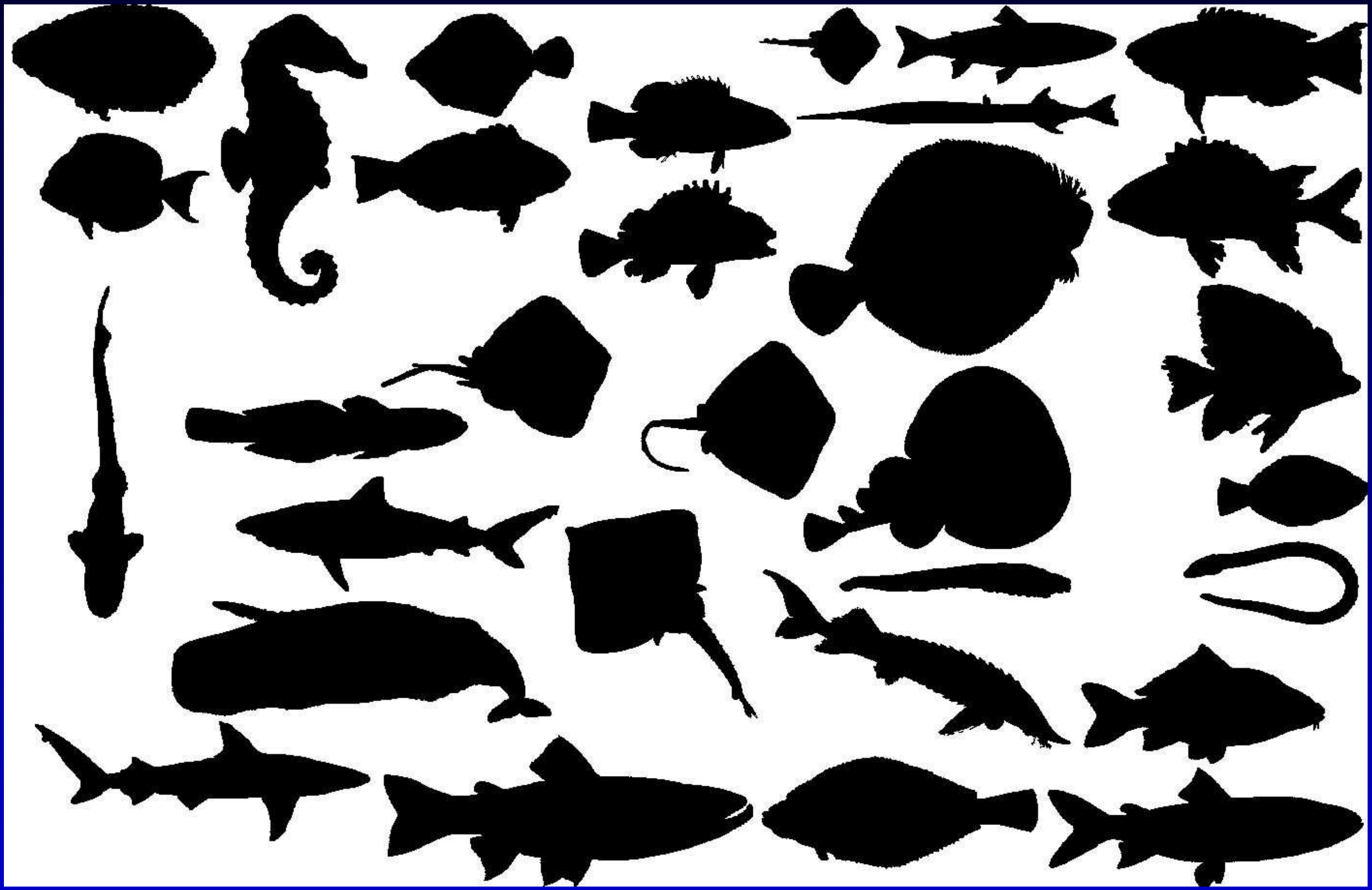


Figure 3: The 31 randomly picked silhouettes from the SQUID database

Experimental Results

If no occlusion, only one reference point required (CG is good)

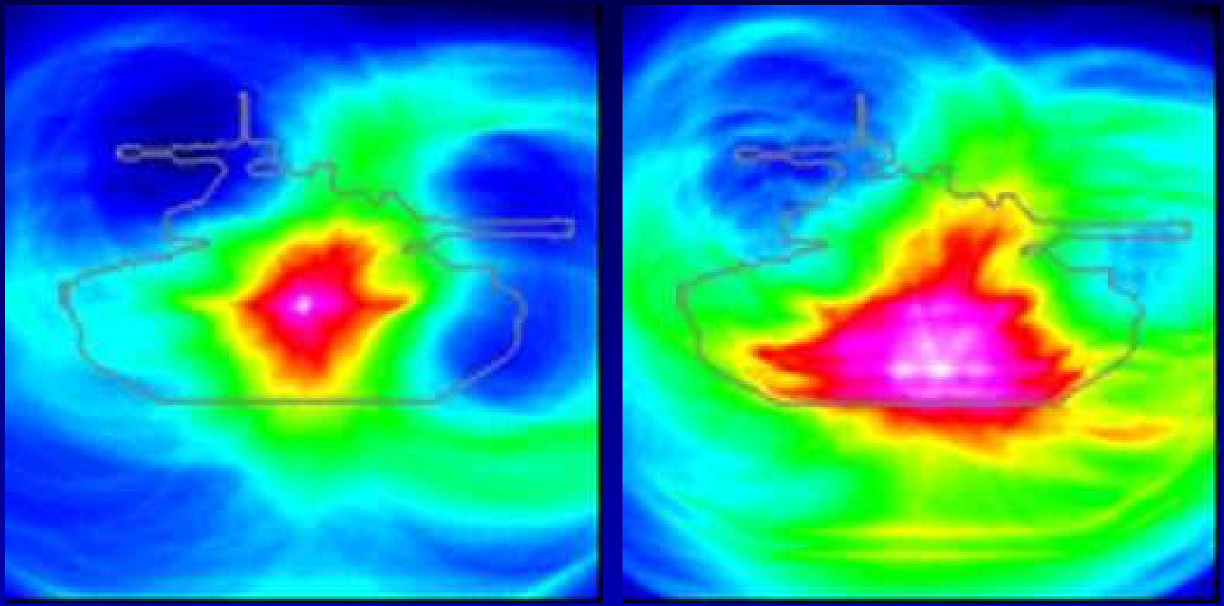


Figure 4: Two examples of matching contours. The one on the left exhibits a distinctive peak representing a good match.

Contour Blur

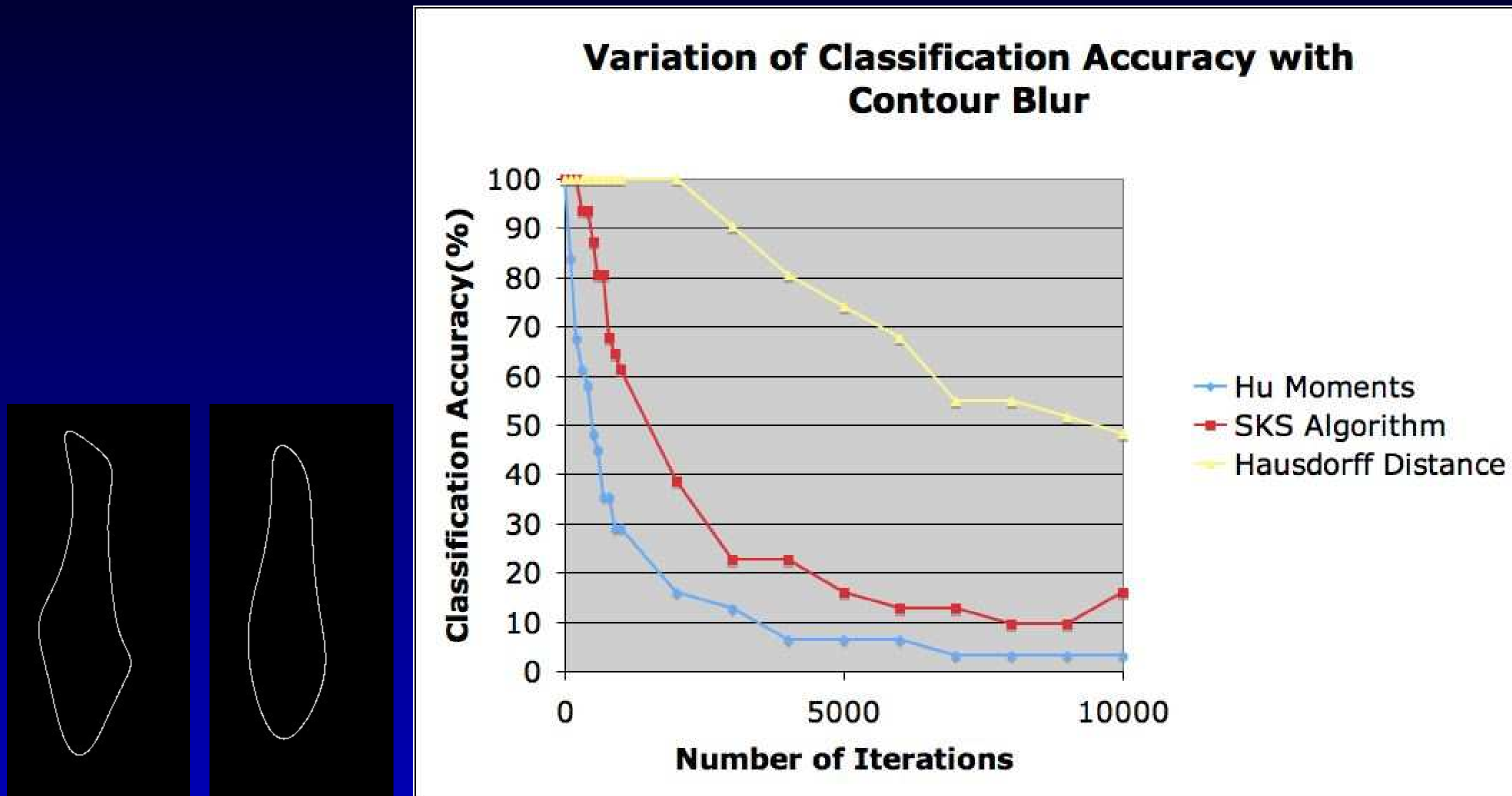


Figure 5: Left: The fish contour blurred by 2000 iterations of smoothing with unit variance Gaussian. Middle: The same contour after 6000 iterations

Invariance to Similarity Transforms

We built models using all the tanks(rotated and scaled) and matched each tank contour with every model. 12 tanks and each tank has 6 rotated and scaled versions of itself, the total number of correct matches is 1728.
Table below shows the retrieval results of the four algorithms.

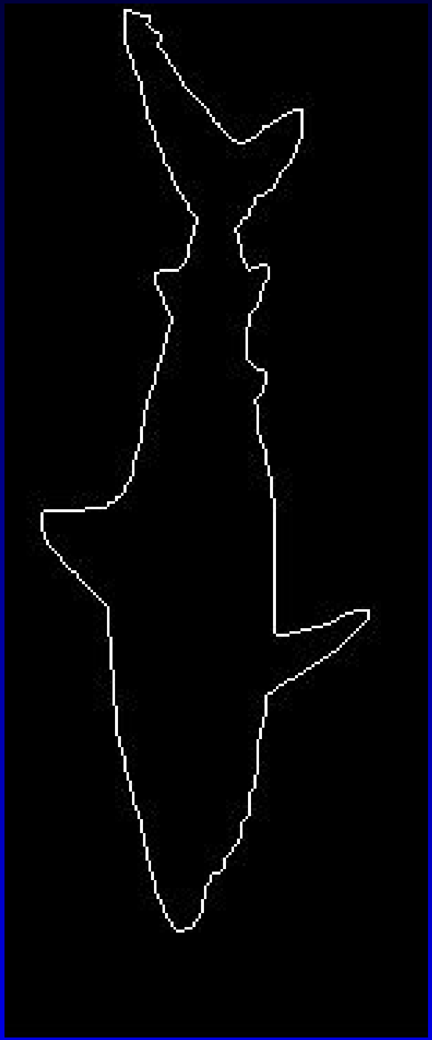
Algorithm	Correct Retrievals(%)
SKS	98.26
Shape Context	99.13
Hu Moments	77.6
CSS	75.1

Table 1: Invariance to Similarity Transforms

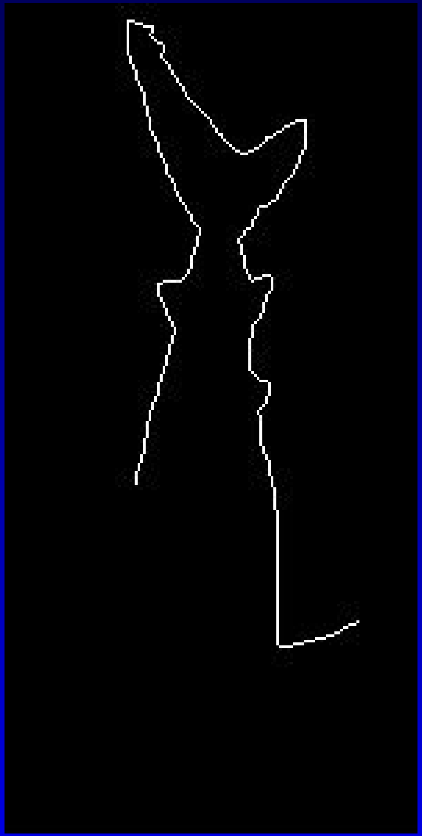
Generating Occluded Contours

- 31 fishes from the SQUID database, shown earlier.
- Partially occlude each fish by retaining 10-90% of the points.
  algorithm : Starting at point C_1, K points are chosen such that $K/N = \delta/100$. Remove those K points.
- Move the starting point from 1 to 2 , then 3, etc, generating a new occluded boundary with same percent occlusion. An example of this is shown in figure 6.
- The performance reported is the average of all the occlusions of that particular boundary.

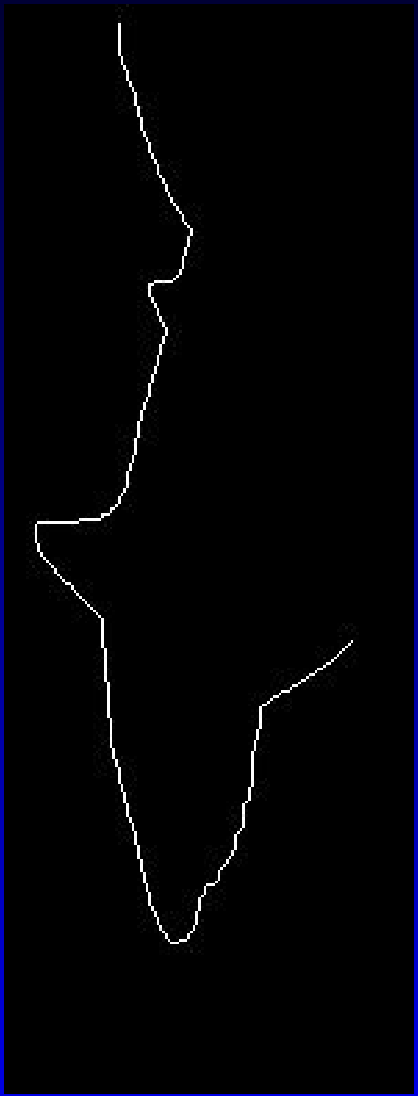
Occluded Contours



(a) Original Contour



(b) Contour at 40% occlusion



(c) Same contour at 40% occlusion

Robustness to Occlusion

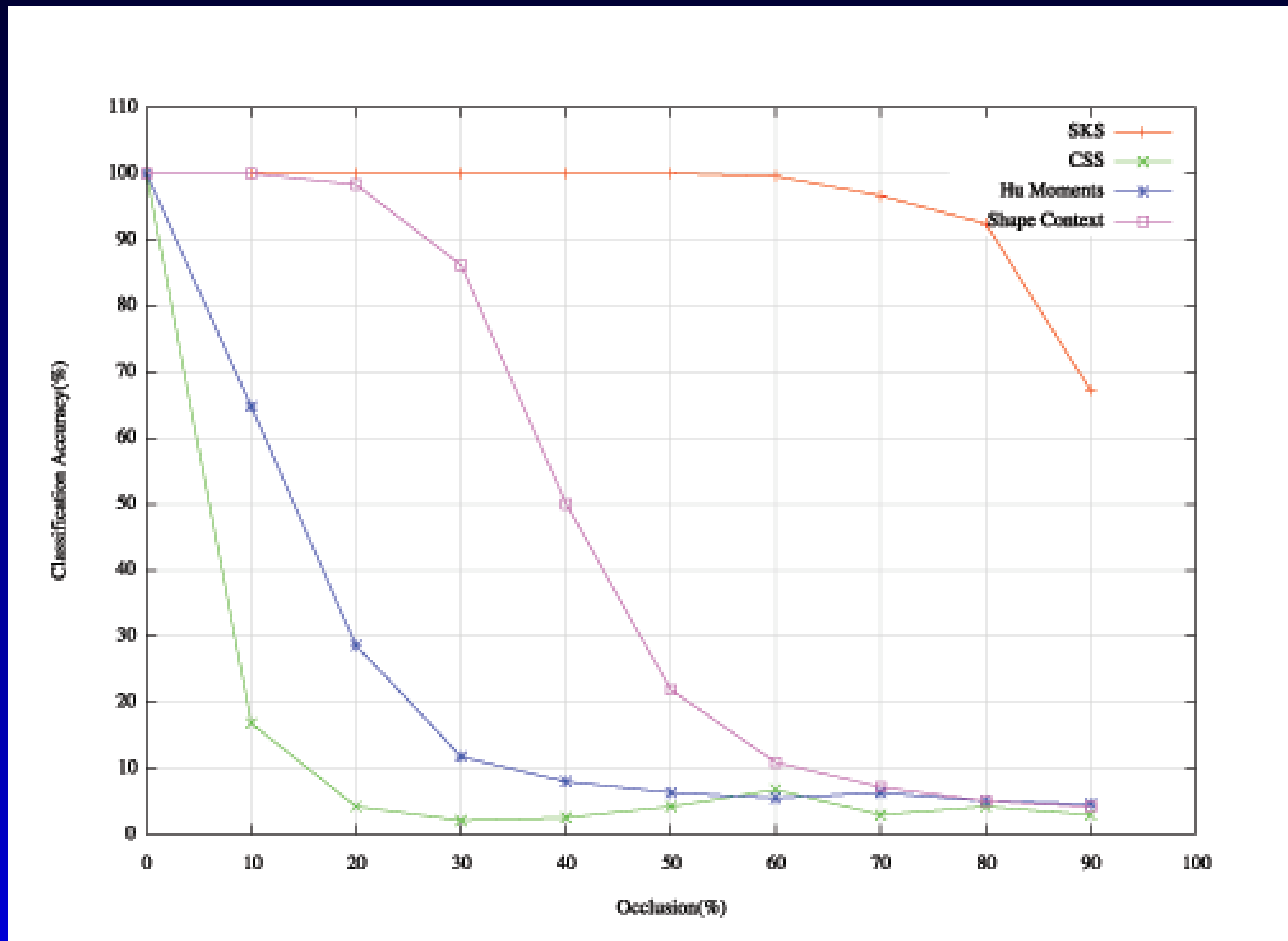
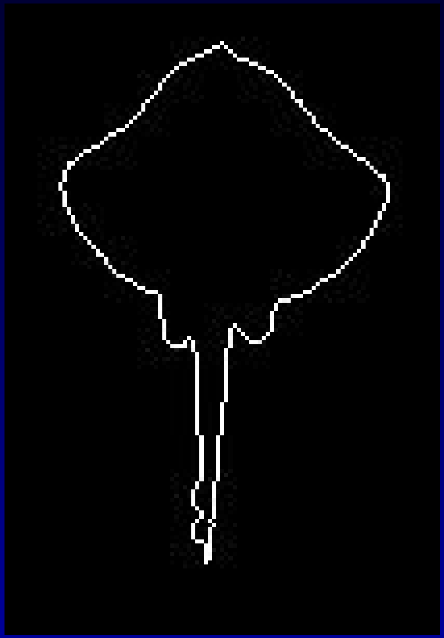


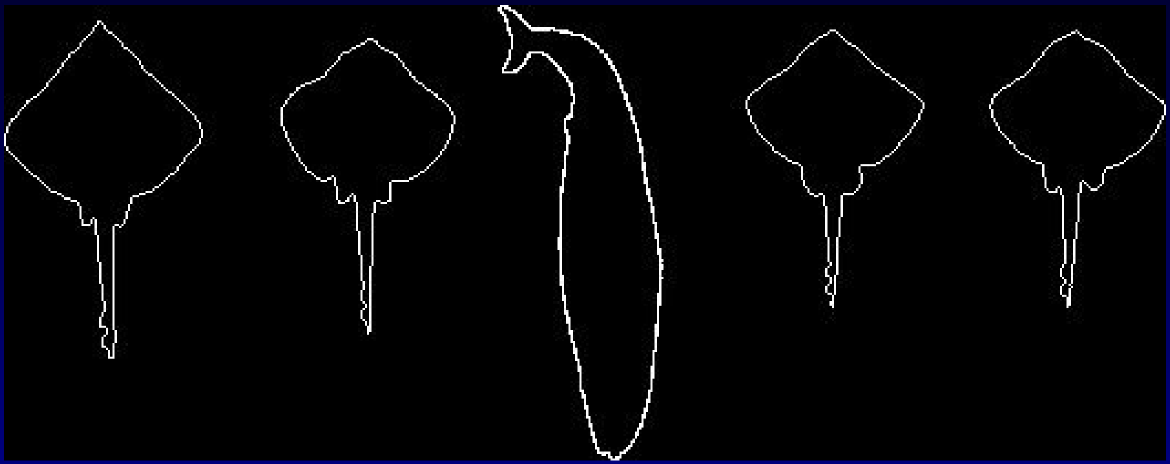
Figure 7: Classification accuracy at various levels of occlusion.

Content-based Image Retrieval

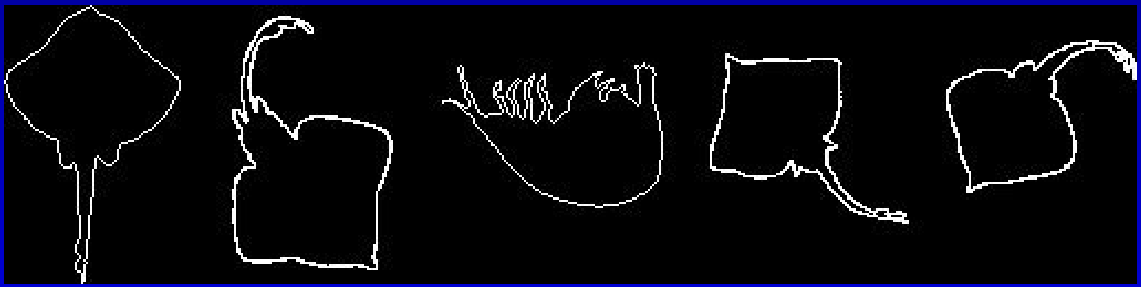


(a) kk732 - Query shape.

Content-based Image Retrieval

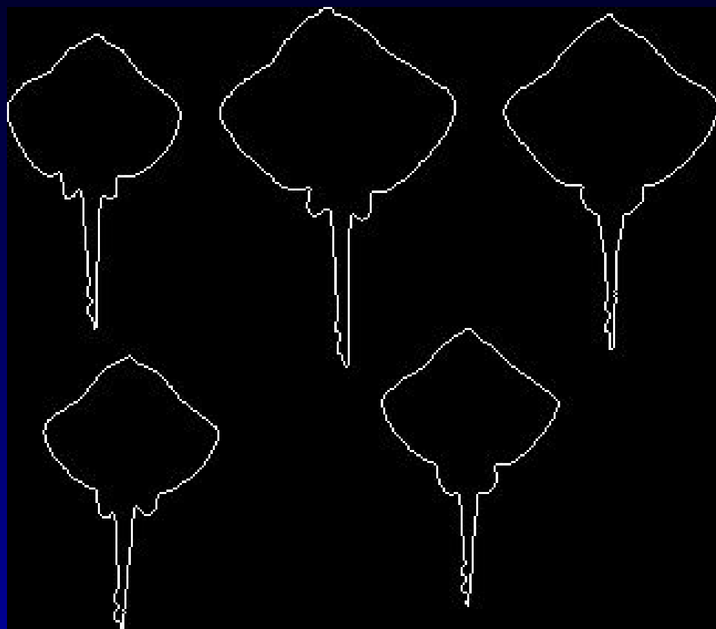


(b) Top 5 matches in random order for the shape kk732 using CSS.

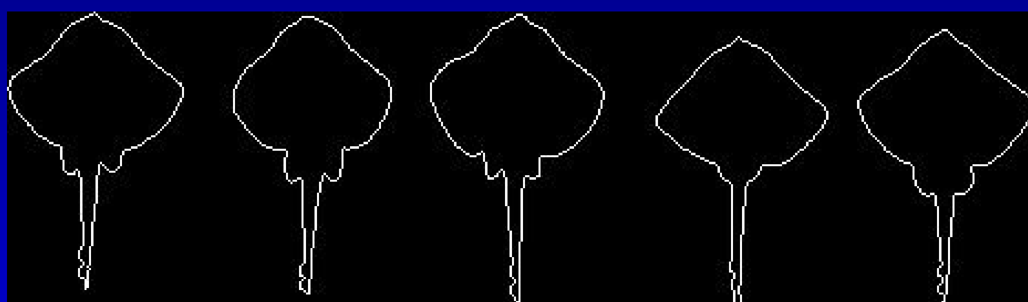


(c) Top 5 matches in random order for the shape kk732 using Hu Moments.

Content-based Image Retrieval



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



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

Conclusion

Is the gestalt approach a good one?

Belongie et al.: "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:“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 objectsÕ 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.’

Conclusion

But, humans use key points

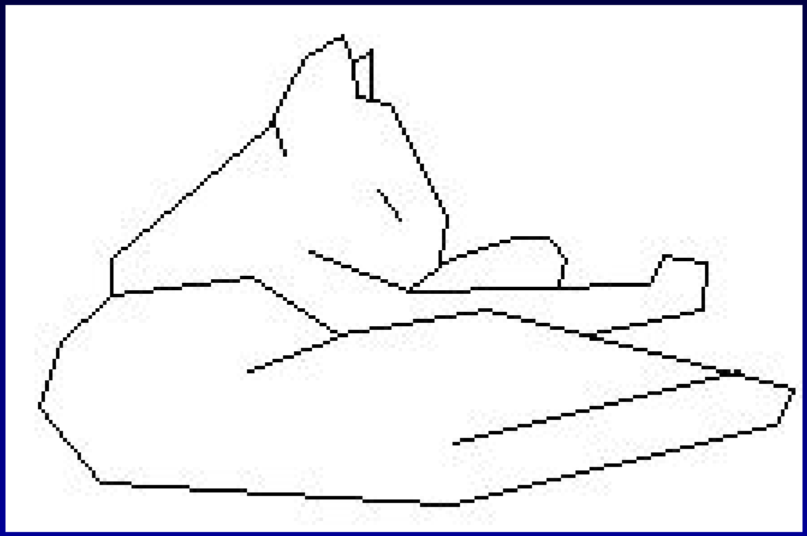


Figure 8: A line drawing, without color or texture information, and consisting of only straight lines connecting salient points.