Shape Similarity by Piecewise Linear Alignment[‡]

Jon Sporring,[*] Xenophon Zabulis[*] ,[†] Panos E. Trahanias[*] ,[†] and Stelios C. Orphanoudakis[*] [†]
{sporring,zabulis,trahania,orphanou}@ics.forth.gr

[*] Institute of Computer Science,
Foundation for Research and Technology – Hellas,
Vassilika Vouton, P.O. Box 1385,
GR-71110 Heraklion, Crete, Greece

[†] Department of Computer Science,
University of Crete,
P.O. Box 1470,
GR-71409 Heraklion, Crete, Greece

ABSTRACT

A key problem when comparing planar shapes is to locate corresponding reference points, such as inflection or high curvature points. High curvature points are to be preferred, since they are psychophysically more important and are shown to be computationally more reliable. Using a scale-space of curves such 'corners' are located, and an empirical analysis demonstrates that high curvature points are indeed least sensitive to noise. Using reliable 'corners', a shift invariant and piecewise linear alignment of shapes is defined, and a similarity measure between two curves is defined using lossless compression. The measure is chosen to be proportional to the area between two shapes after piecewise alignment and to the number of data points. The proposed shape similarity is validated using a small shape database.

Keywords: Scale-Space, Mean Curvature Motion, Minimum Description Length, Procrustes Distance, Shape Similarity.

1. Shape from Invariance

Boundaries are important features, when searching for objects in an image database [1, 2]. A boundary may be characterized by the sequence of turns encountered as its path is traversed, and this sequence is invariant to translation and rotation of the boundary. Hence, the shape of a boundary is an intrinsic property with respect to a group of transformations. Typical transformations studied in image processing are [3]: The euclidean group, $E(n)$, of rotation, reflection, and translation, the similarity group, $SO(n)$, of euclidean transformations and scaling, and the affine group, $A(n)$, of similarity transformations and scaling in two orthogonal directions. These groups define different notions of shape. For example, circles of all sizes have the same shape under the similarity and affine groups but not under the euclidean group. Conversely, all ellipses are the same shape under the affine group but not under euclidean or similarity transformations.

As elaborated upon in [4, 5], each of the above groups can be associated with a set of invariant (and semi-invariant) functions that defines the specific notion of shape. While such an invariant or semi-invariant function is a useful basis for comparing shapes, a number of problems must be solved, before a working system can be built:

1. When comparing invariant functions, the shape must be assigned an origin for point to point comparison. To be intrinsic the choice must be a singular point of the invariant function. Unfortunately, for near symmetrical shapes this is an ill-posed problem: For example, the curvature function of a rectangle, has four global maxima, but an arbitrary small noise signal will result in a curvature function with four points of high curvature, but only one global maximum.
2. Shapes will almost never have the same circumference, and their invariant functions must be stretched so that they can be compared point to point. While linear stretching is optimal when shapes differ in size, nonlinear stretching seems more appropriate in other cases. As an example, the shapes in Figure 1 differ by a dent. Some of the the extremal points of curvature function A matches those of function B but their lengths differs. Unfortunately, linearly stretching function A to match the length of function B causes all the extremal points to mismatch.
3. A sensible similarity measure must depend on the number of samples. For example, three points on a circle will look more like a triangle than a circle, but when the number samples is increased, the points will quickly be recognized as coming from circle and not from a triangle.

Finally, it is observed that curvature difference functions are not the only sensible notion of shape differences as illustrated in Figure 2. The two pieces appear to be different, since no euclidean transformation can make the bent piece straight. However, the curvature functions differ only at one point, otherwise appearing to be rather similar.

2. From points to shapes

The euclidean or affine curvature functions are intrinsic descriptions of shapes under euclidean or affine transformations. This implies that singular points, such as zero-

[‡] This work was funded in part by EC Contract No. ERBFMRX-CT96-0049 (`http://www.ics.forth.gr/virgo`) under the TMR Programme.

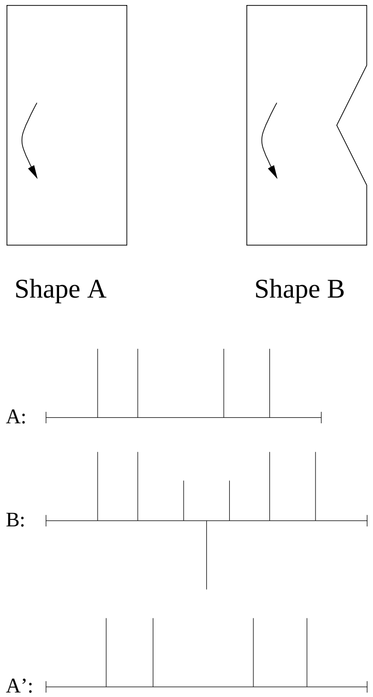


Figure 1. Linear stretching of curvature function is not optimal for all shapes. TOP: Two similar shapes. BOTTOM: The corresponding idealized curvature functions and function A linearly stretched to match length of function B.

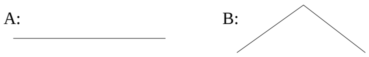


Figure 2. Two different curve pieces with curvature functions only differing in one point.

crossings, extrema, etc., are intrinsic properties of shapes. Therefore, singular points of invariant functions are excellent candidates for shape comparison. An elegant distance measure between ordered point sets is the Procrustes distance [6]: Given two ordered point sets in the plane, $P = \{p_j\} = \{(x_j, y_j)^t\}$ and $Q = \{q_j\}$, the squared Procrustes distance is given by $D^2 = \min_\theta \sum_j |p_j - T_\theta q_j|^2$, where $|\cdot|$ is the length operator and T_θ the rotation operator. There exists a simple, closed form solution of the minimization [6]. Invariance under similarity, euclidean, or affine transformation may be obtained by prior normalization.

Singular points on curves are naturally ordered (except for a possible mirroring). However, in order to calculate the Procrustes distance between two shapes, we need the same number of points on each shape. Further, for similar shapes these points should be chosen at similar positions. In the following, we will show how a scale-space of curves can be used to order a set of singular points according to stability, and we will show how an increasing number of singular points on a curve describes a decreasing number of shapes.

2.1. Stable points of curves

Scale-space and group invariance are very much related concepts [7]. For example, the Mean Curvature scale-space [8] is invariant under euclidean transformation and is non-increasing in the number of extrema and inflection points of the curvature function. Likewise, the Affine Curvature scale-space [9] is invariant under affine transformation and non-increasing in the number of extrema and inflection points of the affine curvature function.

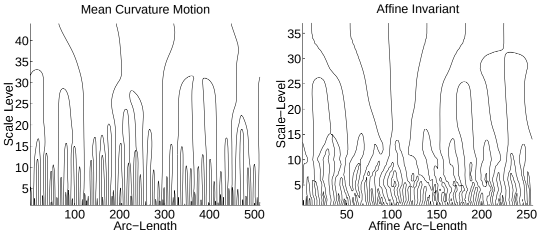


Figure 4. The fingerprint images of the evolutions in Figure 3 MIDDLE and RIGHT respectively.

We have used the coordinate-wise implementation of Mean Curvature scale-space [8], and extended this implementation to the Affine Curvature scale-space by replacing euclidean arc-length with the affine arc-length. A random shape is shown in Figure 3 together with snapshots of the Mean Curvature and the Affine Curvature scale-spaces. It can be verified that the shape tends to a circle in the Mean Curvature scale-space and to an ellipse in the Affine Curvature scale-space. Another way of representing the evolution of the extreme curvature points is by the finger print images shown in Figure 4. Some extrema survive a long time and will henceforth be called stable extrema.

The only catastrophe that generically occurs for zero crossings of the euclidean and affine curvature is pairwise annihilation. The same result holds for extrema of the euclidean and affine curvature, in which case the pairs are made of a maximum and a minimum curvature extremum. These properties are well preserved for the selected implementation of the Mean Curvature scale-space, but it has not been possible to find a well-behaved implementation of the Affine Curvature scale-space. It is noted that global affine invariance can be achieved by a prior global normalization of the affine components followed by the euclidean method presented in the rest of this paper.

To increase localization precision, stable extrema are tracked to zero scale. However, not all stable extrema are equally sensitivity to noise as illustrated in Figure 5. In these 3 experiments, almost invisible noise was added to the coordinate functions of the shape shown in Figure 3(LEFT), using i.i.d. normal noise of standard deviation of 0.05. Comparing the results, it is observed that the 4 most stable extrema are all traced to similar part of the shape, and that some locations are more persistent than others. Such behavior is typical. When a curvature function locally is close to constant, the locations of the stable extrema are very noise sensitive, regardless of how small the noise level may be.

A notion of certainty can be obtained by examining the curvature at a corner over scales. Since the curvature function scales inversely proportional with length, we can compare

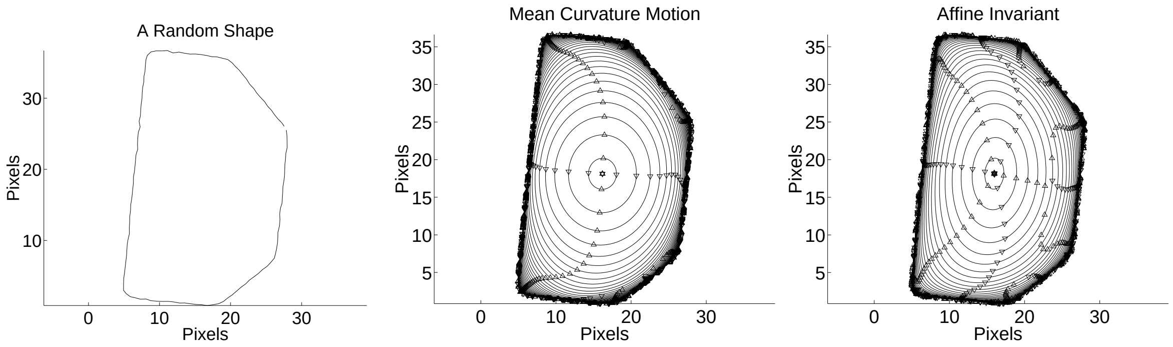


Figure 3. A shape evolves differently in different scale-spaces. LEFT: A shape. MIDDLE: Snapshots from the Mean Curvature evolution. RIGHT: Snapshots from the Affine Curvature evolution. Triangles denote euclidean and affine curvature extrema.

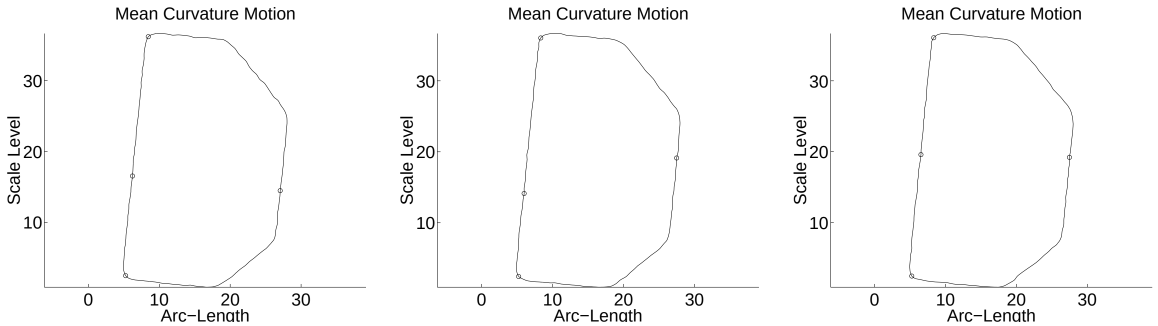


Figure 5. The accuracy of a trace depends on the curvature and the neighboring structure. The shapes are given by Figure 3(LEFT) plus normal noise of standard deviation 0.05.

curvature values of different scales by multiplying by scale,

$$\kappa^* = \max_s \operatorname{sign}(\kappa(0))\sigma^*(s)\kappa(s).$$

The parameter s denotes a parameterization of the fingerprint line. The value κ^* intuitively relates to the spatial extend of the corner [10]. In the following are corners of shapes examined when ordered by κ^*.

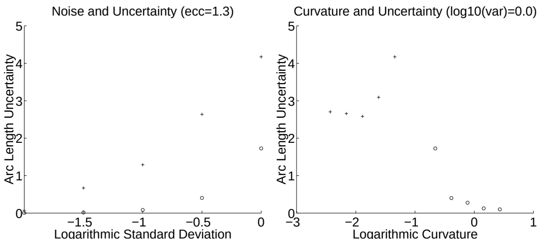


Figure 6. Noise makes corner location more uncertain, whereas sharpness of corners reduces uncertainty. LEFT: the logarithmic standard deviation of the noise versus the standard deviation of the located corner arclength. RIGHT: the logarithmic curvature versus the standard deviation of the located corner arclength.

2.2. A study of stable curvature extrema of ellipses

In this section, a study the ellipse is presented. A one parameter family of ellipses is

$$\begin{bmatrix} x_e(s) \\ y_e(s) \end{bmatrix} = 10 \begin{bmatrix} \frac{1}{e}\cos(s) \\ e\sin(s) \end{bmatrix},$$

where e denotes eccentricity. It follows that the major and minor axis are given by $10e$ and $\frac{10}{e}$, the area is 100π independently on e, and that the circumference increases proportional to the absolute value of e. The ellipse has 4 curvature extrema $s^* = \{0, \frac{\pi}{2}, \pi, \frac{3}{4}\pi\}$, and the curvature values in these points are given by $K(s^*) \in \{\frac{1}{10e^3}, \frac{e^3}{10}\}$. Hence, $K(s^*) \to \frac{1}{10}$ for $e \to 1$, and $K(s^*) \in \{0, \infty\}$ for $e \to \infty$.

In practice, shapes will be subject to noise. A simple model of a noise source is normal distributed noise perpendicular to the curve,

$$\begin{bmatrix} \tilde{x}_e(s) \\ \tilde{y}_e(s) \end{bmatrix} = \begin{bmatrix} x_e(s) \\ y_e(s) \end{bmatrix} + \mathcal{N}(0,\sigma) \begin{bmatrix} -y'_e(s) \\ x'_e(s) \end{bmatrix},$$

where prime denotes differentiation with respect to arclength and $\mathcal{N}(0,\sigma)$ is an independently and identically normal distributed stochastic source of standard deviation σ. This is similar to adding noise directly on the curvature function.

As noted above, the statistical reliability of tracing stable extrema to zero scale depends on the local structure of a shape. In Figure 6 is shown the standard deviation of the resulting position after tracing the 4 most stable corners to zero scale for various values of σ and ϵ. Each data point in the figure is based on 1000 experiments. In Figure 6(LEFT) it is observed that the uncertainty is monotonically increasing with noise, and that the low curvature points are more noise sensitive than high curvature points. Figure 6(RIGHT) is interpreted as follows. High curvature points tend to dominate neighboring structure hence increasing certainty. In other words, the uncertainty tends to zero as curvature tends to infinity. In contrast, uncertainty is maximal for straight lines and perfect circles, since they have constant curvature functions. Thus there is a singularity at curvatures 0 and 0.1. Finally, the uncertainty is reduced by the neighboring structure: A minimum must lie in between two maxima curvature points. Therefore, the uncertainty is inversely proportional to the distance of the maxima, and a minimum at approximately curvature 0.01 is obtained.

From psychophysical experiments it is known that corners are the most important points on curves [11]. From a statistical point of view we conclude that these are also the most certain points on curves.

3. Taking the number of data points into account

Interpretation of shapes depends on the number of data points. As an example, consider data points sampled from a circle. With three points, the circle cannot be distinguished from a triangle, and in general, n points might as well come from an n'th ordered polygon than from a circle. Naturally, the polygon will increasingly resemble a circle as n is increased, and for some large n it seems reasonable to assume that the points stem from a circle rather than a polygon. In this paper, 'reasonable' is defined using lossless compression: In the above example, the points will be interpreted as a circle, if this is a shortest description of the data points.

When comparing models of data in terms of minimal compression, we must compare the coding cost of the model and the deviation of the model from the data [12]:

$$L(D, M) = L(M) + L(D|M),$$

where D is the data set, and M the model parameters. The optimal model is the one that minimizes $L(D, M)$.

In the present case, a given shape is to be compressed using one shape from a database. Hence, the codelength of the model consists of identifying, which shape from the database is being used, and how many confident points are used in the alignment:

$$L(M) = -\log(\text{number of elements in database})$$
$$+ \log^*(\text{number of confident points}).$$

The code is designed such that all shapes in the database are equally probable, and coded the number of confident points by the Universal Distribution of Integers [12], $\log^*(i) = c + \log(i) + \log(\log(i)) + \ldots$, summing over all positive terms.

The deviation from the model is given by the point to point alignment. In order for the decompression to be successful, we will need to describe how the segments of the curve in the database are to be sampled, and what are the deviations from these sample points:

$$L(D|M) = \sum_i \log^*(\text{no. of samples in piece } i)$$
$$+ \log^*(10\sigma) - \sum_j \log\left(G(M_j, D_j, \sigma)\right).$$

The deviations are coded as a two dimensional Gaussian distribution,

$$G(M_j, D_j, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{(X_j - x_j)^2 + (Y_j - y_j)^2}{2\sigma^2}\right).$$

The point (x_j, y_j) is coded by the inferred point (X_j, Y_j) from the shape in the database. We have arbitrarily chosen to compress the standard deviation with 0.1 precision.

We emphasize that in contrast to [5, 13] we do not code the difference between two curves as the difference in their curvature functions. This has two reasons: Firstly, taking the difference between two curvature functions is essentially as noisy as computing third order derivatives. Secondly, the difference between two curvature functions does not coincide very well with our notion of shape difference as explained in Figure 2.

4. Recognizing shapes

In the following, the algorithm for shape recognition invariant under euclidean transformation is presented.

Given a shape as a list of points on its boundary and a database, the following steps are performed:

1. Compute the scale-space until only 4 curvature extrema are present.
2. Track the extrema along scale.
3. Choose the most stable extrema.
4. Find the maximum scale-normalized curvature values for extrema tracked along scale.
5. Sort the extrema at zero scale by their maximum scale-normalized curvature.
6. For each of the $n = 2, 3, \ldots$ points with highest maximum scale-normalized curvature sort them according to location on the curve.

Each subset of points constitutes a segmentation of the boundary. Two shapes are then compared by aligning their segmentations using Procrustes distance. To obtain invariance under origin, we:

7. Generate all shifts of one subset and calculate the Procrustes distance to the other, and select the shift minimizing the distance.

At this point the shape is coded relative to a shape from the database. Since two segmentations are aligned, a piecewise linear point to point correspondence between the boundaries is known.

8. Generate the piecewise linear point to point correspondence between the boundaries.
9. Calculate the codelength using the correspondence, and minimize this codelength with respect to translation, rotation, and scaling.

The minimal codelength over segmentations and euclidean transform is taken as the distance between the curves. As a consequence of using non-linear stretching respecting relative arc-length of one curve, the distance between two curves is only symmetric when the curves have the same segmentation and number of points. Further we note that the distance of a shape to itself is always larger than zero. Given n points of a shape, t scale-levels, and m stable extrema, the computational complexity of the above algorithm is $\mathcal{O}(nt\log(nt) + m^2 \log m^2)$. The first term is typically the largest and refers to the scale-space computation, while the later refers to Step 6 in the algorithm.

We have implemented a system that uses the stable points of boundaries to perform piecewise linear stretching. This system has been tested on a small database of 39 boundaries. The shapes have been selected from an automatic segmentation of images from the Columbia University Image Library (Coil-100)[‡]. In Figure 7 a sample set of the shapes is shown. The distance of each of these has been calculated to the 38 other shapes, and in Figure 8 the 3 closest matches are shown. These results were selected as a fair representative of the shape similarity ordering produced by the system. The shapes in the left column refer to the best match in the database. From top to bottom we see that the codelength of the best match is increasing downwards, and observe that this correlates well with the quality of the match. The same observation is made for each row individually.

5. Conclusions

Shape similarity and invariance are two closely coupled concepts, and invariant signature functions, such as the curvature, are good starting points for shape comparison. In this paper, a linear and shift invariant alignment of signature functions has been studied with emphasis on the euclidean group. The alignment can conveniently be performed using singular points of the signature functions, and it has been have demonstrated that a scale-space of curves together with scale-selection can be used to identify stable and reliable singular points. The problem of shape comparison depends on the number of sample points used. We have therefore defined a shape distance using modeling by compression, and the distance measure has been designed to be closely related to the area between two curves.

Some authors have suggested to use lossy compression algorithms that preserves ‘semantics’ in order to efficiently compare shapes [1]. Although psychophysical experiments on mammals may lead to a useful lossy compression algorithm, these algorithms cannot easily be compared in an objective manner. In contrast, lossless compression algorithms are build with a common yardstick, in the sense that no part of the original data is discarded. Hence, the results can be directly compared. For example, we may determine, if a finite sum of cosine waves is a better model of a boundary than a cubic spline. By isolating the subjectivity to the selection of a model class, we gain a framework by which we can discuss semantics in an objective manner.

References

[1] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[2] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steel, and Peter Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, pages 23–32, September 1995.

[3] P. Olver, G. Sapiro, and A. Tannenbaum. Differential invariant signatures and flows in computer vision: a symmetry group approach. In B. M. ter Haar Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, number 1 in the series Computational Imaging and Vision, pages 255–306, Dordrecht, Netherlands, 1994. Kluwer Academic Publishers.

[4] T. Moons, E. J. Pauwels, L. J. van Gool, and A. Oosterlinck. Foundations of semi-differential invariants. *ijcv*, 14:25–47, 1995.

[5] E. J. Pauwels, T. Moons, L. J. van Gool, P. Kempenaers, and A. Oosterlinck. Recognition of planar shapes under affine distortion. *International Journal of Computer Vision*, 14:49–65, 1995.

[6] Fred L. Bookstein. Shape and the information in medical images: A decade of morphometric synthesis. *Computer Vision and Image Understanding*, 66(2):97–118, 1997.

[7] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Archive Rational Mechanics and Analysis*, 123(3):199–257, September 1993.

[8] Farzin Mokhtarian and Alan Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):34–43, January 1986.

[9] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11(1):25–44, 1993.

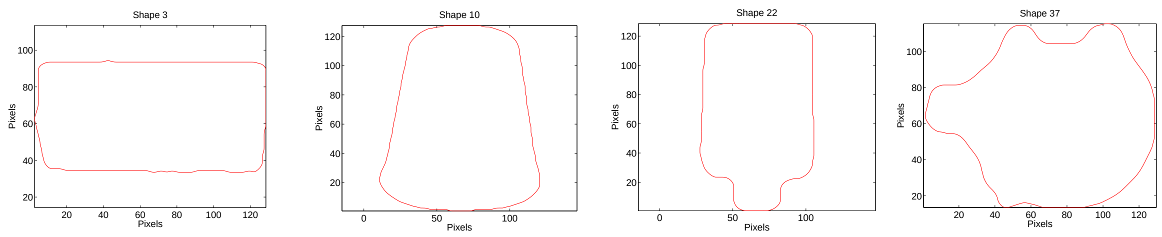
[10] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, USA, 1994.

[11] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[12] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.

[13] Michael Kliot and Ehud Rivlin. Invariant-based shape retrieval in pictorial databases. *Computer Vision and Image Understanding*, 71(2):182–197, 1998.

[‡] `http://www.cs.columbia.edu/CAVE/coil-100.html`



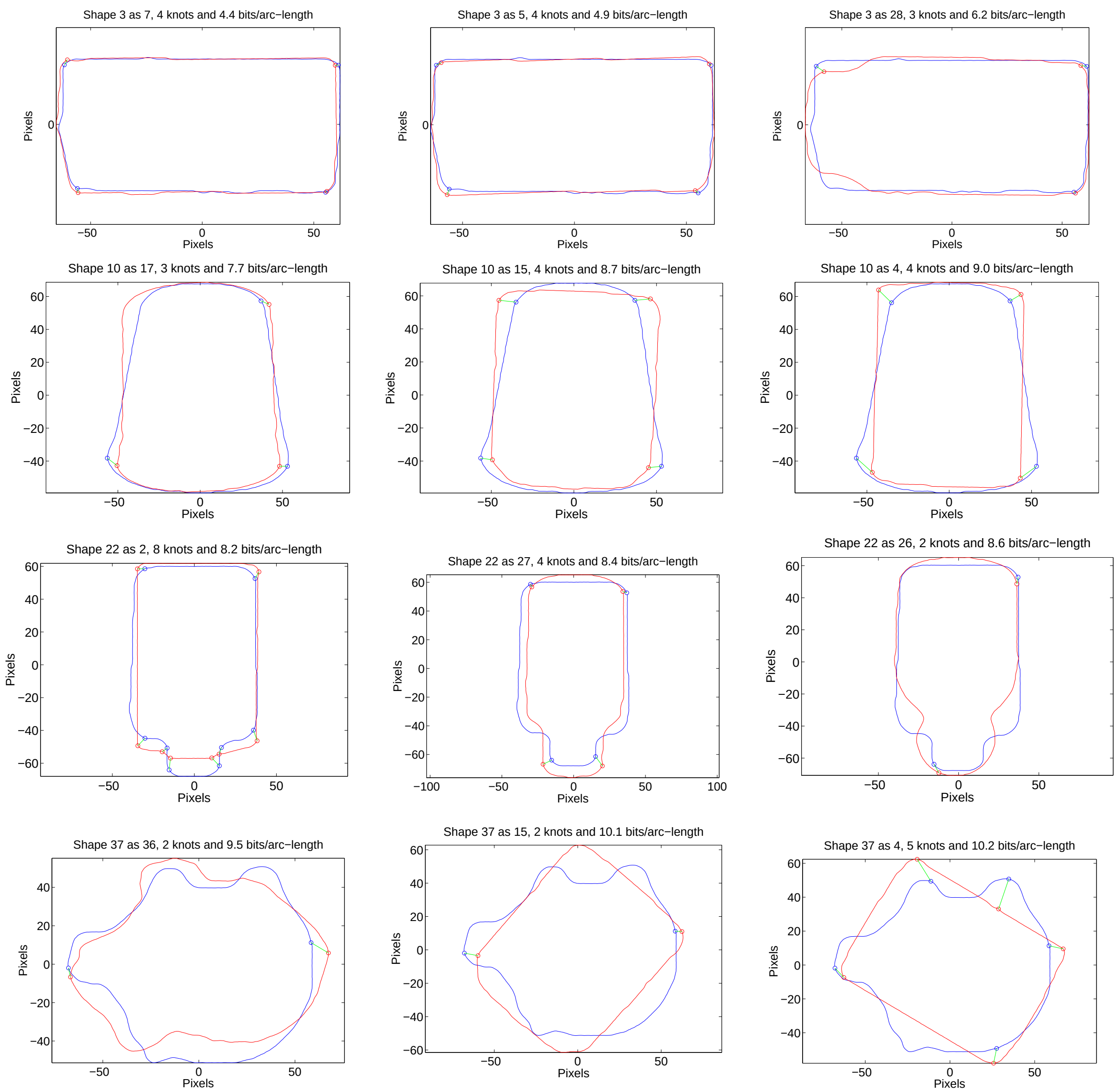
Shape 3

Shape 10

Shape 22

Shape 37

Figure 7. Some boundaries of objects (from left to right): A gum package, a reusable food container, a tube with a screw-lid, a pig.



Shape 3 as 7, 4 knots and 4.4 bits/arc−length

Shape 3 as 5, 4 knots and 4.9 bits/arc−length

Shape 3 as 28, 3 knots and 6.2 bits/arc−length

Shape 10 as 17, 3 knots and 7.7 bits/arc−length

Shape 10 as 15, 4 knots and 8.7 bits/arc−length

Shape 10 as 4, 4 knots and 9.0 bits/arc−length

Shape 22 as 2, 8 knots and 8.2 bits/arc−length

Shape 22 as 27, 4 knots and 8.4 bits/arc−length

Shape 22 as 26, 2 knots and 8.6 bits/arc−length

Shape 37 as 36, 2 knots and 9.5 bits/arc−length

Shape 37 as 15, 2 knots and 10.1 bits/arc−length

Shape 37 as 4, 5 knots and 10.2 bits/arc−length

Figure 8. Three best matches for each of the shapes in Figure 7. Best is shown in left column and worst (out of 3) in right column.