Straightening and Partitioning Shapes

Paul L. Rosin

Department of Computer Science, Cardiff University, UK.
Paul.Rosin@cs.cf.ac.uk

Abstract. A method for partitioning shapes is described based on a global convexity measure. Its advantages are that its global nature makes it robust to noise, and apart from the number of partitioning cuts no parameters are required. In order to ensure that the method operates correctly on bent or undulating shapes a process is developed that identifies the underlying bending and removes it, straightening out the shape. Results are shown on a large range of shapes.

1 Introduction

Shape obviously plays an important role in biological vision. However, the task of shape perception is inherently complex, as demonstrated by the slow developmental process of learning undertaken by children to recognise and use shape. At first, they can only make topological discriminations. This is then followed by rectilinear versus curvilinear distinctions, then later by angle and dimension discrimination, and then continuing to more complex forms, etc. [11].

The application of shape in computer vision has been limited to date by the difficulties in its computation. For instance, in the field of content based image retrieval, simple methods based on global colour distributions have been reasonably effective [19]. However, although attempts have been made to incorporate shape, they are still relatively crude [6,14].

One approach is to simplify the problem of analysing a shape by breaking it into several simpler shapes. Of course, there is a difficulty in that the process of partitioning will require some shape analysis. However, to avoid a chicken and egg problem, low level rules based on limited aspects of shape understanding can be used for the segmentation.

There are numerous partitioning algorithms in the computer vision literature. Many are based on locating significant concavities in the boundary [16,18]. It has been shown that humans also partition forms based on, or at least incorporating, such information [4,8,10]. While this lends credence to such an approach, the computational algorithms employed to detect the concavities generally depend on measuring the curvature of the shape's boundary. Unfortunately curvature estimates are sensitive to noise. Although the noise can be reduced or eliminated by smoothing for instance, it is not straightforward to determine the appropriate degree of filtering. In addition, purely local boundary-based measures do not

C. Arcelli et al. (Eds.): IWVF4, LNCS 2059, pp. 440–450, 2001.

capture the important, more global aspects, of shape. An alternative approach that can incorporate more global information operates by analysing the skeleton of the shape [1,3]. Nevertheless, it remains sensitive to local detail and tends to be error prone since even small amounts of noise can introduce substantial variations into the skeleton. Thus substantial post-processing generally needs to be applied in an attempt to correct the fragmentation of the skeleton.

2 Convexity-Based Partitioning

To overcome the unreliability of the curvature and skeleton based methods an approach to partitioning was developed based on global aspects of the shape [12]. The criterion for segmentation was convexity. Although convexity had been used in the past, previous algorithms still required various parameters to tune their performance, generally to perform the appropriate amount of noise suppression [7,15]. In contrast, the only parameter in Rosin's formulation was the number of required subparts. Moreover, an approach for automatically determining this number was also suggested.

The convexity of a shape was measured as the ratio of its area to the the area of its convex hull. The total convexity of a partitioned shape was defined as the sum of the individual convexity values of the subparts, each weighted by their area relative to the overall shape's area. Thus the convexity and combined subpart convexity values range from zero to one. Partitioning was performed by selecting the decomposition maximising convexity. As with most partitioning schemes, straight lines were used to cut the shape into subparts, and the cuts were constrained to lie within the shape.

The convexity measure is robust since small perturbations of the shape boundary only result in small variations in the convex hull. Thus noise has a minor effect on the areas of the shape and its convex hull, and therefore on the convexity measure itself. Although the measure is based on global properties of the shape it produces good partitions, often locating the cuts at significant curvature extrema even though no curvature computation is necessary.

3 Shape Straightening

Despite its general success, there are also instances in which the convexity based scheme fails [12]. In figure 1 the effect of the ideal cut (shown dotted) would be to split the crab shape into the inner convex part and the outer non-convex part. The latter would score very poorly according to convexity, and so the crab would actually receive a better score without performing any partitioning. This is counterintuitive since we would expect that partitioning should always lead to simplification. In general, we make the observation that many bent objects will be given a low convexity rating even though human perception might suggest that they are suitable representations of simple, single parts.

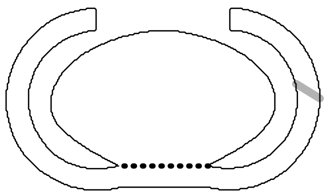
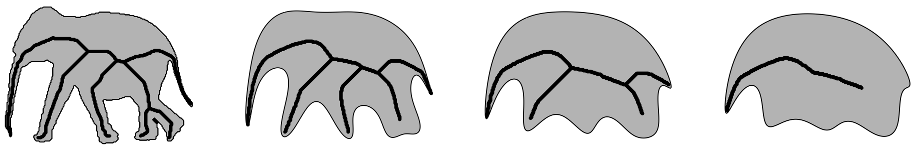


Fig. 1. Convexity is not always appropriate for partioning as demonstrated in the crab figure. Instead of the ideal cut (shown dotted) the gray cut is selected.

In this section we describe a solution to overcome the deficiency of convexity in this context. Since it is the bending of the shape that creates the problem, the bending is removed. Conceptually, if the outer portion of the crab were straightened it would receive a high convexity score since there would be no concavities. Of course, the straightening process should not eliminate all concavities *per se* since these are required to enable the convexity measure to discriminate good and bad parts. Instead, the most basic underlying bending should be removed while leaving any further boundary details unchanged.

Some work in computer vision and computer graphics has looked at multiscale analysis and editing of shapes. For instance, Rosin and Venkatesh [13] smoothed the Fourier descriptors derived from a curve in order to find "natural" scales. The underlying shape was then completely removed by modifying the lower descriptors such that on reconstruction just the fine detail occurring at higher natural scales was retained and superimposed onto a circle. Another approach was taken by Finkelstein and Salesin [17] who performed wavelet decompositions of curves and then replaced the lower scale wavelets extracted from one curve with those of another. Although both methods enabled the high resolution detail to be kept while the underlying shape was modified there were several limitations. The Fourier based approach operates globally, and therefore assumes uniform detail spatially distributed over the curve, which is not necessarily correct. Wavelets have the advantage that they cope with spatial localisation, but Finkelstein and Salesin did not provide any means for automatically selecting which wavelets to retain such that they correspond to significant curve features.

The approach taken in this paper is to determine the appropriate straightening of a shape by first finding its medial axis. Its sensitivity to noise can be overcome since the axis is only required to describe of the shape at a very coarse level, and so heavy smoothing can be applied to eliminate all branches as shown in figure 2. More precisely, the boundary is repeatedly smoothed, and at each step the branches in the resulting medial axis are identified by checking for vertex pixels. If no vertex pixels are found the smoothing terminates and the final boundary and axis is returned. Our current implementation uses: Gaussian blurring of the boundary, Zhang and Suen's thinning algorithm to extract the medial axis [20], and vertices are identified by checking at each axis pixel for three of more black/white or white/black transitions while scanning in rotation around its eight neighbours.



(a) (b) (c) (d)

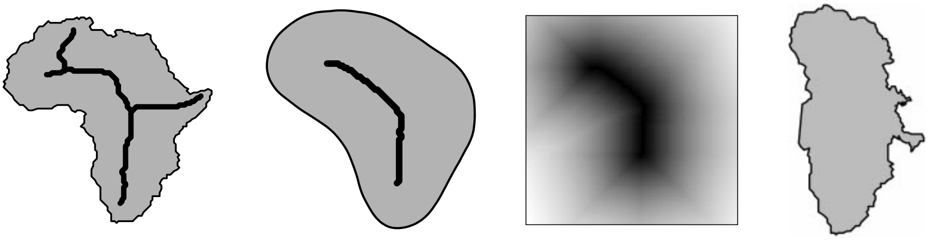
Fig. 2. Repeated boundary smoothing applied until all medial axis branches are eliminated.

Once the axis is found it is used to straighten the shape. First each boundary point needs to be assigned to a point on the axis. Conceptually this can be performed by regenerating the shape by growing the axis. In practise we just run a distance transform [2] taking the axis as the feature set. In addition to propagating the distances the originating co-ordinates of the closest feature are also propagated. These then provide the corresponding axis points for each boundary point. At this stage the smoothed boundary points are still used (after integer quantisation) rather than the original boundary set.

Second, a local co-ordinate frame is determined for each boundary point. The frame is centred at the corresponding axis point and orientated to align with the local section of axis. The orientation is calculated by fitting a straight line to the ten axis pixels on either side of the centre. The position of each point in the original shape boundary is now represented in polar co-ordinates with respect to the local co-ordinate frame determined for its corresponding smoothed point.

The third step performs the straightening of the boundary by first straightening the medial axis. The axis points $(x_i, y_i)_{i=1...n}$ are mapped to $(i, 0)$, giving the straight line $(0, 0) \rightarrow (n, 0)$. Transforming the local co-ordinate frames to be appropriately centred and oriented the transformed boundary points have now been straightened.

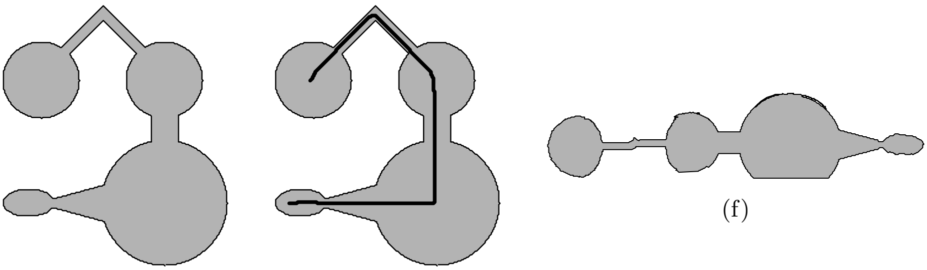
An example of the full process is given in figure 3. The irregular map of Africa is smoothed until its medial axis represents just the underlying bent shape. The distance transform of the medial axis is shown in figure 3c where low intensities represent small distances. The final, straightened map of africa in figure 3d clearly demonstrates that the original major bend has been removed



(a) (b) (c) (d)

Fig. 3. The straightening process. (a) The irregular outline of the input shape produces a skeleton with several branches. (b) The shape is iteratively smoothed until its skeleton consists of a single spine. (c) The distance transform of the skeleton is generated, and the X and Y co-ordinates of the closest axis point are recorded. This enables the appropriate transformation to be applied to the original shape, resulting in (d).

while the local boundary features have been retained, although slightly distorted in some instances.



(d) (e) (f)

Fig. 4. Examples of shape straightening. The first column contains the original shape; the second column contains the smoothed shape with the medial axis; and the third column contains the straightened shape.

The validity of the approach is demonstrated on the synthetic examples in figure 4. Due to the nature of the data the medial axis is easily found, and reliably represents the basic form of the underlying shape. The final results show that the straightening is performed correctly.

Further examples of shape straightening are provided in figure 5. For simple elongated shapes the technique is generally successful as the medial axis is representative of the bending underlying the shape. Cases in which there are several competing axes are more problematic. For instance, in the donkey there is a dominant elongated horizontal portion as well as three vertical elongated portions (the donkey's fore feet and rear feet, and the rider). No single unbranch-

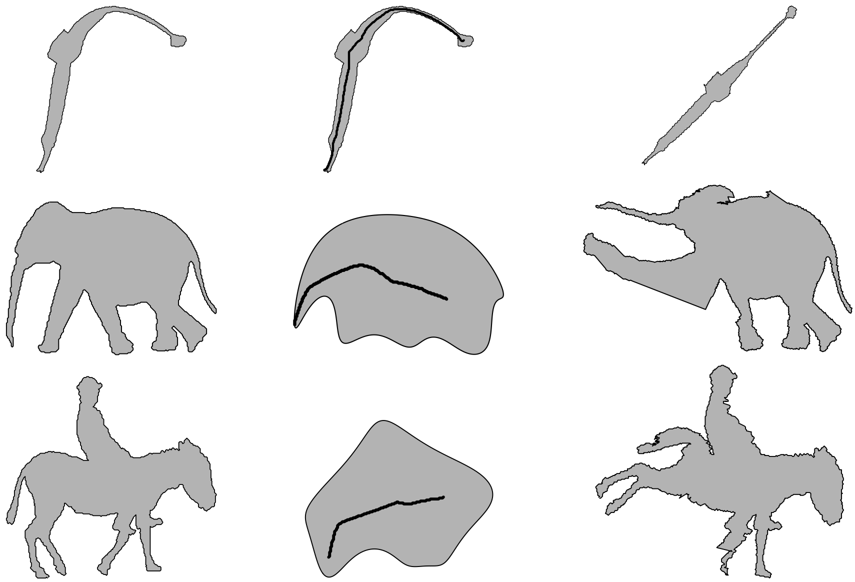


Fig. 5. Examples of performing shape straightening of natural data

.

ing axis can capture all this. Nevertheless, the result successfully straightens the donkey's rear feet and tail even though the rider and fore feet remain protruding. A similar partial straightening is seen on the elephant. In some cases local distortions are evident. Such errors creep in from a combination of sources such as the distance transform approximation, fitting of the local co-ordinate frame, and the mapping itself.

4 Partitioning

The partitioning algorithm is now complete. Its operation is much as before: candidate cuts are assessed and the one maximising the weighted sum of convexities is selected. However, before calculating convexity the subpart is first straightened out. Since this transformation can distort the size as well as shape of the subpart the total convexity of the set of subparts is combined using the individual subpart convexities weighted according to the relative area of the *unstraightened* subparts.

The results of the two algorithms are compared in the following figures in which the different levels of performance have been grouped. It can be seen that in many cases both methods produce the same or very similar results (figures 6 and 7). Sometimes the original algorithm is still successful despite the shape containing significant bending. For instance, although the fish's tail wiggles back and forth it is thin. Thus its low area causes its low convexity to only contribute weakly to the overall high convexity generated mainly by the highly convex fish body. The results in figure 7 verify that the incorporation of straightening does not prevent the new algorithm from performing satisfactorily.

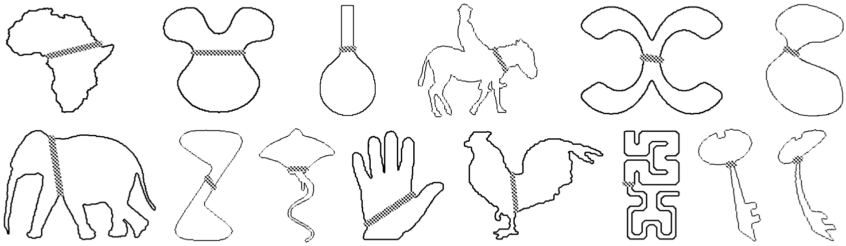


Fig. 6. Similar partitioning using convexity alone

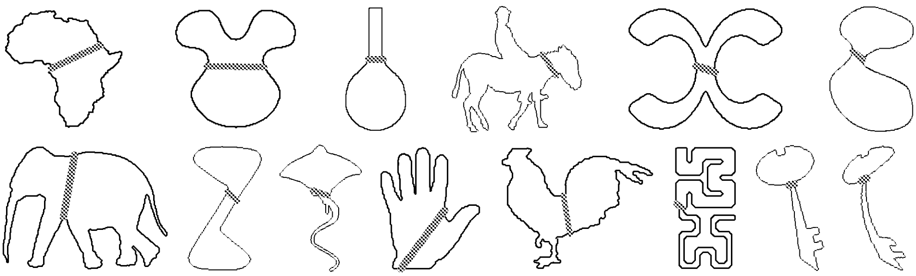


Fig. 7. Similar partitioning using convexity in combination with straightening

Figures 8 and 9 contain results that differ significantly between the algorithms, although it is not clear that either one is superior. For instance, the original algorithm has cut off one of the curved arms in the second shape. By incorporating straightening the new algorithm has managed to successfully combine both arms into a single part.

In some cases we find that the addition of straightening worsens the effectiveness of the method (see figures 10 and 11). The head is better partitioned by the old algorithm, although the new algorithm's result is still reasonable. By making a cut from the nose to the back of the head it has created a region that was straightened into a fairly convex shape. On the last shape the new algorithm's

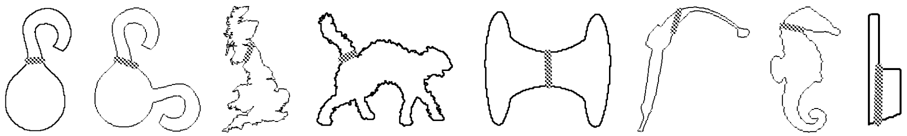


Fig. 8. Different partitioning using convexity alone

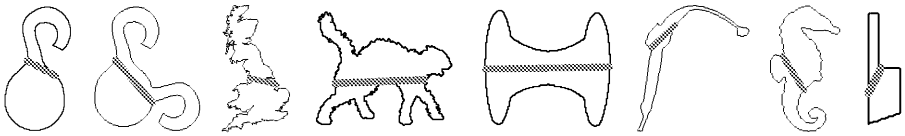


Fig. 9. Different partitioning using convexity in combination with straightening

result is poor, although a contributing factor is that it needs to be partitioned into more than two subparts.

Fig. 10. Better partitioning using convexity alone

Fig. 11. Worse partitioning using convexity in combination with straightening

Finally, examples in which straightening has provided a clear benefit are given in figures 12 and 13. In most cases the failings of using convexity alone are self-evident – sections are chopped off with no regard for their fitness as subparts.

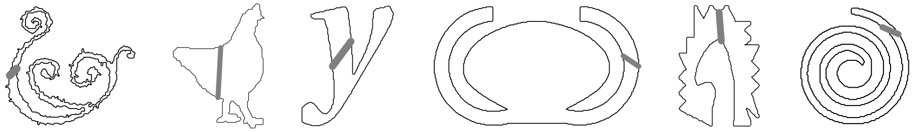


Fig. 12. Worse partitioning using convexity alone

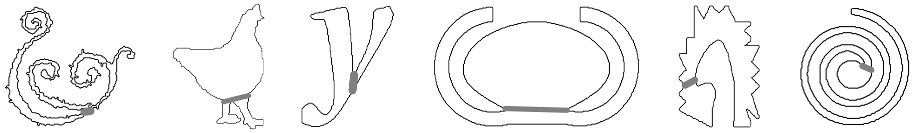


Fig. 13. Better partitioning using convexity in combination with straightening

5 Discussion

In this paper we have shown how shapes can be straightened, and how this can be applied to aid partitioning. Several issues remain, relating to the efficiency and effectiveness of the technique.

Currently, the straightening process is time consuming, and can take several seconds. Since it is applied repeatedly as part of the evaluation of many candidate cuts this slows down the overall analysis of a shape containing a thousand points to several hours. The actual time depends on the shape since if it contains many concavities such as the spiral then many of the trial cuts will lie outside the shape and can therefore be rejected without requiring the more time consuming straightening and convexity calculations.

Two approaches to speeding up the process are possible. The first is to improve the efficiency of the straightening process. The current implementation involves some image based operations (for the medial axis calculation and axis branch checking). A significant improvement could be made by determining the medial axis directly from the shape boundary. Efficient algorithms exist, in particular, Chin *et al.* [5] recently described an algorithm that runs in linear time (with respect to the number of polygon vertices).

Another complementary approach is to apply the convexity calculation only at selected cuts. Rather than exhaustively considering all possible pairwise combinations of boundary points as potential cuts, a two stage process can be employed. For example, the cuts can be restricted to include only a subset of boundary points such as dominant (i.e. corner) points. Although corner detectors are typically unreliable, if a low threshold is used then the significant points will probably be detected, at the cost of also including additional spurious points. Alternatively, a simpler but less reliable partioning algorithm can be used to produce a set of candidate cuts by running it over a set of parameter values. These can then be evaluated and ranked by convexity.

At the moment the only speed-up implemented is to process the data at multiple scales. First the curve is subsampled, typically at every fourth point. The best cut is determined and this initialises a second run at full resolution in which only cuts around the best low resolution cut are considered. Currently the window centred at the first cut is six times the sampling rate

Regarding the effectiveness of the straightened convexity measure some improvements could be made. As discussed previously, the measure does not explicitly take curvature extrema into account. Nevertheless these are important local features even though their reliable detection is problematic.

On the issue of the saliency of alternative partitionings, Hoffman and Singh [9] ran psychophysical experiments to determine three factors affecting part salience: relative area, amount of protrusion, and normalised curvature across the part boundary. Previously the convexity measure was demonstrated to match fairly well on a simple parameterised shape with these human saliency judgements [12]. However, there remain examples in which the basic convexity and the straighted convexity measures cannot discriminate between alternative partitions of different quality. For instance, most humans would judge the first segmentation in figure 14 as more intuitive than the second, but after straightening both receive perfect convexity scores.

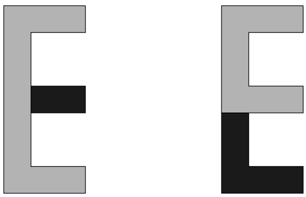


Fig. 14. Alternative partitions with identical straightened convexity ratings

Finally, the straightening process works well for elongated, bent shapes, but can run into problems with shapes containing several competing dominant axes. Simplifying the axes in order to remove all the vertices requires a large amount of smoothing leading to distortion of the shape.

References

1. C. Arcelli and G. Sanniti di Baja. An approach to figure decomposition using width information. *Computer Vision, Graphics and Image Processing*, 26(1):61–72, 1984.
2. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):344–371, 1986.
3. G. Borgefors, G. Ramella, and G. Sanniti di Baja. Permanence-based shape decomposition in binary pyramids. In *10th Int. Conf. on Image Analysis and Processing*, pages 38–43, 1999.

4. M.L. Braunstein, D. Hoffman, and A. Saidpour. Parts of visual objects: an experimental test of the minima rule. *Perception*, 18:817–826, 1989.
5. F. Chin, J. Snoeyink, and C. Wang. Finding the medial axis of a simple polygon in linear time. *Discrete Computational Geometry*, 21(3):405–420, 1999.
6. M. Flickner *et al.* Image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.
7. A. Held and K. Abe. On the decomposition of binary shapes into meaningful parts. *Pattern Recognition*, 27(5):637–647, 1994.
8. D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.
9. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63:29–78, 1997.
10. C. Lamote and J. Wagemans. Rapid integration of contour fragments: from simple filling-in to parts-based shape description. *Visual Cognition*, 6:345–361, 1999.
11. J. Piaget and B. Inhelder. *The Child's Conception of Space.* Routledge and Kegan Paul, 1956.
12. P.L. Rosin. Shape partitioning by convexity. *IEEE Trans. on Sys., Man, and Cybernetics*, part A, 30(2):202–210, 2000.
13. P.L. Rosin and S. Venkatesh. Extracting natural scales using Fourier descriptors. *Pattern Recognition*, 26:1383–1393, 1993.
14. S. Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):627–641, 1997.
15. L.G. Shapiro and R.M. Haralick. Decomposition of two-dimensional shapes by graph-theoretic clustering. *IEEE Trans. PAMI*, 1(1):10–20, 1979.
16. K. Siddiqi and B.B. Kimia. Parts of visual form: Computational aspects. *IEEE Trans. PAMI*, 17(3):239–251, 1995.
17. E. Stollnitz, T. DeRose, and D. Salesin. *Wavelets for Computer Graphics.* Morgan Kaufmann, 1996.
18. K. Surendro and Y. Anzai. Decomposing planar shapes into parts. *IEICE Trans. Inf. and Syst.*, E81-D(11):1232–1238, 1998.
19. M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Computer Vision*, 7(1):11–32, 1991.
20. T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns. *Comm. ACM*, 27(3):236–240, 1984.