

A Symmetric Convexity Measure

Paul L. Rosin and Christine L. Mumford

Cardiff School of Computer Science, Cardiff University, Cardiff, UK
{Paul.Rosin@cs.cf.ac.uk, C.L.Mumford@cs.cardiff.ac.uk}

Abstract

A new area-based convexity measure for polygons is described. It has the desirable properties that it is not sensitive to small boundary defects, and it is symmetric with respect to intrusions and protrusions. The measure requires a maximally overlapping convex polygon, and this is efficiently estimated using a genetic algorithm. Examples of the measure's application to medical image analysis are shown.

1. Introduction

Convexity is a useful attribute of shape, and has applications in classification, image segmentation, figure/ground separation, etc. Whereas its mathematical definition provides a binary property, in image analysis we prefer a continuous measure, so that a shape can be assigned a degree of convexity. This provides more information, enabling classification for example to be more discriminating, and also allows the concept of convexity to be applied more usefully and robustly to real life irregular and noisy shape data.

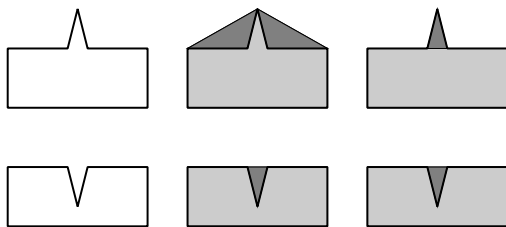


Figure 1. Measuring convexity of two shapes (left); middle: standard convex hull based approach, right: shows proposed approach.

Many convexity measures for polygons in the literature are either area [1, 11] or perimeter based [15]. The most common computes the ratio of the area of the polygon to the area of its convex hull [10], $C = \frac{A(P)}{A(CH(P))}$. This measure is nominally area-based, and so narrowing the intrusion (i.e.

decreasing its area) in the polygon in the lower row in figure 1 increases the measured convexity. In the limit, as the area of the intrusion tends to zero (an infinitely thin cut), the convexity tends to one (perfect convexity). However, there is an obvious asymmetry in the measure since a similar but opposite defect, namely the protrusion in the upper row in figure 1 produces a lower measured convexity. Even worse, in this example as the protrusion narrows the measured convexity decreases rather than increases, disagreeing with our perception. Thus, the measure inherits the convex hull's sensitivity to protrusions. Such sensitivity to thin protrusions is more appropriate to perimeter based convexity measures such as Zunic and Rosin's [15] method which is also sensitive to thin intrusions.

2. The New Convexity Measure

In this paper we propose an alternative area-based convexity measure related to C which has the advantages that it is not sensitive to small (in area) boundary defects, and it is symmetric with respect to intrusions and protrusions. The basic idea is to replace the convex hull of the polygon P by its "robustified" version which we define as the convex polygon Q that best fits P in the sense of maximising the overlap of P and Q , or equivalently minimising the area of $P \text{ XOR } Q$. The right column in figure 1 highlights $P \text{ XOR } Q$, which is the same for the two shapes. To make the measure scale invariant it needs to be normalised with respect to size. This can be done using the area of the input polygon $E_P = \frac{A(P \text{ XOR } Q)}{A(P)}$, or alternatively by the area of the fitted convex polygon $E_Q = \frac{A(P \text{ XOR } Q)}{A(Q)}$. The latter ensures greater symmetry (see figure 1) but is potentially less robust than the former if Q is not estimated accurately. To make the measure maximal for convex shapes E_P is modified to form the convex measure $C_P = 1 - \frac{A(P \text{ XOR } Q)}{A(P)}$. We have tested C_P on hundreds of polygons and confirmed that it lies in the range $[0, 1]$, although this has not been proven analytically. Although E_P lies in $[0, 1]$, E_Q is not bounded from above. This can be demonstrated by the star shaped polygon in figure 2. For a star with very short arms Q is a

regular polygon centred on the star. However, as the arms grow outwards the space between them grows much faster than the increased area of the arms. This leads to Q flipping to fit one of the arms, and so for an n -sided star (n is odd) $E_Q = n - 1$. We therefore normalise E_Q to form the measure $C_Q = \frac{1}{1+A(P \text{ XOR } Q)/A(Q)}$ to keep it in $[0, 1]$.

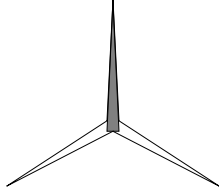


Figure 2. The convex polygon Q is fitted to one arm of the star.

Given the polygon Q it is now also possible to use it to define two new shape measures. The amount of protrusions and intrusions can be quantified by $\frac{A(P \setminus Q)}{A(Q)}$ and $\frac{A(Q \setminus P)}{A(Q)}$. Only the former needs to be normalised, and so the measures of protrusiveness and intrusiveness are $C_{\text{prot}} = \frac{A(P \setminus Q)}{A(P \setminus Q) + A(Q)}$ and $C_{\text{int}} = \frac{A(Q \setminus P)}{A(Q)}$.

3. A Genetic Algorithm for Estimating Q

Finding the convex polygon Q is not a trivial task, and for convenience we estimate Q using a general purpose optimiser, namely a genetic algorithm. This approach also enabled us to easily experiment with alternative versions of the criteria for defining Q . In our current formulation, for the input polygon P , the GA's function is to search the space of vertex subsets for a derived polygon Q with the following properties:

1. Q is convex (a hard constraint), and
2. Q produces a minimum value when its area is XORed with P (i.e. the functional).

Vertex subsets are represented in the GA by bit strings in which bit i is set if and only if vertex i , from the original polygon, is included in the subset.

Figure 3 gives an outline of our simple genetic algorithm. It is an example of a steady state GA [13], and it uses the weaker parent replacement strategy first described by Cavicchio in [2]. The current GA, first used in [9], is particularly easy to implement because it lacks a requirement for global selection probabilities: in our GA all parents are selected from a uniform distribution, and all pressure advancing the genetic search is applied at the replacement stage of the GA. Similar ideas have been used successfully for multi-objective GAs [14].

Procedure GA

begin

Generate and repair a population of N individuals
 Evaluate the objective function for each individual and store it
 Store *best-so-far*

Repeat

For each member of the population

This individual becomes the first parent

Select a second parent at random

Apply crossover to produce offspring

Apply a single mutation to the offspring

Repair the offspring

Evaluate the objective function produced by offspring

If offspring is a duplicate, delete it

Else

If offspring better than weaker parent then
 it replaces it in population

If offspring better than *best-so-far* then it replaces
best-so-far

Endfor

Until stopping condition satisfied

Print *best-so-far*

End

Figure 3. A steady state genetic algorithm.

Each time the GA iterates through its inner loop, it selects two parents and applies *one-point crossover* to the bit strings to produce a single offspring. A single *point mutation* is then applied to the offspring which involves selecting a bit at random and flipping it, either from zero to one or from one to zero. The pairs of parents are selected in the following way: the first parent is selected deterministically in sequence, but the second parent is selected uniformly, at random.

A problem specific repair routine has been developed for the present application, and this ensures that all candidate solutions are valid. The routine is needed because without it the GA will waste vast amounts of time generating and evaluating large quantities of infeasible solutions. The repair routine is applied to each individual in the initial population, and also to every offspring. It accepts, as arguments, an arbitrary subset of vertices from a polygon generated by the GA, and returns the convex hull of this subset. The vertices of the convex hull are then written back into the relevant bit string, guaranteeing a valid candidate solution. The area of the convex derivative polygon is finally XORed with the area of the original polygon to generate a fitness value.

A final feature of the GA is that no new offspring is allowed to replace its weaker parent until it has been checked as a duplicate. If a duplicate of the offspring is present in the population then the new offspring will die immediately. Our GA deletes *phenotypic* duplicates i.e., new offspring are deleted if they duplicate a current population member's objective function (although we also check that both individuals also have the same number of vertices). Ideally

we should delete *genotypic* duplicates (i.e. matching bit strings), but this would involve performing time consuming pairwise comparisons on long chromosomes.

4. Computational Complexity

The overall running time of the genetic algorithm is dependent on the computational complexity of generating and evaluating each candidate solution. The crossover and mutation operations have linear complexity, as is generating the polygon from the vertex bit string defining an individual. To repair the offspring requires computing the convex hull. For simple (non-intersecting) polygons the computation complexity is $O(n)$ [8], however, the process of generating individuals does not guarantee that the ordered sequence of vertices is simple. The more general case of computing the convex hull of a set of points is $O(n \log n)$. To avoid this extra overhead the candidate polygon Q could be tested for simplicity. In theory this can be performed in linear time as a consequence of Chazelle’s linear-time triangulation algorithm [4]. However, in practise we have found experimentally that non-simple polygons are generated fairly rarely, and it is therefore more efficient to use the linear complexity convex hull algorithm, and check the final solution produced by the GA. If it self-intersects (this occurred in less than 1% of the cases during our testing) then the GA is run again with a more general $O(n \log n)$ convex hull algorithm.

Computing the XORed area for evaluating individuals is the most complex operation in the processing pipeline. Performing polygon intersection operations takes $O(n \log n + k \log n)$ time, where the two polygons have n_1 and n_2 vertices, $n = n_1 + n_2$, and k is the complexity of the output [5]. The number of intersection between a concave and convex polygon is $\max(2n_1, 2n_2)$ [12], and so the overall complexity of intersection is $O(n \log n)$.

5. Qualitative Evaluation

The new convexity measure was applied to 250 polygons from a variety of sources, covering both large and small shapes (between 200 and 5000 pixels), real and synthetic, and computation typically took a few seconds. A sample of the results is shown in figure 4, ranked in order of increasing convexity. The first point is that the results appear intuitively perceptually correct. It is also of interest to note that for polygons consisting of a mainly convex part with intrusions then the fitted polygon Q (shown shaded) is roughly the convex hull of P . Likewise, for polygons consisting of a mainly convex part with protrusions, Q is roughly the convex skull (i.e. the largest convex subpolygon) of P . Although the concept of the convex skull might be considered

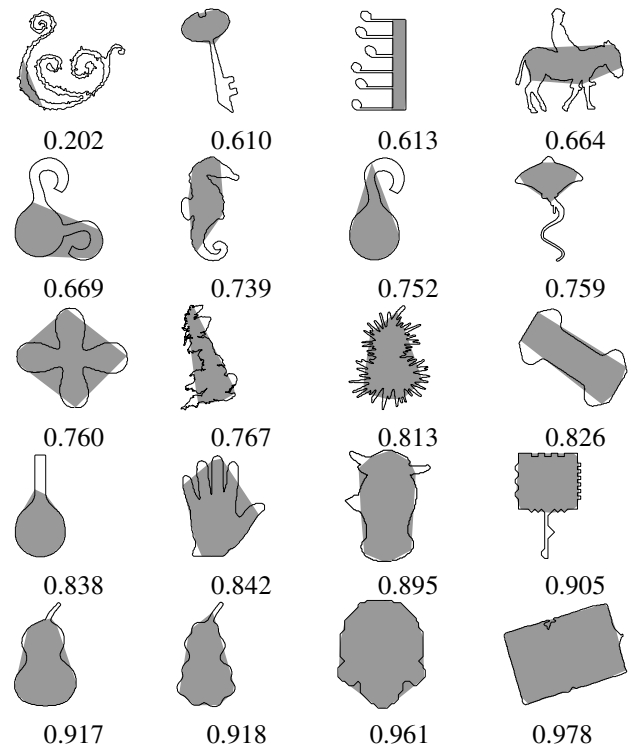


Figure 4. Some polygons P with their convexity values C_Q shown below. P is overlaid on a filled version of Q .

similar to the convex hull, its computation is much more difficult and expensive – the fastest known algorithm [3] runs in $O(n^9)$ time!

6. Experiments

6.1. Lesions

We demonstrate the convexity measure on a medical image analysis task. Lee *et. al* [7] presented results for classifying lesions as either benign or malignant melanomas based on the irregularity of their boundaries. They had 14 dermatologists rate a set of 40 lesions from a four point scale according to their probability of being a melanoma. These ratings were averaged over the 14 experts and then compared against various shape measures by computing the Spearman rank correlation between them. Their “overall irregularity index” achieved a correlation value of 0.88 outperforming the alternative measures they considered. However, we have found that the standard convexity measure C performs just as well. Moreover, applying our new measures we see that they do substantially better (table 1). It is interesting to note that in this application intrusions appear to be substantially more salient than protrusions.

C	C_Q	C_P	C_{int}	C_{prot}
0.888	0.952	0.959	0.952	0.745

Table 1. Absolute Spearman rank correlation scores for lesion data.

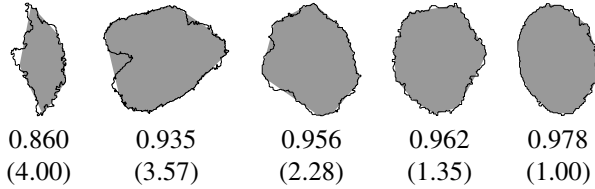


Figure 5. Examples of lesions with C_Q values and the mean expert score underneath.

6.2. Greebles

Among the 250 polygons tested were 49 “greebles”; some examples are shown in figure 6.¹ Since the appearance of members within this class is qualitatively similar, despite their individual differences, their measured shape attributes should also be similar. C was compared with C_Q by determining the ranks of the greebles in the set of 250 polygons ordered by each of the convexity measures. The standard deviation of the greeble ranks was 27.66 for C and 22.99 for C_Q ; the lower value demonstrating the improved stability and consistency of C_Q .

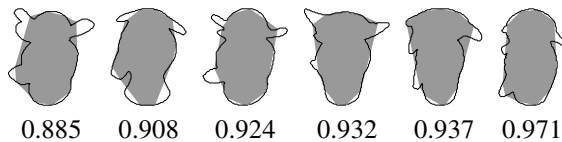


Figure 6. A selection of greebles covering their full range of convexity values.

7. Conclusions

A new area-based convexity measure for polygons has been defined, and a genetic algorithm based solution has been described to efficiently compute the measure. Compared to the traditional convexity measure it is more robust

¹Greebles are a popular source of test objects used in psychology, e.g. [6]. They are objects synthesised to a standard configuration: a vertically-oriented body with four protrusions: two “boges”, a “quiff” and a “dunth”.

and symmetric, and was shown to outperform it on a medical analysis task involving the estimation of the likelihood of melanoma from lesion boundaries.

8. Acknowledgements

I would like to thank Alan Murta for his “Generic Polygon Clipper” code, and Ben Kimia for providing some of the polygon data.

References

- [1] L. Boxer. Computing deviations from convexity in polygons. *Pattern Recognition Letters*, 14:163–167, 1993.
- [2] D. Cavicchio. *Adaptive Search Using Simulated Evolution*. PhD thesis, University of Michigan., 1970.
- [3] J. Chang and C. Yap. A polynomial solution for the potato-peeling problem. *Discrete Comput. Geom.*, 1:155–182, 1986.
- [4] B. Chazelle. Triangulating a simple polygon in linear time. *Computational Geometry*, 6:485–524, 1991.
- [5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2nd edition, 2000.
- [6] I. Gauthier, P. Williams, M. Tarr, and J. Tanaka. Training greeble experts: a framework for studying expert object recognition processes. *Vision Research*, 38:2401–2428, 1998.
- [7] T. Lee, D. McLean, and M. Atkins. Irregularity index: A new border irregularity measure for cutaneous lesions. *Medical Image Analysis*, 7(1):47–64, 2003.
- [8] D. McCallum and D. Avis. A linear algorithm for finding the convex hull of a simple polygon. *Inform. Process. Lett.*, 9:201–206, 1979.
- [9] C. Mumford-Valenzuela, J. Vick, and Y. Pearl. Heuristics for large strip packing problems with guillotine patterns: An empirical study. In D. Du and P. Pardalos, editors, *Metaheuristics: Computer Decision-Making*. Kluwer Academic Press, 2003.
- [10] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Chapman and Hall, 1993.
- [11] H. Stern. Polygonal entropy: a convexity measure. *Pattern Recognition Letters*, 10:229–235, 1989.
- [12] S. Suri. Poygons. In J. Goodman and J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.
- [13] G. Syswerda. Uniform crossover in genetic algorithms. In *Proc. Third Int. Conf. on Genetic Algorithms*, pages 2–9. Lawrence Erlbaum Associates, 1989.
- [14] C. Valenzuela-Mumford. A simple evolutionary algorithm for multi-objective optimization (SEAMO). In *Congress on Evolutionary Computation (CEC)*, pages 717–722, 2002.
- [15] J. Žunić and P. Rosin. A convexity measurement for polygons. In *Proc. British Machine Vision Conf.*, pages 173–182, 2002.