Tools for Asymmetry Rectification in Shape Design

R. R. Martin*
Department of Computing Mathematics,
University of Wales College of Cardiff
and
D. Dutta
Department of Mechanical Engineering,
University of Michigan

August 25, 1994

Abstract

This paper considers the task of asymmetry rectification. We start by giving various reasons why the possession of symmetry may be beneficial for designed shapes, and mention how various construction methods may produce shapes which are less symmetric than desired. We then consider how to make asymmetric parts symmetric, a process we call asymmetry rectification. In detail, we break this process into three stages—the detection of existing partial- or almost-symmetries, the choice of the parameters of the symmetry to be imposed, and finally construction of the symmetric shape from the original shape. Various possible approaches to each of these steps are considered, and their relative merits discussed.

1 Introduction

This paper concerns itself with the question of how to make an asymmetric shape symmetric; often before this can be done it is also necessary to decide what symmetry should be imposed. In this introduction we commence by briefly justifying why such an operation is useful in the context of CADCAM. This will be done in two stages: firstly, we will discuss why symmetric shapes may be more desirable than asymmetric ones. Secondly, we will consider how asymmetric shapes might arise when symmetric ones would be desired.

The simplest reason a symmetric shape might be preferred is for aesthetic reasons—the human mind finds such shapes more orderly and acceptable to look at. People expect asymmetries to be present in designed components only when there is some particular need for them.

Manufacturing considerations also favor symmetric components. Firstly, symmetric components are more regular and are thus often easier to make, requiring simpler tooling and less complex set-ups. If several slots are all the same, they may be manufactured with the same cutter, for example.

---

[*] This work was done whilst this author was visiting the University of Michigan.

Secondly, as explained in greater detail elsewhere [16], when assembling complex objects from their parts, advantages are to be obtained if some of those parts are symmetric. For example, such parts may be assembled in one of several equivalent orientations, and thus time need not be spent in determining the correct orientation of the part before assembly. In particular, objects which are *slightly* asymmetric present a greater problem than those which are more asymmetric, and indeed, measures of difficulty of assembly dependent on asymmetry have been proposed [14]. Thus, symmetrizing components of an assembly may well save manufacturing time and costs.

Other reasons for ensuring that parts are symmetric include that finite element analysis need only consider a fraction of the shape, making analysis quicker (or more accurate) [16]; similarly if a prototype is to be made, it may be possible to use one which is a fraction of the shape, saving material costs and labour time. If an object is sufficiently symmetric, it may be possible to find analytical rather than numerical solutions to various questions concerning the object's properties. In a different area, object recognition [24] or reconstruction [27] by a computer vision system become easier tasks. Indeed, if a vision system captures an image of an occluded object which is known (or guessed) to be symmetric, reconstructing the overall shape from the visible part of the profile is one application of methods of constructing symmetric shapes from asymmetric ones.

This example leads to the more general question of when asymmetric shapes may arise if symmetric ones are desired. While many shape designs are carefully constructed by human operators, others may be generated in whole or in part by automatic methods. For example, shapes may be derived from optimization processes, such as homogenization design [4]. They may arise as a result of computer interpretation of freehand sketched input [13]. They may be produced by methods which fit curves and surfaces to data captured by such devices as 2D and 3D scanners, and coordinate measuring machines. For example, it may be desired to capture an existing font using a scanner. That font may contain various characters which should be symmetric. For all such data sources there may be both aesthetic and practical reasons for making such shapes more symmetric.

We refer to the process of making asymmetric shapes more symmetric as *asymmetry rectification.* The rest of this paper considers how such a task might be performed. Before an object can be made symmetric, two problems must be solved. The first is to decide what type of symmetry to impose (for example, 3-fold rotational symmetry), and the second is to choose the parameters defining that symmetry (for 3-fold rotational symmetry, where the axis should be). As might be expected, using an approach which treats these problems as completely separate may not be optimal. Ultimately we will propose a combined solution, but it will help initially to keep these two ideas separate. Finally, after the symmetry and its parameters have been chosen, a method is needed to create a new object with the desired symmetry from the original object. Each of these three main steps is considered in turn in the following Sections of this paper.

2 Partial- and Almost-Symmetry Detection

In some cases, the user may decide in advance what symmetries are to be imposed on an object, but in many other cases, such as when software is automatically tidying freehand sketched input, these symmetries must be determined. Here we will consider various approaches to detecting inexact symmetries, but first we describe what we mean by two main types of inexact symmetry, partial-symmetry and almost-symmetry. We will also show how symmetry measures may be used to decide which symmetry an object most nearly has.

2.1 Symmetry, Partial-Symmetry and Almost-Symmetry

A symmetric object is one which is invariant under a given transformation. Thus, an object has a mirror symmetry if it is invariant on reflection in a line (in 2D) or in a plane (in 3D). An object has an n-fold rotational symmetry if it is invariant upon rotation through an angle of $2\pi/n$ about some point (in 2D) or axis (in 3D); a special case arises when an object has circular (or spherical) symmetry and is invariant under rotation through any (i.e. an infinitesimal) angle. An *infinite* object may also have translational symmetry whereupon it is invariant under certain translations, as is the case for the pattern of a tiling. In practice, a weaker form of translational invariance may be of interest, where a *finite* object has certain features which are congruent to each other under translation, such as the teeth on a rack. We consider this to be an example of a *partial-symmetry.* This term will also be used to denote other cases where just part of the boundary of an object is invariant under transformation, while other elements of the boundary violate that symmetry. For example, a complex piece of equipment may have a pair of mirror-image handles on either side for picking it up, even though the rest of the equipment is definitely not symmetric. Another example is shown later in Figure 6.

The idea of symmetry can also be extended to include self-similarity, whereupon the object is invariant under a scaling. Strictly, this too can only be a property of an infinite object, but again the idea can be applied to parts of a finite object. We will ignore similarities in the rest of this paper, as they can in principle be dealt with by first normalizing the size of the object's features being compared, and then searching for congruences rather than similarities.

An object may only have certain combinations of symmetries. For example, when considering finite objects in 2D, the only possible symmetries are C_n, n-fold rotational symmetries for $n \geq 2$, and $D_n, n \geq 1$, where n-fold rotational symmetry is combined with n mirror-axes of symmetry through the center of rotation spaced at equal angles. The case of D_1 symmetry corresponds to simple mirror symmetry; if an object has more than one line of mirror symmetry it must also possess a rotational symmetry.

There are important differences between the allowable symmetries in 2D and 3D. For example, regular n-sided polygons exist for any value of n in 2D, whereas regular n-faced polyhedra only exist for the specific values $n \in \{4, 6, 8, 12, 20\}$. Some of the ideas given in this paper will refer particularly to symmetry in 2D. In some cases it may be easy to infer how the ideas extend into 3D; other cases may require more careful consideration. A clear introduction to symmetry in two and three dimensions can be found in Weyl [25], where it is shown how the transformations on a symmetric object leaving it invariant form a group. Further details can be found in Lockwood [18].

When attempting to enforce symmetry on an object, a necessary initial step will often be to detect partial-symmetries as described above, and *almost-symmetries.* By almost-symmetries, we mean that when an object is transformed in one of the above ways, the resulting shape is in some sense close to the untransformed shape, rather than identically equal to it. We will consider in detail various measures of closeness later, but it should be noted that in terms of boundary-representation geometric modeling, two similar objects may be topologically the same, and only geometrically different, or they may be both topologically and geometrically different. Note that, as Figure 1 shows, a topological difference may be less important for almost-symmetry than a geometric one: shape A is symmetric, while many persons would agree that shape B is intuitively more like A than C is, despite being topologically different.

If an object which is almost-symmetric has been created by an optimization procedure which

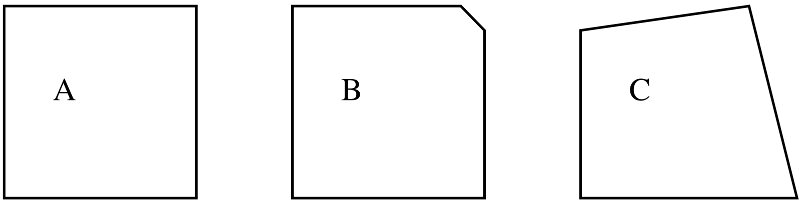

Figure 1: A symmetric shape and others which have nearly the same symmetry

is free to alter both the topology and geometry of a shape, it is quite possible that small edges and facets, as depicted in Figure 1B, can be removed by a preprocessing step. On the other hand, it is quite likely that the shape produced by an automatic freehand sketch capture program, for example, will only have errors of a geometric nature, and the qualitative (i.e. topological) aspects of the sketch are likely to be correct. Methods of detecting almost-symmetries will obviously be simpler if it can be assumed that only geometric differences occur from a symmetric shape.

We end this Section by noting that the ideas of partial- and almost-symmetries can be combined to give partial-almost-symmetries.

2.2 Use of Exact Symmetry Detection Methods

One approach to detecting almost-symmetries is to take existing symmetry detection algorithms, and to modify them by replacing 'equal' in suitable places by 'almost equal'. Also, when attempting to find partial-symmetries, it is clear that existing symmetry detection algorithms will be useful for comparing parts of objects rather than whole objects. Thus, we briefly review here various algorithms for determining if an object is symmetric, and consider when and how they may be applicable.

A useful survey of symmetry detection algorithms is provided by Eades [7]. He reviews algorithms which work on a variety of types of input: point sets, polygons and polyhedra, and line segments, for example. Some of these algorithms were originally proposed for checking congruence between two objects, but these can readily be used to check for symmetry, which is (non-trivial) *self*-congruence.

A basic approach used by many 2D symmetry algorithms [19, 26] starts by constructing an ordered set of elements representing the object. For example, for a polygon, this set might consist of pairs (α_i, l_i) where α_i is the angle between the i^{th} and $i+1^{st}$ sides of the object considered cyclically, and l_i is the length of the i^{th} side. These elements are arranged as a string L. To find a rotational symmetry, a string search is carried out for (non-trivial) occurrence of the string L in the string LL made by concatenating L with itself. This search can be done in linear time using the Knuth-Morris-Pratt algorithm [15], leading to a test for any rotational symmetry in linear time. A simple modification of this method also allows for the detection of mirror symmetries by using strings in both forwards and reverse order [10]. Given a suitable representation, such algorithms can also be used to detect symmetries of objects having more complex sides than straight lines, as is for example done in [2]; in this particular case the algorithm takes $O(n \log n)$ time because it deals with collections of objects, rather than the naturally ordered boundary of a polygon.

In 3D symmetry detection algorithms become more complex. This is because the elements of the boundary of a 3D object no longer simply form a cycle, and also because of the more complex nature of rotation in 3D: the axis about which rotational symmetry can occur can now be any direction in space. If an axis is given, essentially the same methods as in 2D can be applied, using cylindrical coordinates or other methods to represent the object [3, 26]. More generally, special graph isomorphism algorithms which run quickly on particular types of graphs can be used to check for symmetries of particular classes of polyhedra in either $O(n)$ time (see Wolter [26]) or $O(n \log n)$ time (see Sugihara [24]). At least the former is unattractive to implement because of the large constant involved in the $O(n)$, leading to the publication of a simple $O(n^2)$ method by Jiang [12] which has better average-case behavior than its worst-case behavior suggests. Again, this method is based on graph isomorphism tests.

A wide range of algorithms has been devised by the computer vision community for matching object representations captured from images to stored object descriptions [20]; many of these work by comparing specific points, edges or faces. A frequently used approach represents the problem as a tree search, with local constraints (such as the angle between adjacent faces) being used to control the potentially exponential growth of the tree. Such algorithms could also form the basis of a symmetry detection method, by matching the object to itself, although the more symmetry there is in the object, the bigger the search tree becomes.

Turning now to the issue of detecting partial-symmetries, we note that string-matching based symmetry algorithms can also be used to detect partial-symmetries. While for a completely symmetrical object the string L will occur in LL, for an object with a repeated feature around its boundary some *sub*-string of L will occur at more than one place in LL. Thus, as a string matching problem it is now necessary to find all substrings of one string occurring within another. A brute-force approach to this problem can solve it in $O(n^2)$ time. In a similar vein, Sugihara [24] notes that his graph-based algorithm can be used to search for partial congruences between polyhedra in $O(n^2)$ time.

Both the two-dimensional string matching symmetry algorithms, and the three-dimensional graph isomorphism based approaches rely strongly on connectivity to search for symmetries, and thus almost-symmetries which only differ geometrically and not topologically can readily be detected by such algorithms by replacing appropriate tests for equality with tests for almost equality. Even in such cases, however, care must be taken. Algorithms for finding symmetries of point sets, for example, essentially rely on constructing a topology, or ordering, and as noted by Wolter [26] small geometric errors in the input can can cause a difference in the constructed topology, and hence failure of the algorithm. For example, if the input points are sorted first by angle about the centroid, then by distance, a small error in the location of point a in point set A in Figure 2 can result in the (implicit) construction of shape B, which is no longer symmetric, rather than C.

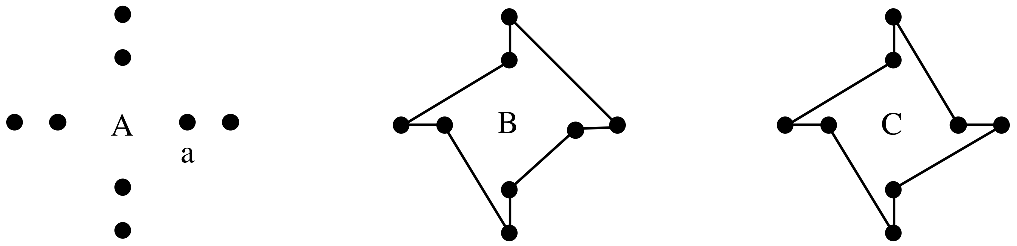

Figure 2: A topological error destroying symmetry

In conclusion, symmetry detection algorithms of the type discussed so far can be readily modified to search for partial-symmetries, and to detect almost-symmetries where only geometric rather than topological differences from a symmetric shape exist. We return to such approaches later when considering how to find the parameters of a desired symmetry.

2.3 Existing Almost-Symmetry Detection Methods

Other types of method have been proposed for detecting symmetries in noisy real-world data, such as image data. Often they are $O(n^2)$ methods even in 2D.

Parui [23] gives a hierarchical approach for finding planar mirror symmetries. Local candidate axes of symmetry are found, and a symmetry measure is computed for each appropriate pair of edges on either side of the axis, depending on the relative directions of the axis and each edge. This is summed over all pairs to give a total measure for that axis; the axis with lowest total measure is chosen as the global symmetry axis. For efficiency, searching for symmetry is done by using various simplified versions of the polygon, with coarser descriptions used first.

Davis [5] gives a similar method also based on hierarchical descriptions of symmetry of a polygon—'microsymmetries' between pairs of sides are sought, and these are clustered into larger aggregates. As well as dealing with polygons, Davis suggests how his technique may be extended to objects bounded by smooth curves using measures for straightness, convexity/concavity, and S-shapedness, with methods for deciding where corners should go.

Kilani [14] gives a rule-based approach to finding exact symmetries of a polygon. However, his methods are akin to those just described, and are thus included in this Section. His rules are

- A center of rotational symmetry must be the center of gravity.
- The minimum possible angle of rotational symmetry is equal to the minimum angle between two vertices at the same radius from the center.
- The number of vertices to the left of an axis of mirror symmetry is equal to the number on the right.
- The maximum number of axes of symmetry is equal to the number of edges.
- An axis of symmetry passes through a vertex or bisects an edge.
- An axis of symmetry passing through a vertex bisects the angle there.
- An axis of symmetry bisecting an edge is perpendicular to it.

Clearly some of these rules could be suitably modified to give an algorithm for detecting almost-symmetries. The modifications needed would depend on whether partial-almost-symmetries or global ones are being sought, and whether topological as well as geometrical differences from a symmetric shape exist.

Any of the above methods could be modified to seek partial-symmetries by selectively considering the elements of an object's boundary rather than incorporating all of them. Elements which are not in agreement with the a candidate partial-symmetry should be left out during the clustering stage of the method.

A different approach to those given above is the application of Woodwark's Method to symmetry detection [22]. This works from a set-theoretic (CSG) description of the shape, which is used to produce a multidimensional model representing the object in all possible orientations and positions in a configuration space. A template version of the original object (or its mirror-image) is also extruded in this space. A maximal intersection of the two multidimensional objects indicates in what pose the transformed object is most like the untransformed one; such intersections are located by a recursive division technique in the multidimensional space. This method is attractive in that it not only reports any almost-symmetry, but also gives the associated axis, center, etc. of the symmetry. It is clearly capable in principle of handling objects with complex boundary curves and surfaces. Unfortunately, this method seems to have very heavy computational requirements.

2.4 Symmetry Measures

A somewhat different approach can be taken to almost-symmetry detection by using methods which compute a symmetry *measure* for an object. Typically, a different symmetry measure is used for each type of symmetry under consideration; the lowest overall measure can then be used to determine what type of symmetry the object most nearly possesses.

Zahn [28] shows how the boundary of a plane closed curve can be described by a set of Fourier descriptors, which are the discrete Fourier transform of the cumulative angle turned through as a function of normalized arc-length. He goes on to show that for rotationally- or mirror-symmetric objects, the Fourier coefficients must satisfy certain conditions. In principle, as the Fourier series is an infinite series, an infinite number of coefficients must be considered. However, Zahn suggests that by considering a limited number of coefficients of low order, a good idea may be obtained of the amount of symmetry of the shape. This may readily be understood by noting that the higher-order coefficients represent changes of shape at smaller length scales, while the lower-order coefficients are better descriptors of overall shape.

Very closely related is the use of moments to describe shapes in 2D and 3D. It can be shown that a shape can be reconstructed from its infinite set of moments, where each moment $M_{i,j}$ in two dimensions is defined as

$$M_{i,j} = \iint\limits_{\text{object}} x^i y^j \, dx dy,$$

and i and j are non-negative integers. It can be shown that various combinations of these moments are invariant with respect to changes of orientation and position (and scale) of the object. Such relationships are generally made simpler by describing them in terms of *central* moments, computed in a coordinate system with origin at the centroid of the object. Hu [11] gives various relations that must exist between moments if the object is to be symmetric, but again an infinite series of moments must be considered to demonstrate exact symmetry. As before, it is possible to propose candidate symmetries by considering a certain number of these relations for lower-order moments, or to use them to construct an approximate (i.e. not always reliable) symmetry measure.

Grünbaum [9] describes various symmetry measures for convex shapes in 2D, all of a similar type. One such example is Winternitz's measure, which is defined as follows. Choose some point in the shape, and a line through that point, so the shape is split into two parts. Find the ratio of the area of the smaller part to the larger part. Minimize this ratio over all lines through the point. Maximize that value over all points in the shape. This gives the measure of symmetry of the set. Unfortunately such measures do not readily generalize to non-convex shapes; also no algorithms are given for efficiently computing these measures, although they could probably be devised at least for simple shapes like polygons using methods akin to those given by Martin [21].

Zabrodsky [27] defines a symmetry measure on a discrete set of points as follows. Let the distance between two sets of n points $\{P_i\}$ and $\{Q_i\}$ be given by

$$\frac{1}{n}\sum_{i=0}^{n-1}(P_i - Q_i)^2.$$

The *symmetry transform* of a set of points $\{P_i\}$ is the symmetric set of points $\{Q_i\}$ nearest to the set $\{P_i\}$ (for some given symmetry); the *symmetry distance* of a shape is the distance between these two sets. To find $\{Q_i\}$, Zabrodsky proves that the symmetry transform can be computed by using a method called *folding*, originally conceived by Lin [17]. Suppose a set of n points $\{P_i\}$ is given. To find the n-fold rotational symmetry transform of this set, the set is translated so that its centroid is at the origin, and each P_i is rotated by $-2\pi i/n$ radians. A new point $\bar{P}$ is created by averaging the resulting points, and then finally the Q_i are generated by creating n copies of $\bar{P}$ rotated through $2\pi i/n$. If the original set has n points and m-fold symmetry is being sought, rather than n, the method can be extended provided n is a multiple of m. Let $n = mr$. Then the input points are divided into r sets of m points, by taking $P_1, P_{1+r}, P_{1+2r}, \ldots$ in the first set, $P_2, P_{2+r}, P_{2+2r}, \ldots$ in the second set, and so on. Each subset is now symmetry transformed independently about the common centroid, and then finally the transformed sets are put back together into the appropriate order to form the overall transform of the original set.

An analogous approach can also find the mirror symmetry transform with respect to a particular axis. Zabrodsky goes further and shows how to determine the axis which has the corresponding best symmetry measure. It passes through the centroid of the points and has orientation given by

$$\tan 2\theta = \frac{(x_0y_1 + y_0x_1) + \cdots + (x_{m-2}y_{m-1} + x_{m-1}y_{m-2})}{(x_0x_1 - y_0y_1) + \cdots + (x_{m-2}x_{m-1} - y_{m-1}y_{m-2})}$$

where θ measures the angle the axis makes with the x-axis, and corresponding points on either side of the axis are successive pairs $P_0, P_1; P_2, P_3; \ldots;$ with $P_i = (x_i, y_i)$. However, a corresponding analytical result is not available in 3D and numerical methods must be used.

Although not explicitly considered by Zabrodsky, it is clear that this type of approach can also be extended to *translational* partial-symmetry. For two sets of points, it is simple to show that the points match best when the translation vector is equal to the vector between their centroids; if there are more than two sets of points these translation vectors should be averaged. Using this vector, the sets can then be folded (translated) into correspondence, the corresponding points from each set can be averaged, and then the set of averages replicated in positions determined by the vector.

Zabrodsky's method has the obviously useful side-effect that while computing the symmetry measure, the nearest symmetric point-set is also found. However, although he criticizes the measures in Grünbaum as only applying to convex shapes, Zabrodsky's approach suffers from a serious defect itself when applied to a set of points which represent the vertices of a non-convex polygon: when the points representing the symmetry transform are joined, the result may be a self-intersecting polygon if the original shape is too far from symmetric. Thus, its application for constructing new symmetric polygons from existing ones by finding the symmetry transforms of their vertices would appear to be limited, or at least to require some care.

We also note that a further problem arises on attempting to use Zabrodsky's approach to determine the symmetry of polyhedra in 3D. In 2D, an *ordered* set of points which are the vertices of a polygon is sufficient to uniquely define a polygon, and if that set of points is symmetric, so must be the polygon. Unfortunately, in 3D, as is well known, an ordered set of vertices and the corresponding edge graph is *not* sufficient to define a unique shape—face information is needed as well. Thus, in 3D, a polyhedron may have a lower degree of symmetry than its corresponding vertex set, and so the vertices by themselves are not a sufficient set of points of a polyhedron to use for symmetry detection.

Another symmetry measure proposed by Kulkarni [16] for mirror symmetry is based on symmetric set-difference. Suppose an axis (in 2D, or plane in 3D) of symmetry has already been chosen; let the parts of the object on either side of the axis be A and B. Then A is reflected in the axis to give A', and the symmetric difference $A' \ominus B$ of A' and B is found using $A' \ominus B = (A' \cup B) - (A' \cap B)$. This is an empty set if A' and B are the same; if they are not it gives a set representing their mutual regions of non-overlap. The degree of symmetry is defined as

$$\text{degree of symmetry} = 1 - \frac{\text{area}(A' \ominus B)}{\text{area}(A') + \text{area}(B)}.$$

Note that this measure is one if and only if the original shape is symmetric, and hence gives an exact test for symmetry, unlike measures based on moments, for example. It is trivial to generalize this measure to apply under a proposed translational symmetry. It is not much more difficult to cater for rotational symmetries, where the concept of symmetric difference is applied to the n folds of the object each rotated through an appropriate angle, using the multi-way symmetric difference of $A, B, C, \ldots$ defined as $(A \cup B \cup C \cup \ldots) - (A \cap B \cap C \cap \ldots)$. One slightly counter-intuitive aspect of this symmetric difference measure is that distances are not taken into account (i.e. larger distances are not given a greater weight), either of distances of the non-overlapping parts from the axis of symmetry, or of how far various points in the non-overlapping parts are from the nearest overlapping points.

3 Choosing the Symmetry Operation

Even after the most appropriate type of symmetry to impose on the object has been specified by the user or automatically determined, there remains the problem of choosing the parameters of that symmetry. For example, it may be decided that a shape is to be given 3-fold rotational symmetry. The point to be used as the center of the rotational symmetry still has to be determined in this case. We will consider two approaches to choosing these parameters, based on computing various statistics for the whole object, with and without optimization, and on using selected details of the object.

3.1 Whole Object Methods

A widely used method of finding a canonical set of orthogonal axes for an object is the method of principal axes. For an object with a single mirror axis in 2D, one of the principal axes is the axis of symmetry; similar remarks hold in 3D. If the object is asymmetric, the appropriate principal axis can be used as the axis about which to impose symmetry. To find this axis, the object is translated so that its centroid is at the origin; the principal axes now pass through the origin and make an angle θ with the x-axis where

$$\tan 2\theta = \frac{2M_{1,1}}{M_{2,0} - M_{0,2}},$$

and $M_{i,j}$ are central moments.

The basic method of principal axes has disadvantages. Firstly, if the object has any exact rotational symmetry, this expression gives 0/0. Instead, higher order moments must be computed using a method given by Hu [11]; the formulae become increasingly complex with order. On the other hand, just because this expression comes out to 0/0, it cannot be assumed that the object has *any* rotational symmetry—there are quite asymmetrical objects objects for which this happens. As remarked earlier, an infinite series of moments must be considered to guarantee exact symmetry.

Zabrodsky's method shows that for the symmetry transform of a set of points in 3D, any axis of rotational symmetry or plane of mirror symmetry must pass through the centroid of the original points. In 2D the formula given earlier may be used to find an axis of mirror symmetry; he also gives a corresponding formula for the case when an object has both rotational and mirror symmetries.

In fact, Zabrodsky's formula bears some similarities to the principal axis formula. Consider an object consisting of just one pair of points, P_0, P_1. Then, after the object is transformed to its centroid, $x_0 + x_1 = 0$ and $y_0 + y_1 = 0$. Thus,

$$\tan 2\theta = \frac{x_0 y_1 + x_1 y_0}{x_0 x_1 - y_1 y_0} = \frac{2x_0 y_0}{x_0^2 - y_0^2},$$

which is the same as the principal axis formula applied to just P_0. Nevertheless, in general, Zabrodsky's formula does not give the same axes as the principal axis method, and as already noted, has a rather simpler generalization to handle the D_n symmetry cases.

Lin [17] originally devised the folding method as a means of finding key lines through each fold of an n-fold rotationally symmetric shape; the method gives the axes of mirror symmetry if the object has D_n symmetry. Here we note that the method can also be used to suggest axes for almost-symmetric shapes. Let points in the initial shape S be given by their polar coordinates (r, θ), with the origin at the centroid of S. A new shape E, the fold-expanded shape, is constructed from S by mapping (r, θ) to $(r, n\theta)$. The centroid of E in polar coordinates, $(\bar{r}, \bar{\theta})$ is now computed, and finally mapped back to the point $(\bar{r}, \bar{\theta}/n)$. The desired axis goes through this point and the centroid of the original shape; this axis is independent of the choice of direction for $\theta = 0$. (Lin also gives a variant method which finds the radius-weighted-average point in E rather than its centroid.) For an object which is rotationally symmetric, the same result can be obtained by mapping just a single fold; in the current case this short-cut cannot be taken, but rather all the folds are averaged. Unlike Zabrodsky's use of folding which just uses a discrete set of points, Lin's method is applied to the solid area of a 2D object.

Unfortunately, all methods of the type described above share one disadvantage for choosing symmetry parameters, for example, a possible axis of mirror symmetry. Although the axis computed is the axis of symmetry if the object *is* symmetric, when the object is *not* symmetric, the data may be considered to be defective in some way. Because *all* the data, good and bad, are used to compute the axis, the resulting axis is not the one which is really desired. Thus, rectifying the shape using such an axis as a basis is unlikely to give the result which is sought. An example shape and the axis computed are shown in Figure 3A; the results of rectifying it are shown in Figure 3B. (The rectification method chosen in this example is to discard the upper part of the shape, and to reflect the part of the shape below the axis. Choice of rectification method will be discussed in Section 4.)

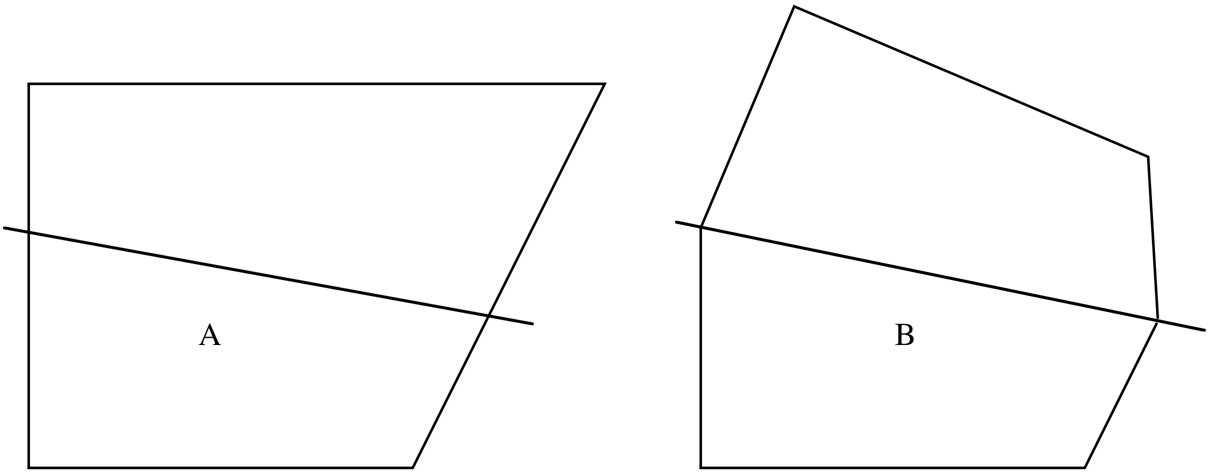

Figure 3: A poor choice of rectification axis

3.2 Optimization Methods

One way in which the above problem may be alleviated to some extent is given by Zabrodsky. He considers a problem in image processing, where an occluded star-shape is extracted from an image, the visible boundary being similar to that shown in Figure 4A. The symmetry distance for D_5 symmetry is low, determined by sampling a large number of points at equal angles around the boundary of this figure relative to its centroid; this indicates a candidate symmetry for the original object. However, if the corresponding symmetry transform is computed, a shape like that in Figure 4B is reconstructed. Zabrodsky notes that if folding is done not about the centroid of the original points, but about some other point, a different value is obtained for the symmetry measure. If an optimization method is used to find the point about which the symmetry measure takes on its lowest value, the point obtained is nearer to the center of symmetry of the original, unoccluded, object than is the centroid of the occluded shape. Thus, if this optimal point is used

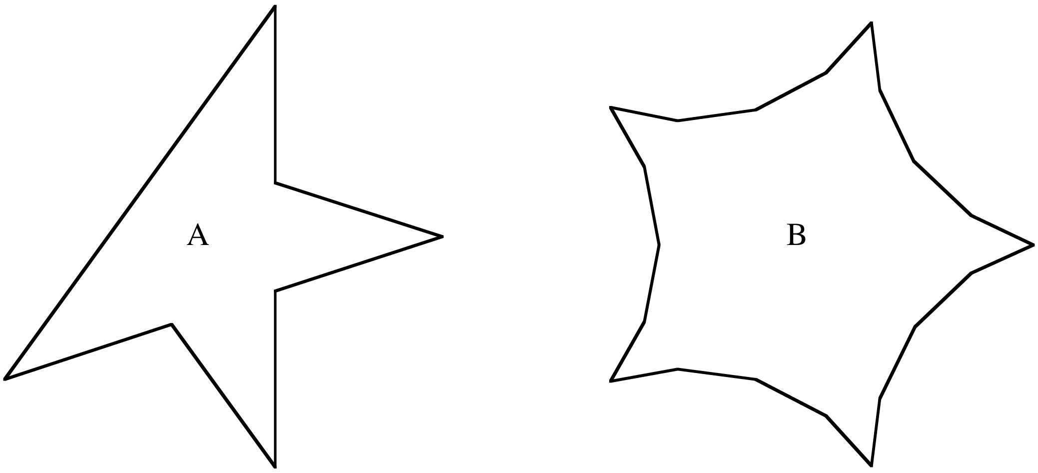

Figure 4: Reconstructing a star

for reconstructing the complete object, rather than the centroid, a shape much more similar to the original object is obtained than that in Figure 4B.

Kulkarni [16] also describes an optimization approach for proposing an axis for 2D mirror symmetry rectification; this axis is the one which minimizes the symmetric set-difference measure of symmetry given earlier. In practice the optimization is carried out in two stages. Firstly, the best axis is found restricted to ones which pass through the centroid of the object. Then, using this axis as a starting point, this restriction is removed to find the best overall axis. A problem noted in experiments with this approach is that the non-smooth nature of the symmetry measure (consider what happens when a candidate axis is slightly moved so that a vertex passes from one side of it to the other) means that the optimization method must be chosen with care. Unfortunately, this method gives results which seem less successful than Zabrodsky's optimization method in terms of finding 'reasonable' symmetry axes. This may be partly due to the different symmetry group that each was considering, rather than the particular symmetry measure and method used.

Overall, the results presented by these authors lead us to conclude that while such optimization methods may be more successful than simple whole object methods, they still do not give what would intuitively be considered to be the best possible results. Furthermore, a disadvantage of such methods is that, being iterative, they are generally quite slow.

3.3 Selective Methods

Given the limitations of the methods described in the previous two sections, we suggest the use of different methods for choosing centers, axes and planes of symmetry.

The approach we propose in this paper is to use a method of finding partial-symmetries based either on sub-string (or -graph) matching, or on the hierarchical and rule-based methods described earlier. The best partial-symmetry detected can then be used to give both the symmetry group and the parameters of the symmetry. More generally, these methods can be relaxed to search for partial-almost-symmetries where only small differences in geometry and not in topology exist upon transformation. A complementary 'region growing' approach can then be used to build upon these nuclei of almost-symmetry to find larger almost-symmetric regions. Growing such regions is a relatively simple task which typically involves only considering a few parts of the boundary of the object at a time. Finding several nuclei for the same symmetry gives additional evidence of its existence. The approach may be further improved by preprocessing the shape to throw away small edges or faces below a certain size threshold. Such preprocessing will help to increase the success rate of finding almost-partial-symmetries, and in practice can be carried out 'on-line' by such an algorithm performing the deletion as it operates. The choice of whether to use exact partial-symmetry detection or partial-almost-symmetry detection will be dictated to a large extent by the source of the input data describing the original shape.

The partial-symmetry based approach given above has the advantage that parts of the object which are already symmetric with respect to the chosen symmetry will be retained on rectification, and only other parts of the object will change. Consider for example the shape first shown in Figure 3, now repeated in Figure 5. The partial-symmetry at the left-hand end of the shape has now been used to choose the axis shown, which bisects and is perpendicular to the left hand edge. On rectification, this produces the more intuitive result shown in Figure 5B (again discarding the upper half of the object and reflecting the lower half).

Deciding what is meant by the 'best' partial-symmetry is not trivial, and is likely to be applica-

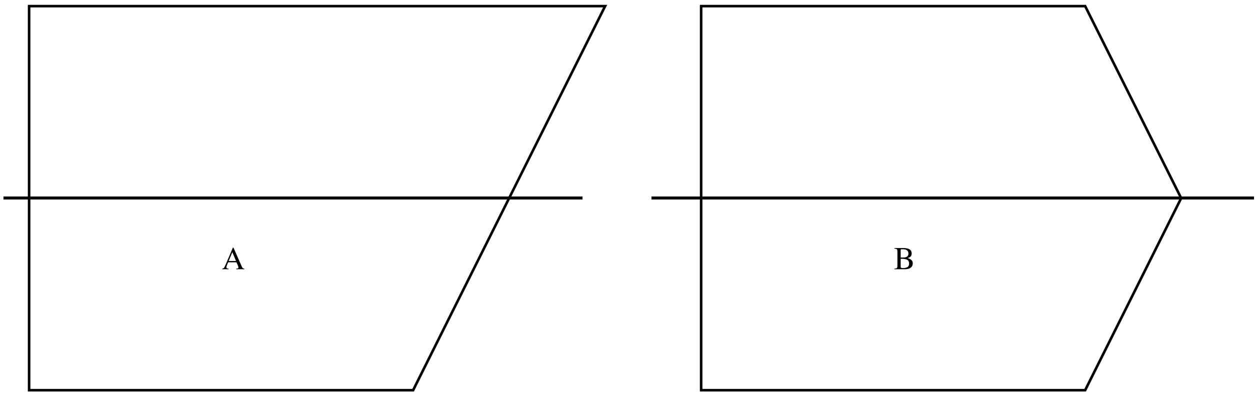

Figure 5: A better choice of rectification axis

tion dependent. It is likely to take into account both the number of boundary elements of the shape sharing a given symmetry, and on the size of those elements. One possible measure might be fractional arc-length of the perimeter of a 2D object bounded by those elements for mirror symmetry, or on angle subtended for rotational symmetry.

In general, it will not just be *adjacent* elements of the boundary of an object which show a given symmetry—for example, both ends of the object in Figure 6 support an axis of D_2 symmetry as shown. Thus, having found *local* partial-symmetries, they must be merged into overall partial-symmetries, and the corresponding evidence must be accumulated.

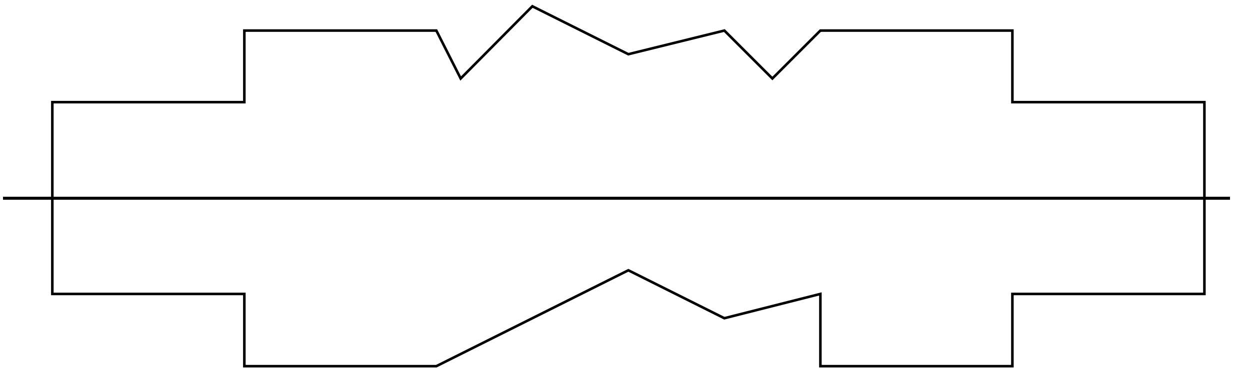

Figure 6: A symmetry axis shared by elements at both ends of an object

A further selective method of choosing axes of 2D mirror symmetry for convex polygons, different from the ideas already described, is given by Kulkarni [16]. The polygon is reduced to its skeleton (Voronoi diagram), which is simpler than the original shape, being a set of connected lines. The skeleton of such a polygon is mirror symmetric if and only if the object itself is mirror symmetric. In such a case the mirror line of symmetry must *either* run along one of the segments of the skeleton *or* must bisect one of the segments at right angles; note also that the axis passes through the centroid of the object. Thus, in an asymmetric shape, spine segments or their bisectors are good candidate axes of symmetry for rectification; the authors suggest using the one which passes closest to the centroid of the object.

Unfortunately, skeletons of non-convex objects no longer simply contain straight line segments. Skeletons are also relatively expensive to compute, especially in 3D. Furthermore, skeletons are relatively sensitive to small variations in shape. These disadvantages would seem to limit the potential of this type of approach.

4 Rectifying Asymmetry

Having determined the symmetry desired for the object, and its parameters, the problem of imposing that symmetry on the object now remains. To start, we note that enforcing the overall symmetry group should be considered as a *single* problem. This avoids the possibility that enforcement of a single symmetry may undo changes made to enforce a previously considered symmetry. Furthermore, a combined approach is likely to be more efficient. Thus, for example, D_n symmetry should be enforced in one go, rather than trying to enforce C_n symmetry and n lines of mirror symmetry separately.

We first consider several methods for rectifying simple objects, and then propose a basis for dealing with more complex objects in terms of features.

4.1 Simple Cases

Whichever type of symmetry is to be imposed, once the symmetry has been identified, the object may then be divided into n folds, i.e. parts of the object which are (hopefully slightly) different, and which must be identical after rectification. The simplest way to impose symmetry is to select just one of the folds, and replicate it n times, replacing the other folds with the suitably transformed replicants. This is the method used for mirror symmetry rectification in [16], but it may be applied to any type of symmetry. Generally, there may be no particular reason to choose any one fold over the others, and so the results of using each in turn could perhaps be offered to the user as choices. Alternatively, the user might specify some criterion in advance to determine the fold to replicate, such as the one of lowest area, the one with the least (or most) vertices, the one which is the median with respect to some property, etc.

When rotational symmetry is considered, there is a further decision to be made about which direction to use as $\theta = 0$ for cutting the object into folds. Either of Lin's methods described earlier would seem to be good choices, but user selection or other possibilities may also be appropriate.

In Kulkarni's original approach for rectifying polygons to enforce mirror symmetry, the *edges* on one fold are kept (ones connecting to vertices in the other fold are clipped) and are replicated with the appropriate transformation. However, an alternative approach is to keep the *vertices* instead, and join them with new edges. This produces the same result except for the edges which cross from one fold into the next. The former case produces a new vertex on the fold boundary, the latter approach produces edges which cross the fold at right angles.

Replicating a particular fold has limitations, and undesirable results can occur. For example, an originally convex polygon may be rendered non-convex if edges instead of vertices are used, as shown in Figure 7, where a vertical mirror axis of symmetry is imposed by reflecting the edges on the right-hand side of the object.

An alternative to selecting one fold and replicating it is to *average* all of the folds. This is akin to what Zabrodsky does when computing his symmetry transform, although he works with

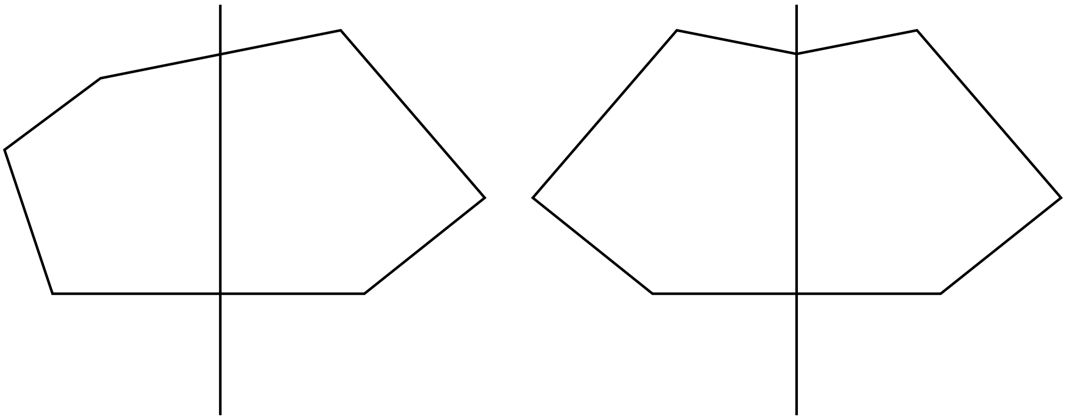

Figure 7: A convex polygon becomes non-convex after rectification

a discrete set of points rather than boundaries of shapes. If a polygon (or polyhedron) is being rectified, and each fold has an equal number of vertices, Zabrodsky's transform can be applied to the vertices. However, this does not work very well in cases where the vertices are not approximately equally spaced around the object's boundary. More generally, the fold boundaries may have different numbers of vertices and they may be more complex curves than straight lines. He proposes sampling the object at regular arc-length or angular intervals, but this leads to a low-level representation of the object and the result after rectification. Instead, we may average the boundary curves themselves.

We first consider 2D mirror symmetry, and suppose without loss of generality that the axis is the y-axis. We also initially assume the shape is a polygon and is convex; these restrictions will be removed later. Let the width profile on each side of the axis be $-x_l(y)$ and $x_r(y)$ respectively. Then each of these width profiles is replaced by $\mp\bar{x}(y) = [x_r(y) - x_l(y)]/2$ respectively. Considering the particular case where the original shape is a polygon, i.e. x_l and x_r are piecewise linear functions, it can be shown that the result is also a polygon with the following properties (see Figure 8):

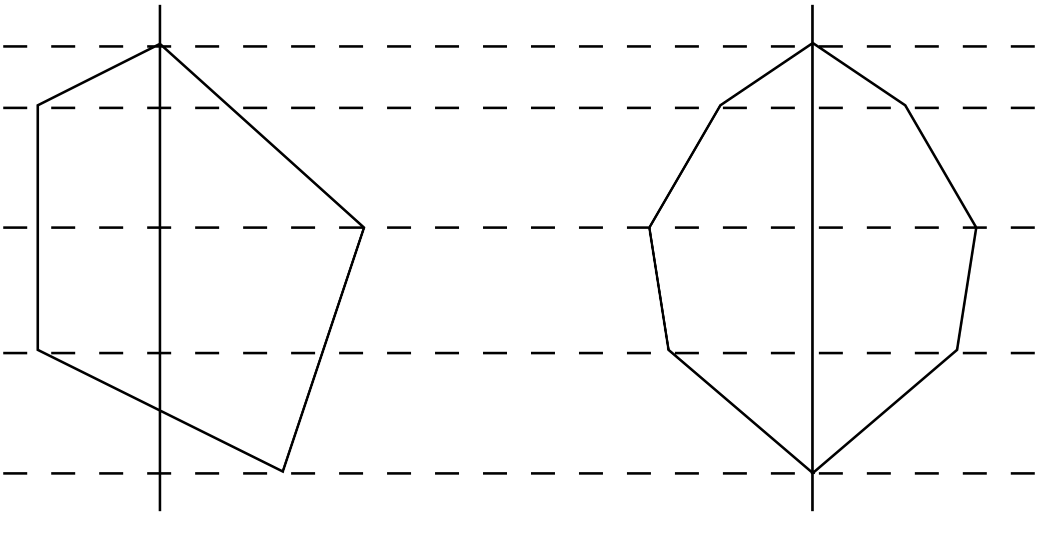

Figure 8: Rectifying a polygon by averaging

- Vertices occur on *both* sides of the new polygon at heights at which there was a vertex on *either* side of the original polygon.
- To find the vertices of the new polygon, the $-x_l$ and x_r values are averaged at these heights. The new edges are just the straight lines drawn between these vertices.
- If any edge of the original polygon crosses the axis, the given formula is still applied even when left and right boundaries lie on the *same* side of the axis.
- The top and bottom vertices of the polygon are considered as being part of both left and right boundaries. From the previous result, the new top and bottom vertices must lie *on* the axis.
- The area of the figure is unchanged; so is the left-to-right width of the figure at any given height.

For more complex boundary curves, provided that they can be expressed in $x = f(y)$ form, analytic solutions can also be found.

While the area preserving property is an improvement over the method based on selecting a single fold, the fact that the number of vertices increases is a disadvantage compared to that method, as in some sense the shape has been made more complex than the original.

The previous ideas may be extended to the case of rotational C_n symmetry; the ideas below and the previous idea can be applied simultaneously in the case of D_n symmetry. In this case, assume the center of symmetry is at the origin. This time, the boundary of each fold after rotating to lie in the range $0 < \theta \leq 2\pi/n$ is expressed in polar coordinates of the form $r_i(\theta)$, and $\bar{r}(\theta) = \sum_0^{n-1} r_i(\theta)/n$ is computed. Any boundary curves which can be expressed as piecewise functions of this type can be analytically averaged between values of θ corresponding to successive vertices in any of the folds.

Unfortunately, when straight edges are averaged in this way, the result is *not* a straight edge. This makes averaging much less attractive as a means of rectifying rotational symmetry than it is for mirror symmetry. Nevertheless, it may still be appropriate for shapes with curved boundaries. Alternatively, for polygons, we can replace exact averaging by an approximate version as follows. For each value of θ for which there is a vertex in at least one fold, the corresponding value of r is found, giving a point from each fold. These points are averaged to give a new vertex which is placed in each fold after rectification; such vertices are joined by straight lines, ensuring that the resulting shape is a polygon. However, some of the advantageous properties of true averaging will be lost by this approximate method.

A further problem arises with averaging on attempting to use it for rectifying (partial-) translational symmetries. Although the parts of the object can be folded into correspondence, the choice of how the boundaries are to be averaged has more than one plausible solution—are the boundary curves to be averaged as position functions of fractional arc-length? As radius functions about the common centroid? Whilst natural choices for averaging exist for mirror and rotational symmetries, it is not so clear how points on the boundaries of each fold should be put into correspondence in this case.

For both mirror rectification and rotational rectification, the above restriction to convex objects is not completely necessary. Instead, it can be seen that the method will produce a well defined and meaningful result, if, for the mirror symmetry case, $x_l(y)$ and $x_r(y)$ are single valued functions with respect to y, and for the rotational symmetry case, each of the $r_i(\theta)$ is single valued with respect to θ. In fact, it is possible to go even further than this, and to allow each fold to be represented by multi-valued functions provided that at each value of y or θ, each fold has the *same* number of multiple values. In such a case corresponding values from each fold are averaged. This would allow the rectification of an approximate ⊔-shape about a vertical axis of symmetry, for example.

Unfortunately, it is not obvious how best to average shapes whose folds do not meet these criteria, or what results would be considered to be the meaningful result of such an operation—see Figure 9. One possibility is to use the approach described earlier, where the side of the axis on which each edge lies is ignored, and the outer pair of edges is averaged, then the next innermost pair, and so on. This would lead to centering the slot in the shape in Figure 9, but this may not be what the user expects. Alternatively, it is possible to consider the lower part of the left hand side of Figure 9 as being degenerately three-valued, with all three values being the same. There are now three values on each of the left- and right-hand sides which can be averaged independently. This would produce two narrow slots, one on each side of the axis, rather than the single slot as before.

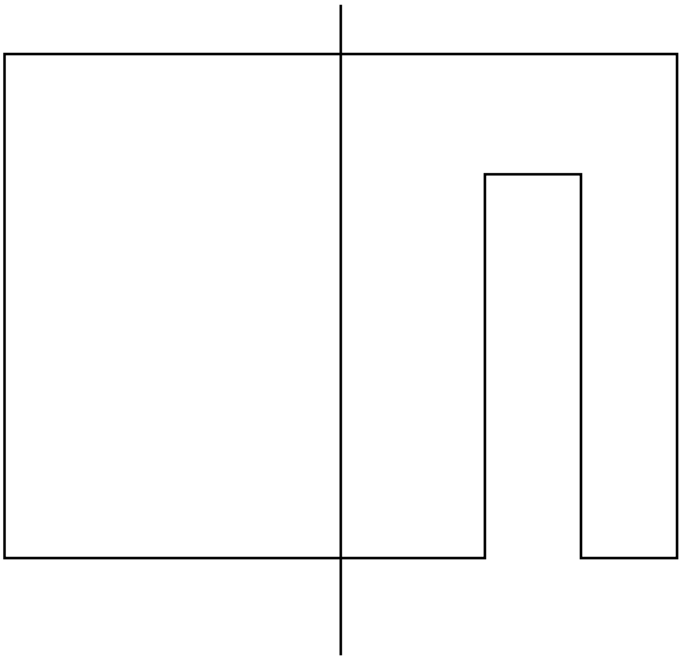

Figure 9: An object for which the desired result of rectification is not obvious

If such approaches to rectification are applied carelessly to non-convex objects the result may be an object of different topology to the original, for example, a disconnected object, or an object with a self-intersecting boundary. As already noted, the latter problem may occur if points on the boundary of a non-convex shape are sampled at equal intervals, and Zabrodsky's symmetry transform is applied directly.

Averaging is in some ways more appealing than selecting one fold and replicating it, especially for mirror symmetry. However, this is not always the case. For example, consider Figure 1B, where replication of the left side of the shape leads to a more natural outcome.

Whilst averaging has certain useful features for rectification, it also has various disadvantages noted above. On the other hand, as all folds are taken into account, not just one, makes the result of rectification is more representative of the input data. Thus, other methods based on using all folds are also worth seeking. One such example is to transform all of the folds by the operations of the chosen symmetry, and then to take some set-theoretic combination of the folds as the master to be replicated in the appropriate positions. For example, one might take the union of all the folds, or the intersection of the folds, for example. This type of method has the advantage that even for rotational symmetry rectification, straight boundaries remain straight, but other properties like the area-preserving one are lost.

A rather different approach to rectification is to treat it as an optimization problem. Here the boundary of the shape is represented by a set of parameters. These parameters may describe positions of vertices on the boundary of the object or the shapes of boundary curves, for example. The initial shape corresponds to some point in the parameter space. Rectification is now a matter of searching this parameter space; this can be done by assigning a symmetry measure to the object at each point of the space and using an optimization method. More generally, the objective function could also incorporate other criteria desired of the sought-for shape, such as constraints between the parameters, on the area, and even on structural or mechanical properties. Several drawbacks are apparent for this type of approach, however. Firstly, the computational requirements are very high. Secondly, if too few free parameters are used, there may be *no* symmetric shape in the search space. Thirdly, if too many parameters are used, there may be a whole spectrum of symmetric shapes, some of which are very different from the original shape; the exact shape returned would depend on the nature of the optimization technique. It would thus appear to be a difficult problem to choose a good search space and search technique.

We end this section with a method of shape rectification briefly mentioned by Kulkarni [16]. This is to rectify the skeleton of the object, using the relation between the symmetry of an object and its skeleton. This then requires the reconstruction of the object from the new skeleton, a problem discussed in [8]. Given the disadvantages of the skeleton approach mentioned earlier, and the extra computational time involved in the reconstruction, this can hardly be considered an appealing approach.

4.2 Complex Shapes

It may seem premature to consider the rectification of complex shapes when we do not have a single best method for dealing with simple cases. Nevertheless, many complex shapes of practical interest may be considered to be made up of simple shapes which can be handled.

We assert that the key to dealing with such complex shapes is to decompose them into a basic object, and features which modify that object, doing so recursively if needed. The user may then specify the rectification procedure to be carried out in the following terms

- Is it the symmetry of the basic object which is to be rectified?
- Are the features to be rectified with respect to the symmetry of the basic object?
- Are the features to be rectified with respect to each other?

where it is understood that these questions are to be applied recursively. Such an approach requires the object to either have been designed from features [6], or the use of a method of feature extraction [1]. Alternatively, if the object has been constructed from primitives using set-theoretic (CSG) modeling, the primitives and internal nodes of the tree may be considered as features for the purposes of rectification. Generally, however, there is no reason to assume that the structure of the tree will naturally lead to internal nodes representing sub-parts which should symmetrically related. Reordering the tree so that internal nodes do is probably as difficult a problem as feature recognition.

We outline here how a rectification method based on recursively dealing with features might work. In order to carry out the above rectification steps, it is useful for each type of symmetry to identify *key lines*—for mirror symmetry, the axis of symmetry, for rotational symmetry, a line through each fold as given by Lin's method, for translational symmetry, the line between centroids of each fold. In further detail, the first question asks whether the parent object (ignoring the features) is symmetric. The second question asks whether the features are identical under the the symmetry transform of the plate. A weaker version of the second question considers whether the centroids of the features lie on the key lines of the parent object, or are symmetrically placed with respect to them. It also asks whether the orientations of the key lines of the features are in agreement with those of the parent object—if the parent object is mirror symmetric, parallel to the key line of the parent object, if the parent object is rotationally symmetric, aligned with a radial line. The third question asks whether the features are identical under some other transform than the parent object's symmetry, or again whether centroids or key lines of features are so related.

Now consider what is required if the user asks for each type of rectification in turn. In the first case, the problem reduces to the problem of rectifying a single object, albeit a less complex object than the original object. In the second case, a set of points (the centroids of the features) and orientation vectors must be rectified with respect to the symmetry of the parent object; having done that the shapes of the features can also be rectified if required under that symmetry, by treating the set of features as a single disconnected object. The third case can also be treated in a similar way, except that now the symmetry to be used is not that of the parent shape, but must be determined before it can be imposed. In practice, it is desirable that any keylines or centers of this symmetry should also be brought into agreement with the parent shape's symmetry.

A trivial example of these ideas is shown in Figure 10, where an almost square plate with an eccentric hole is rectified according to the first two criteria—the outer shape is rectified, as is the placement of the feature (the hole).

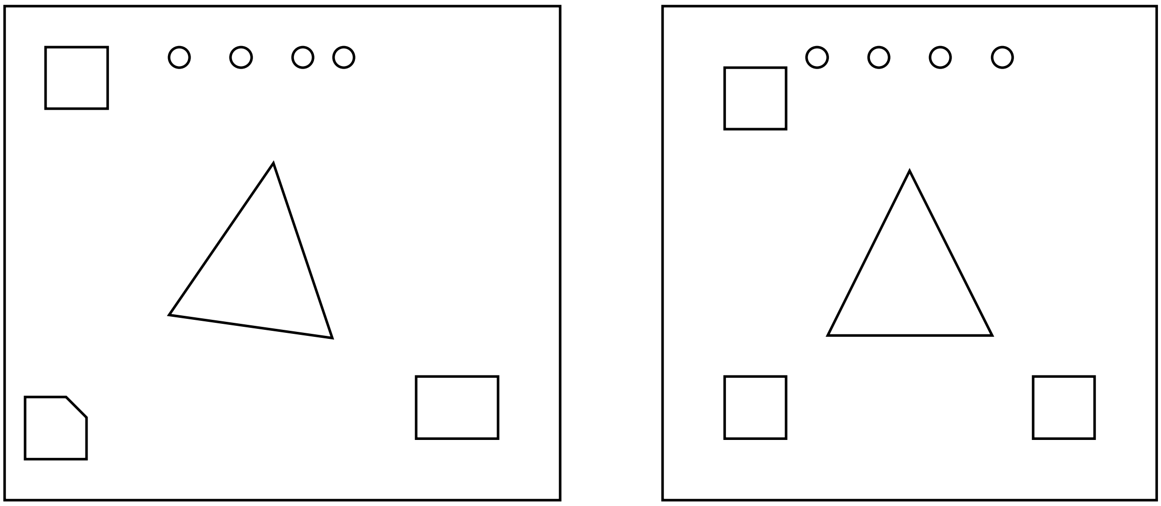

Figure 10: Rectification of a complex object by decomposition

Thus, we have briefly justified here that rectifying the symmetries of 'complex' shapes may be broken into a series of steps which are in principle no more conceptually difficult than solving the problem of what to do for 'simple' shapes.

5 Conclusions

Asymmetry rectification is a problem with a variety of applications. A range of methods of quantifying asymmetry and detecting symmetry has been explored, and it is suggested that methods based on detecting maximal partial-symmetries (or partial-almost-symmetries) should be used to determine both symmetries to be enforced and suitable parameters such as centers and axes of symmetry. Various methods of rectification have also been considered, and simple methods have been put forward which work reasonably well for simple shapes. Finally, some initial ideas for the rectification of more complex shapes have been explored. It is clear that much work remains to be done to evaluate these ideas on practical examples, and it is intended that this will be the subject of future research.

6 Acknowledgements

R. R. Martin would like to thank D. Dutta for inviting him to the University of Michigan to collaborate on this work. R. R. Martin received partial support from the Office of the Vice-President for Research, University of Michigan and from ONR Grant N00014–94–1–0022. D. Dutta received partial support from ONR Grant N00014–94–1–0022. Both authors would like to thank Gabriel Taubin of IBM for helpful discussions about moments during the course of this work.

References

[1] S. Ansaldi, B. Falciendo, Extracting and Completing Feature Information in Process Planning Application, in *Geometric Modeling for Product Engineering*, eds. M. J. Wozny, J. U. Turner, K. Preiss, North-Holland, 1990.

[2] M. J. Atallah, On Symmetry Detection, *IEEE Transactions on Computers* **34** (7) 663–666, 1985.

[3] M. D. Atkinson, An Optimal Algorithm for Geometrical Congruence, *Journal of Algorithms* **8** (2) 159–172, 1987.

[4] M. P. Bendsoe, N. Kikuchi, Generating Optimal Topologies in Structural Design using a Homogenization Method, *Computer Methods in Applied Mechanics and Engineering* **71**, 197–224, 1988.

[5] L. S. Davis, Understanding Shape: II. Symmetry, *IEEE Trans. on Systems, Man and Cybernetics* **7**, 204–212, 1977.

[6] J. R. Dixon, E. C. Libardi, E. H. Nielsen, Unresolved Research Issues in Development of Design-with-Features Systems, in *Geometric Modeling for Product Engineering*, eds. M. J. Wozny, J. U. Turner, K. Preiss, North-Holland, 1990.

[7] P. Eades, Symmetry Finding Algorithms, in *Computational Morphology*, ed. G. T. Toussaint, Elsevier, 1988.

[8] S. Gelston, D. Dutta, *Boundary Surface Reconstruction from Skeleton Curves and Surfaces*, Technical Report UM-MEAM-92-03, Dept. of Mechanical Engineering and Applied Mechanics, University of Michigan, 1992.

[9] B. Grünbaum, Measures of Symmetry for Convex Sets, *Proc. Symp. Pure Math. AMS* **7** 233–270, 1963.

[10] P. T. Highnam, Optimal Algorithms for Finding the Symmetries of a Planar Point Set, *Information Processing Letters* **22** 219–222, 1986.

[11] M.-K. Hu, Visual Pattern Recognition by Moment Invariants, *IRE Trans. on Information Theory* **8**, 179–187, 1962.

[12] X. Y. Jiang, H. Bunke, A Simple and Efficient Algorithm for Determining the Symmetries of Polyhedra, *CVGIP Graphical Models and Image Processing* **54** (1) 91–95, 1992.

[13] D. L. Jenkins, R. R. Martin, Applying Constraints to Enforce Users' Intentions in Free-Hand 2-D Sketches, *Intelligent Systems Engineering* **1** (1) 31–49, 1992.

[14] M. I. Kilani, R. H. Sturges, *Detection and Evaluation of Orientation Features for CAD Part Models*, Technical Report EDRC 24–66–91, Engineering Design Research Center, Carnegie-Mellon University, 1991.

[15] D. Knuth, J. Morris, V. Pratt, Fast Pattern Matching in Strings, *SIAM Journal of Computing* **6** 322–350, 1977.

[16] P. Kulkarni, D. Dutta, R. Saigal, *On Asymmetry Rectification*, Technical Report UM-MEAM-94-02, Dept. of Mechanical Engineering and Applied Mechanics, University of Michigan, 1994.

[17] J. Lin, S. Chou, W. Tsai, Detection of Rotationally Invariant Symmetric Shape Orientations by Fold-Invariant Shape-Specific Points, *Pattern Recognition* **25** (5) 473–482, 1992.

[18] E. H. Lockwood, R. H. Macmillan *Geometric Symmetry*, Cambridge University Press, 1978.

[19] G. Manacher, An Application of Pattern Matching to a Problem in Geometric Complexity, *Information Processing Letters* **2** 6–7, 1976.

[20] A. D. Marshall, R. R. Martin, *Computer Vision, Models, and Inspection*, World Scientific, Singapore, 1992.

[21] R. R. Martin, P. C. Stephenson, Putting Objects into Boxes, *Computer Aided Design* **20** 506–514, 1988.

[22] S. Parry-Barwick, A. Bowyer, Symmetry Analysis and Geometric Modeling, *Proc. DICTA 93, Digital Image Computing: Techniques and Applications*, Australian Pattern Recognition Society, ISBN 0–646–16522–4, 1993.

[23] S. K. Parui, D. Dutta Majumder, Symmetry Analysis by Computer, *Pattern Recognition* **16** (1) 63–67, 1983.

[24] K. Sugihara, An $n \log n$ Algorithm for Determining the Congruity of Polyhedra, *J. of Computer and System Sciences* **29**, 36–47, 1984.

[25] H. Weyl, *Symmetry*, Princeton University Press, 1952.

[26] J. D. Wolter, A. C. Woo, R. A. Volz, Optimal Algorithms for Symmetry Detection in Two and Three Dimensions, *The Visual Computer* **1** 37–48, 1985.

[27] H. Zabrodsky, *Computational Aspects of Pattern Characterization—Continuous Symmetry*, Ph.D. Thesis, Hebrew University in Jerusalem, 1993.

[28] C. T. Zahn, R. Z. Roskies, Fourier Descriptors for Plane Closed Curves, *IEEE Trans. on Computers* **21** (3) 269–281, 1972.