

Classifying Symmetry Sets

Margaret M. Fleck

Dept. of Engineering Science *
University of Oxford
Parks Rd
Oxford OX1 3PJ

This paper presents a fast algorithm for computing local symmetry descriptions of region shape. Like previous algorithms, it extracts groups of edge points tangent to a common circle. However, by examining the number of distinct points of tangency, the new algorithm separates groups belonging to round regions from those belonging to elongated regions. Thus, when these relations are connected to form extended regions, round and elongated regions can be processed differently.

The new implementation uses an edge-tracking algorithm to build extended regions. This handles the effects of limited precision and shape irregularities better than axis-tracking methods. In particular, Blum's idea of locating SAT branch points can be converted into a practical method of detecting locations at which three or more regions join. By combining a new density constraint with constraints used previously, the output and much of the processing is made linear in the image area.

One method of representing region shapes is to extract *symmetry sets*,¹ i.e. sets of edge points tangent to a common circle (the *symmetry circle*), and then group these sets into extended regions (Figure 1). These *local symmetry* relationships form an intermediate representation between raw edge points and descriptions of whole regions. Thus, a wider variety of regions can be identified than with whole region models as in [1].

However, existing local symmetry algorithms [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and theoretical descriptions [14, 15, 16, 17, 18, 19] are all unsatisfactory. Some algorithms are slow [8, 9]. Some handle only restricted classes of inputs [5, 6, 7] or detect only restricted classes of symmetries [10]. Some are hard to analyze [3]. Converting the raw symmetry sets into nice-looking regions requires complex post-processing [2, 8, 9, 12, 13]. Some formulations [10, 11] are sensitive to clutter and occlusion because they do not allow edges to lie within the tangent circle.

*The author is supported by a junior research fellowship funded by BP.

¹Terminology in this area varies considerably and a few authors define this term differently.

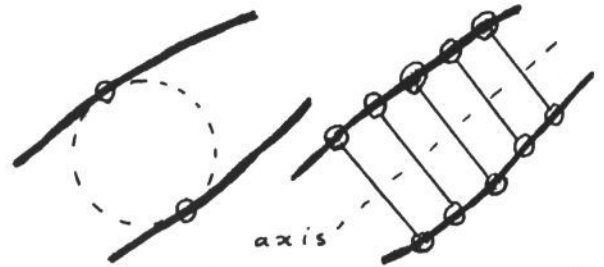


Figure 1: *Pieces of boundary opposite one another in an elongated region are tangent to a common circle (left). Many such pairs can be joined to represent the whole region (right), with their midpoints forming its axis.*

Underlying these superficial difficulties are deeper problems. None of these proposals cleanly models the distinction between round and elongated regions. Some descriptions concentrate on axis or center locations, which fragment badly in irregular regions. Furthermore, the number of symmetry relations among edges in an image grows faster than the area of the image, slowing down both the symmetry algorithm itself and any subsequent processing.

This paper presents a new algorithm that classifies symmetry sets as belonging to either elongated or round regions, so later processing can be adjusted for the type of region. Construction of extended regions uses edge tracking, not center or axis clustering. The algorithm can detect not only extended round regions, but also *connectors* where several regions join. Symmetry relations are constrained so that the number of raw symmetry relations, the number of output regions, and much of the processing is linear in the image area.

THE IDEAS

Classifying symmetries

In order to compute stable, intuitively acceptable local symmetry descriptions of shape, it is essential to distinguish round regions from elongated ones. In an elongated region, the symmetry pairs contain points opposite one another on two extended edges, as in Figure 1. The points midway between the two points in each symmetry pair form a natural axis for the region. The distance

between the points supplies a local width.

The local relationships in a round region can also be represented using symmetry sets. However, such a region is most naturally represented by relating many edge points to a common center, as in Figure 2. This center is at or near the centers of the symmetry circles relating the region's edge points to one another. With each edge point is associated a radius and an angular position.

When symmetry sets from round and elongated regions are given the same processing, this favors one of the two types of regions. Treating round regions as if they were elongated [7] results in large numbers of unstable axes. Treating elongated regions as round [3, 10, 11, 14, 15, 16] yields counter-intuitive descriptions of the width of, and the area enclosed by, tapered regions. Leyton's PISA points [18, 19] invite both sorts of problems. Previous algorithms for separating the two cases [2, 8, 9] have been messy and *ad hoc*.

Therefore, the two types of regions should be distinguished after computing symmetry sets, but before symmetry sets are linked together into regions. I claim that the two types of symmetry sets can be distinguished by how many times the symmetry circle is tangent to the edges. Symmetry sets with two patches of tangency belong to elongated regions, whereas those with three or more patches of tangency are parts of round regions.

More precisely, in order to be considered multiply-tangent, I require that a symmetry set contain at least 3 edge segments, each pair at least 40° apart. Failing this, the symmetry set is classified as doubly-tangent if it contains a pair of edge segments at least 80° apart. All other symmetry sets are discarded. These thresholds prevent slight perturbations in edge orientation from generating symmetry relations.

Doubly-tangent symmetry sets occur in two intuitively distinct environments: elongated regions and corners. Multiply-tangent symmetry sets occur both in round regions and at *connectors*, where several regions join. The images shown in Figures 3-4 illustrate all four types of regions. In irregular round regions, the centers of the symmetry circles scatter, but the algorithm described below still collects them into coherent regions.

The idea of classifying symmetry sets by patterns of tangent points goes back to [10, 11]. In particular, connector regions are closely related to Blum's SAT branch points. His original proposal, however, contained three fatal flaws. First, the cloud of symmetry sets generated by an irregularly circular region were not merged into a single region description. Second, resolution was not adjusted to prevent small irregularities from sabotaging the descriptions of large regions. Finally, his descriptions focus on the axis or circle center points, rather than on

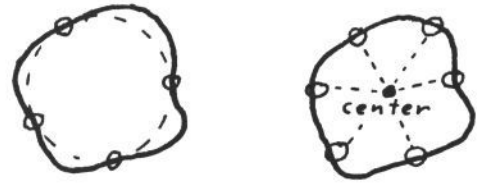


Figure 2: In a round region, sets of three or more pieces of boundary are tangent to a common circle (left). Sets with similar centers are merged to represent the whole region in terms of a common center (right).

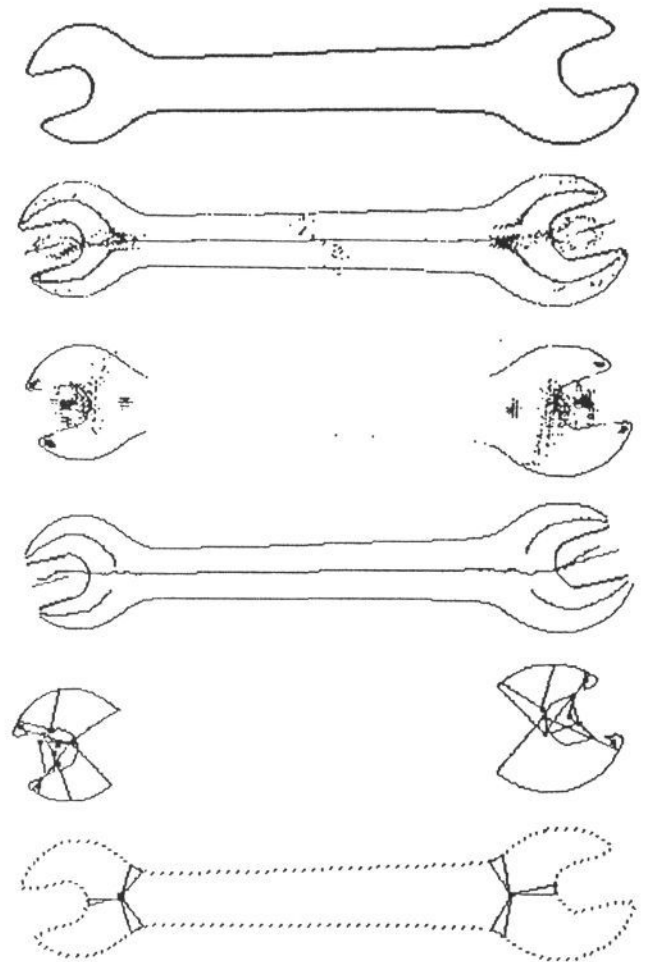


Figure 3: The boundary of a wrench (top), axis points for elongated regions and circle centers for round regions (upper middle), connected partial regions whose widths are not too large for their boundary lengths (lower middle), and possible connector regions (bottom).

the relationships among edge points.

Good and Bad Symmetries

An image containing L edge segments can generate at least $O(DL)$ symmetry sets, where D is the diameter of the image (i.e. the maximum distance between any pair of points in it): consider a set of parallel line segments. It can generate at most L^2 symmetry sets, although it is unclear whether this maximum can be attained. Many of these theoretical possibilities, however, do not represent parts of intuitively important regions. The new algorithm uses four constraints to limit the number of symmetries to $O(L)$. The underlying principle is not to waste effort computing symmetries that are doomed to be pruned by later processing.

The first constraint (“minimum angle”) is that the edge segments in each symmetry set must span a minimum angular range (currently 80 degrees), a consequence of the classification thresholds given above. As a consequence, the algorithm need not consider circle radii larger than $2\cos(40)D$, where D is the image diameter, because they cannot generate acceptable symmetries. Similar restrictions were imposed in [7, 8, 9].

The second constraint (“aspect ratio”) is that a good symmetry region must relate edge curves that are long compared to the region’s width [2, 8, 9]. Short regions are less acceptable intuitively and statistically more likely to occur by chance. For this reason, as the symmetry circle radius is increased, edge orientations are computed using wider support along the edge curve. This constraint does not apply to connector regions.

Segments near a sharp bend in an edge have a wide range of orientations, due to smoothing in the orientation computation. Very sharp bends are excluded from the symmetry computation (as in [7]) because they would generate large numbers of purely accidental symmetry relations. From this “curvature limitation”, it follows that computed normals in other sections of the edges change only slowly, so the symmetry computation can consider only a sampling of the edge segments.

The final, “density” constraint on the symmetry computation requires that symmetry sets must become sparser as their radius is increased. This is essential to ensuring linear output and is enforced implicitly by some algorithms [2], but it has not been articulated before. As Figure 5 illustrates, parallel edges generate many symmetries, violating this constraint. However, only narrow ones, or those that are locally minimal or maximal (e.g. that between the extreme ends of the radiator) seem intuitively interesting (compare the fork example in [12]).

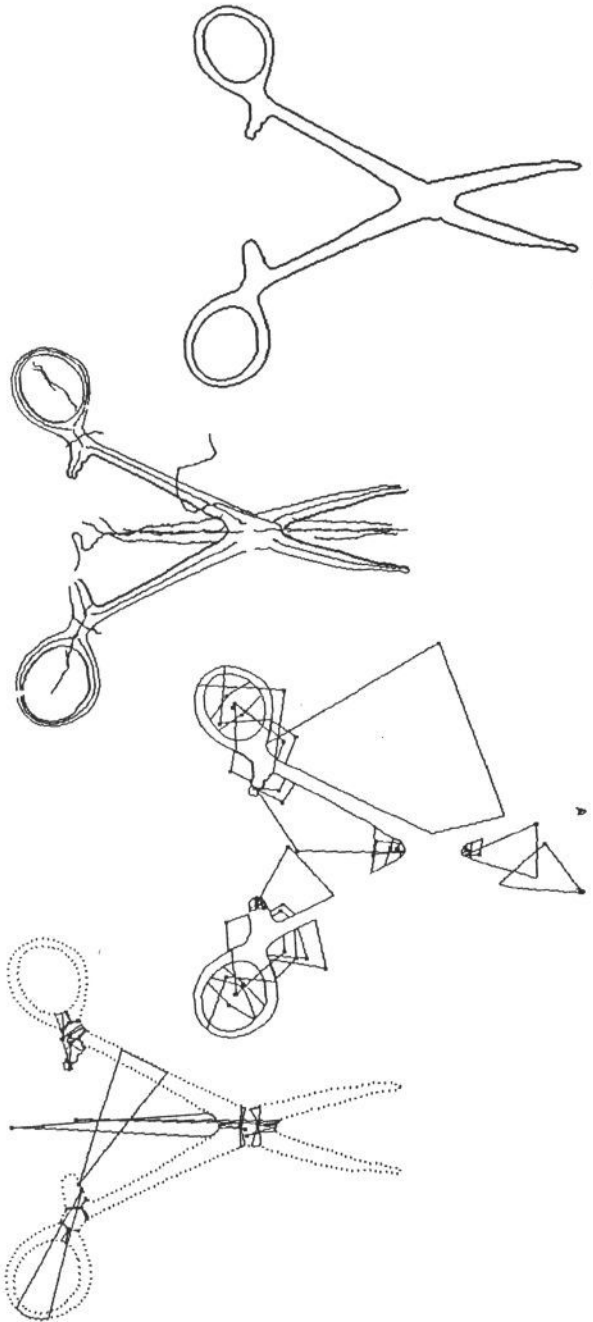


Figure 4: *The boundary of a hemostat, elongated regions, round regions, and possible connectors.*

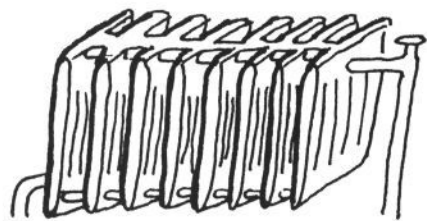


Figure 5: *The n parallel edges in the radiator generate n^2 symmetry regions, but few are perceptually significant.*



Figure 6: *The chain-link length of a curve is the length of a set of line segments approximating the curve, each no shorter than a constant c .*

THE ALGORITHM

The constraints described above make it possible to build a fast algorithm for computing and classifying symmetry sets. This initial stage of the computation is done separately for each circle radius r , $0 \leq r \leq 2\sin(40)D$. To prevent the description of a region from depending on its size, all thresholds used in the shape computation are proportional to the current radius. To enforce the density constraint, only selected radii are considered, though quantization in the algorithm allows symmetries at other radii to be detected. The sampling rate varies smoothly with the radius and is 1 radius value per $\max(1, \frac{r}{25})$.

Basic symmetries

The input to the algorithm is a set of circular, doubly-linked lists of edge segments, provided by the edge finder described in [20] and the tracker described in [21]. Intuitively, an edge segment is a section of boundary with a designated inner and outer side. This designation makes the edge segments easier to manipulate than raw boundary segments: they form simple closed curves and each belongs to at most one symmetry set per radius.

Various parts of the computation must measure the length of an edge. Under the standard definitions, fine texture increases curve length, which changes coarse-scale shape analysis. To avoid this problem, I use a new *chain-link* measure of curve length, an easily computed approximation to the methods in [2]. The curve is approximated with a series of line segments, ending at points on the curve (Figure 6). The endpoints are chosen as close together along the curve as possible, but each segment must be at least c pixels long. c is a constant proportional to the symmetry radius, currently $\max(1, \frac{r}{10})$. As c is increased, any fine texture is smoothed away in the approximation, so it does not affect edge length.

At each radius, the lists of edge segments are sampled at a rate of one element per $\max(1, \frac{r}{10})$ units of chain-link distance along the edge. The sampled list is represented by adding a second set of links to the input list. The orientation at each sampled segment x is then computed. It is the direction of the vector from the point D cells before x along the sampled contour to a point D cells after

it, for some constant D (currently 5 cells). To implement the curvature limitation described above, x is excluded if the straightline distance between $x-5$ and $x+5$ is too small ($\leq 10\sin(5)r$), or if its sampled contour contains fewer than 40 edge segments.

The algorithm then computes the symmetry center at radius r , for each edge segment x . This is just the point at distance r from x , in the direction normal to the edge, moving away from the boundary. The edge segments associated with each center are stored in a collector. The x - and y - components of center locations are quantized at a rate of $\max(1, \frac{r}{10})$ image cells per collector cell. Each segment is placed in the appropriate collector cell, plus the eight immediately adjacent ones. The collector is currently an array, but could also be implemented as a hash table (more efficient for sparse images).

Classification

Each symmetry set in the collector is classified as doubly-tangent, multiply-tangent, or no-symmetry using a bucketing technique. The algorithm determines which orientations are present in the symmetry set, but quantizes them to the nearest 20 degrees. The pattern of quantized orientations is then analyzed to determine whether they meet the angular criteria for either type of symmetry set.

A doubly-tangent symmetry set may actually contain many more than two edge segments, due to the quantization of center positions. If all pairs meeting the orientation criteria were returned, all possible symmetry regions would be found, but this would increase both the running time and the number of output symmetries above $O(L)$. Therefore, only a fixed number of pairs is selected to represent each symmetry set.

The current implementation selects one pair per doubly-tangent symmetry set, each of whose edge segments is at the maximum distance from the symmetry center among all edge segments at that quantized orientation. Alternatively, all pairs maximizing the distance could be chosen. Also, minimum distance pairs could be used. Any of these options, combined with the sampling of radii, enforces the density constraint. This constraint eliminates only intermediate alternatives in wide regions with close, parallel edges.

When classification for a particular radius is complete, symmetry sets are removed from the collector and attached to their edge segments. As computation proceeds through the range of radii, each edge segment accumulates two lists. One contains the symmetry centers and radii for each multiply-tangent symmetry to which the segment belongs. The other contains the other edge segment and symmetry circle radius for each segment in each doubly-tangent symmetry set it belongs to.

The work for all radii in this first stage of processing takes $O(L)$ time. At each radius r , processing at edge segments requires $O(\frac{L}{r})$ time. Searching or clearing the collector requires $O(\frac{L}{r})$ time for a hash table or $O(\frac{D^2}{r^2})$ for an array. Because radii are sampled, however, the total work is proportional to $\sum_{n=0}^{\infty} \frac{L}{K^n}$ or $\sum_{n=0}^{\infty} \frac{D^2}{K^{2n}}$, where K is a constant. Since both series converge, the total work is simply $O(L)$ (hash table) or $O(L + D^2)$ (array). The maximum number of output symmetry relations is $O(L)$, and $O(L)$ or $O(L + D^2)$ space is required.

Building regions

Once symmetry sets are computed for all radii, they are linked together to form connected partial regions. Local symmetry representations require regions to have connected edges: any gaps (e.g. due to attachment or occlusion) must be filled explicitly. This connectivity requirement is what distinguishes local symmetry representations from models such as the Hough transform [22, 23]. Irregular regions, particularly round ones, generate loose clouds of symmetry centers which make center-clustering or axis-tracking techniques difficult. As in [5, 6, 7, 8, 9], the new algorithm tracks along edges, using their connectivity to correctly identify related symmetry sets.

The tracker works its way through the edge curves twice, once building round regions and once building elongated regions. It uses the original linked lists of edge points, without subsampling, and considers symmetry sets obtained at all radii. The sampling used in previous parts of the algorithm means that only selected points along the edges of each region will have been assigned symmetries. In the gaps between them, symmetry relations must be inferred as regions are constructed.

For each type of region, the main tracker loop repeats the following sequence of actions:

- Find the next edge segment with a non-empty symmetry list,
- Extend the region forwards along the edge,
- Extend the region backwards along the edge, and
- Connect the ends of the region together, if possible.

Each region is represented as a list of ordered pairs of points: two edge points for an elongated region, or a edge point and a center location for a round region. Thresholds used by the tracker are proportional to the symmetry circle radius of the pair on the current end of the partial region.

Each extension of a region requires three steps:

- Find the next symmetry relation “similar” to the current end of the region,
- Interpolate symmetries for intermediate points,
- Add the new points to the region, and

- Remove from the edges all symmetries used in this extension, together with any “similar” ones.

In an elongated region, both edges are extended at the same rate, until a symmetry is found relating edge segments in the two extensions.

Two elongated-region symmetries are considered similar if their radii are similar and their edge segments are connected to and near one another along the edges (both in segment count and chain-link terms). For round region symmetries, the center locations must also be similar. These quantities also must not change too quickly if one looks a short distance back along the region. In case of multiple “next” elongated region relations, the one with the smallest width is chosen. For a round region, all similar center locations are averaged.

In searching for elongated region extensions and in removing similar elongated region relations, the algorithm must examine the symmetry lists attached to a number of edge segments, searching for symmetries to segments in or near the current extension. To do this efficiently, special markers are attached to segments in the extension as it is created. Since each symmetry list contains pointers to the original edge segments, it is then simply scanned for marked segments.

Most work in building regions is clearly $O(DL)$: in processing symmetry sets of radius r , the sampling of boundary length ($\frac{L}{r}$) cancels the effects of the search radius (r). The exception is deleting marked symmetry relations attached to edge segments. In this case, note that each edge segment can have at most $\sum_{r=1}^D \frac{1}{r}$ symmetry relations and can belong to at most $O(\sum_{r=1}^D \frac{1}{r})$ regions. Thus, the total number of segment examinations is $O((\sum_{r=0}^D \frac{1}{r})^2)$. This is $\sum_{r=0}^D \sum_{i=1}^{D-r} \frac{1}{i(i+r)}$ which is at most $\sum_{r=0}^D \sum_{i=1}^{D-r} \frac{1}{i^2}$, which is at most $\sum_{r=0}^D \sum_{i=0}^{\infty} \frac{1}{i^2}$. Since the infinite sequence converges, this is just $O(D)$ for each edge segment, i.e. $O(DL)$ total.

Thus, this second phase of symmetry computation requires $O(DL)$ time in the worst case. However, since the actual number of symmetries rarely hits the theoretical limits, it usually runs faster than the initial phase of computation. Since each symmetry pair from the first stage belongs to one and only one extended region, the final output contains only $O(L)$ regions, and the total space required is $O(DL)$.

CONNECTING REGIONS

The output partial regions are easily converted to either displays or traditional symbolic descriptions (e.g. length, width). To suppress clutter in the results displayed in this paper, regions with short boundaries were pruned. Specifically, round regions were required to cover at least

80 degrees and the shorter side of each elongated region was required to be at least half the average width of the region. This output is as clean than that of previous programs (e.g. [8, 9]), but would require further processing to form a final shape description.

An individual symmetry region almost never contains enough information to suggest a particular object. Thus, for object recognition, it is necessary to group nearby regions together into larger configurations. Previous algorithms [4, 5, 8, 9, 12, 13] have divided regions when they contain sharp changes in properties (e.g. axis orientation). Two regions are then merged if one smoothly continues the other. Looser associations ("subpart" relations) are created between pairs of regions whose ends connect, but with a sharp change in properties.

However, when three or more regions join, it may be impossible to connect the ends from any pair of regions. As Figures 3-4 illustrate, the connector regions introduced above could fill the gap between the other regions and provide a pair of edge contours for each region to attach to. Potential connectors are currently identified using a crude algorithm that finds sets of partial regions with a common center and at least three gaps, all under 180 degrees. Better algorithms are under development, but even these preliminary results are much more promising than those of [10, 11].

CONCLUSIONS

Many people may have been discouraged from using local symmetry representations because previous algorithms have either been slow or produced poor quality output. By contrast, the new algorithm is fast and produces clean output. For example, the wrench in Figure 3 (315 pixels long, radii up to 283 pixels) required only 5 minutes on a Sun-4 workstation. The new algorithm cleanly represents the similarities and differences between round and elongated regions in local symmetry representations. It can also detect a new type of region, connectors, which should prove useful in grouping regions to form object descriptions. Thus, the algorithm should allow local symmetry representations to be used in a wider variety of applications and with greater success.

References

- [1] Brooks, R. A. "Symbolic Reasoning Among 3-D Models and 2-D images" *Artif. Intell.* 17 (1981) pp. 285-348.
- [2] Saund, E. "The Role of Knowledge in Visual Shape Representation" Ph.D. thesis, Dept. of Brain and Cogn. Sci., MIT (1988) also *Artif. Intell. Lab. TR-1092*.
- [3] Scott, G. L., S.C. Turner, and A. Zisserman "Using a Mixed Wave/Diffusion Process to Elicit the Symmetry Set" *Im. and Vis. Comp.* 7/1 (1990) pp. 63-70.
- [4] Bagley, S. C. "Using Models and Axes of Symmetry to Describe Two-Dimensional Polygonal Shapes" MS thesis, Dept. of Elect. Eng. and Comp. Sci, MIT (1985).
- [5] Heide, S. S. "A Hierarchical Representation of Shape from Smoothed Local Symmetries" MS thesis, Dept. of Mech. Eng., MIT (1984).
- [6] Bookstein, F. L. "The Line-Skeleton" *Comp. Graph. and Image Proc.* 11 (1979) pp. 123-137.
- [7] Brady, J. M. and H. Asada "Smoothed Local Symmetries and Their Implementation" *Intern. Journ. Robot. Res.* 3/3 (1984) 36-61.
- [8] Fleck, M. M. "Local Rotational Symmetries" M.S. thesis, Dept. Elec. Eng. and Comp. Sci., MIT, also *Artif. Intell. Lab. TR-852* (1985).
- [9] Fleck, M. M. "Local Rotational Symmetries" *Proc. IEEE Conf. on Comp. Vis. and Patt. Rec.* 1986, pp. 332-337.
- [10] Blum, H. "Biological Shape and Visual Science (Part I)" *Jour. of Theor. Biol.* 38 (1973) pp. 205-287.
- [11] Blum, H. and R. N. Nagel "Shape Description using Weighted Symmetric Axis Features" *Patt. Recogn.* 10 (1978) pp. 167-180.
- [12] Connell, J. H. "Learning Shape Descriptions" MS thesis, Dept. of Elec. Eng. and Comp. Sci, MIT, also *Artif. Intell. Lab., TR-853* (1985).
- [13] Connell, J. H. and J. M. Brady "Generating and Generalizing Models of Visual Objects" *Artif. Intell.* 31/2 (1987) pp. 159-183.
- [14] Giblin, P. J. and S. A. Brassett "Local Geometry of Plane Curves" *Amer. Math. Monthly* 92/10 (1985) 689-707.
- [15] Bruce, J. W., P. J. Giblin, and C. G. Gibson "Symmetry Sets" *Proc. Roy. Soc. Edinburgh* 101A (1985) 163-186.
- [16] Giblin, P. J. and D. B. O'Shea "The Bitangent Sphere Problem" *Amer. Math. Monthly* 97/1 (1990) 5-23.
- [17] Brady, J. M. "Criteria for Representations of Shape" pp. 39-84 in A. Rosenfeld and J. Beck, eds., *Human and Machine Vision*, Academic Press, NY (1983).
- [18] Leyton, M. "A Process-Grammer for Shape," *Artif. Intell.* 34/2 (1988) 213-247.
- [19] Leyton, M. "Symmetry-Curvature Duality," *Comp. Vis. Graph. and Im. Proc.* 38 (1987) 327-341.
- [20] Fleck, M. "Multiple Widths Yield Reliable Finite Differences" Report 1827/90, Oxford Dept. Eng. Sci. (1990).
- [21] Fleck, M. "Some Defects in Finite Difference Edge Finders" Report 1826/90, Oxford Dept. Eng. Sci. (1990).
- [22] Ballard, D. H. "Generalizing the Hough Transform to Detect Arbitrary Shapes" *Patt. Recogn.*, 13/2 (1981) 111-122.
- [23] Davis, L. "Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms," *Patt. Recogn.*, 15 (1982) 277-285.