Skeletonisation using an extended Euclidean distance transform

Mark W. Wright, Roberto Cipolla and Peter J. Giblin

CUED/F-INFENG/TR 196

October 1994

Department of Engineering
University of Cambridge
Trumpington Street
Cambridge CB2 1PZ
England

Email: markwrsaifh.ed.ac.uk

Abstract

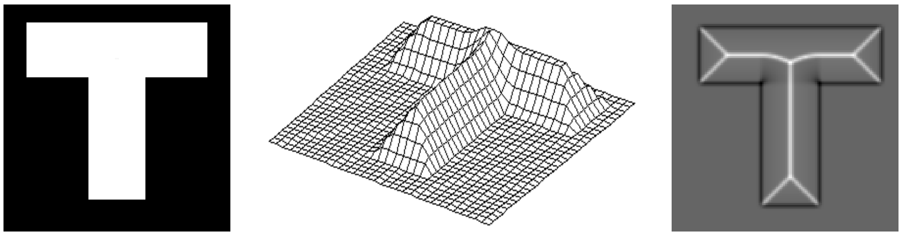
A standard method to perform skeletonisation is to use a distance transform. Unfortunately such an approach has the drawback that only the Symmetric axis transform can be computed and not the more practical smoothed local symmetries or the more general symmetry set. Using singularity theory we introduce an extended distance transform which may be used to capture more of the symmetries of a shape. We describe the relationship of this extended distance transform to the skeletal shape descriptors themselves and other geometric phenomena related to the boundary of the curve. We then show how the extended distance transform can be used to derive skeletal descriptions of an object.

Keywords: Symmetry, Skeleton, Distance transform, Singularity theory

1 Introduction

Skeletonisation is an image processing operation which reduces input shapes to axial "stick like" representations. It has many applications ranging from preprocessing for optical character recognition [1] to use as a shape descriptor in complete object recognition systems. Shape descriptions of this kind are also the subject of renewed interest in the context of robot manipulation and grasping of objects [2]. There are many types of skeletal shape descriptor and many ways in which they can be computed. Of particular note are the family of shape descriptors based on bitangent circles which include the Symmetric Axis Transform (SAT)[3], Smoothed Local Symmetries (SLS)[4], Process Inferring Symmetry Analysis (PISA)[5] and the Symmetry Set (SS)[6]. Differences within this family occur due to what bitangent circles are allowed to contribute to the skeleton and where the skeletal point is drawn in relation to each circle. Of the many methods which can be used to generate these and other skeletal shape descriptors most can be put into four categories: Iterative erosion of the shape boundary *i.e.* thinning, [7], wave propagation from the boundary [3], detection of "local maxima" on a distance transform [8], and analytical methods, for simple shapes, following some form of function approximation, *e.g.* polygon approximation or spline fitting [4, 9].

1.1 The distance transform approach to skeletonisation



(a) (b) (c)

Figure 1: The relationship of the SAT to the distance transform

Figure 1a shows a Tee shape. In figure 1b we see its distance transform, this assigns a value to each pixel within the shape equal to the minimum distance from that pixel to the shape boundary. The Symmetric Axis Transform of a Tee shape, figure 1c, corresponds to the "local maxima" of the distance transform.

The idea behind the distance transform approach is explained in figure 1. If we wish to obtain the symmetric axis transform of the tee shape in figure 1a then we first obtain its distance transform as in figure 1b. This assigns a value to each pixel within the shape equivalent to the distance of that pixel to the nearest point on the border. The symmetric axis of the figure corresponds to the "local maxima" of the distance transform as can readily be seen by comparing figures 1b and 1c.

The use of a distance transform is a general approach and specific techniques for the detection of "local maxima" vary widely. These include simple peak detection [10], thinning operations [8] and filtering [11]. Another interesting development is the use of active contours [12].

1.2 Limitations of the distance transform

The main drawback with the distance transform approach to skeletonisation is that it can only produce the Symmetric axis transform (SAT) which does not capture all the potential symmetries of a shape. The SAT is the locus of centres of circles bitangent to a shape boundary and *contained completely within that boundary.* In figure 2a we see the SAT for a rectangle, in figure 2b we see the extra symmetries created if all bitangent circles are allowed to contribute to the skeleton. In the SAT certain skeletal features are suppressed by the proximity of other boundary segments in a way which can seem counter intuitive. This can be important if the skeleton is to be used in an object recognition system. The presence of an occluding boundary for example can suppress symmetries which would otherwise contribute to the skeleton. This is one reason why the computation of skeletons which capture more symmetries than the SAT is a worthwhile goal.

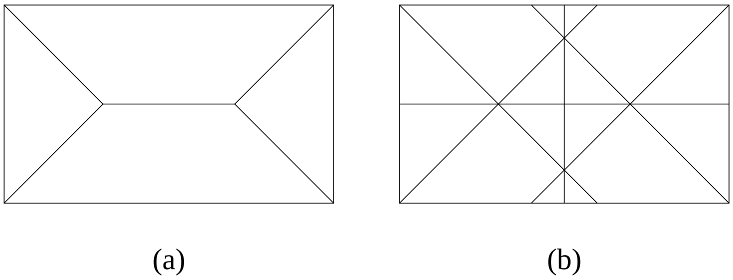


Figure 2: Limitations of the Symmetric Axis Transform
Figure 2a shows the SAT of a rectangle which is the locus of bitangent circles which are contained within the boundary. Figure 2b demonstates that there are many other symmetries which could be expressed.

1.3 Contribution

We address the problem of performing skeletonisation using a distance transform. Our main contribution is to present an *extended* distance transform derived using standard techniques from singularity theory. We demonstrate how this extended distance transform can be used to derive skeletal shape descriptors which capture more symmetries of a shape than can be acquired by the standard distance transform approach. We suggest that studying distance transforms and their role in skeletonisation from the point of view of singularity theory provides an interesting perspective with many valuable new insights.

2 The extended distance transform

2.1 An intuitive explanation

A central assumption of the distance transform approach is that skeletal axes correspond to the "local maxima" of the Euclidean distance transform. A mathematical analysis of this assumption from the point of view of singularity theory [13] is given below which leads to the introduction of the extended distance transform. However, before embarking on a formal theoretical analysis we feel it is instructive to give an intuitive description of the extended distance transform. Consider a point on a smooth planar curve as in figure 3. We can imagine tracking this point as it traverses the curve and making note of its inward normal which would sweep across the plane. Now consider this normal raised at an angle of 45 degrees to the plane so that its height above the plane encodes distance along the normal from the boundary. As this new inclined normal is swept along the curve it acts as a generator defining a surface in the space above the plane. It is this surface which is the *extended* distance transform or, as will be defined shortly, the *discriminant.*

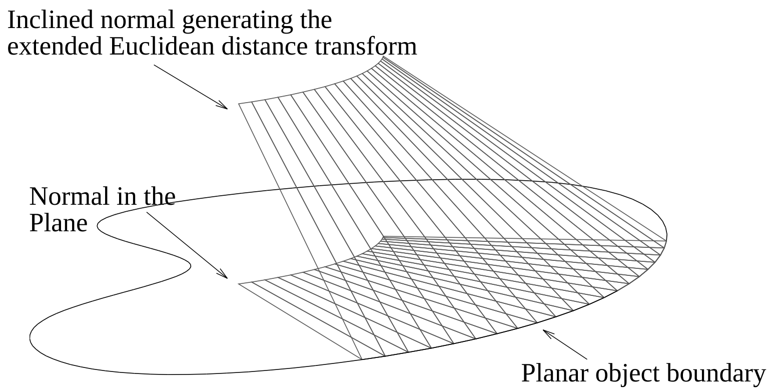


Figure 3: Intuitive explanation of the extended distance transform
We can imagine the extended distance transform being generated by a normal inclined at 45 degrees to the plane as it is swept around the boundary of a shape.

Figure 4 shows the extended distance transform or discriminant of a circle. It consists of a lower cone which corresponds to the ordinary Euclidean distance transform but extends above this to form another inverted cone. It is this upper portion which has the potential to interact with surfaces from other parts of the contour and thus define symmetries other than those of the SAT. In figure 5 we see the surface generated by a parabola. There are two significant type of "event" on this surface, *i.e* points where the surface appears "creased" and points where two smooth portions of the surface appear to "intersect". The creases correspond to the *evolute* of the curve which is the locus of centres of curvature and the intersections correspond to the skeletal symmetry points.

2.2 Theoretical Analysis

Let $\gamma(t)$ be a regular parameterization of a curve, which we can assume to be unit speed for simplicity of calculation. (That is, $\| (\gamma'(t)) \| = 1$, where the prime denotes differentiation.) The function

$$f(t,x,y) = \| \gamma(t) - (x,y) \|$$

measures the distance from the point $\gamma(t)$ of the curve to the general point (x,y) in the plane. (Compare [13, p.33], where the *square* of f (the 'distance-squared' function) is used. The only purpose of squaring is to make the function differentiable when (x,y) lies on the curve, and for our purposes we can ignore this problem.)

It is convenient to make the target space of the map f of the same dimension as the domain and we do this by 'enlarging' it to the map F given by the following formula:

$$\mathbb{R} \times \mathbb{R}^2 \xrightarrow{F} \mathbb{R} \times \mathbb{R}^2$$

$$t,x,y \longmapsto \| \gamma(t) - (x,y) \|, x, y$$

This can be thought of as taking a point $\gamma(t)$ on the curve, together with a point (x,y) in the plane, and mapping it to a point of $\mathbb{R} \times \mathbb{R}^2 = \mathbb{R}^3$ 'over' the plane point (x,y), at a 'height' equal to the distance of (x,y) from $\gamma(t)$.

We can use vanishing of the 3×3 Jacobian determinant of F to determine when F fails to be a 'local diffeomorphism', i.e. when it fails to be 'locally one-to-one'. The set of points (t,x,y) where this happens is called the **critical set** of F and is denoted Σ. Writing down the determinant, it is easy to check that it vanishes precisely when $\partial f / \partial t = 0$. This can be interpreted as saying that, for *fixed* (x,y), the distance from this point to the curve is a maximum or minimum, and geometrically this means that (x,y) is on the normal to the curve at $\gamma(t)$ (for a formal proof, see [13, p.33]).

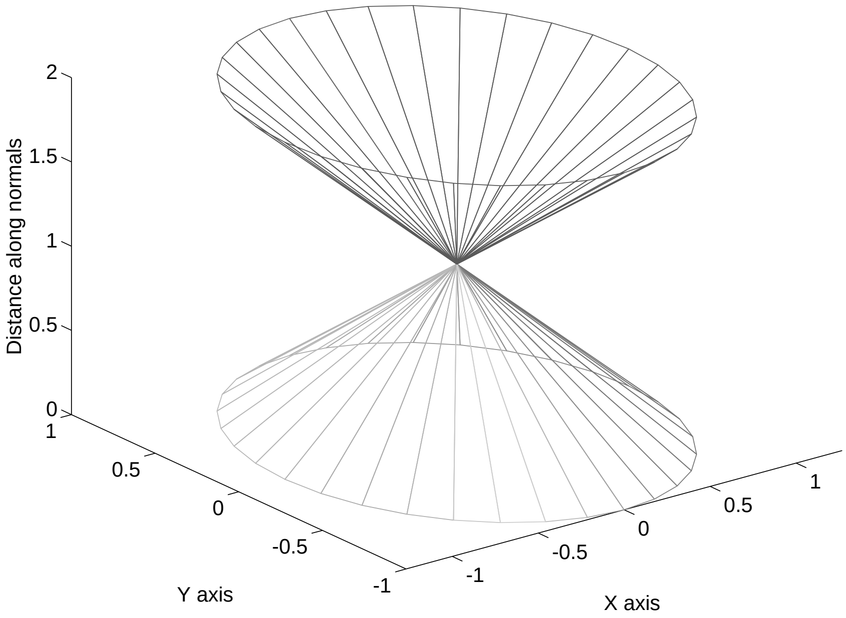


Figure 4: Discriminant for a circle
The discriminant can be thought of as normals to the original curve rising from the plain at 45 degrees. For a circle the discriminant looks like two cones with a solitary singularity at their common apex.

Thus the critical set

$$\Sigma = \left\{(t, x, y) \mid \frac{\partial f}{\partial t} = 0\right\}$$

can be visualised as the normals to γ, where the normal at $\gamma(t)$ is 'lifted up to height t' above the (x, y) plane.

The **critical locus** or **discriminant** Δ of F is defined by

$$\Delta = F(\Sigma).$$

This is a subset of the target space $\mathbb{R} \times \mathbb{R}^2$.

To see how this relates to the intuitive explanation above, fix t. In Σ there is a line l of points, namely the normal at $\gamma(t)$ but raised to height t. The image $F(l)$ is a line whose projection to the (x, y)-plane is still the normal line. But $F(t, x, y)$, for $(t, x, y) \in l$, is raised above this normal line to a height equal to its distance from the point $(0, \gamma(t))$ at the foot of the normal. Hence the image $F(l)$ is a line through $(0, \gamma(t))$ at $45°$ to the base plane $t = 0$.

The surface formed by the $45°$ lines is a *ruled surface* and will usually have singularities. The circle (Fig. 4) is a rather untypical case here in that all the lines actually pass through a common point. A more typical case is that shown in Fig. 5, where the surface has cusp edges and a swallowtail point.

It is our task first to characterise the singularities of the discriminant surface and then to outline methods for detecting them.

We use Singularity Theory to define the type and position of singularities which occur on Δ. The **critical set** Σ is a smooth surface and so Δ can have only (i) 'multi-local' and (ii) local (or intrinsic) singularities. We proceed to describe these in turn.

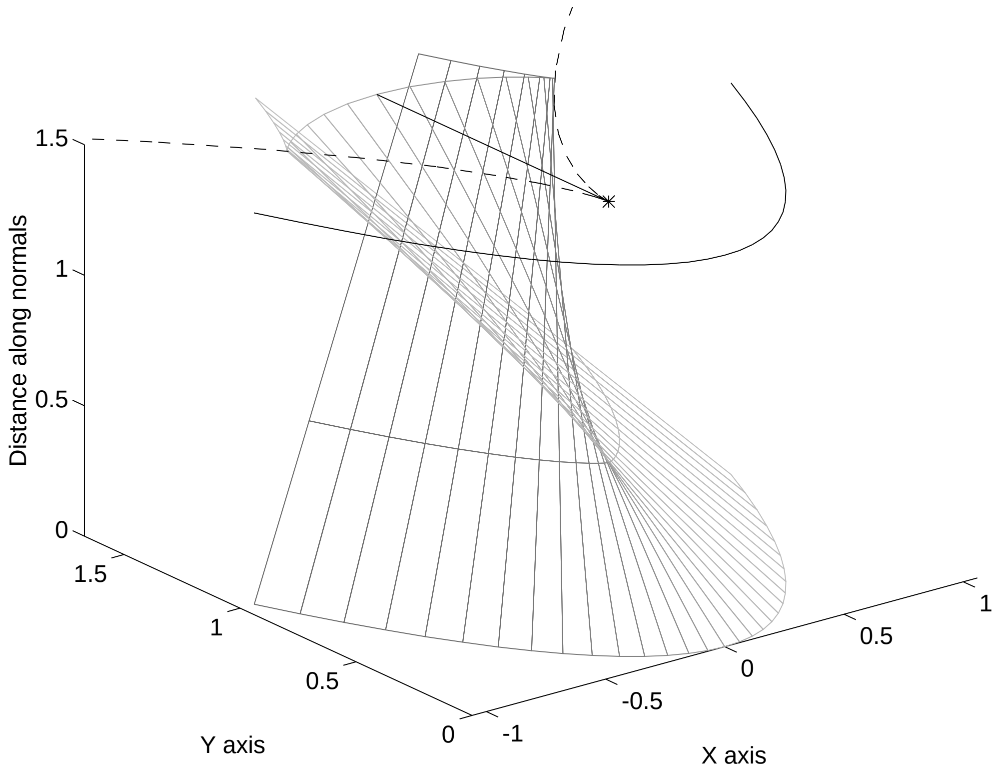


Figure 5: Discriminant for a parabola
The discriminant for a parabola shows a "swallow tail" surface. Above the discriminant is drawn the parabola complete with evolute (dashed line) and symmetric axis (straight solid line). The central symmetric axis corresponds to points of multi-local symmetry on the discriminant, the evolute corresponds to intrinsic singularities which are along the cusped edges of the "swallow tail". The only intrinsic singularity to have a minimum normal distance and therefore to be on the symmetric axis is that corresponding to the cusp on the evolute (asterisk).

2.3 Multi-local singularities

Multi-local singularities are caused by two points of Σ mapping to the same point of Δ. That is, there exist points $(t_1, x_1, y_1), (t_2, x_2, y_2) \in \Sigma$ with $F(t_1, x_1, y_1) = F(t_2, x_2, y_2)$. This clearly requires $x_1 = x_2$ and $y_1 = y_2$, so corresponds to two normals *of the same length*, intersecting at (x_1, y_1). (The lengths are measured from the 'feet' $\gamma(t_1), \gamma(t_2)$ of the normals to the intersection point.)

Equal length normals intersect in the centres of bitangent circles, that is, circles tangent to γ at two points, namely $\gamma(t_1)$ and $\gamma(t_2)$. Thus the points (x_1, y_1) generated in this way are part of the Symmetry Set ([6]). The point (x_1, y_1) will also be part of the Symmetric Axis Transform, if the equal normals are the shortest normals from (x, y) to the curve.

2.4 Local singularities

Local (or intrinsic) singularities of Δ are images of points $(t, x, y) \in \Sigma$ where the map $F|\Sigma$ (that is, F restricted to the smooth surface Σ) is singular *i.e.* fails to map Σ to a smooth surface in the image. It turns out that $F|\Sigma$ is singular when $(t, x, y) \in \Sigma$ and (x, y) is at the centre of curvature of γ at $\gamma(t)$. (The calculation for this is almost identical to that in [13, p.33].) This means that (x, y) is on the evolute which is the locus of centres of curvature (See figure 5). Note that $F(t, x, y)$ only belongs to the Euclidean distance transform when the distance along the normals from (x, y) to $\gamma(t)$ is the shortest among all normals through (x, y). In fact this never happens on the evolute except at cusps (corresponding to maxima and minima of curvature).

2.5 Relevance of analysis

The above analysis is of relevance for three reasons:

1. Singularity theory tells us exactly what type of singularity can exist, under what conditions they occur and their local structure [13, 14, 15, 16].

2. In the literature the artifacts on the Euclidean distance transform which correspond to skeletal points have been called "local maxima" [8]. This is a misnomer as strictly speaking they are not local maxima of any variable in the usual sense. Using singularity theory in this context these artifacts can be described simply and formally as singularities of the distance function.

3. The discriminant can be thought of as an *extended* Euclidean distance transform as it contains the EDT but also describes distances along normals which are not minimal. The description of singularities in such cases enables us to detect smoothed local symmetry and symmetry set points. This fact is exploited to advantage in the next section.

2.5.1 Interpretation of the standard Euclidean distance transform

In the analysis we described multi-local singularities as the intersection of equal length normals. Equal length normals intersect in the centres of bitangent circles, that is, circles tangent to γ at two points, namely $\gamma(t_1)$ and $\gamma(t_2)$. Thus the points (x_1, y_1) generated in this way are part of the Symmetry Set ([6]). The point (x_1, y_1) will also be part of the Symmetric Axis Transform, if the equal normals are the shortest normals from (x, y) to the curve.

This confirms that the so called "local maxima" of the Euclidean distance transform do indeed correspond to the SAT but we can now define them more formally not as "local maxima" but as singularities. We can also see that the standard Euclidean distance transform is a subset of the discriminant, specifically it is that portion of the discriminant consisting of normals running from the boundary of the shape and terminating at the first singularity they encounter. The standard distance transform is single valued for any position x, y on the plane where as the extended distance transform can be multi-valued. The only intrinsic singularities on the standard distance transform are points corresponding to cusps of the evolute at maxima of curvature whereas large sections of the extended distance transform have intrinsic singularities corresponding to the entire evolute.

2.5.2 Interpretation of wave propagation algorithms

The relationship between wave based methods such as Brady and Scott's algorithm and the extended Euclidean distance transform can be described using the concept of level sets. A familiar example of level sets are the contours on a geographic map, each individual contours tells us what points on the earths surface are at the same height above sea level. In a similar way we can construct level sets of the discriminant. Firstly a curve is defined by the intersection of a horizontal plane of height c with the discriminant and then this curve is projected down to the plane.

Formally the level set for c is the set (c, x, y) where $c = f(t, x, y)$ and (x, y) is on the normal at t. Thus (x, y) lies on the parallel of the curve at a distance c. The level set is this parallel lifted up to the discriminant surface.

In other words a wave front which has propagated a distance c from the boundary is the parallel to the original curve at distance c and can be thought of as the projection of the level set of the discriminant at height c to the plane. The entire wave propagation process thus computes the projection to the plane of all level sets of the discriminant from $c = 0$ to some positive upper bound in ascending order.

We demonstrate this relationship for a parabola in figure 6. It is easy to see that collisions of wave fronts in the plane correspond to multi-local singularities of the discriminant occurring at a particular level set. It is perhaps less intuitively obvious that the wave fronts should form cusps and that as the wave front progresses the cusps follow the evolute. These cusps correspond to local

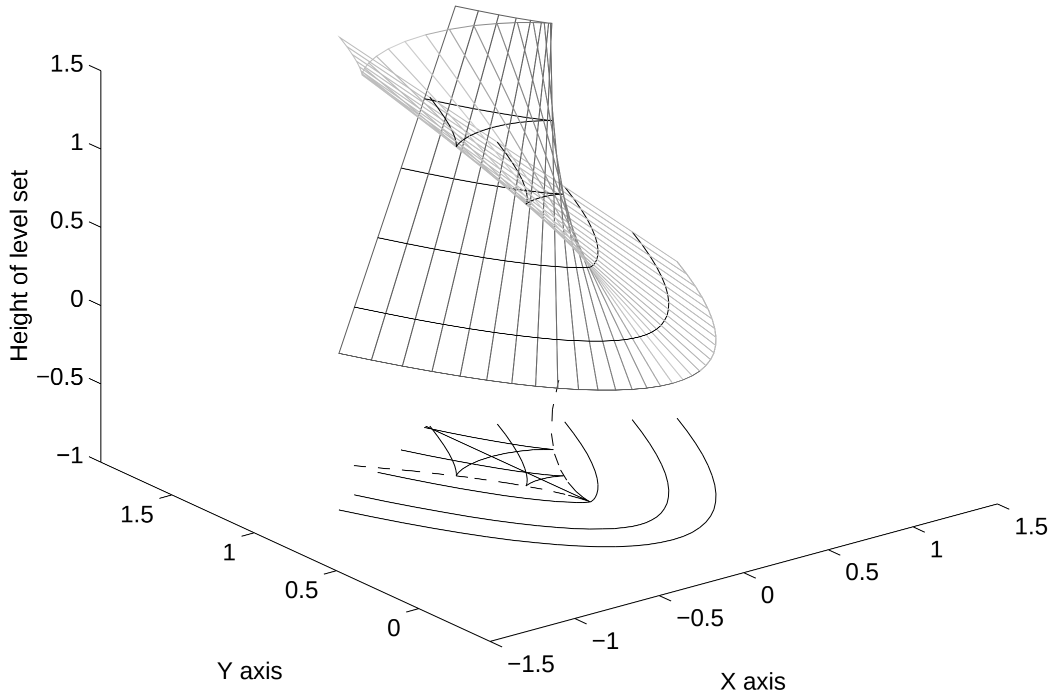


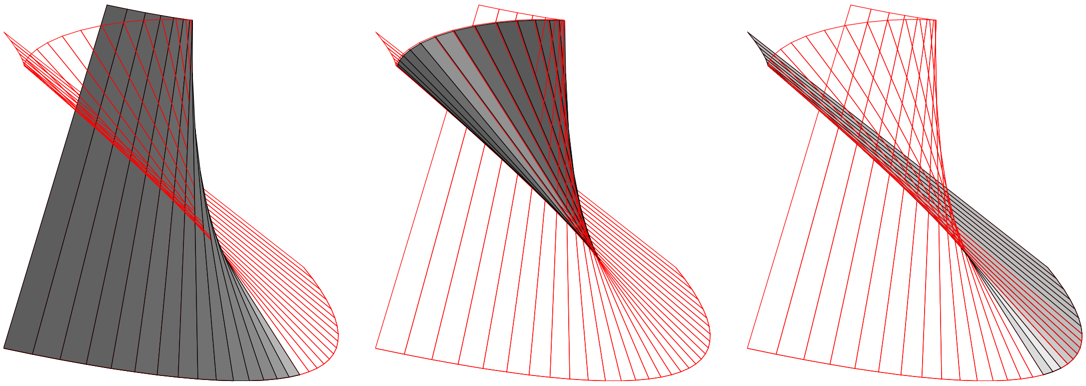
Figure 6: Discriminant level sets and wavefronts for a parabola
The discriminant for a parabola is a "swallow tail" surface. Below the discriminant is drawn the parabola complete with evolute (dashed line) and symmetric axis (straight solid line). Four waves or parallels of the parabola are seen to correspond to four distinct cross sections of the discriminant. Note that once the wave front encounters the evolute it forms cusps which propagate along the evolute.

singularities of the discriminant at a particular level set. It is interesting to note that as the wave propagates from the shape boundary the twin cusps of the "swallow tail" are created as the wave hits the evolute cusps associated with and pointing towards maxima of curvature on the boundary. These cusps then follow the evolute until they are annihilated at a cusp of the evolute associated with and pointing away from an adjacent minima of curvature unless there is an inflection in this case they become asymptotic to the normal at the point of inflection and propagate to infinity.

3 Principle behind the algorithm

In the previous section we have determined that skeletal points correspond to multi-local singularities on the discriminant where smooth portions of its surface appear to "intersect" one another. The goal of the algorithm therefore is to firstly compute the discriminant and then to detect these intersections.

A natural decomposition of this problem is as follows: Firstly, to dissect the discriminant into separate *single valued* sheets, (A single valued sheet is one where only one point on the sheet is above any one point on the plane). Secondly, to compute the distance from the boundary to every point on these sheets. Thirdly, to compare sheets to see if any occupy the same point over the plane and have the same distance from the boundary at that point. If this is the case then we can conclude that they intersect and their points of intersection can be marked as skeletal points. The key question in this approach is how can the discriminant be dissected into separate single valued sheets? This can be achieved by constructing sheets from curve segments bounded by two consecutive maxima of curvature and their corresponding evolutes.



(a) (b) (c)

Figure 7: The dissected discriminant of a parabola.
The discriminant of a parabola can be dissected into three separate sheets. These are of two types, pre-evolute sheets as in figure 7a and figure 7c and post-evolute sheets as in figure 7b.

Figure 7 shows how the discriminant can be thought to consist of separate sheets. Figure 7 shows a parabola which has a maximum of curvature in the middle which bisects the curve. Figure 7a shows one of the three separate sheets formed on the discriminant of the parabola. It consists of all inclined normals originating from the left half of the parabola and terminating at the first "crease" or local singularity they encounter on the discriminant. Figure 7c shows another sheet which due to the symmetry of the parabola can be considered a mirror image of the first. This consists of all inclined normals originating from the right half of the parabola and again terminating at the first "crease" or local singularity they encounter. It is the intersection of these two sheets projected down the plane which forms the skeleton of the parabola. Figure 7b shows the third sheet of the parabola discriminant. This consists of all normals which have passed beyond the "crease" in the discriminant. In the case of the parabola this doesn't intersect with any other sheet and therefore doesn't contribute to a skeletal branch. If a single sheet was constructed from the whole curve then the portions in figure 7a and figure 7c would together form a self intersecting sheet.

4 Implementation of the algorithm

The process of going from raw image to skeletal description involves three main stages: preprocessing, computation of discriminant and detection of singularities on the discriminant.

4.1 Preprocessing

The input to the algorithm is a grey level image. Edge segments are found using a Canny edge detector and the edge segments from this are chained together using a simple linker. These edge chains are then approximated by cubic B-Splines. These preprocessing stages are very similar to that found in the work of Saint-Marc and Medioni [9] on symmetry detection and that of Chong *et al* [17] on fingerprint data compression. The output from the preprocessing stage is a list of separate spline chains. Each chain conists of the coordinates of the splines control polygon vertices as floating point numbers and additional information such as if the curve is open or closed.

4.2 Computation of Discriminant sheets

This part of the algorithm computes the extent of the separate discriminant sheets over the plane. A pseudo code listing is given in figure 8. The input to this stage are the separate spline chains. The output is a list of blocks of pixels which lie under particular sheets.

```
For each spline chain{
    Compute curvature at sample intervals.
    Compute maxima of curvature along chain.
    Compute evolute for spline chain.
    Break chain into segments between maxima of
    curvature and spline boundaries.
    For each segment{
        Create a polygon consisting of the spline boundary
        within the segement and its corresponding evolute.
        Call a scanline conversion algorithm to find what
        pixels lie within the polygon.
        Output segment number and list of pixel coordinates.
    }
}
```

Figure 8: Pseudo code for computation of discriminant sheets

Firstly the curvature κ along the splines is computed, this is defined by the following equation:

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}$$

First and second derivatives of the splines required for this calculation can be obtained by differentiating the polynomials which make up the spline basis matrix to obtain new matrices. Maxima of curvature are found by a simple search and then the evolute E is computed which is defined as:

$$E : \left(x - \frac{y'(x'^2 + y'^2)}{x'y'' - x''y'}, y + \frac{x'(x'^2 + y'^2)}{x'y'' - x''y'} \right)$$

The curve is then broken into segments between maxima of curvature and spline boundaries. This ensures that no sheet can overlap itself. For each of these sheets a polygon with floating point vertices is created consisting of the spline boundary segment and its corresponding evolute. These are two open curves so the polygon is closed by defining sides between their two end points. A scanline conversion algorithm due to Hackbert [18] is then called. Scanline conversion is a standard computer graphics technique which when given an arbitrary polygon returns the pixels that lie within the polygon. This is done in order to registered the sheets to the underlying pixel grid. This means that when distances on separate sheets are compared they are sampled at precisely the same point over the plane for each sheet.

Figure 9 depicts a very simple shape to explain how the discriminant is dissected. The numbers denote the control polygon of the round spline boundary. The evolute is the cusped curve inside the shape. One sheet of the discriminant is shaded in between two consecutive positive curvature extrema (asterisks) and the evolute. This comprises of two shaded areas because there is a spline boundary between the extrema. There are a further two partially overlapping segments to be shaded.

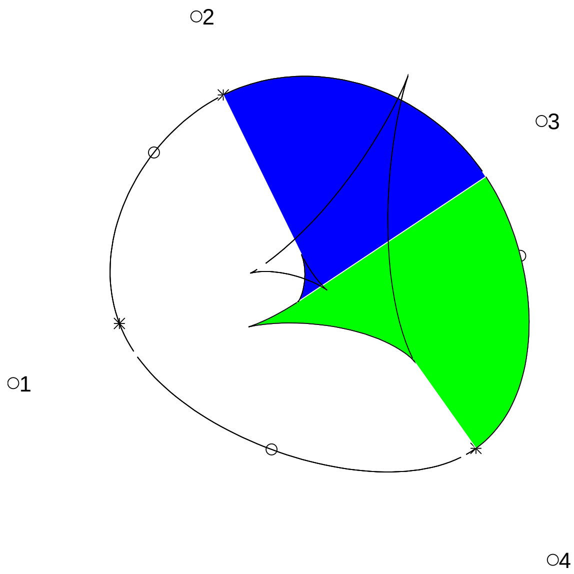


Figure 9: Dissection of Discriminant
Simple spline object (round shape) with control polygon denoted by numbers. Within the shape we see the evolute and one section of discriminant shaded. Section is bounded by positive curvature extrema (asterisks) and evolute.

4.3 Detection of singularities on the discriminant

The preceding stage of the algorithm gave us separate smooth sheets of the discriminant. These consisted of a number of spline segments each with an associated block of pixels. We now have to find where these sheets intersect and to mark these intersections as skeletal points. The basic requirement is to compute the distance of every pixel from its associated spline segment and then to search to see if any other segment has a pixel of same coordinates and of the same distance from that spline segment.

This part of the algorithm could be very inefficient in terms of time and memory if care is not taken in its implementation. The memory requirement varies over the image with some pixels having many sheets and others having few. The distances from pixels to spline segments are computed using a root finding algorithm due to Schieder, full details of which are given in [18]. This is rather slow and so should only be applied when necessary. Large portions of typical images only have a single sheet over them and so distance calculation is unnecessary.

It was decided that an efficient solution would be to use an insertion sort implemented with linked lists [19]. A pseudo code listing of this part of the algorithm is given in figure 10. Firstly an array of pointers to list records is created which is the size of the image, *i.e.* there is a pointer for each pixel. In the beginning this array is empty but as the algorithm progresses lists of arbitrary size can be hung from each pixel pointer. Each pixel for each sheet is considered in turn. A list record is created. If there is no list at this pixel location the record is inserted to start a list but the distance to the spline segment is *not* computed as it may not be needed. Only when a second record is added is the distance for the first record computed thus saving much computation time. The record is added to the list using an insertion sort based on the distances contained in each record. If adjacent records have distances within a certain threshold, typically 1 to 2 pixel widths then the pixel is signalled as a skeletal point. This avoids the need for a further search through the lists for intersections. As large portions of typical images only have one or a low number of sheets covering them it was thought that more efficient sorts [20] were unnecessary.

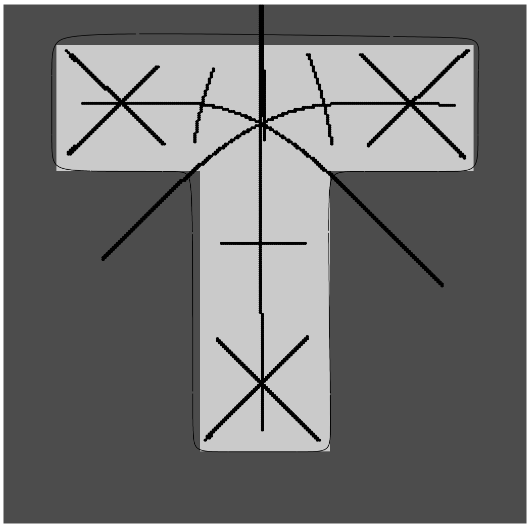
```
Allocate memory for image sized array of pointers to link lists
For each segment of spline{
    Input list of pixels within segment polygon.
    For each pixel in list{
        Perform insertion sort using linked lists.
    }
}

Insertion_sort(pixel_coordinates)
    Create a single linked list record.
    Insert pixel coordinates and spline segment into record.
    Check linked list image array at pixel coordinates.
    If(No list at this location)
      insert new record and return.
    elseif(Only one record in linked list)
        Compute distance of pixel to spline segment of existing record.
        Insert distance into existing record.
        Compute distance of pixel to spline segment of new record.
        Insert distance into new record.
        Traverse list performing insertion sort based on distance.
        If(neighbouring record distances < threshold)
            Mark Skeletal point in output image
    Else(More than one record in linked list)
       Compute distance of pixel to spline segment of new record.
       Insert distance into new record.
       Traverse list performing insertion sort based on distance.
       If(neighbouring record distances < threshold)
           Mark Skeletal point in output image
```

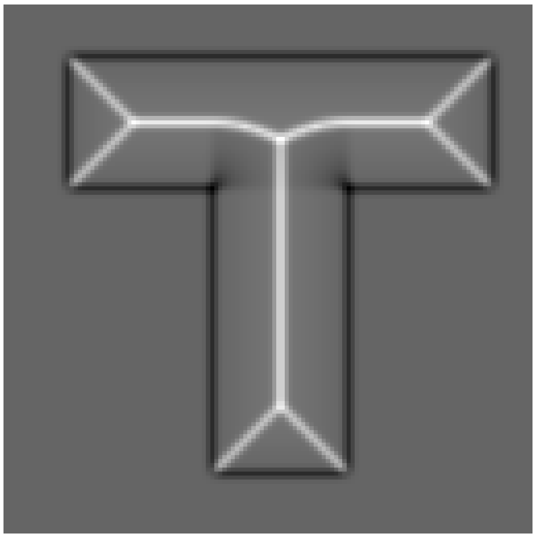
Figure 10: Pseudo code for detection of singularities on the discriminant

4.4 Experimental results

Figure 11 gives a comparison of the skeletons produced by the extended distance transform and the standard approach. The skeleton of a tee shape in figure 11a produced by the extended distance transform can be seen to make more symmetries explicit than the symmetric axis transform (SAT) in figure 11b derived from the standard distance transform. In particular it depicts more of the vertical axis of symmetry than the SAT and more of the diagonal symmetries emanating from corners. It also marks the symmetry between the base and the top of the tee which is entirely omitted in the SAT. However the new skeleton does not reach to the corners of the shape and there appear to be additional branches near the cross of the tee which at first do not appear to have any intuitively obvious cause. These artifacts arise because the discontinuous corners of the tee shape are represented by smooth spline sections of finite radii. At convex corners the skeleton can only approach as near to the border as the cusp on the evolute. The concave corners interact with themselves and straight edge segments to produce skeletal branches which are long compared with the smooth corner sections which contribute to them.



(a)



(b)

Figure 11: The additional information aquired by the extended distance transform

Figure 11a shows the skeleton for a Tee shape obtained using the extended distance transform whereas figure 11b shows the symmetric axis transform of the same shape obtained using the standard distance transform. We can see that many more symmetries are made explicit using the extended distance transform. Skeletal branches do not touch the shape boundaries in figure 11a, this is due to the fact that the discontinuous corners of the original shape are approximated by smooth splines.

Figure 12 shows the result of applying the extended distance transform to the skeletonisation of a wrench. In addition to the standard SAT branches running inside the wrench other symmetries are made explicit. Extra branches are formed in the jaws and the end to end symmetry of the jaws is captured by large lateral branches. The SAT is a subset of this skeleton and it is interesting to compare their local structure. Note the crossing of branches in the ends of the wrench which would be a three branch junction in the SAT.

Figure 13 displays the remarkably complex symmetries of the human hand. These arise partly from the multiple symmetries created between the fingers and thumb. The best example is perhaps the four "echoing" skeletal branches to the right of the thumb. This figure also highlights a limitation of the thresholding technique used to detect the intersection of the discriminant sheets. Boundary sections with normals pointing towards one another create thin skeletal branches as in the fingers. Boundary sections where normals point in a similar direction stay close for longer and produce broad skeletal branches. The arcs at the base of the fingers are examples of such boundaries.

The example of the screwdriver in figure 14 shows how the problem of broad skeletal branches can be alleviated to some extent. This is done by detecting a sign change or zero crossing in the subtraction of distances between sheets. We have indicated positive and negative results by two different shades. The border between the two regions on the branches marks the zero crossing and the position of the skeleton to within one pixel. In this example the skeletal branches protruding from the handle appear to mark important geometric relationships. The branches extending laterally from the shaft seem less important and are due to minor fluctuation of the spline which are probably artifacts of the spline fitting process.

5 Discussion

5.1 Efficient discriminant computation

The method we have used for computing the discriminant is accurate but rather inefficient. Our main concern to date has been to prove the principle of our algorithm, however if this technique is to be viable we need a more efficient method for deriving the discriminant surfaces and detecting their intersection. In this section we discuss two possible alternatives for these tasks.

One method would be to compute each smooth surface of the discriminant using separate *constrained distance transforms*. Constrained distance transforms are used in the literature for tasks such as path planning. These algorithms tend to be more complex than ordinary distance transforms particularly when the distance is propagated from many source pixels. This problem may be circumvented if we take advantage of certain constraints which apply in our case. If an evolute has an inflexion then the corresponding involute (the shape boundary in this case) will have a cusp. By the converse argument if the involute is smooth (which is one of our assumptions) then the evolute will never have an inflexion. So for any section of evolute running between two cusps the curvature of the evolute will have constant sign. This means we can use an ordinary distance transform if we use the following computational sleight of hand. We can initialise the pixels on the evolute with their corresponding radius of curvature. This would mean that the evolute would act as a kind of "wave guide" for the distance transform as it propagated from the source pixels.

For the second method, if we sample the boundary finely enough using closely spaced normals, the discriminant appears to be made up of many thin rectangular strips. Discriminant values within each strip can be interpolated from radii of curvature calculated at their vertices. This kind of process is common in graphics applications such as shading calculations. Indeed many workstations have dedicated hardware to perform this operation in real-time for thousands of quadrilaterals. In principle this would open the possibility for real-time computation of the discriminant for a moving or deforming object.

5.2 Detection of discriminant intersection

Currently detection of discriminant intersection points is due to a simple thresholding method. We have demonstrated that this can lead to broad skeletal branches in some cases. The detection of zero crossings in the subtraction of distances between sheets does offer a solution but there may be better techniques. Leymarie and Levine [12] use active contours to detect the SAT on the standard Euclidean distance transform. It may be possible to use this type of technique on the extended distance transform thus producing a continuous skeleton.

5.3 Advantages and limitations

The quality of skeletons produced by our algorithm are good. This is partly due to the production of a dense and precise distance map made possible in part by the use of splines to approximate the boundary. The algorithm produces much more information than the SAT and it is not adversely effected by incomplete segmentation in the preprocessing stages. Comparisons with the SAT and other skeletonisation routines is difficult as most assume segmented binary data as input and this presupposes preprocessing of unspecified nature and parameters.

The use of splines does however impose limitations on the approach. The repeatable fitting of splines to data is a difficult problem. Residuals can lead the spline to oscillate and produce false skeletal branches. The detection and modelling of discontinuities is another important issue

6 Conclusion

We have introduced an extended distance transform and demonstrated that it can be used to generate more general skeletal shape descriptors than can be produced using the ordinary distance transform. The current algorithm is promising and although inefficient in terms of speed and memory usage we hope to implement several major improvements shortly. In closing we hope this paper has demonstrated that singularity theory provides a powerful tool in the study of distance transforms and their use in computer vision.

7 Acknowledgements

Mark Wright would like to thank Edinburgh University and in particular Bob Fisher for the use of computer resources in the preparation of this paper and Andrew Fitzgibbon for practical help and proof reading. He would also like to remember the late Prof F. Fallside for his encouragement at the very beginning of this work.

References

[1] L. Lam and C.Y. Suen. Evaluation of thinning algorithms from an ocr viewpoint. In *International Conference on Document Analysis and Recognition*, pages 287–290. IEEE, October 1993.

[2] A. Blake, M. Taylor, and A. Cox. Grasping visual symmetry. In *Fourth International conference on computer vision*, pages 724–733, 1993.

[3] H. Blum. Biological shape and visual science (part 1). *Journal of Theoretical Biology*, 38(2):205–287, February 1973.

[4] M. Brady and H. Asada. Smoothed local symmetries and their implimentation. A.I.Memo 757, MIT, February 1984.

[5] M. Leyton. A process-grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[6] P.J. Giblin and the late S.A. Brassett. Local symmetry of plane curves. *American mathematical monthly*, 92(10):689–707, 1985.

[7] C. J. Hildich. Linear skeletons from square cupboards. In B.Meltzer and D.Michie, editors, *Machine Intelligence IV*, chapter 22, pages 403–420. Elsevier,New York, 1969.

[8] C. Arcelli and G. S. di Baja. Finding local maxima in a pseudo-Euclidean distance transform. *Computer Graphics Vision and Image Processing*, 43:361–367, 1988.

[9] P. Saint-Marc and G. Medioni. B-spline contour representation and symmetry detection. Technical Report IRIS#262, Institute for robotics and intelligent syutems, University of Southern California, Los Angeles, California 90089-0273, 1990.

[10] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 1966.

[11] M. W. Wright. Skeletonistion as model based feature detection. In *4th International Conference on Image Processing and its Applications*, Maastricht, Holland, April 1992. IEE.

[12] F. Leymarie and M.D.Levine. Simulating the grassfire transform using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):56–75, January 1992.

[13] J.W Bruce and P.J. Giblin. *Curves and singularities.* Cambridge University Press, second edition, 1992.

[14] J.W. Bruce and P.J. Giblin. Growth, motion and 1-parameter families of symmetry sets. *Proceedings of the Royal society of Edinburgh*, 104A:179–204, 1986.

[15] T.F. Banchoff and P.J. Giblin. Global theorems for symmetry sets of smooth curves and polygons in the plane. *Proceedings of the Royal Society of Edinburgh*, 106A:221–231, 1987.

[16] V.I. Arnol'd. Wave front evolution and the equivalent morse lemma. *Communications on pure and applied mathematics*, 29:557–582, 1976.

[17] M. M. S. Chong, R. K. L. Gay, H. N. Tan, and J. Liu. Automatic representation of fingerprints for data compression by B-spline functions. *Pattern Recognition*, 25(10):1199–1210, 1992.

[18] Andrew. S. Glassner. *Graphics Gems.* Academic Press, 1990. ISBN 0-12-286165-5.

[19] J. F. Korsh and L. J. Garrett. *Data structures, algorithms and program style using C.* PWS-Kent, 1992. ISBN 0-87150-099-X.

[20] R. Sedgewick. *Algorithms in C.* Addison Wesley, 1990. ISBN 0-201-51425-7.

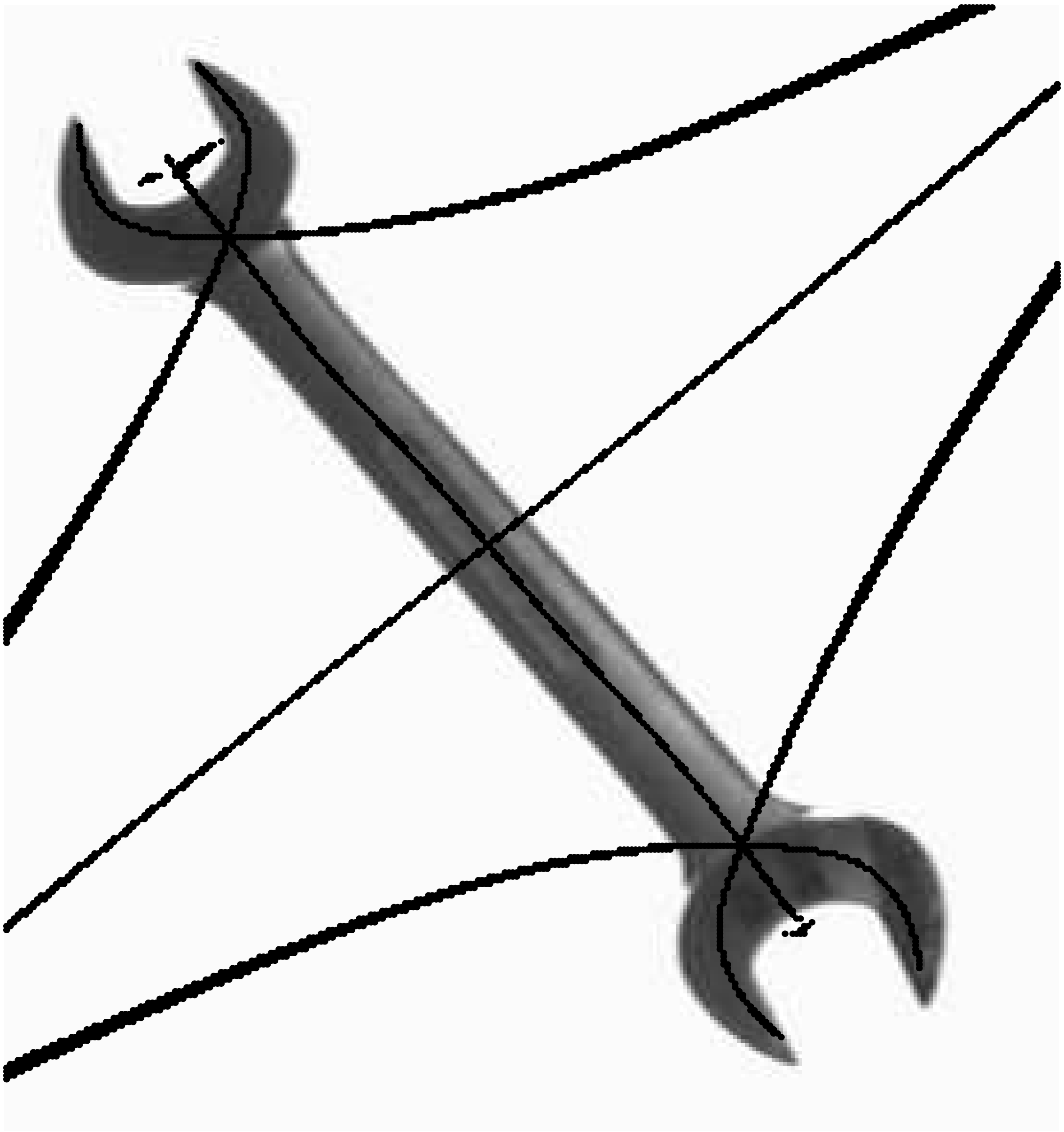


Figure 12: Skeleton of wrench produced by the extended distance transform
The skeleton is superimposed on input image. Large lateral branches are produced denoting the end to end symmetry of the jaws. These would be suppressed in the SAT.

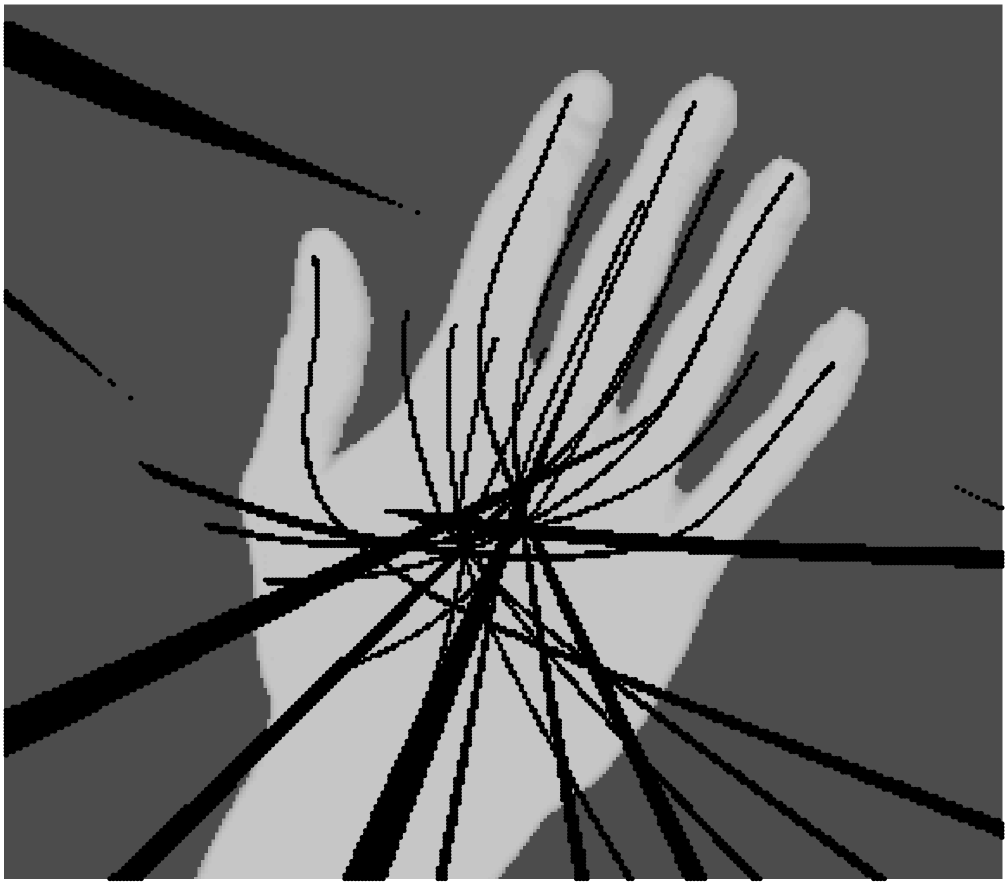


Figure 13: The symmetries of a hand using the extended distance transform
The complex symmetries of the hand are due in part to multiple symmetries between the fingers and thumb. Note the wide skeletal branches caused by intersecting sheets of the discriminant meeting at oblique angles.

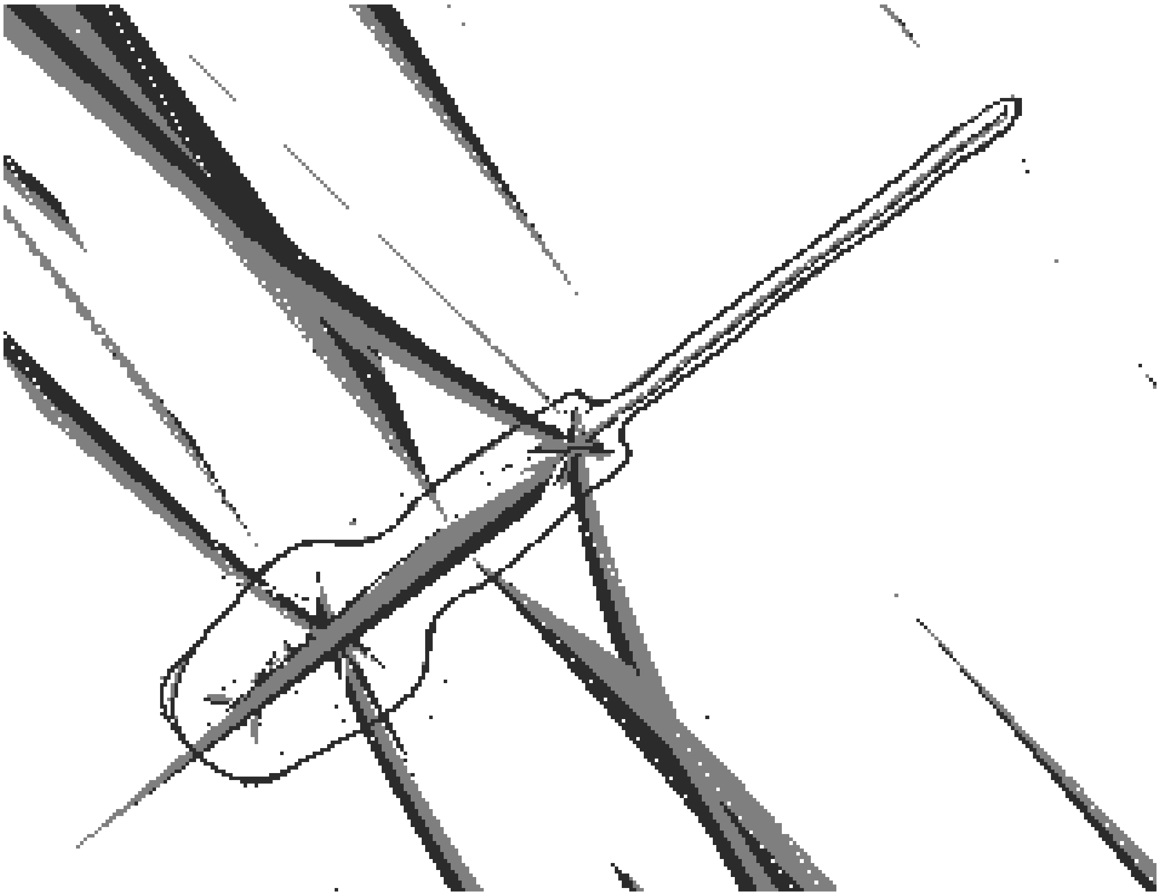


Figure 14: The skeleton of a screwdriver using the extended distance transform

We can determine the center of a broad skeletal branches by detecting the sign change in the subtraction of distances between sheets. Note that even fairly flat boundaries can produce skeletal branches some distance from the object.