MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1397 January, 1993

Complexity as a Scale-Space for the Medial Axis Transform

Ronald D. Chaney

Abstract

The medial axis skeleton is a thin line graph that preserves the topology of a region. The skeleton has often been cited as a useful representation for shape description, region interpretation, and object recognition. Unfortunately, the computation of the skeleton is extremely sensitive to variations in the bounding contour. Tiny perturbations in the contour often lead to spurious branches of the skeleton. In this paper, we describe a robust method for computing the medial axis skeleton across a variety of scales. The resulting scale-space is parametric with the complexity of the skeleton representation. The complexity is defined as the number of branches in the skeleton. A set of curves is computed to represent the bounding contour across a variety of complexity measures. The curves possessing larger complexity measures represent greater detail than curves with smaller measures. A medial axis skeleton is computed directly from each contour. The result is a set of skeletons that represent only the gross structure of the region at coarse scales (low complexity), but represent more of the detail at fine scales (high complexity).

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-91-J-4038.

1 Introduction

The medial axis skeleton is a thin line graph that preserves the topology of a region. The skeleton has often been cited as a useful representation for shape description, region interpretation, and object recognition. The skeleton provides a decomposition of the region into salient subparts. It also provides a description of the connectivity of the subparts.

Unfortunately, the computation of the skeleton is extremely sensitive to variations in the bounding contour of a region. Tiny perturbations in the contour often lead to spurious branches of the skeleton. It is non-trivial to determine which of the branches are spurious and which correspond to significant subregions.

There have been numerous attempts to find a robust algorithm for computing the medial axis skeleton (see, for example, [1], [2], [4], [5], [6], [9], [10], [13]). Most algorithms use some deviation of morphological thinning. Often, spurious branches are eliminated based upon some approximate property of the bounding contour, or based upon some property of the branch itself.

One common problem with previous approaches is that the resulting skeleton is inconsistent with the bounding contour or the region from which it was computed. Inconsistencies between the representations of the skeleton and the contour may lead to inconsistent inferences in higher level processes. If such inconsistencies could be eliminated, the performance of higher level processes would be improved.

Another problem with previous approaches is that the results are typically mediocre. Most algorithms are only capable of handling simple objects like pseudopods, for example[9]. These algorithms fail because they are incapable of distinguishing between "noise" in the data and subtle features that may exist on the contour. As a result, most algorithms tend to produce skeletons with spurious branches, or they tend to provide skeletons that are unduly simplified.

Because computation of the skeleton is so sensitive, it is desirable to represent the skeleton across a variety of scales. The multiple scale description eliminates the need to determine the "optimal" scale by some artificial means. Attempts to find an optimal scale parameter in this and other contexts typically yield only marginal results.

Ideally, a scale-space for the medial axis skeleton would provide representations of the skeleton with varying levels of detail. At finer scales, the skeleton would have a larger number of branches; a greater number of features would be represented. At more coarse scales, the skeleton would have fewer branches; the skeleton would represent only the gross structure of the region.

The key to obtaining a multiple scale representation for the skeleton is to determine which branches should be eliminated as the algorithm moves from fine to coarse scales. Furthermore, it is necessary to determine the appropriate position of the skeleton branches so that they accurately depict the structure of the region. Finally, it is desirable to modify the bounding contour of the region, simultaneously, so that each skeleton in the scale-space corresponds to a consistent bounding contour.

In this paper, we consider a robust method for computing the medial axis skeleton across a variety of scales. The scale-space is parametric with the complexity of the skeleton. The complexity measure is defined as the number of branches of the skeleton.

The complexity of the skeleton is related to the complexity of the bounding contour. The complexity of the contour is measured by the number of extrema of curvature contained in the contour[3]. As we shall see, there is a formal relationship between the complexity measure of the contour and that of the skeleton. Thus, minimizing the contour complexity is tantamount to minimizing the skeleton complexity.

A set of curves is computed to represent the bounding contour across a variety of complexity measures. The curves possessing larger complexity measures represent greater detail than curves with smaller measures. A medial axis skeleton is computed directly from each contour. The result is a set of skeletons that represent only the gross structure of the region at coarse scales (low complexity), but they represent more of the detail at fine scales (high complexity).

In Section 2, we discuss the concept of complexity in greater detail. In Section 3, we briefly consider an analytical representation for contours; the contour representation paradigm is essential for computation of the scale-space. In Section 4, we consider the computation of the medial axis skeleton directly from the bounding contour. In Section 5, we define a scale-space for the medial axis skeleton that is based on the complexity measure. In Section 6, we discuss the benefits of the complexity scale-space for the medial axis skeleton and compare the scale-space with the minimum description length approach.

2 Complexity

The complexity of an object may be viewed as the number of primitive components of the object. Similarly, the complexity of a representation of an object may be measured by the number of subparts contained within the representation. In this section we seek to formalize this notion of complexity for contours and the medial axis skeleton.

Hoffman and Richards[7] have proposed the use of *codons* to decompose a contour into salient parts. They observe that minima of curvature of a contour serve as natural break points of the curve. Therefore, the curve is broken into sections that are bounded by extrema of curvature. These sections are called codons. Pairs of codons typically correspond to subjective parts of the region bounded by the contour.

Given this insight, the number of codons contained in the contour is a reasonable measure of the number of subjective features of the region bounded by a contour. Therefore, the number of codons contained in the contour is a suitable measure of the complexity of the curve. Conveniently, the number of codons contained in the contour is equal to the number of extrema of curvature of a closed contour.

Similarly, the complexity measure of the medial axis skeleton is the number of branches of the skeleton. Each branch of the skeleton corresponds to a subregion of the region bounded by the contour. A region possessing a larger number of subregions is more complex than a region possessing a smaller number of subregions.

The complexity measure for contours is related to the complexity measure of the medial axis skeleton. Each branch of the skeleton that terminates into the contour, rather than into a node of the skeleton, does so at a positive maximum of curvature. Therefore, an upper bound for the number of branches in the skeleton may be obtained from the number of extrema of curvature of the bounding contour. If the number of extrema of a particular curve is M, then there are at most $M/2$ positive maxima of curvature. Therefore, there are no more than $M/2$ branches that terminate into the contour. Each of these terminal branches intersects another branch at a node and a third branch emanates from the node. This branch may or may not be a terminal branch. In the worst case, the number of non-terminal branches is $M/2-2$. Therefore, the complexity of the skeleton, B, is no larger than $M-2$.

A similar argument may be made for a region with holes. First, consider the skeleton associated with the bounding contour without holes. From above, the skeleton associated with the bounding contour has complexity $M-2$. Now add the holes one by one and consider the resulting skeleton. Each time a hole is added, the number of branches increases by no more than three. Thus, the maximum number of skeleton branches for a region with holes is M + 3H -2, where H is the number of holes.

More importantly, if the bounding contour is deformed continuously in such a way that the complexity measure decreases, the complexity measure of the corresponding skeleton almost always decreases. Equivalently, reducing the number of extrema of curvature of the bounding contour almost always causes the number of branches of the skeleton to decrease.

There is a tradeoff between the descriptive power of a representation and the complexity of the representation. If the complexity is allowed to be arbitrarily large, any set of data may be represented. On the other hand, limiting the complexity restricts the class of shapes and objects that may be represented. In Section 5, we exploit this tradeoff to define a scale-space for the medial axis skeleton that is similar to a complexity scale-space for contours. In the next two sections, we consider an analytical representation for contours and the computation of the medial axis skeleton directly from the representation.

3 Analytical Representation of the Bounding Contour

The complexity scale-space for the medial axis skeleton is based upon the complexity scale-space for contours[3]. In this section, we briefly consider an analytical representation for contours that makes computation of the scale-space possible. We define the contour representation and consider primitive operations for deforming a contour. A scale-space for contours based upon the complexity measure of the contour is also described.

Any reasonably well-behaved contour may be approximated by a list of pairwise tangent circular arcs. This representation provides the ability to represent the position, orientation, and curvature of the contour explicitly. The representation facilitates the computation of a variety of mathematical properties of the contour. In particular, the contour representation facilitates the computation of an analytical representation of the skeleton.

We assume that a list of data points representing the location of points along the bounding contour is available. The algorithm constructs an arbitrary, initial contour that passes through each point. The initial contour is transformed into a more desirable one by applying a set of deformations to the contour, as described below.

There are three local deformations of particular interest. The first operation is the deformation of the curvature of a single arc. The second operation is the rotation of two neighboring arcs. The third operation is the splitting of a single arc into two arcs. These operations are illustrated in Figure 1.

The deformation of the curvature of a single arc is accomplished under the constraint that its neighboring arcs remain fixed. This operation has one degree of freedom. As the radius of curvature of the arc changes, the center of curvature is constrained to move along a curve that is a conic section. This constraint is a result of the fact that the arc of interest must remain tangent to its neighbors. The deformation of the curvature of a single arc is illustrated in Figure 1a.

The rotation of two neighboring arcs is accomplished under the constraint that the neighbors of the two rotating arcs remain fixed. The radii of curvature of the arcs of interest also remain fixed; only the position of centers of curvature are modified. Because the arcs must remain tangent to their respective neighbors, the center of curvature of each of the arcs is constrained to lie on a circle whose center is coincident with the center of the respective neighbor. Furthermore, the positions of the two arcs of interest must be modified in such a way that they remain tangent to each other. The rotation of two neighboring arcs is illustrated in Figure 1b.

Splitting an arc into two is accomplished under the constraint that the two arcs must be tangent to each other and each of the arcs is tangent to one of the neighbors of the original arc. As stated, the deformation has three degrees of freedom. An additional constraint is imposed to reduce the complexity of the calculation. The point of tangency between the two new arcs is constrained to lie on a line specified by the algorithm. The choice of the constraint line is dependent on the context of the computation. Splitting an arc into two arcs is illustrated in Figure 1c.

The complexity of a contour, M, is defined as the number of extrema of curvature present on the contour. Subjective parts of a region are delimited by negative extrema of curvature on the bounding contour[7]. The number of such parts is limited by the number of extrema of curvature. Therefore, the number of extrema of curvature on a contour is related to the number of subjective parts of the interior of the contour.

A scale-space is constructed based on the complexity measure. At finer scales, the contour is depicted with a higher complexity measure. Thus, more detail is represented. At more coarse scales, the contour is depicted with a lower complexity measure. Fewer extrema of curvature are present in the contour. Consequently, less detail is present and only the gross structure of the contour is represented. At each scale, the contour is chosen to minimize the square-error under the constraint that it has the appropriate complexity measure.

The computation of the scale-space is performed in two stages. In the first stage, a curve with the minimum complexity measure is computed under the constraint that the curve passes within a specified tolerance, δ, of each data point. The value, δ, acts as the scale parameter. If δ increases, the complexity of the curve decreases and less detail is represented. In the second stage, the curve that is closest to the data in the square-error sense is computed under the constraint that the curve has the complexity measure found in the first stage.

In the first stage, deformations of the curve are chosen to minimize the difference in curvature between neighboring extrema. The curvature of each arc associated with a maximum is decreased until it is not possible to do so without moving the curve outside the tolerance of one of the data points. Similarly, the curvature of each arc associated with a minimum is increased. During the course of this operation, the number of extrema is reduced when neighboring maximum-minimum pairs are modified such that their respective curvature values are equal.

In the second stage, deformations of the curve are chosen to minimize the square-error between the curve and the data points. The computation proceeds under the constraint that the complexity is not changed by any of the deformations. During each iteration, all the arcs are modified locally to reduce the square-error. The algorithm iterates until it is no longer possible to reduce the error.

The result of the two-stage computation is the minimum complexity, least square-error contour. The contour has the minimum complexity possible under the constraint that the curve lies within δ of each data point. The curve has the least square-error of any curve that has the same complexity measure.

A multiple scale representation of the contour is achieved by computing contours with a variety of complexity measures. This is accomplished by varying the scale parameter, δ. At larger values of δ, the curve has a lower complexity measure and only the gross structure of the contour is represented. At smaller values of δ, the curve has a higher complexity measure and more of the details are represented. The silhouette of an airplane at two scales is depicted in Figure 2.

The analytical representation paradigm provides a robust method for describing a contour and its mathematical properties. In particular, the curvature of the contour is represented explicitly. This facilitates the computation of a novel scale-space for contours. Furthermore, the paradigm facilitates the computation of a novel scale-space for the medial axis skeleton. The contour representation paradigm is described in more detail in an earlier paper[3].

4 Computation of the Medial Axis Skeleton

The medial axis skeleton may be computed directly from the analytical contour representation. In this section, we consider the mechanics of the computation. First, we consider a number of useful general properties of the skeleton and its bounding contour. Next, we consider properties of the skeleton when the contour is made up of pairwise tangent circular arcs. Finally, we consider the computation of the skeleton from the analytical contour representation.

The medial axis skeleton is usually defined as the locus of points where wavefronts propagating inward from the bounding contour meet (see, for example, [9]). The skeleton points are locations where two or more wavefronts have propagated the same distance from their respective starting locations. This definition suggests morphological operators that approximate the propagation of the wavefront.

The medial axis skeleton may also be defined by the following properties: Each point on the skeleton is equidistant from two or more points on the bounding contour. There are no points on the boundary closer to the skeleton point than these equidistant points. And, each skeleton point lies in the interior of the bounding contour.

This alternate definition is mathematically equivalent to the wavefront definition. The distance from each skeleton point to the closest points on the contour is the distance traveled by the associated wavefronts. As we shall see, the alternate definition is constructive; it leads to a novel method of computing the skeleton.

Each point that is on a branch of a skeleton, but not a node, is equidistant from exactly two points on the contour. Each node point is equidistant from three or more points on the bounding contour. Typically, a node is equidistant from exactly three boundary points. The case where the node is equidistant from more than three points is a zero measure condition.

For each point on the branch of the skeleton there is a circle that is tangent to the contour in two places. The center of the circle is coincident with the point on the branch. The radius of the circle is the distance from the center to the two nearest points on the contour. Aside from the two tangent points, the circle does not contact the contour. We call such a circle the *interior circle* of the point of interest. We call the two points of tangency between the interior circle and the contour the *tangent points* of the interior circle.

Similarly, for each node point, there is a circle that is tangent to three (or more) points on the contour. The center of the circle is coincident with the node point and the radius is the distance from the node to the three nearest points on the contour. Aside from these tangent points, the circle does not contact the contour. Such a circle is called the *interior circle* of the node.

As a branch of the skeleton is traversed, the radius of the interior circle varies continuously. Stated another way, the distance from a skeleton branch to the contour varies continuously along the branch. Furthermore, as the branch is traversed, each tangent point of the interior circle moves continuously along the contour. In the case where the branch terminates into the contour, rather than into a node, the two tangent points converge with the branch at a point of maximum curvature on the contour.

Each branch of the skeleton that terminates into the contour does so at a positive maximum of curvature. It is not the case, however, that each positive maximum of curvature is associated with a branch termination. There is a simple test to determine if a positive maximum is associated with the termination of a branch. If the osculating circle associated with the curvature maximum lies completely in the interior of the contour or on the contour, there is a branch that terminates at the maximum. Otherwise, there is no terminus. The terminus test is illustrated in Figure 3.

Now, consider a contour that consists of pairwise tangent circular arcs, as described in Section 3. At any point on the skeleton, the tangent points lie on two particular arcs of the contour representation. Locally, the points on the skeleton branch are equidistant from these two arcs. A locus of points that is equidistant from two circles is a conic section. Therefore, the medial axis skeleton consists of segments of curves that are conic sections. We call such a curve segment a *conic segment* of the skeleton branch. The analytical contour representation leads directly to an analytic representation for the medial axis skeleton.

Because the conic segments of the skeleton are well characterized, it is convenient to compute the skeleton in a piecewise fashion. The key to computing this representation is finding the end points of the branch segments. At an end point of a branch segment, one of the tangent points of the interior circle is guaranteed to be coincident with the point of tangency of two neighboring contour arcs. Therefore, the segment end point must lie on the line determined by the radius of the circle corresponding to the end angle of the arc.

Assume that in some intermediate stage of the computation, a branch segment end point has been found. The arcs associated with the next branch segment are called arc1 and arc2, arbitrarily. The line determined by the center of arc1 and the end point of arc1 is called line1. The line determined by the center of arc2 and the end point of arc2 is called line2. Assume, without loss of generality, that the branch segment is hyperbolic. There are two possibilities for the location of the next segment end point. The end point may coincide with the intersection of the hyperbola and line1 (candidate point1). Or, the end point may coincide with the intersection of the hyperbola and line2 (candidate point2). The appropriate choice of the two candidate points is the one closest along the hyperbola to the known end point. Note that the same reasoning would also apply to a branch segment that is an ellipse or a parabola. This geometric situation is illustrated in Figure 4.

The choice is made between candidate point1 and candidate point2 by determining which point is closer to the previous branch segment end point along the conic curve. Conveniently, there is a simple computational test to determine the appropriate point. If candidate point1 is within the sector of arc2, point1 is the appropriate choice. Similarly, if candidate point2 is within the sector of arc1, point2 is the appropriate choice. The case that both of these conditions are true is zero-measure. Furthermore, in that case candidate point1 and candidate point2 are coincident.

Once the end points of the segments have been determined, it is possible to characterize the segment between the end points. The branch segment is known to be a conic section. It is possible to determine the type of the conic section (hyperbola, ellipse, or parabola) by considering the relationship of the associated contour arcs and their respective curvatures. The foci of the conic section are coincident with the centers of the contour arcs. Because the end points of the branch segment lie on the conic section, they provide the remaining information necessary to construct the segment analytically.

Each branch of the skeleton is computed in a piecewise fashion as described above. Each segment of the branch corresponds to two arcs on the curve; the tangent points associated with each point in the branch segment lie on these two arcs. Two neighboring segments always share one arc; the other arcs associated with the two neighboring segments are neighbors on the contour. In the example shown in Figure 4, the segment of interest is associated with arc1 and arc2. The neighbor of this segment is associated with arc1 and the neighbor of arc2.

In effect, as the branch is traversed during computation, the tangent points on the contour are implicitly traversed as well. The tangent points associated with each point on a branch segment are easily computed. One of the tangent points is simply the projection of the branch point onto one of the arcs associated with the segment. The other tangent point is the projection of the branch point onto the other arc.

Each point on the contour is associated with exactly one point on the skeleton. It is not possible for two distinct skeleton points to have the same tangent point. This property of the skeleton is useful for determining the location of node points on the skeleton, as we shall see.

The skeleton computation begins by determining starting points for candidate skeleton branches. Because it is known that branches terminate into the bounding contour at positive maxima of curvature, these locations are chosen for the starting points. As these candidate branches are extended, the locations of intersections of the branches are found. At the intersections of two branches, a candidate node is formed and an additional candidate branch is created that emanates from the node. During the computation, some of the candidate branches are eliminated when it is determined that no branch exists at its location.

At each positive maximum of curvature on the bounding contour, a candidate skeleton branch is created. By convention, the initial branch segment is the bisecting radius of the arc associated with the maximum of curvature. Strictly speaking, such a segment is not part of the medial axis skeleton as defined mathematically. However, these segments are included by convention because doing so yields more intuitively pleasing results.

Each segment is extended in a piecewise fashion as described above. As the computation proceeds, the algorithm must determine the locations where the branches intersect to form nodes. Each time a branch is extended, the algorithm determines if the branch is overextended relative to another branch. In addition, the algorithm must determine if the other branches are overextended relative to the branch of interest. Ultimately, the algorithm must determine the locations of intersections of branches, that is, the nodes of the skeleton. The following set of conventions achieve these goals.

After a branch has been extended by a single segment, the algorithm determines what interaction, if any, occurs between the branch and the other branches. Conceptually, the algorithm determines if the tangent points of the branch have crossed any of the tangent points associated with any other branch. In practice, the algorithm considers only those branches that have arcs of the contour in common with the branch of interest. More specifically, the algorithm only considers branches whose tail segments (i.e. end segments) have an arc in common with the tail segment of the branch of interest.

If the tail segments of two branches have a contour arc in common, the systematic application of a simple test determines the interaction between the branches. The test determines if either or both of the branches have been extended beyond the intersection between branches. Furthermore, these tests are used to find the node point which is located at the intersection of two segments. This test, described below, is illustrated in Figure 5.

By convention, the arc associated with both of the tail segments is called the common arc. We arbitrarily refer to one of the branches as branch1 and the other as branch2. Similarly, segment1 and segment2 are the current tail segments of the respective branches. Arc1 and arc2 are the arcs associated with segment1 and segment2 that are not the common arc.

The intersection test determines if an arbitrary point on candidate segment1 is beyond the intersection of segment1 and segment2. The interior circle associated with the point is constructed. By definition the interior circle is tangent to arc1 and the common arc; the center is located at the point of interest on segment1. If the distance from the point to arc2 is greater than the radius of the interior circle, the point of interest is beyond the intersection point. Conversely, if the distance from the point of interest to arc2 is less than the radius of the interior circle, the point is not beyond the intersection of segment1 and segment2. Of course, if the distance from the point of interest to arc2 is equal to the radius of the interior circle, the point is the intersection point.

The intersection test is illustrated in Figure 5. In Figure 5a, point p_2 is beyond the intersection because the interior circle intersects arc2. In Figure 5b, point p_1 is not beyond the intersection because the interior circle does not contact arc2. In Figure 5c, point p_3 is the intersection of the two segments; the interior circle is tangent to all three contour arcs.

This test may be used to determine if two tail segments with a common arc intersect. The test is applied to both endpoints for each segment. The segments intersect if and only if the intersection test provides opposite answers for each endpoint of both segments. That is, one endpoint of segment1 is beyond the intersection and the other endpoint is not. Similarly, one endpoint of segment2 is beyond the intersection and the other is not.

When two segments intersect, the node point may be found using the intersection test recursively. Initially, the intersection is known to be between the two original endpoints of segment1. We arbitrarily call these points the upper and lower bound points of the node. The intersection test is applied to a point midway between the bound points. If the midpoint is beyond the intersection, the midpoint becomes the new upper bound. Conversely, if the midpoint is not beyond the intersection, the midpoint becomes the new lower bound. This process is repeated until bound points converge to the node.

It is also possible during the computation that the entire tail segment of a particular branch has been extended beyond the intersection of the branch (branch1) with another branch (branch2). Again, the intersection test is used to distinguish this situation. The test is applied to both endpoints of the tail segment of branch1. If the test determines that both endpoints are beyond the intersection, the entire tail segment is beyond the intersection. In that case, the tail segment is removed from the branch representation.

Note that extending a particular branch is a local operation. It is not necessary to consider the entire bounding contour to perform the computation. In fact, only two arcs of the contour are required for each step. This suggests that the branches could be computed independently, in parallel. Of course, if a branch is extended such that its tail segment has an arc in common with another tail segment, the branches must interact in the manner described above. That is, they must determine if either of the branches is overextended and if the branches intersect at a node.

In our discussion, we have tacitly assumed that the region is bounded by a single simply connected curve. That is, we have assumed that there are no holes in the region. It is straightforward to generalize the algorithm to handle regions with holes. To do so it is necessary to find initial branches such that each segment has one contour arc on an interior curve. (An interior curve is the bounding curve of a hole.) The algorithm extends these initial branches to find nodes similar to the extension of branches described above.

The initial branches associated with the interior curves are found in the following manner: The point on the interior curve that is closest to the bounding curve is determined. Simultaneously, the point on the bounding contour that is closest to the interior curve is found. The midpoint of these points lies on the initial candidate segment for the initial branch. An interior circle is constructed such that the center is the midpoint and the radius is the distance between the midpoint and either of the contour points. The midpoint lies on the skeleton if and only if this interior circle does not intersect another of the interior curves.

In the case that the interior circle does intersect another interior curve, an alternate starting point must be found. The alternate starting point is determined in the same manner described above, except that the closest points between the two interior curves are found. The midpoint between these two points is the new candidate skeleton point. The new candidate point lies on the skeleton if and only if no other interior curve intersects the interior circle. In the event that the interior circle does intersect another interior curve, the process is repeated until an appropriate skeleton point is found. This procedure is guaranteed to provide a skeleton point that is associated with each of the holes.

Once a skeleton point has been found for each hole, a branch segment is constructed that extends in both directions from each of the initial skeleton points. This segment serves as the starting segment for an initial branch. Each of these branches is extended in both directions to find the appropriate nodes with other branches. Figure 6 illustrates the initialization of a skeleton corresponding to a region with holes.

Given the piecewise conic description of the skeleton, it is possible to reconstruct the bounding contour exactly. For each segment, it is possible to reconstruct the two contour arcs associated with the segment. If the segment is hyperbolic or elliptic, the centers of the arcs are determined by computing the locations of the foci of the hyperbola or ellipse. If the segment is parabolic, the focus of the parabola is the center point of one arc; the other arc is a straight line segment. The radii of the two contour arcs may be determined by considering the radii of any two interior circles along the conic segment. The endpoints of the contour arcs are determined by projecting the endpoints of the segment onto each arc.

The medial axis skeleton may be computed directly from the analytical contour representation. The contour representation leads naturally to the analytic representation of the skeleton. Each branch of the skeleton is piecewise elliptic, hyperbolic, or parabolic. The bounding contour may be reconstructed exactly from the skeleton.

5 The Medial Axis Skeleton Complexity Scale-Space

In an earlier paper[3], we considered a complexity scale-space for contours. In this section, we extend the concept to define a complexity scale-space for the medial axis skeleton. We consider the computation of the skeleton scale-space from the contour scale-space.

A scale-space is a set of descriptions that differ in their level of detail. At coarse scales the descriptions are relatively simple and, presumably, contain only the most important aspects of the description. At fine scales, the descriptions are relatively complicated and contain the details.

A scale-space representation is desirable because it provides alternative descriptions for subsequent processing. If a higher level process requires accuracy and dense information, a fine scale is appropriate. However, if accuracy is not as critical and sparse information is sufficient, a coarse scale is appropriate. In the latter case, the computational burden is often significantly reduced because the algorithm is required to process a smaller quantity of data.

The construction of a scale-space requires a tradeoff between the accuracy and the level of detail in the description. At fine scales, the tradeoff is skewed toward the accuracy of the description. At coarse scales the tradeoff is skewed toward the simplicity of the description.

The measure of this tradeoff is typically a smoothing parameter such as the spatial width of a Gaussian filter applied to the data (see, for example, [12]). In such a case, an increased spatial width of the filter reduces some of the existing detail. The description is simplified, but the ability to localize the remaining components of the description (the accuracy) is reduced.

In this paper, we propose an novel method of quantifying the accuracy versus simplicity tradeoff. The tradeoff yields a set of descriptions with varying complexity measures, as defined in Section 2. Each description is chosen such that it is as close as possible (in the square-error sense) to the data under the constraint that it has a particular complexity measure.

In the case of a contour, we assume that a set of data points along the contour has been provided. A set of contours is constructed such that each contour has a different complexity. That is, each contour has a different number of extrema of curvature. Each of the contours is chosen such that it minimizes the square-error between data points and the contour under the constraint that the complexity measure is equal to a particular value.

In the case of the medial axis skeleton, we also assume that a set of data points along the bounding contour has been provided. Again, a set of skeletons is constructed such that the skeletons have different complexity measures. The bounding contour is chosen such that the square-error between the data and the contour is minimized under the constraint that the associated skeleton possesses the appropriate complexity measure. As we shall see, the contour scale-space is very similar to the skeleton scale-space.

Now, consider the computation of the contour scale-space. If the contour is constrained to pass through each data point exactly, there is a particular minimum complexity measure, M_0. It is not possible to construct a curve that passes through every data point and has a complexity measure smaller than M_0. If the constraint is relaxed such that the curve must pass within some tolerance, δ_1, of each data point, another minimum complexity measure, M_1, is obtained. Of course, $M_1 \leq M_0$. Therefore, as the tolerance, δ, increases, the associated minimum complexity measure, M, decreases. Thus, the tolerance, δ, acts as a scale parameter for the complexity scale-space.

For any tolerance, δ_i, there are infinitely many contours that meet the tolerance requirement and possess the minimum complexity measure, M_i. It is desirable to choose the curve that minimizes the square-error between the data and the contour from the class of minimum complexity curves. This suggests a two-stage algorithm for determining the desired minimum complexity/least square-error curve.

In the first stage, an instance of the minimum complexity contour is computed under the constraint that the curve passes within δ_i of each data point. The curve found in the first stage is used as the starting point for the second computational stage. The output of the first computational stage is illustrated in Figure 7 with the silhouette of an airplane as test case.

In the second computational stage, the minimum complexity contour is modified such that square-error between curve and the data points is minimized. The modification is performed under the constraint that the complexity measure does not change. The result is the minimum complexity/least square-error curve. The final results for the airplane silhouette are depicted in Figure 8. The computation of the minimum complexity/least square-error curve is described briefly in Section 3 and more fully in an earlier paper[3].

As the tolerance parameter increases from scale to scale, contours with smaller complexity measures are found. Some of the details in the contour are eliminated in the process; the more coarse representation is simpler. The features that are present in the representation are depicted as accurately as possible. However, the square-error is guaranteed to increase because the contour is unable to account for all of the detail in the data. Thus, as the contour becomes less complex, the accuracy of the representation decreases.

In the case of the airplane silhouette, the position of the tip of each wing is accurately depicted across the entire scale-space. In contrast, at coarse scales, the protrusions on the back of the wings disappear because they are smaller than the scale-parameter. The representation is simpler, because only the gross structure of the wing is depicted. However, the error between the data and the contour is significantly greater in the proximity of the protrusions. This is an example of the tradeoff between simplicity and accuracy.

It would be reasonable to base the skeleton scale-space directly on the contour scale-space. One option is to compute the contour at multiple complexities. For each contour, the corresponding skeleton is computed. As the complexity of the contour decreases, the complexity of the skeleton is guaranteed to decrease. This yields a reasonable scale-space for the medial axis skeleton.

However, there is at least one case where the above definition of the skeleton scale-space leads to a counter-intuitive result. This definition does not penalize the magnitude of the curvature; only the number of extrema of curvature is considered. Thus, the algorithm prefers a curve with relatively large curvature if such a choice reduces the square-error, even if the reduction is small. In some cases, this results in an additional skeleton branch that terminates into the curvature maximum. This phenomenon is illustrated in Figure 9.

Fortunately, a minor change in the contour smoothing criteria eliminates this effect. The first stage of the contour smoothing algorithm is identical; the curve is modified to minimize the number of curvature extrema. In the second stage, an additional constraint is placed on the computation. No deformations of the curve are allowed that would introduce an additional branch terminus. The complexity of the skeleton is preserved in the second stage, as well as the complexity of the contour.

The change in the smoothing criteria has only a small effect on the flow of computation. In fact, the computation need only be altered for arcs corresponding to positive maxima of curvature. The computation for all other arcs is identical to that for the original criteria.

At each maximum of curvature that is not associated with a branch termination, it is necessary to ensure that the osculating circle extends to the exterior of the contour. Recall that whenever the osculating circle does not extend to the exterior of the contour, a branch terminus is guaranteed to occur at the maximum. Furthermore, whenever the osculating circle extends to the exterior of the contour, no branch terminus occurs at the maximum.

The most obvious method to avoid the introduction of additional skeleton branches is to disallow deformations of the curve that would create a spurious branch. However, this method places an unnecessary burden on the computation. It would be necessary to test every candidate deformation at each iteration of the smoothing process against this condition.

A more efficient method is to compute the least square-error curve as before, then modify that curve to eliminate any spurious branches that have been introduced. This method requires that the locations of the branch termini are determined after the first stage of the smoothing algorithm. After the second stage of the contour smoothing algorithm, each maximum of curvature is tested to determine if there is a branch present. If a branch has been introduced during the second stage of computation, the curve must be deformed locally to eliminate the spurious branch.

To eliminate a spurious branch, it is necessary to decrease the curvature of the maximum curvature arc. The arc is deformed under the constraint that the neighbors remain fixed, as described in Section 3. The arc is modified until its osculating circle contacts another portion of the curve; the branch is eliminated. We call the additional point of contact of the osculating circle with the contour the *alternate contact point.*

Once the branch is eliminated, it is desirable to modify the curve locally to reduce the square-error. Such modifications are carried out under the constraint that the osculating circle does not lose contact with the contour at the alternate contact point. This computation may be accomplished locally in the sense that operations need information only about arcs that are near the maximum and the arc associated with the alternate contact point. No other arcs of the contour need be considered.

The result of these computations is a scale-space representation for the medial axis skeleton. At a relatively coarse scale, a skeleton that represents only the gross structure of the interior of the bounding contour is provided. At a relatively fine scale, a skeleton that represents more of the details is provided. The scale-space is illustrated in Figure 10 for the silhouette of an airplane.

In the case of the airplane silhouette, the gross structure of the airplane is depicted across the entire scale-space. At the coarse scales, the protrusions of the wings are not present in the skeleton. However, the gross structure of the wings is accurately represented. At the finest scale, the protrusions are present and accurately depicted.

Often, the contours obtained from the contour scale-space are identical with the bounding contours from the skeleton scale-space. However, there are cases where a contour may be modified such that an additional branch appears (or disappears) in the skeleton without changing the number of extrema of curvature in the curve. In these cases, the bounding contour of the skeleton scale-space differs slightly from that of the contour scale-space.

For each type of scale-space, a set of descriptions with varying levels of detail is obtained. The tradeoff between complexity and proximity to the data is quantified in terms of a tolerance about each data point. Each representation is as accurate as possible in the square-error sense under the constraint of minimum complexity. The contour scale-space and the skeleton scale-space are similar; the difference lies in a slight inconsistency of the complexity measures.

6 Discussion

Most "scale-space" representations would be more accurately described by the term *resolution-space*. Typically, such resolution-spaces are parameterized by the spatial width of some filter, (usually a Gaussian filter, for example [12]). Subjective features are eliminated from the representation as the data is blurred. For example, at a fine scale a particular feature may be represented accurately. At a coarse scale, the feature may be eliminated, as desired. However, at an intermediate scale, the feature may exist with degraded accuracy. There is no advantage to representing particular atomic features with varying degrees of accuracy.

A true scale-space provides descriptions of the data with varying levels of detail. At each scale, all features are represented as accurately as possible. Given that a feature is present at two scales, there is no advantage in reducing the accuracy of localization of the feature from one scale to the next. It is more desirable to retain the accuracy of the representation from one scale to the next, until a feature is eliminated altogether.

The complexity measure yields a true scale-space for the medial axis skeleton. At finer scales, greater detail is represented. At coarse scales, only the gross structure of the skeleton is represented. The complexity criterion explicitly guarantees that the skeleton becomes simpler at coarse scales.

The medial axis skeleton complexity scale-space offers a novel approach to the trade-off of accuracy and simplicity. The tradeoff occurs between the number of subjective features and the square-error between the data and the representation. Whenever a feature is present in the representation, it is depicted as accurately as possible. For example, the location of the tip either wing of the airplane depicted in Figure 8 is represented accurately across the entire scale-space. Of course, there is still a reduction in the overall accuracy of the representation at coarse scales because, for example, the protrusions on the back side of each wing are not depicted at the coarse scales.

In general, it is more difficult to obtain a true scale-space representation than a resolution-space. It is nontrivial to obtain a formal tradeoff between the simplicity of a description and the accuracy. Whenever possible, it is desirable obtain a true scale-space representation. Of course, resolution-spaces are useful to the extent that they approximate the desired behavior of a scale-space.

Furthermore, at each scale, the branches that are present are represented as accurately as possible. That is, each branch corresponds to a portion of the curve that is as close as possible to the data in the square-error sense. The accuracy of the representation of each branch is not diminished from one scale to the next. For example, the ability to localize the tip of either airplane wing in Figure 10 is not diminished at any scale.

Another advantage of the complexity scale-space for the medial axis skeleton is that the skeleton is consistent with the contour representation. The mapping between the contour and the skeleton is unique and invertible. In contrast, most methods of computing the medial axis skeleton yield a result that is inconsistent with the bounding contour; construction of a curve from the skeleton would lead to a curve that differs from the original bounding contour.

Consistency among representations is desirable because higher level processes may make inferences based upon properties of the contour or the skeleton. If the representations are consistent, such a higher level process is less likely to make incompatible inferences about the data. Such incompatible inferences would lead to degradation of the overall system performance.

The complexity scale-space is similar to the minimum description length (MDL) approach at an intuitive level. The idea of providing the simplest possible representation is present in both approaches. However, the formal definition of complexity is different in the two approaches. As a result, there are significant conceptual differences between the complexity scale-space and the MDL paradigm.

The MDL criterion requires that the number of parameters employed by a model to account for the data should be minimized[11]. More precisely, the number of bits required to encode the data is minimized. Thus, an important component of the MDL approach is the choice of an efficient model for the representation. Formally, the MDL approach requires that the representation correspond to an optimal code in the information theoretic sense. A theory for choosing the representation from *a priori* probability distributions is well known (see Leclerc[8], for example).

Unfortunately, the determination of such an optimal code for a general application is difficult in practice. Typically, minimum description length approaches assume a particular form of the representation. The number of parameters employed by the representation is minimized. Interestingly, the MDL theory provides a simple objective measure of the performance of a particular representation. The performance measure is, of course, the average length of the code.

For example, Leclerc[8] applies the MDL technique to the image partitioning problem. Each subregion requires a set of parameters to distinguish the boundary and to specify the behavior of image brightness within the subregion. The additional parameters required for each subregion causes the algorithm to favor a small number of partitions. However, when there are relatively few partitions, it is more difficult to account for the variations from the model within each subregions; a larger number of bits per subregion is required. Conversely, this effect favors a larger number of partitions. There is a particular choice of the partitioning that optimizes these two opposing effects.

In contrast, the complexity scale-space seeks to minimize the number of subjective features in the description. The optimization is indifferent to the number parameters used by the representation. In the case of a contour, the contour may be described by arbitrarily many circular arcs; the complexity depends only on the number of extrema of curvature. Similarly, each branch of a medial axis skeleton may be represented by arbitrarily many branch segments; the complexity is specified by the number of branches. An MDL approach would essentially require minimizing the number of arcs in a contour or the number of segments in a skeleton.

Furthermore, the complexity scale-space does not attempt to provide an efficient code for the representation. Rather, the scale-space seeks to make the most relevant information explicit for higher level processes. In fact, there is a loss of information at the coarse scales. We have argued, above, that this loss of information is desirable for computational purposes.

Another important distinction is that MDL approaches typically do not provide scale-space representations. MDL approaches provide a single representation for each data set. In effect, an optimal scale is chosen implicitly by the MDL criterion. The scale chosen by the MDL criterion is the one requires the fewest number of bits to represent the data.

The paradigm described in this paper leads to true scale-spaces for contours and the medial axis skeleton. At first glance, the complexity scale-spaces are very similar to minimum description length approaches; the notion of simplicity of the representation is exploited in each case. However, there are significant philosophical and practical differences between the complexity scale-spaces and MDL approaches.

7 Conclusion

We have considered a novel approach to computing the medial axis skeleton across a variety of scales. Complexity, as defined by the number of branches of the skeleton, is a natural measure of the level of detail of the skeleton description. The multiple complexity paradigm leads to a true scale-space; the ability to localize a feature is not diminished from one scale to another unless the feature is completely eliminated. The representation for the skeleton is consistent with the corresponding representation of the bounding contour; the mapping between the skeleton and the bounding contour is unique and invertible.

References

[1] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[2] H. Blum. A transformation for extracting new descriptions of shape. In *Symposium on Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, MA, 1964.

[3] R. D. Chaney. Analytical representation of contours. AI Memo 1392, MIT Artificial Intelligence Laboratory, October 1992.

[4] R. T. Chin, H. K. Wan, D. L. Stover, and R. D. Iversion. A one-pass thinning algorithm and its parallel implementation. *Computer Vision, Graphics, and Image Processing*, 40(1):30–40, 1987.

[5] A. R. Dill, M. D. Levine, and P. B. Nobel. Multiple resolution skeletons. *Computer Vision, Graphics, and Image Processing*, 40(1):30–40, 1987.

[6] S. B. Ho and C. R. Dyer. Comparison of thinning algorithms on a parallel processor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):512–520, 1986.

[7] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, pages 65–96, 1984. Look this up.

[8] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, May 1989.

[9] 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, Jan. 1992.

[10] M. P. Martínez-Peréz, J. Jiménez, and J. Navalón. A thining algorithm based on contours. *Computer Vision, Graphics, and Image Processing*, 39(2):186–201, August 1987.

[11] J. Rissan. Minimum-description-length principle. In *Encyclopedia of Statistical Sciences*, volume 5, pages 523–527. Wiley, New York, 1987.

[12] A. P. Witkin. Scale space filtering. In *Proc. 7th International Conference on Artificial Intelligence*, pages 1019–1021, Karlsruhe, 1983.

[13] Y. Xia. Skeletonization via the realization of the fire front's propagation and extinction in binary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1076–1086, October 1989.

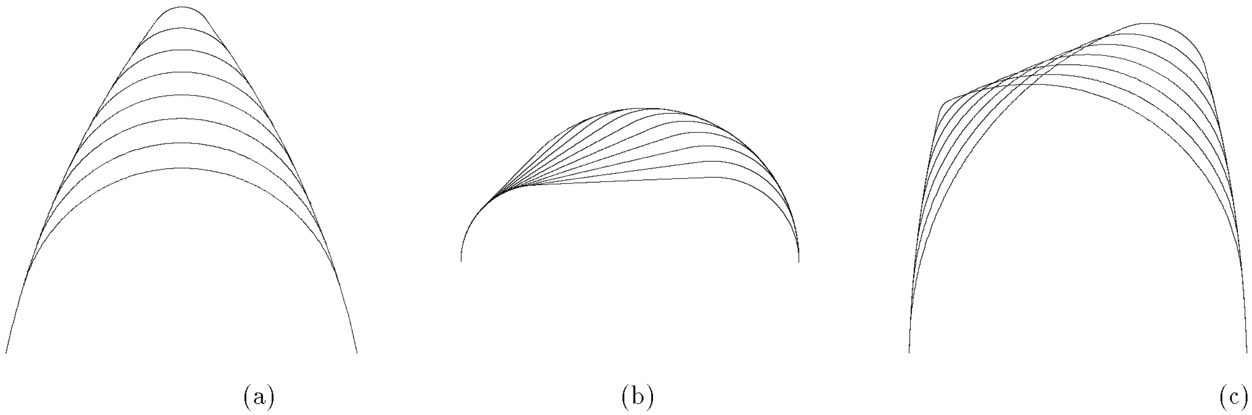

Figure 1: Local deformations of an analytic contour. In (a), the curvature of a single arc is modified under the constraint that it remains tangent to its neighbors. In (b), two neighboring arcs are rotated under the constraint that they remain tangent to their neighbors and to each other. In (c), an arc is split into two arcs. The two new arcs are tangent to each other at a point along a specified constraint line.

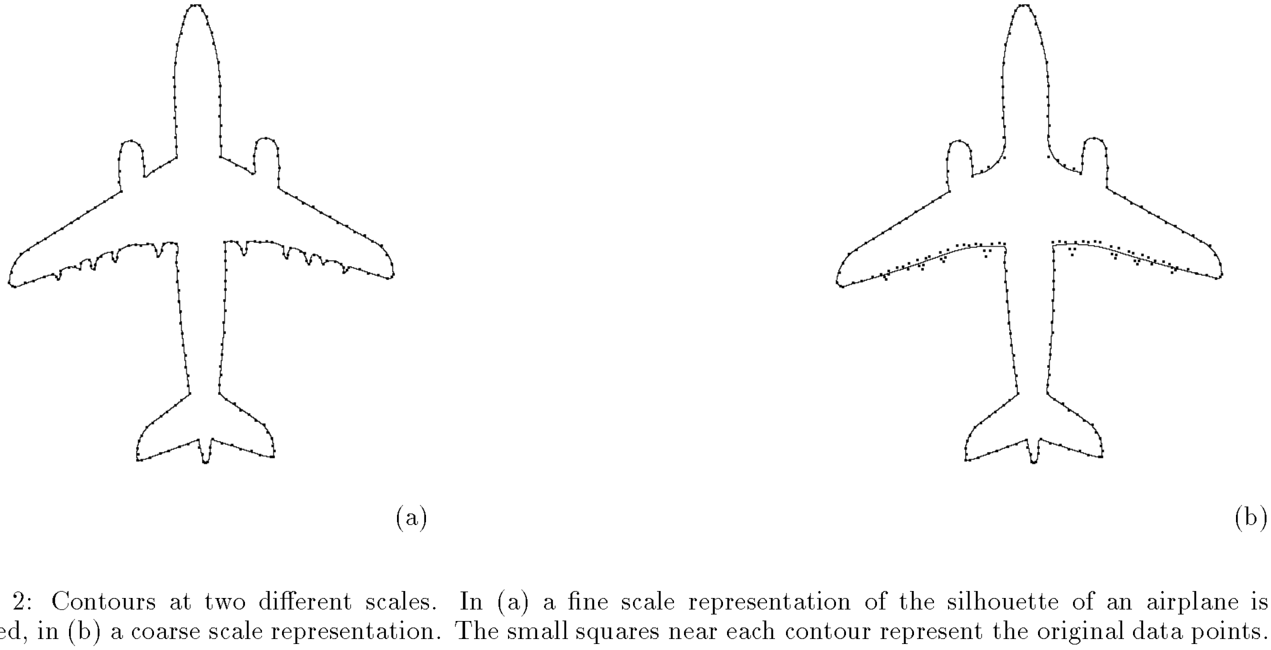

Figure 2: Contours at two different scales. In (a) a fine scale representation of the silhouette of an airplane is depicted, in (b) a coarse scale representation. The small squares near each contour represent the original data points.

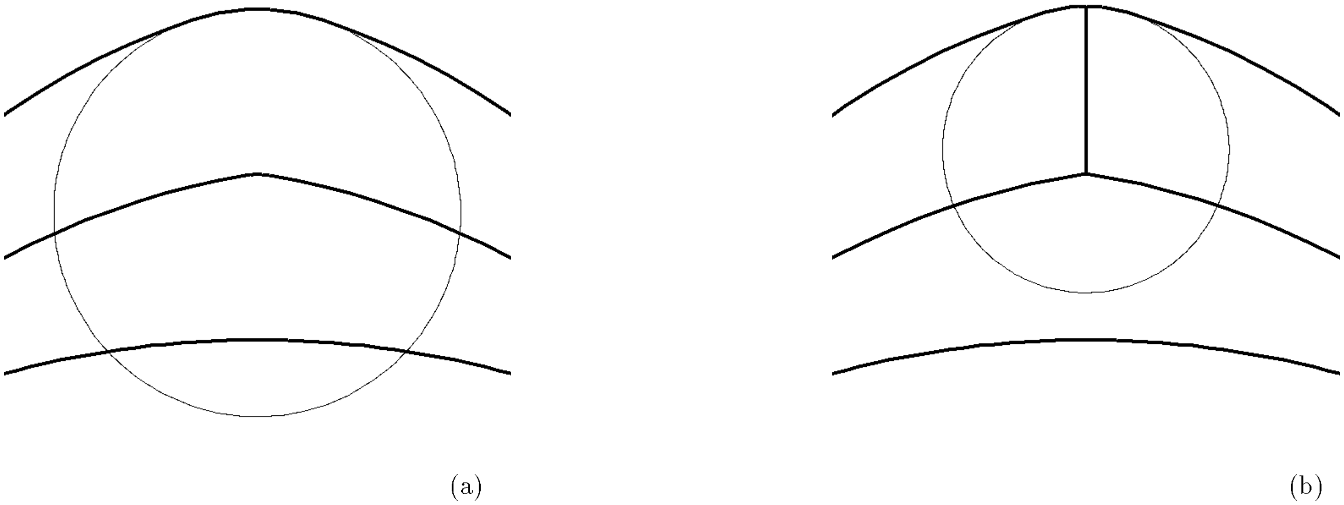

Figure 3: Test for the existence of branch terminating into a maximum of curvature. Each illustration depicts a portion of a contour and its medial axis skeleton; the osculating circle for the maximum of curvature at the top of the curve is also shown. In (a), no branch terminates into maximum of curvature because the osculating circle extends to the exterior of the curve. In (b), a branch does terminate into the maximum of curvature because the osculating circle remains in the interior of the contour.

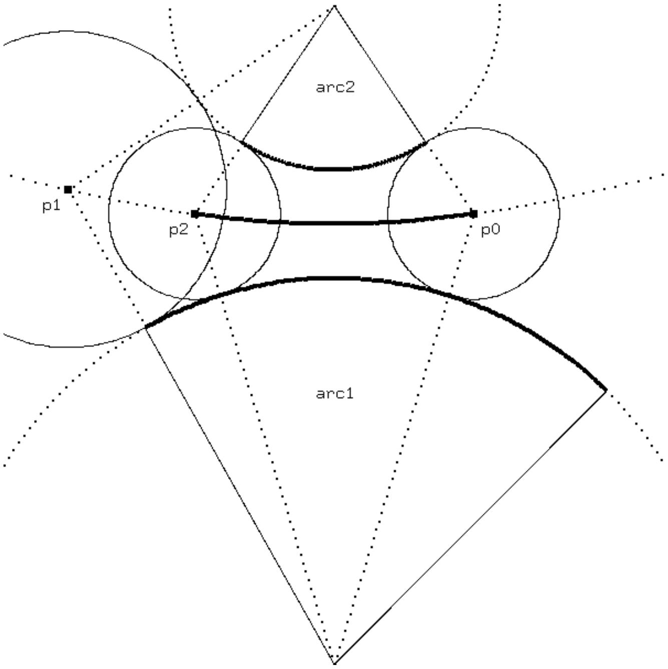

Figure 4: Computation of a branch segment. The branch segment is the hyperbolic curve connecting point p0 to point p2. The hyperbola is defined by the property that each point is equidistant from arc1 and arc2 (two arcs in the contour representation). The candidate end point, p1, is the intersection of the end radius of arc1 and the hyperbola. Similarly, the candidate end point, p2, is the intersection of the end radius of arc1 and the hyperbola. The point, p2, is chosen because it is within the sector of arc1 - the point, p1, is not in the sector of arc2. The next segment that would be computed extends beyond point, p2, and is determined by arc1 and the neighbor of arc2.

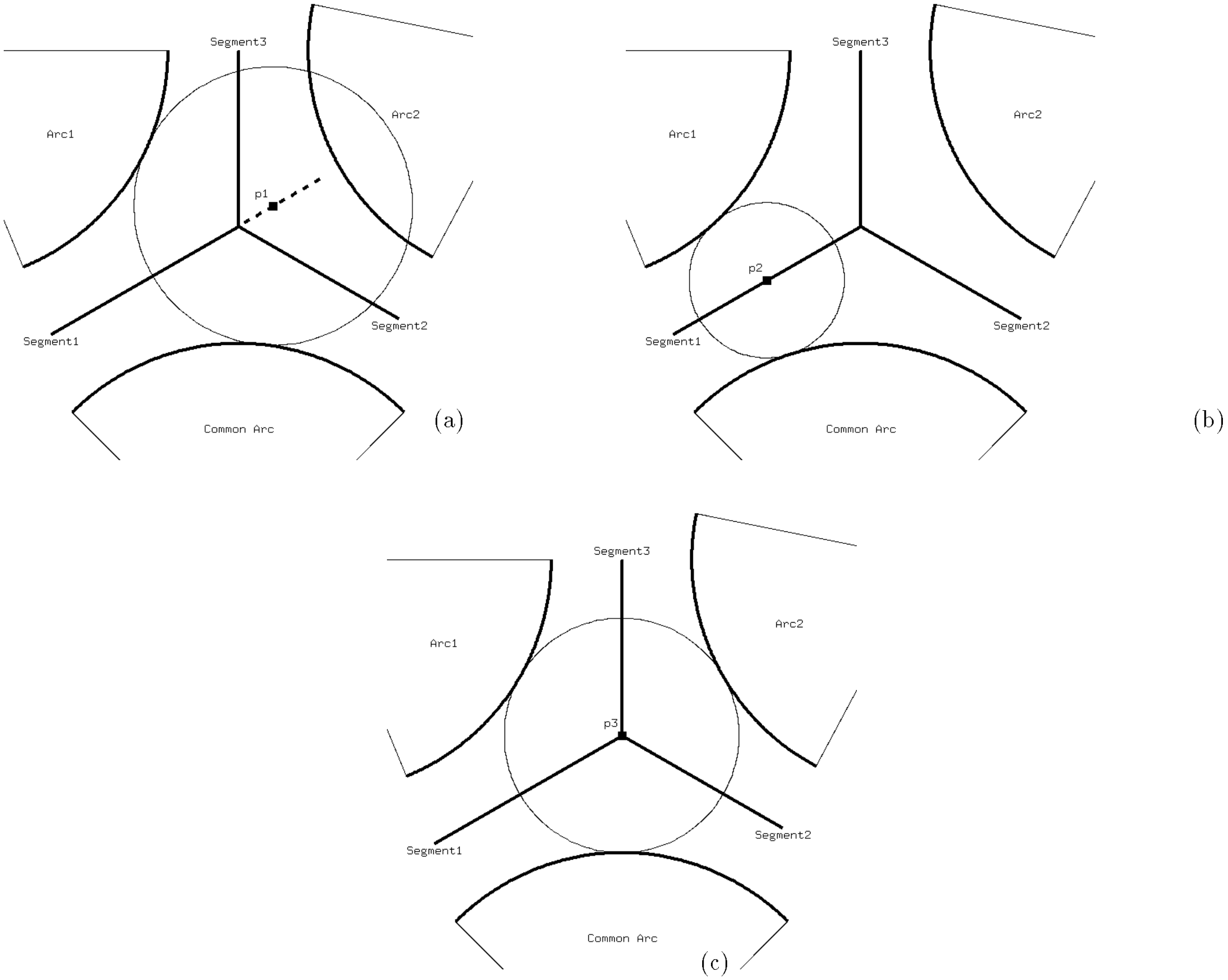

Figure 5: Test for determining the location of node points. In each figure, three segments that intersect at a node are depicted. The contour arcs associated with these segments are also depicted. Segment1 is equidistant from the common arc and arc1; segment2 is equidistant from the common arc and arc2; and segment3 is equidistant from arc1 and arc2. Conceptually, segment1 and segment2 intersect to form a node; segment3 emanates from this node. In (a) point p_2 is beyond the intersection of segment1 and segment2: the interior circle is tangent to the common arc and the interior circle intersects arc2. In (b), point p_1 on segment1: the interior circle is tangent to the common arc and arc1, and the interior circle does not intersect arc2. In (c), point p_3 is the intersection between segment1 and segment2: the interior circle is tangent to all three contour arcs.

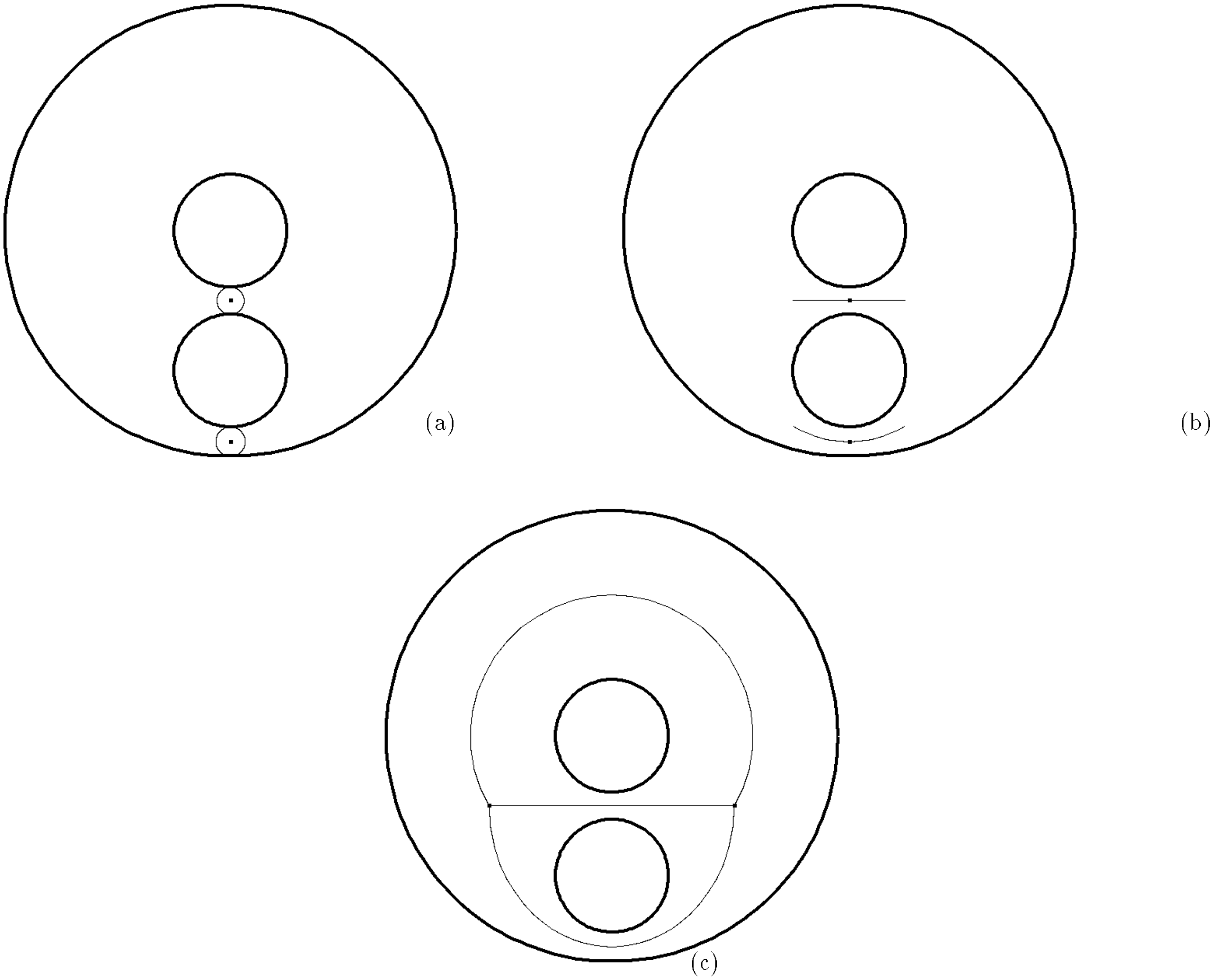

Figure 6: Skeleton for a region with holes. The computation of a skeleton for a region with two holes is shown. In (a), the initial skeleton points associated with the interior curves are depicted along with their respective interior circles. In (b), the initial segments are extended from the initial points. These segments serve to initialize the skeleton branches for the computation. In (c), the skeleton associated with the region is shown.

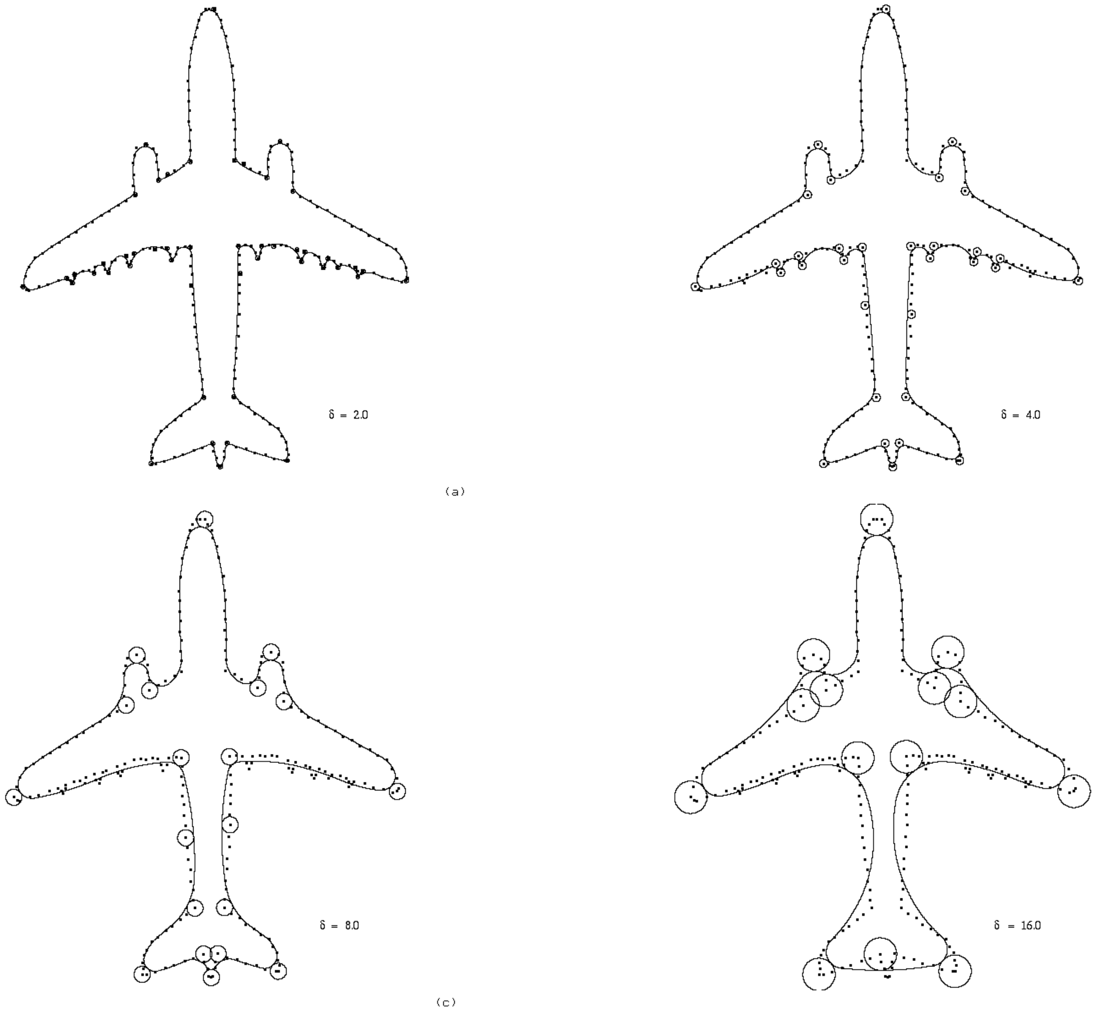

Figure 7: Minimum complexity curves for the silhouette of an airplane. Each curve is an instance of a minimum complexity curve for a particular tolerance value. The circles that are tangent to the curve are centered about a particular critical data point. The radius of each of these circles is equal to δ, the scale parameter. The scale parameter for each case is (a) $\delta = 2.0$, (b) $\delta = 4.0$, (c) $\delta = 8.0$, and (d) $\delta = 16.0$.

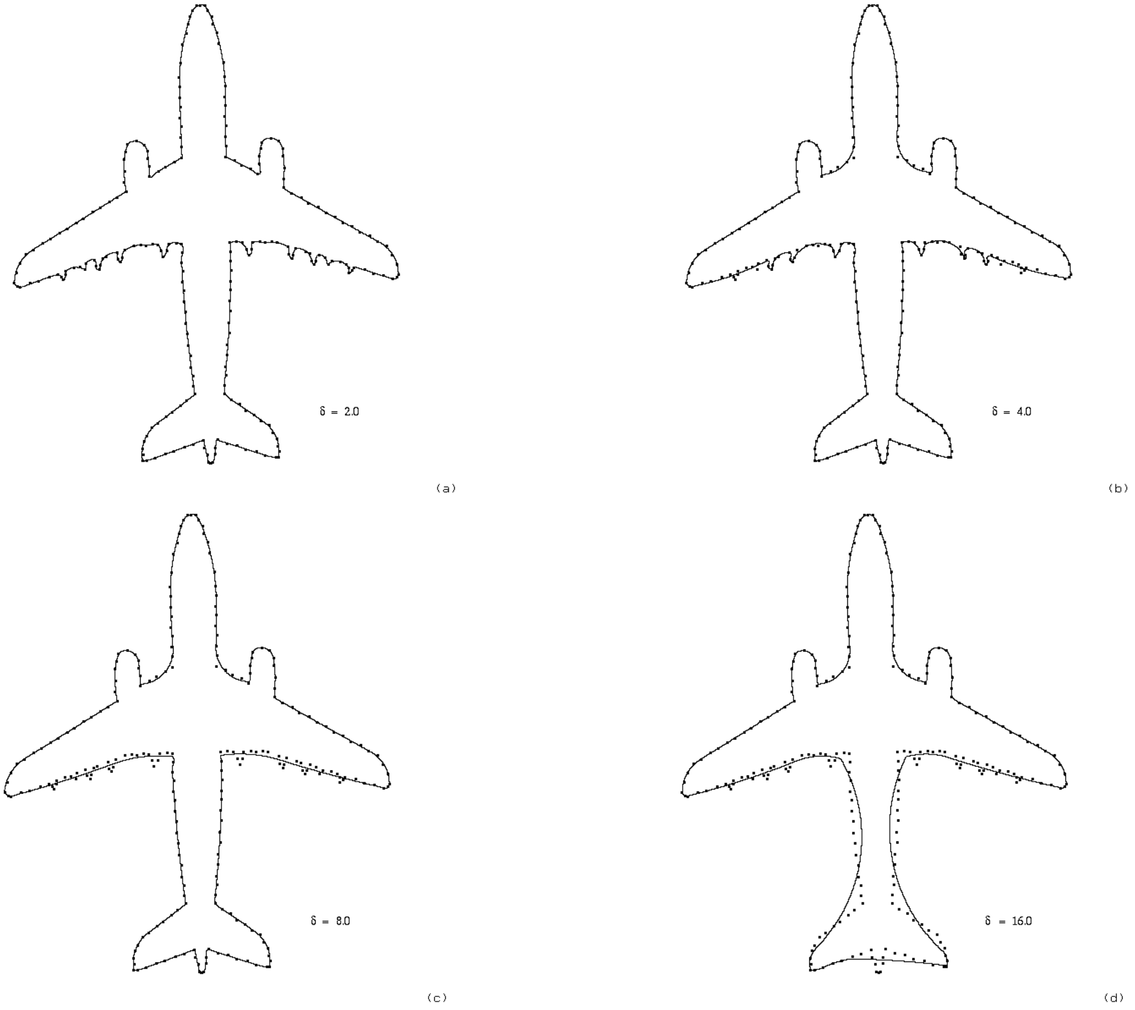

Figure 8: Minimum complexity/least-square-error curves for the silhouette of an airplane. In each case the least-square error curve that has the complexity determined from the first stage of the computation is shown. The scale parameter for each case is (a) $\delta = 2.0$, (b) $\delta = 4.0$, (c) $\delta = 8.0$, and (d) $\delta = 16.0$.

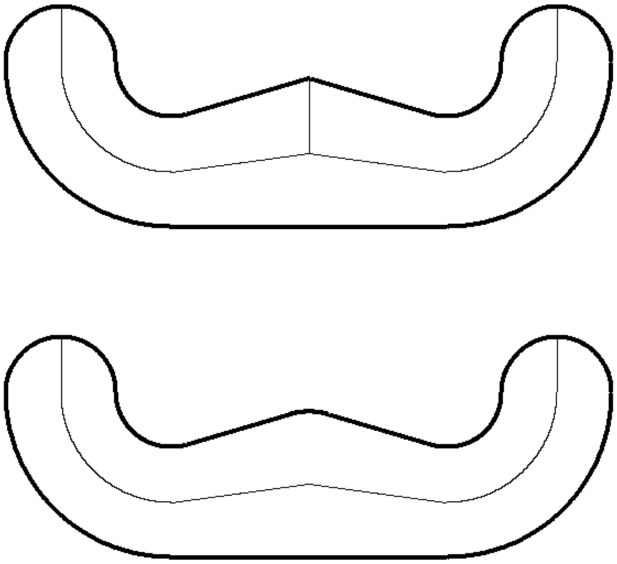

Figure 9: Sensitivity of the skeleton at maxima of curvature. Two curves are shown with their respective medial axis skeletons. The two curves are identical except near the maximum of curvature on the top of each curve. The complexity measures of the curves are equal. The top curve has an additional skeleton branch because the magnitude of curvature at the maximum is much greater than that of the bottom curve. The complexity measure of the skeleton is very sensitive to subtle changes in the bounding contour, particularly near positive maxima of curvature.

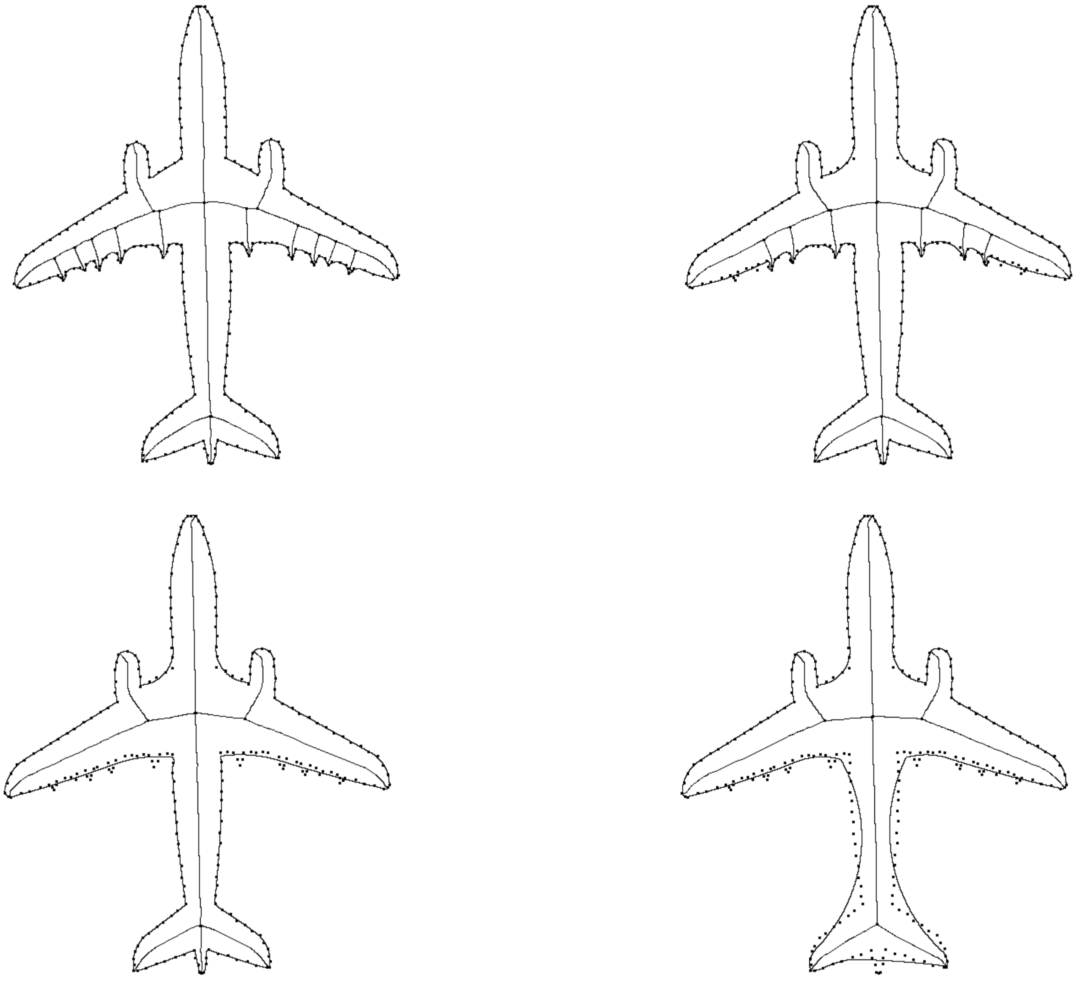

Figure 10: The complexity scale-space for the medial axis skeleton. The medial axis skeleton and the corresponding bounding contour are shown at four scales. The scale parameter, δ, is doubled between each scale. Note that the number of branches, as well as the number of features on the contour, decreases as the scale parameter increases. The features that are present in each representation are depicted as accurately as possible in the square-error sense. The small squares near each contour represent the initial data points of the contour.