Feature-Preserving Medial Axis Noise Removal

Roger Tam and Wolfgang Heidrich

Department of Computer Science,
University of British Columbia
{rtam|heidrich}@cs.ubc.ca

Abstract. This paper presents a novel technique for medial axis noise removal. The method introduced removes the branches generated by noise on an object's boundary without losing the fine features that are often altered or destroyed by current pruning methods. The algorithm consists of an intuitive threshold-based pruning process, followed by an automatic feature reconstruction phase that effectively recovers lost details without reintroducing noise. The result is a technique that is robust and easy to use. Tests show that the method works well on a variety of objects with significant differences in shape complexity, topology and noise characteristics.

1 Background and Motivation

The medial axis [1] is a useful shape representation for many applications in computer vision and computer graphics. The primary drawback of the medial axis is that it is very sensitive to minor perturbations of the object's boundary, such as that caused by discretization, segmentation errors, image noise, and so forth. The goal of most medial axis *pruning* techniques is the removal of branches associated with these artifacts, resulting typically in a much cleaner and more usable medial axis. In addition, the denoised axis can then be used to reconstruct a smoother version of the original object.

Most current medial axis pruning algorithms suffer from the problem that when excess branches are removed, other branches that correspond to fine but perceptually significant features of the object are excessively shortened. This is primarily due to the fact that most pruning methods use a global significance measure (*e.g.*, feature size or frequency) to discern between data and noise. Unfortunately, for most measures there is a significant overlap between what is considered noise and data, and when the noise is removed some data is taken with it. Figure 1 provides a very simple, motivating example. The unprocessed medial axis (Fig. 1b) has many spurious branches, largely because of discretization artifacts. Figure 1c shows the typical result of pruning with a global threshold. In this case the significance measure is noise size. The result is that all of the branches associated with noise are gone, but the remaining branches are also shortened, causing the tip of the pencil to become rounded (Fig. 1d).

So far, proposed solutions to address this issue have proven inadequate. Attempts to overcome the noise/data overlap problem by developing more complicated global measures frequently result in a fuzzy relationship between parameter values and how they correspond to changes in object features, thereby making the estimation of an appropriate threshold more difficult. Another general approach is to recover lost details

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 672–686, 2002.

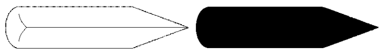
(a) (b) (c) (d)

Fig. 1. Pencil (a) Original object (b) Unpruned axis (c) Typical pruned axis (d) Object reconstructed from typical pruned axis

by *unpruning* the remaining branches after the noisy branches are removed. Current algorithms using this approach typically do not work well because they depend on a global threshold for the unpruning process as well, thereby subjecting it to the same overlap problem.

In this paper, we present a novel approach for medial axis denoising that removes unwanted artifacts while preserving fine features, such as sharp corners and thin limbs. Our method first prunes the axis by using an intuitive global threshold based on noise size, then *automatically reconstructs* the fine features by extending the remaining branches. The use of a simple pruning method based on a physically meaningful parameter followed by an effective feature reconstruction process makes the technique robust and easy to use. The user can determine an appropriate threshold simply by estimating the size of the noise he or she wants to remove and in most cases, rough estimates are adequate because the reconstruction process can automatically correct many errors caused by overly aggressive pruning.

Our feature reconstruction algorithm extends each branch by using local shape information and does not depend on a global threshold. This process localizes the discernment between data and noise to the feature level which significantly reduces the overlap problem. As demonstrated in our results, this localization allows each branch to be extended to an appropriate length and in the correct direction so that each feature is reconstructed accurately without reintroducing noise. Figure 2a shows the medial axis of the pencil after processing with our algorithm; the noise is gone and the tip of the pencil is still sharp.



(a) (b)

Fig. 2. Pencil (a) Axis processed by our denoising algorithm (b) Our reconstructed object

Although simple in its design, we can show that our technique works well for removing artifacts of various sizes and characteristics from objects of arbitrary shape complexity and topology. We have tested our algorithm on a wide variety of data, a number of examples of which are included in this paper to demonstrate the effectiveness of our method.

2 Related Work

Our method for the construction of the medial axis is one of a number of algorithms that use the Voronoi graph of a set of sample points regularly spaced along the object's boundary to form an approximation of the skeleton. The main idea of such algorithms, examples of which include [2,3,4], is to first compute the Voronoi graph of the points, then extract a subgraph to form the skeleton. For example, the subgraph can be extracted by taking only the Voronoi vertices that are inside the boundary of the object.

Given a model or an image of an object, there are two main approaches for producing a clean medial axis. The first approach performs some form of preprocessing on the image or model before computation of the medial axis. Such preprocessing usually consists of blurring (*e.g.*, [5]) or boundary smoothing (*e.g.*, [6]) of the original object to reduce spurious branches. Blurring and smoothing techniques can result in undesirable structural changes to the medial axis [7,8]. In addition, these operations typically use a global scale measure (*e.g.*, size of smoothing kernel) to filter out noise, and smaller object features are often altered or destroyed during preprocessing.

The other main approach is to start with the complete axis and prune the branches using some heuristic (*e.g.*, [3,9]). The general idea is to have a significance measure that assigns an importance value to each branch. During pruning, this value is compared to a user-given threshold to determine how much of each branch gets cut. With an ideal significance measure and threshold, only the parts of the axis associated with noise would be removed, and the rest of the axis would remain unaltered. However, for currently available measures there is usually an overlap between data and noise; a threshold value that completely removes the branches associated with noise will usually shorten the remaining branches as well, often to an undesirable degree. Thus, finding a good threshold value often requires striking a delicate balance between noise removal and feature preservation. In addition, the complexity of some measures makes them seem ad hoc and adds to the difficulty of finding an appropriate threshold. In some cases, even multiple parameters are required (*e.g.*, [10]). To overcome the difficulties in estimating parameter values, a completely automatic method for threshold selection is proposed in [7,11]. The method is able to determine an appropriate value for many shapes, but there are instances in which the algorithm strongly oversegments the shape, resulting in large missing features. Figures 3a and 3b, generated with the algorithm from [7], show two examples. In Fig. 3a, the stem of the leaf is missing from the axis; in Fig. 3b, two of the goat's legs are among the larger features not represented.

A number of researchers have proposed methods that utilize a postprocess to recover small details lost by pruning. Such methods add an unpruning process that extends the branches that remain after pruning (*e.g.*, [8,11]). The typical approach is to use the same significance measure as used for pruning and simply apply a different threshold to extend the branches. This approach forces the user to select two thresholds, and still the problem of overlap between data and noise is not solved. This frequently results in some branches being overextended (*i.e.*, noise is reintroduced) while others are still too short.

Our algorithm is designed to address the problems described above, and consists of the following two main processes:

Fig. 3. Ogniewicz's (a) Leaf (b) Goat

1. A threshold-based pruning technique with a single parameter and a simple significance measure that gives the user intuitive control.
2. An automatic feature reconstruction process that extends each branch using local shape information and does not depend on a global threshold.

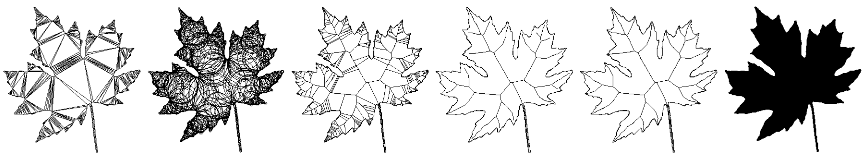
The result is an algorithm that gives the user the freedom to select a threshold that completely removes all noise while providing a reliable feature reconstruction process which brings back the right amount of detail at each branch. This technique gives the user some control, so that large features are not accidentally removed, but hides the more complex data/noise discernment algorithm inside an automatic process so that the user is not burdened with a complicated significance measure.

It should be noted that some pruning methods, such as [11], are hierarchical in nature and can produce results at multiple levels of detail. Thus, at coarser levels, the loss of fine features is considered acceptable, even appropriate. In contrast, our algorithm is designed to remove artifacts of a given size, while preserving as much detail in the rest of the object as possible. However, at finer levels of detail, the goals of the algorithms are essentially the same.

3 Methodology

Given the boundary points of an object, the main steps of our algorithm for medial axis noise removal are as follows:

1. Construct the medial axis from the boundary points (Figs. 4a-c).
2. Prune the spurious branches (Fig. 4d) by using a user-determined global threshold.
3. Extend the remaining branches to recover small details (Fig. 4e) by using a local measure of shape smoothness to distinguish between data and noise.
4. Reconstruct the object with the clean medial axis (Fig. 4f).



(a) (b) (c) (d) (e) (f)

Fig. 4. Maple leaf example (a) Delaunay triangulation (b) Union of Circles (c) Complete medial axis (d) Pruned medial axis (e) Pruned medial axis with details recovered (f) Denoised shape

3.1 Medial Axis Construction

Like most Voronoi-based methods for medial axis construction, our method assumes that the sample points are spaced with sufficient density along the boundary of the object. Our method for the computation of the medial axis from a boundary point set can be divided into three main steps:

1. Compute the Delaunay triangulation of the point set and discard any triangles that are outside of the object (Fig. 4a).
2. Compute the circumscribing circle of each remaining triangle (Fig. 4b). This set of circles is referred to as a *Union of Circles* (*UoC*) [12].
3. Construct the medial axis by connecting the centres of the circles (Fig. 4c).

We represent the medial axis as a directed graph[1] whose root node is the centre of the largest circle in the UoC. Construction of the graph begins by creating a line segment (called an *axial segment*) between the root node and each of its neighbours (two circles are *neighbours* if their corresponding triangles share an edge). The root node is the *parent* node and the neighbours are *child* nodes. This process is then repeated with the neighbours as parent nodes until all circles in the UoC are linked. The result is a medial axis that can be traversed recursively by starting at the largest circle and following the child nodes until they reach the boundary of the object. With this construction method, each node in the axis has a corresponding circle in the UoC, which in turn has a corresponding triangle in the Delaunay triangulation.

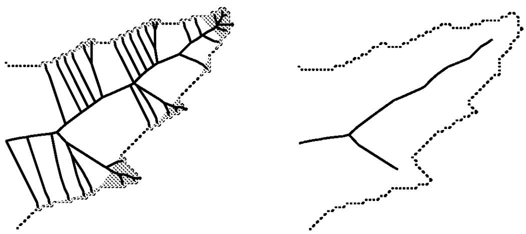
In this paper, a node that has no children is referred to as an *end node*. A node that has more than one child is called a *branch node*. A *branch* is defined as any chain of nodes that has a single branch node at the beginning and an end node at the end.

3.2 Area-Based Pruning

The purpose of the pruning process is to remove parts of the medial axis that are associated with noise on the object's boundary. A significance measure is needed for determining

[1] For objects of genus zero, the graph is naturally a tree. For objects with holes, we break each cycle by imposing an appropriate breakpoint in the loop. This is only for the purposes of traversing the graph without running into infinite loops. In order to preserve the topology of the original object, cycles are never pruned.

whether a feature of the object should be considered as noise. In the context of pruning, a feature can be defined as the set of triangles associated with any subtree of the axis graph. Our significance measure assigns an importance value to a feature based on the surface area it covers. An example is shown in Fig. 5a, which shows a part of the maple leaf from Fig. 4. In this figure, the shaded regions are features that are smaller in area than the user-given significance threshold. The significance value of a feature can be determined by summing the areas of all triangles in the subtree associated with that feature. Any subtree that has a value below the threshold is pruned. Each feature can be seen as being *supported* by the branches of the subtree, so when the subtree is pruned, the feature is eliminated.



(a) (b)

Fig. 5. Area-Based pruning (a) Unpruned axis (b) Pruned axis

Our significance measure has the following two main advantages:

- The pruning is guaranteed not to disconnect the graph, because a parent always has a higher significance value than its child.
- Area is a simple and intuitive significance measure and an appropriate threshold can be estimated via a typically straightforward analysis of the data acquisition method. Even when knowledge of the acquisition method is insufficient, a suitable value can be found by visual inspection of the data more easily than most heuristics-based measures that have a less direct physical meaning.

The result of the pruning process is that all noise below the threshold size is eliminated. With an appropriate threshold, the only branches that remain are associated with significant features of the object. As mentioned, a side effect of pruning with a fixed global threshold is that the remaining branches are typically shorter than they should be and fine but important details are often lost.

3.2.1 Noise Model and Threshold Selection

Like practically all medial axis pruning methods, our noise model focuses on artifacts in the form of relatively small protrusions from a larger body. Pruning techniques are the most effective when applied to this type of noise, called *additive noise*, because they work by removing and shortening branches. A common example of additive noise are the artifacts originating from the dark current in CCD cameras. We define two conditions that must be satisfied in order for a feature to be classified as noise:

1. The size of the feature is smaller than the user-determined threshold. This is the condition used in the pruning phase.
2. The feature is not *smoothly connected* to the rest of the object. The transition into a protrusion is considered smooth if the abruptness of the narrowing does not exceed the changes in width in the parts of the object leading up to the feature. This condition is used in the feature reconstruction phase and is defined more precisely in the next section.

To select an appropriate threshold for pruning, consideration must be given to the data and application at hand. If the noise characteristics are known, an estimate of the artifact size can be made, and selection of the threshold is relatively simple. Otherwise, the value can be set by visual inspection of the data. Our implementation is such that the threshold can be set as an absolute size or as a percentage of the total area of the object. In most of our examples, the value is determined interactively. The feature reconstruction process, as described in the next section, is robust enough to allow a fairly imprecise threshold selection, and a range of appropriate values exist for most objects.

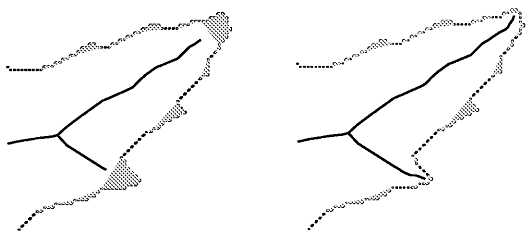
The most important guiding principle in selecting an appropriate threshold is to ensure that each significant protrusion in the object has a single supporting branch. The reason for this is best described by Leyton's Symmetry-Curvature Duality Theorem [13]:

Theorem 1 (Symmetry-Curvature Duality). *Any section of curve, that has one and only one curvature extremum, has one and only one symmetry axis. This axis is forced to terminate at the extremum itself.*

Figure 5b shows an example of how each significant feature is supported by a single branch. The result of pruning should be that each significant extremum in the border has a single remaining branch.

3.3 Feature Reconstruction

The purpose of the feature reconstruction process is to recover significant parts of the object that have been pruned because they fall below the size threshold. For example,



(a) (b)

Fig. 6. Feature reconstruction (a) Before reconstruction (b) After reconstruction

Fig. 6 shows part of the maple leaf before and after reconstruction. The shaded areas show features that would be removed. The axis in Fig. 6b clearly represents the shape of the object better than the axis in Fig. 6a.

Our reconstruction algorithm works on one branch at a time, and its main idea is to use the shape information present in the remaining branches and circles to calculate local smoothness constraints that determine how far each branch can be extended to recover fine features without reintroducing noise. In this scheme, what is classified as noise varies from branch to branch. Figure 7 shows an example in which the branch in one feature (B) is extended further than in another (A), even though the tips of the features have the same angle and both branches reach the boundary in the unpruned axis. In this case, the small sharp point in Feature A is regarded as noise because it falls below the size threshold and violates the local smoothness constraints.

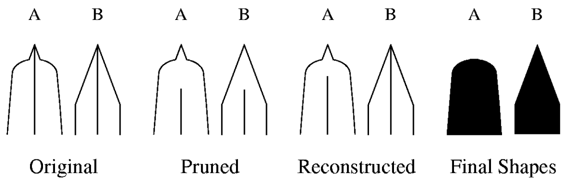


Fig. 7. Effect of smoothness constraints on feature reconstruction

The measure of smoothness that we use is termed the *axial gradient*, which measures the change in the width of the object per unit length of the axis. Each axial segment has an axial gradient value that is mathematically defined as the signed difference in radius between the child circle and the parent circle, divided by the Euclidean distance between the two nodes. Figure 8 illustrates this definition.

The algorithm works by starting at the root node and following the axis until it reaches the current end node of the branch in question, while keeping track of the greatest absolute axial gradient value ($|g_{\max}|$) encountered along the path. This value is used to determine how far the branch can be extended. The reasoning is that if a feature at the end of the branch is below the threshold size and is marked by a narrowing that is more abrupt than any other change in width along the path, then the feature is most likely noise. Because $|g_{\max}|$ is calculated individually for each branch, the data/noise overlap problem associated with global thresholds is significantly reduced.

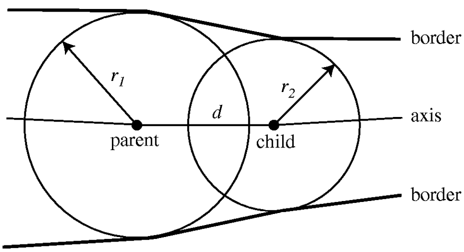


Fig. 8. Axial gradient between two nodes ($g = \frac{r_2 - r_1}{d}$)

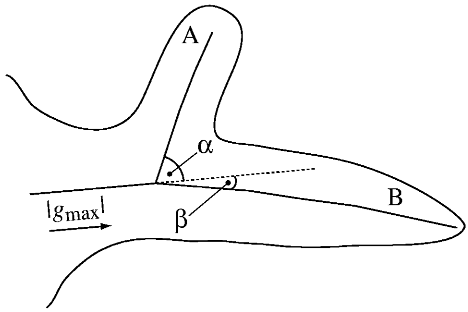


Fig. 9. Axial gradient at a branch node. Because β is smaller than α, $|g_{\mathrm{max}}|$ is more relevant to Branch B than Branch A.

When calculating $|g_{\mathrm{max}}|$ along a path, special consideration must be given to branch nodes, because the degree of *continuity* of a feature across a branch node depends strongly on the branching angles at these nodes. Figure 9 shows an example in which one feature (B) has stronger continuity across a branch node than another (A). In this case, the shape information along the path before the branch node is more relevant to Branch B than Branch A. For any given path, the branching angle at a branch node is a good indicator of how much of the maximum gradient encountered before the node should be "carried over" past the branch node. Intuitively, an angle of 0 degrees (maximum continuity) should impose no change to the maximum gradient, whereas an angle of 90 degrees or greater (no continuity) should cause $|g_{\mathrm{max}}|$ to become 0 at the branch node. For an angle between 0 and 90, we use this formula:

$$g_{\mathrm{max}}(\theta) = |g_{\mathrm{max}}| \times \frac{90-\theta}{90}$$

where $|g_{\mathrm{max}}|$ is maximum axial gradient before the branch node and $g_{\mathrm{max}}(\theta)$ is the maximum axial gradient at the branch node for the segment with angle θ.

Once $|g_{\mathrm{max}}|$ for a given path has been calculated, the next step is to extend the branch at the end of the path. Two main issues need to be considered at this point. First, if there is more than one direction for possible extension (*i.e.*, the current end node is a branch node in the original axis), a decision needs to be made to determine which segment to follow. The second issue is how far to extend the branch.

The first issue is addressed in consideration of the second part of Leyton's theorem, which says that the symmetry axis of a feature should terminate at the extremum of that feature. Given that the objects we are considering have many minor extrema due to noise, we need to distinguish these from the extrema associated with significant features. Again, we use the axial gradient for this purpose. If there is more than one path for possible extension, the segment with the lowest axial gradient is chosen. This method is essentially a greedy algorithm for finding the smoothest path. As shown in the Results section, this gives a high probability of reaching the correct extremum. Figure 10 illustrates an example.

The second issue of how far to extend the branch is addressed by comparing the absolute value of the axial gradient of the chosen candidate segment ($|g_{\mathrm{cand}}|$) with the

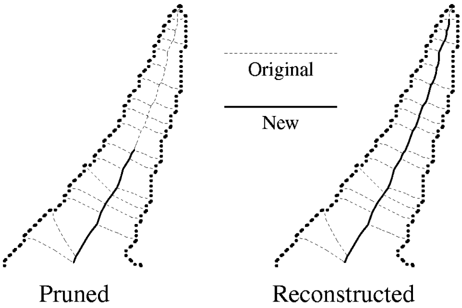


Fig. 10. Branch extension. At each branch node, the segment with the lowest axial gradient is chosen, which extends the branch toward the appropriate extremum.

$|g_{\text{max}}|$ of the branch. If $|g_{\text{cand}}| \leq |g_{\text{max}}|$, the segment is added to the end of the branch. Note that because our noise model defines noise as small protrusions from a larger body, a feature can be classified as noise only if $g_{\text{cand}} < 0$. Segments are added until $g_{\text{cand}} < 0$ and $|g_{\text{cand}}| > |g_{\text{max}}|$, or there are no more candidate segments.

As demonstrated by the examples in the Results section, our branch extension algorithm is very effective in automatically reconstructing features to their appropriate degree of sharpness without reintroducing noise.

3.4 Shape Reconstruction

An advantage to using the UoC method for medial axis construction is that there is a one-to-one correspondence between the nodes of the axis and the set of circles. To reveal the final object shape after pruning and feature reconstruction, we simply take the union of the circles associated with the new axis. The triangles associated with the new axis can be used efficiently for applying error metrics based on area.

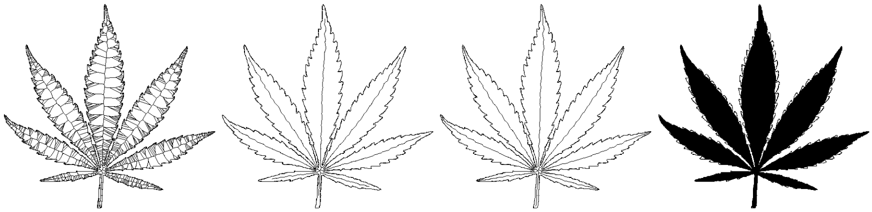
4 Results

We have tested our algorithm on many objects with various amounts of noise. Our data sources include map data, medical images, aerial photographs and specially designed/chosen test models. In this section, a number of examples are used to illustrate the effectiveness of our method. Possible extensions to overcome some of the current limitations of our algorithm are also discussed. The examples include four synthetic objects and one object from an MRI image. The characteristics of the objects are shown in Table 1. All thresholds (t) are specified as a percentage of the total area of the original object.

Figures 11 (Leaf) and 12 (Goat) are good examples of how the algorithm can reconstruct fine features after removing unwanted artifacts. In the Leaf, the small variations along the border are removed, resulting in a much smoother shape. However, the fine features such the tips of the leaflets and the thin stem are nicely reconstructed. This example can be compared to the result by Ogniewicz in Fig. 3a, where the stem is missing

Table 1. Characteristics of our test objects

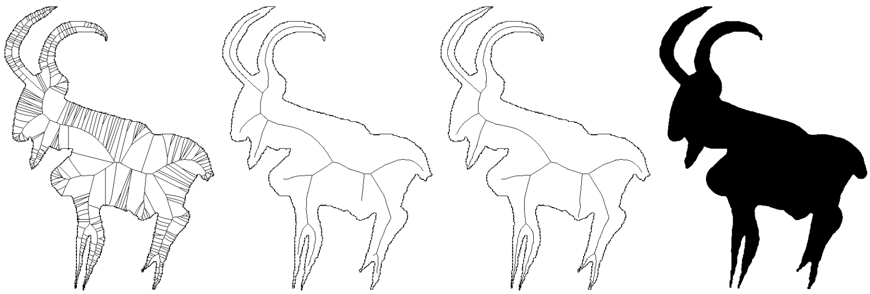
Object	Figure	Dimensions	Threshold (t)
Leaf	11	479×462	0.3%
Goat	12	254×344	0.4%
Lizard	13	443×446	2.0%
Brain	14	255×293	0.1%
Rectangle	15	339×238	2.0%



(a) (b) (c) (d)

Fig. 11. Leaf (479×462, $t = 0.3\%$) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

completely. The Goat shows how the branch extension method can use local shape information to reconstruct features to differing degrees of sharpness where appropriate. For the sharper features, such as the horns, goatee and legs, the branches are extended to the tips. For the more rounded features, such as the mouth, chest, belly and tail, the branches are extended enough to fully reconstruct the features, but not so far as to reintroduce noise.



(a) (b) (c) (d)

Fig. 12. Goat (254×344, $t = 0.4\%$) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape

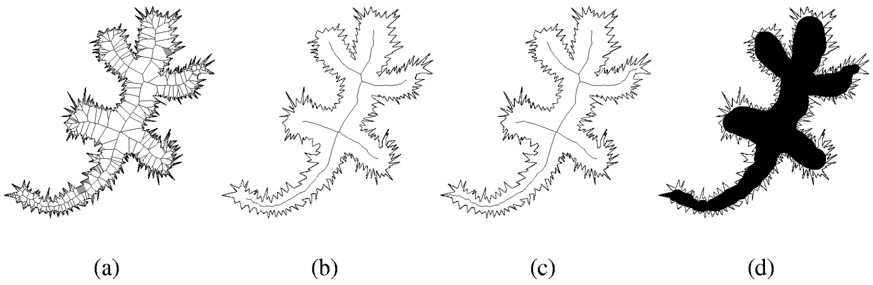


Fig. 13. Lizard (443×446, $t = 2.0\%$) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

Figure 13 (Lizard) is an example where the "noise" artifacts are quite large. Our algorithm still results in a nicely denoised shape in this case. In the head and the four legs, the reconstructed branches do not extend into any of the spurious spikes along the border. The shape of the tail causes its supporting branch to be extended to the tip. It is somewhat debatable whether the spike at the end should be considered as a significant feature or noise, but in this case its inclusion seems appropriate.

Figure 14 (Brain) shows an object from an MRI image. The noise in this case is a combination of image noise, segmentation errors and discretization artifacts. The shape of this object is significantly more complex than in the other examples. This object is also of a different topology in that it has two holes. Our algorithm is able to effectively remove the various types of noise from all areas of the object.

Figure 15 (Rectangle) is an example of an object that has a heterogeneous distribution of noise. In this case, the left side of the rectangle is very noisy, whereas the right side is clean. Again, the reconstruction algorithm is able to perform well, extending branches to their appropriate lengths so that the corners on the right side are sharp, while the artifacts on the left side are removed.

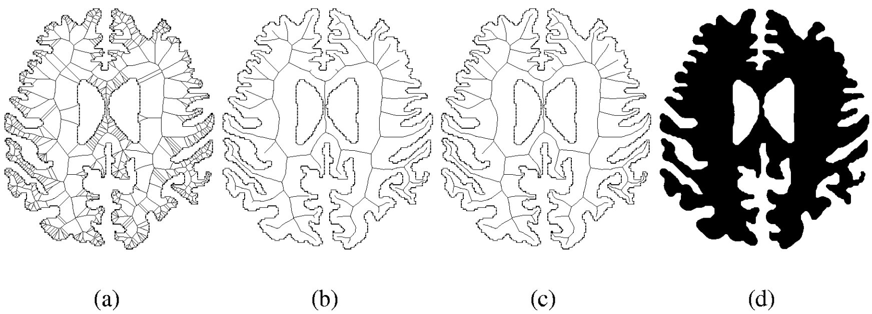


Fig. 14. Brain (255×293, $t = 0.1\%$) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape

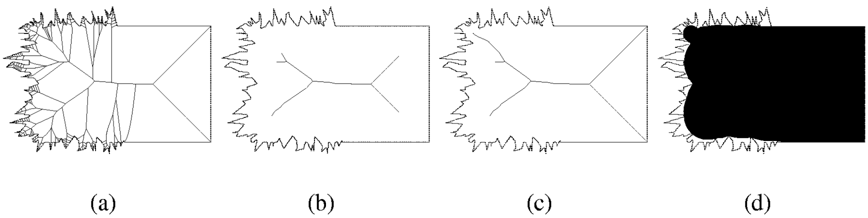


Fig. 15. Rectangle (339×238, $t = 2.0\%$) (a) Unpruned (b) Pruned (c) Features reconstructed (d) Final shape with original boundary superimposed

4.1 Extensions

We are considering a number of extensions to our algorithm to overcome some of the current limitations:

- Our use of the maximum axial gradient as a local smoothness constraint inherently assumes that the axial gradient does not vary greatly within a single feature, at least not relative to the gradient of any noise at the end of the branch. Although such variations are not encountered frequently, they certainly can exist. Figure 16 shows an example of such a situation. In this case, a thin *neck*, or local narrowing, in the object causes the branches in the noisy square to be overextended into the corners. A possible solution to this problem is to impose a limit on the length of the path used for computing the maximum gradient. For example, instead of only having one root node from which to start, we can break the graph down into subgraphs using features such as necks to do the division. Although theoretically straightforward, this has not been implemented at the time of writing of this paper.

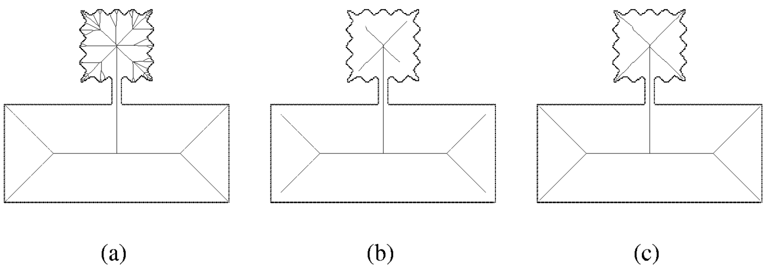


Fig. 16. Narrow neck example (a) Original axis (b) Pruned axis (c) Axis after feature reconstruction. The branches are extended too far.

- The branch extension algorithm may not work well in the case of a very short branch with a large branching angle, because there would be very few axial segments and, therefore, a very limited amount of local shape information with which to calculate an appropriate maximum axial gradient. Increasing the sampling density

of the boundary points may be a possible solution, but this would increase the computational costs considerably.
- Although our tests show it to be largely effective, the greedy algorithm for finding the smoothest path for branch extension is not *guaranteed* to reach the correct extremum. A solution would be to search further into the tree before making a path selection.

5 Summary and Future Work

We have presented a novel algorithm for medial axis noise removal. Our algorithm consists of a threshold-based pruning method that uses a simple significance measure, followed by an automatic feature reconstruction process that extends the remaining branches to recover fine features without reintroducing noise. Our method is easy to use and our results show that the approach has strong potential to be viable in practical applications.

There are a number of areas, in addition to the extensions described in Sect. 4.1, that we intend to include as part of our future research:

- Although in our experience the method presented is robust with respect to changes in the user-selected threshold (Sec. 3.2), a more formal evaluation of the algorithm's sensitivity should be done.
- We intend to further investigate the axial gradient approach by testing alternate formulations, such as using the average gradient with a tolerance instead of the maximum gradient.
- We would like to develop a method for automatically determining the pruning threshold. Although our current technique makes threshold selection relatively easy, an automatic method would be useful for processing large data sets. To achieve this goal, a more precise noise model may be required (*e.g.*, [10]).
- We would like to extend the method to 3D.

Acknowledgments. We would like to thank Robert Ogniewicz for making his pruning program and test data publicly avaiable. This research is supported by the Natural Sciences and Engineering Research Council, British Columbia Advanced Systems Institute, and the University of British Columbia. This paper is dedicated to the memory of Alain Fournier.

References

1. Blum, H., Nagel, R.: Shape description using weighted symmetric axis features. Pattern Recognition **10** (1978) 167–180
2. Attali, D., Montanvert, A.: Semicontinuous skeletons of 2D and 3D shapes. In: Proceedings of the International Workshop on Visual Form, Capri, World Scientific (1994) 32–41
3. Brandt, J., Algazi, V.: Continuous skeleton computation by voronoi diagram. CVGIP: Image Understanding **55** (1992) 329–337

4. Ogniewicz, R., Ilg, M.: Voronoi skeletons: Theory and applications. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (1992) 63–69
5. Pizer, S., Oliver, W., Bloomberg, S.: Hierarchical shape description via the multiresolution symmetric axis transform. IEEE Trans. Pattern Analysis and Machine Intelligence **9** (1987) 505–511
6. Mokhtarian, F., Mackworth, A.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE Trans. Pattern Analysis and Machine Intelligence **14** (1992) 789–805
7. Ogniewicz, R.: Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory (1995)
8. Shaked, D., Bruckstein, A.: Pruning medial axes. Computer Vision and Image Understanding **69** (1998) 156–169
9. Attali, D., Sanniti di Baja, G., E., T.: Pruning discrete and semicontinuous skeletons. In De Floriani, C., Braccini, C., Vernazza, G., eds.: Lecture Notes in Computer Science, Image Analysis and Processing. Volume 974. Springer-Verlag (1995) 488–493
10. Attali, D., Montanvert, A.: Modeling noise for a better simplification of skeletons. In: Proc. of the International Conference on Image Processing. Volume III., Lausanne, Switzerland (1996) 13–16
11. Ogniewicz, R.: Skeleton-space: A multiscale shape description combining region and boundary information. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, WA (1994) 746–751
12. Ranjan, V., Fournier, A.: Matching and interpolation of shapes using unions of circles. Computer Graphics Forum (Proceedings of Eurographics '96) **15** (1996) 35–42
13. Leyton, M.: Shape and causal-history. In Arcelli, C., Cordella, L., Sanniti di Baja, G., eds.: Visual Form: Analysis and Recognition. Plenum Press (1992) 379–388