

An Image Space Algorithm for Morphological Contour Interpolation

William Barrett, Eric Mortensen, and David Taylor
Department of Computer Science
Brigham Young University
Provo, Utah

e-mail: barrett@cs.byu.edu
Telephone: (801)-378-7430

Abstract

An image space algorithm for morphological interpolation between contours is presented. Image space interpolation avoids the need to represent or store contour data using intermediate data structures. The algorithm makes use of basic morphological transforms such as dilation and erosion and interimage operations such as XOR and union. Morphological interpolation is applied successfully to a variety of synthetic contours as well as naturally occurring contours such as those found in medical images or topographic maps [17]. The algorithm interpolates between nested, overlapping, nonoverlapping, or branching contours in a general way although nonoverlapping or minimally overlapping contours require initial registration. The algorithm is particularly appropriate for generation of digital elevation maps or whenever the original contour data is derived from a regular sampling grid. Image space morphological interpolation exploits pipeline architectures allowing simultaneous generation of interpolated contour values while making essential use of neighboring contour morphology. In addition, there is a logarithmic gain in the number of interpolated points when processing a contour interval exhaustively.

keywords: contour interpolation, morphological transforms, parallel, height grid, DTM, DEM, cartography

1. Introduction

Many real world objects are effectively and succinctly represented by contours. For example, geologic terrain surfaces can be represented by nested, usually nonintersecting, isocontours found in topographic maps. Isocontours often are used to generate Digital Terrain Models (DTMs) or (discrete) Digital Elevation Models (DEMs) in automated cartography [1-4]. Contours also may be extracted from and used to represent closed three-dimensional objects such as medical anatomy. For example, contours of anatomical boundaries may be detected automatically from serial transaxial cross sections in CT or MRI scans [5]. In the case of closed three-dimensional

objects, contours from two spatially adjacent slices frequently intersect or overlap when superimposed.

Because it may not be economically or physically practical to densely sample the object of interest, contours usually provide only a sparse representation of the object(s) from which they were extracted. As a result, interpolation schemes [6-10, 18] often are necessary to recover the original three-dimensional surface geometry. Thus, connection of, or equivalently, interpolation between contours is a general problem in computer graphics. The most significant advancements in approaching this problem have come through algorithms which address the correct mapping or correlation of contours points at one level with those at an adjoining level [6, 18].

Some of the greatest difficulties associated with a general solution to the problem of contour interpolation arise due to topological changes and/or overlap between adjacent contours, such as when contours differ in both position and number from one level to an adjacent level. Such is the case when there is a natural branching of the surface geometry. Even without branching, if there is a striking disparity in contour shape or position between two adjacent levels, interpolation algorithms may fail to provide a smooth transition of object geometry at intermediate levels. Thus, efficient and robust contour interpolation algorithms which make essential use of contour morphology at both the local and the global level are still needed.

A new image space contour interpolation algorithm which exploits both local and global contour morphology has been developed. The algorithm makes use of morphological transforms (such as dilation and erosion) and other image-level logical operations (AND, OR, or XOR), all of which operate directly in image space. The main idea of the algorithm is to find the midline between two contours and use it to split the intercontour space into two halves, each of which can be processed in the same way. This process is repeated recursively until the intercontour space is exhausted (i.e. filled with midlines). If the initial two contours overlap, the first midline simply passes through the point of

intersection and the algorithm proceeds as usual.

When compared to existing techniques, image space morphological contour interpolation offers advantages in robustness, accuracy, and computation. These include

1. Robustness: Handles any number of contours of any shape including branching or overlapping geometries. Nonoverlapping contours must be registered.
2. Accuracy: Makes essential use of contour morphology; local shape is extracted using dilation and erosion operations while global shape is represented by midlines.
3. Computation: All operations are performed directly in image space which avoids intermediate contour representation and storage while exploiting pipeline architectures. For nested contours this results in massively parallel speedup since *all* contour intervals as well as entire families of interpolated contour

In particular, for each contour interval processed in parallel the parallelism increases $O(2^{m-1})$ with each recursion while the number of operations decreases $O(\log_2 n)$. This is because contour intervals are essentially split in two with each iteration.

Morphological contour interpolation is very well suited to generation of height grid DEMs from discrete isocontours as is illustrated in this paper. However, the algorithm is also extensible and applicable to other 3D objects whose contours originate *on the grid* such as heart contours from CT scans [5].

A brief introduction to mathematical morphology is given in Section 2. This is followed in Section 3 with a definition of and a distinction between the midline and the medial axis of a region. Section 4 presents the algorithm for morphological contour interpolation followed by results from both simulated and real world contours in Section 5. Section 6 contains a summary of algorithm features with suggestions for future work.

2. Mathematical Morphology

One of the strengths of mathematical morphology lies in its ability to decompose complex shapes into their meaningful parts. In fact, some morphological transforms result directly in structures, such as the medial axis, which have powerful intrinsic shape-describing content. The intent here is to present briefly some morphological operations which are integral to the problem of contour interpolation. For a more detailed treatment of mathematical morphology see references [11-12].

2.1 Dilation and Erosion

The most basic morphological transforms are dilation and erosion. Dilation and erosion transforms exist for both binary and grayscale images. We first define binary dilation, τ_b , and erosion, s_b . Let X be a binary-1 object such as shown in the image in Figure 1a. (Black dots indicate binary-1 pixels; empty squares have value 0.) Let B be a structuring element of binary-1 pixels such as shown in Figure 1b. (A structuring element is similar to a convolution kernel in image processing.) Let B_x be the translation of B so that the origin of B is located at position x in the object image. The binary dilation of X by B ($X \tau_b B$, Figure 1c) is obtained by passing B over the object image and ORing B to the (initially 0) output image whenever the origin of B is over a binary-1 pixel $x \in X$. Formally,

$$X \tau_b B = \{x \mid B_x \cap X \neq \emptyset\} \quad (1)$$

The binary erosion of an object X by B , ($X s_b B$) is obtained by passing B over the object image and plotting the origin of B in the (initially 0) output image whenever B is completely contained in X . Formally,

$$X s_b B = \{x \mid B_x \subset X\} \quad (2)$$

If X were represented by the binary-1 pixels in Figure 1c, binary erosion of X by B would result in the object in Figure 1a. Thus, binary erosion is the dual of binary dilation.

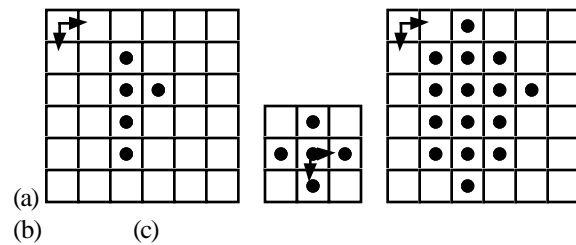


Fig. 1. (a) Binary-1 object pixels $X = \bullet$ (b) Structuring element, $B = \bullet$ (c) $X \tau_b B = \bullet$. \blacktriangledown denotes origin.

Grayscale erosion and dilation can be considered an extension of the binary case. However, rather than being defined only in terms of binary set operations, grayscale erosion and dilation are defined in terms of minimum and maximum over grayscale image pixels I covered by a grayscale structuring element, G_x , with its origin translated to position x in I . Specifically, grayscale dilation of I by G ($I \tau_g G$), is defined as

$$I \oplus_g G = \{x \mid x = \max_{z \in G_x \cap I} (I(z) + G_x(z))\} \quad (3)$$

where $I(z)$ are the pixel values in image I corresponding to values of $G_x(z)$ with its origin at x . Algorithmically, for each translation of G in image I , the pixel value x corresponding to the origin of G_x is calculated as follows for grayscale dilation:

Grayscale Dilation

Input: Grayscale image I , grayscale structuring element, G

Output: Dilated grayscale image I

```

for each pixel  $x$  in  $I$            {only process pixels
  for each pixel  $z$  in  $G_x \cap I$  {covered by  $G$ }
     $I(z) \leftarrow I(z) + G_x(z)$  {add  $I$  value to corresponding  $G$  value}
   $I(x) \leftarrow \max [I'(z) \text{ in } G_x \cap I]$  {output max in  $I$ }
```

In contrast, grayscale erosion of I by G ($I \ominus_g G$), uses minimization over the intersect region. Specifically, ($I \ominus_g G$) is defined as

$$I \ominus_g G = \{x \mid x = \min_{z \in G_x \cap I} (I(z) - G_x(z))\} \quad (4)$$

and can be computed with the appropriate changes in the dilation algorithm above. In the trivial case, the values of the structuring element G are all zero and add no bias to either the minimization or the maximization. Grayscale dilation of a simple image (Figure 2a) by a 4-connected grayscale structuring element composed only of zero values (Figure 2b) results in the image in Figure 2c, (ignoring values outside of the indicated region). Conversely, grayscale erosion of the image in Figure 2c would result in the original image in Figure 2a. This simple illustration is provided because the structuring element in Figure 2b is the one used for morphological contour interpolation described in the algorithm below.

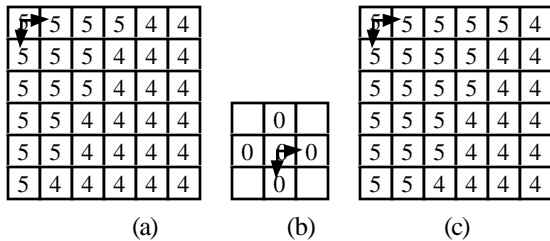


Fig. 2. (a) Grayscale image I (b) Structuring element, G (c) $X \oplus_g G$. \blacktriangledown denotes origin.

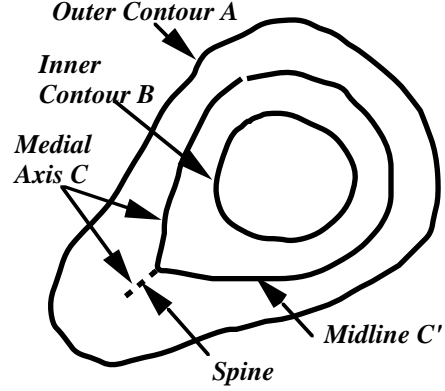


Figure 3. Medial axis, C , of region bounded by outside contour A and inside contour B . The medial axis includes the spine(s) (dashed) while the midline, C' , does not.

2.2 Medial-Axis versus Medial Line

The medial axis [15] of a two-dimensional space-filling region, R , is defined as the locus of points, $x \in R$ which have two or more points on the boundary of R which are equidistant from x (Figure 3). If the distance, d from x to the boundary is saved with each medial axis point, the region can be regenerated from the union of all disks of radius d centered at their respective medial axis points, x . Thus, the medial axis provides intrinsic and powerful global shape description capability. Note that the medial axis as used for shape regeneration is not necessarily connected. In this application the medial axis itself, rather than shape recovery, is our goal and it will be convenient for us to depict the medial axis as a connected line.

Figure 3 shows the medial axis, C , of the region defined by the intercontour space between two contours A and B . Note that the medial axis contains an extension referred to here as a spine. We define the medial line or midline, C' , to be the medial axis without the spine. For interpolation purposes, the medial line is preferable to the medial-axis since use of the medial axis (with spine(s)) results in an undesirable webbing artifact when used in the interpolation algorithm described below.

3. Morphological Contour Interpolation

The objective of the morphological contour interpolation algorithm which follows is to identify the height or elevation of points between contours initially labeled by height, z , thereby producing a continuous interpolated grid of contour values which defines explicitly the discrete surface represented by the original contour lines. The basic idea of the algorithm is to

expand (dilate) contours into the intercontour space until they collide. The collision front defines the medial line of the intercontour space. The medial line is labeled with the average of the original two contour labels and the process is repeated until all intercontour space pixels are labeled. This process is illustrated in Figure 4. The algorithm is given below.

Morphological Interpolation Algorithm.

Input: Labeled contours C_i (intensity = elevation)

Output: Interpolated height grid A

Structures:

- G 4-connected gray-scale dilation/erosion structuring element (Fig. 2b)
- A Accumulator Image (init with C_i , else zero)
- T, U, V, W, X - Work Images
- M maximum label/height

while $\min(A) = 0$ **do**

W \leftarrow A {T, W = A with inter-contour spaces set to M}

repeat

W \leftarrow W[M] \circledast G {Grayscale erode W until
until $\max(W) < M$ all M pixels are gone}

U \leftarrow (W \circledast G) \cap T {Dilate & mask to overlap}

V \leftarrow W XOR U {Overlap V=medial line(s)}

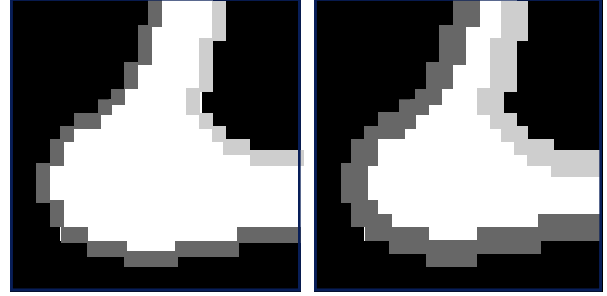
for all V > 0

X \leftarrow M {X \leftarrow medial line mask}

W \leftarrow [(U+W) \cap X]/2 {Get medial line height}

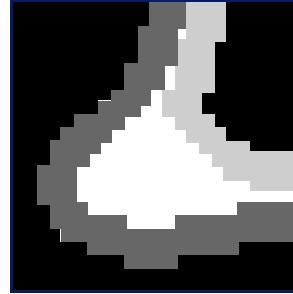
A \leftarrow A \cup W {A \leftarrow new medial pixels}

Input to the algorithm consists of the original contours C_i all labeled by height. The algorithm terminates when all regions in the accumulator array A are filled with a connected set of interpolated values. The algorithm begins by copying contour lines (initialized in A) into a work buffer W and mapping intercontour pixels to high values for subsequent erosion. It then erodes the contour lines into the intercontour space until the eroded lines meet. This is almost equivalent to dilating the contours into the intercontour space with the exception that when a choice between a larger or smaller value must be made, the smaller value is chosen. This is done because the next step favors the higher pixel values and therefore chooses a medial line closer to the center than does using dilation in both instances. After the intercontour space is filled, W is dilated once and placed in a temporary buffer U. The overlap (i.e. difference) between the two buffers contains the medial lines which

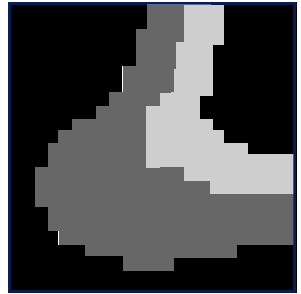


(a) Original labeled contours with intercontour space, S, (white).

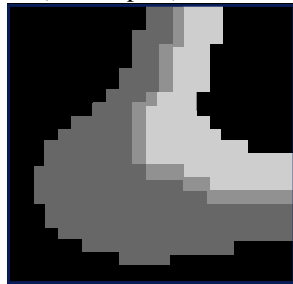
(b) Grayscale erosion of S (first pass). Note: only S is processed.



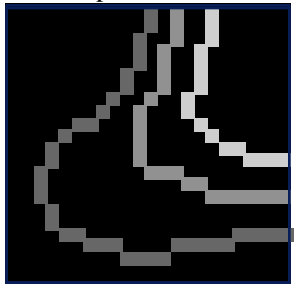
(c) Grayscale erosion of S (second pass).



(d) Erosion of S is complete.



(e) Dilation of intercontour space XORed with (d) gives overlap = medial line.



(f) Label for medial line = average of original contours.



(g) Intercontour spaces are relabeled for next iteration.

Figure 4. (a) Original contours labeled by height. Intercontour space S is labeled white. (b) Grayscale erosion of S erodes S and dilates original contours in the direction of S since only S is processed. (c) Original contours begin to meet with second erosion of S. (d) Erosion of S is complete. (e) Position of medial line is at collision front in figure (d). (f) Medial line is output to accumulator array = interpolated grid. Label = average of original contour labels. (g) Relabeling of new intercontour spaces allows process to be repeated until entire intercontour space is filled with interpolated values.

are extracted using an XOR operation and stored in work image V. The elevations are then computed and the medial lines are added to the accumulator.

It is worth pointing out that all interpolated values are calculated simultaneously for several families of pixels (i.e. contours) since the morphological operations are pipelined through all pixels in the image, targeting specifically the intercontour space pixels.

The process is illustrated in Figure 5 for two simple nested contours. Figure 5 also shows three medial lines resulting from the first two iterations. The interpolated height grid is shown in Figure 6. Figure 7 shows the height grid rendered using an algorithm for polygonalization of the discrete surface [17] (GI '94). The polygon rendering shows some roughness at the base and the top due to the discrete nature of the height grid and the corresponding polygonal approximation. This could be

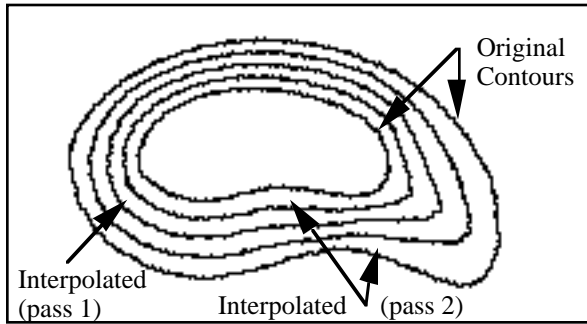


Fig. 5. Interpolation of 3 medial lines (2 passes) from two simple nested contours.

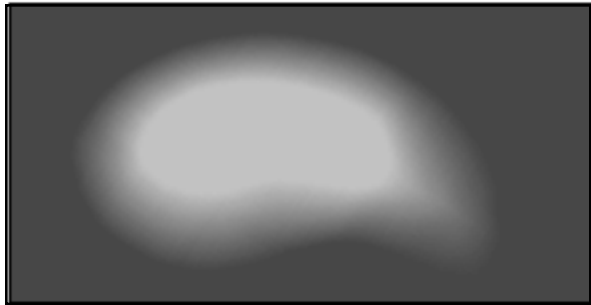


Fig. 6. Complete densely interpolated height grid.

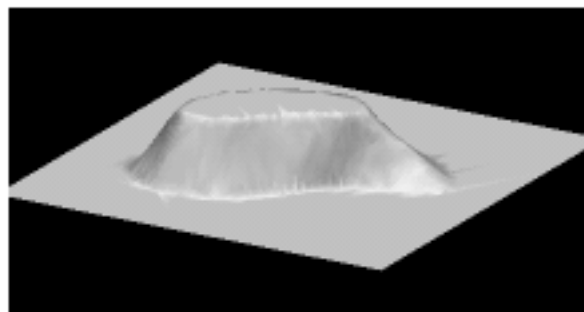


Fig. 7. Polygonal rendering of height grid in Fig. 6.

overcome by finer (perhaps) subpixel polygonalization of the discrete surface. However, that is not the objective of this paper; the polygon rendering is included here and in subsequent examples to demonstrate the surface geometry associated with the original contours.

Figures 8 and 9 illustrate application of the algorithm to multiple nested contours with the corresponding surface rendering. Note the “ghosts” associated with the original contours and their respective levels. This illustrates the independent processing of each contour interval since each yield a different slope based on the (x,y) distance as well as the difference in height (z) between contours. If the medial line is identical to the medial axis, the slope is linear. If not, (i.e. spines exist) the slope or surface between two contours has negative curvature in the region of the spine, as illustrated in the top third of Figure 10. (Perhaps this is appropriate for terrain.)

One of the strong features of the algorithm is that branching contours are handled automatically without any modification to the algorithm. This is illustrated for a simple set of nested contours in Figures 10 and 11. The height of the saddle between the two contours is a function of the distance between contours and the distance to the surrounding contour. As with slopes in general this need not be linear. In fact, the depth of the saddle or the pitch of the slope could be attenuated simply through a lookup table.



Fig. 8. Multiple nested contours.

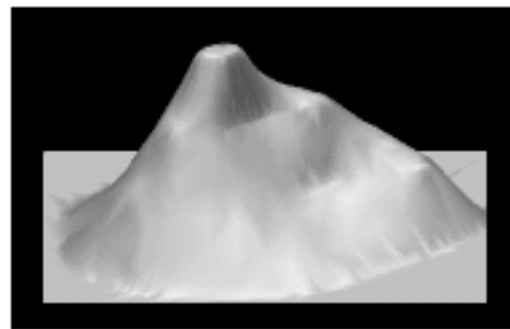


Fig. 9. Polygonal rendering for Fig. 8.

The algorithm also handles overlapping contours in a general way. Figure 12 shows two overlapping contours (black) with the interpolated medial line resulting from the first iteration (gray). In this case the initial contours are simply taken “as is.” That is, no effort was made to register or align the contours. Figure 13 shows two overlapping contours which were registered by their center of gravity prior to interpolation. Registration is appropriate for minimally overlapping contours and essential if the contours do not overlap at all. The resulting interpolated contour (gray) is superimposed by offsetting it half way between the corresponding centers of gravity. The algorithm also produces reasonable results for contours which overlap and branch as evidenced in Figure 14. The indentations in the interpolated (gray) contour correspond to and agree with the branching contours (B) and become more pronounced as subsequent interpolated contours between the gray line and B gravitate towards B.

4. Results and Discussion

Figures 15-17 demonstrate application of the algorithm to a set of contours extracted from a topographic map. The height grid (DEM) in Figure 16 was created by interpolating between the isocontour lines in Figure 15. The corresponding surface rendering in Figure 17 demonstrates that the algorithm performs well on real world data as well. Figure 18 is a height grid of Mount Rundle (near Banff, of course) computed from a topographic map of the region. Polygonalization and rendering for Figure 18 can be seen in [17].



Fig. 10. Nested branching contours.

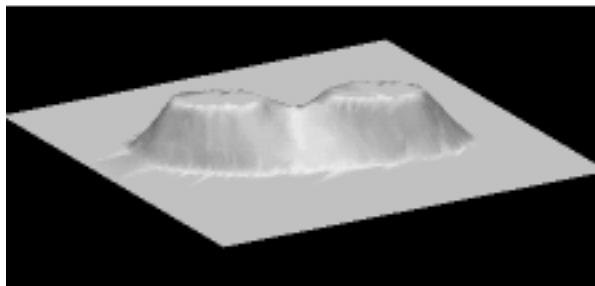


Fig. 11. Polygonal rendering for branching contours.



Fig. 12. Overlapping contours (black). Interpolated medial line (gray).

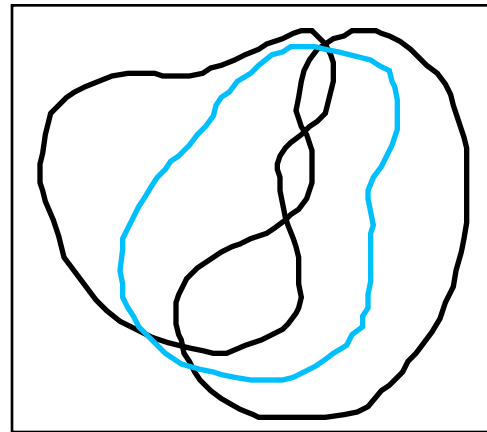


Fig. 13. Overlapping contours registered to produce interpolated medial line (gray).

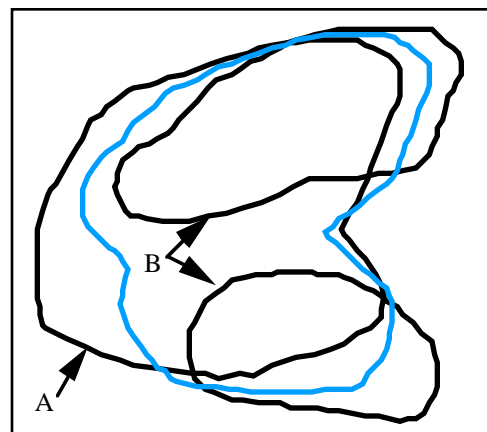


Fig. 14. Overlapping and branching contours A and B. A is at one level and B (two separate contours) at another level. The gray contour is the interpolated result.

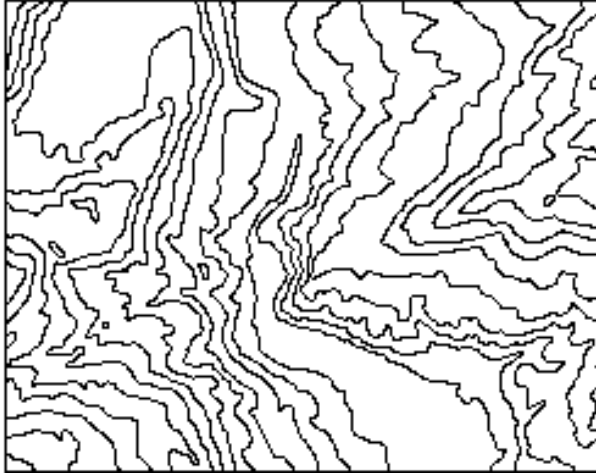


Fig. 15. Isocontour lines from a topographic map of the mouth of Provo Canyon, Utah.

As an additional illustration of robustness the algorithm was applied to digital elevation contours of Taiwan (Fig. 19). The resulting DEM (Fig. 20) preserves terrain features with striking detail.

The 500x400 DEM in Fig. 16 was created in about 2 minutes using a MATROX Image Series board hosted by a 386 processor. The 1024x1024 DEMs in Figures 18 and 20 were generated in about 5 minutes.

The algorithm has also been successfully applied to medical imaging. Figure 21 shows contours interpolated from cross-sectional outlines of the left ventricular chamber of the heart derived from Cine' CT scans. Unlike the nested isocontours used to generate terrain models, the left ventricular contours demonstrate both overlapping and branching situations. However, these present no special problems because of the general nature of the morphological interpolation as illustrated above. The resulting 3D surface of the left ventricle in Figure 22 helps to substantiate this claim.

Specific contributions of this research to applications which require generation and use of an arbitrarily large three-dimensional (3D) data base include:

1. Automated vs. manually-assisted creation of the 3D database
2. Interpolation of the height grid directly from the contour image in image space without the need to explicitly extract contour data.
3. No additional data structures necessary for intermediate representation or storage of contour data
4. It makes essential use of contour morphology.

Several computational advantages also emerge as an inherent part of the algorithm:

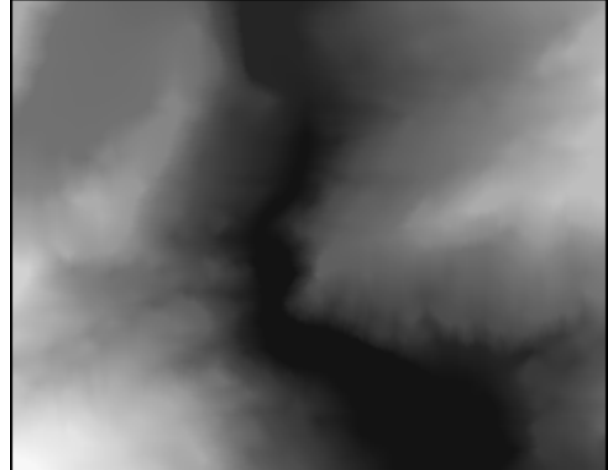


Fig. 16. DEM created by interpolating between isocontour lines in Figure 15.

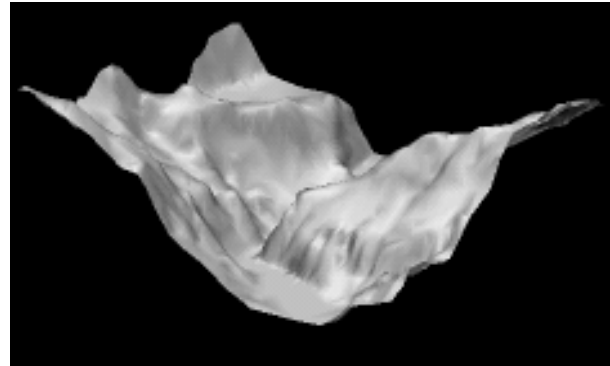


Fig. 17. Polygonalized rendering of DEM in Fig. 16.

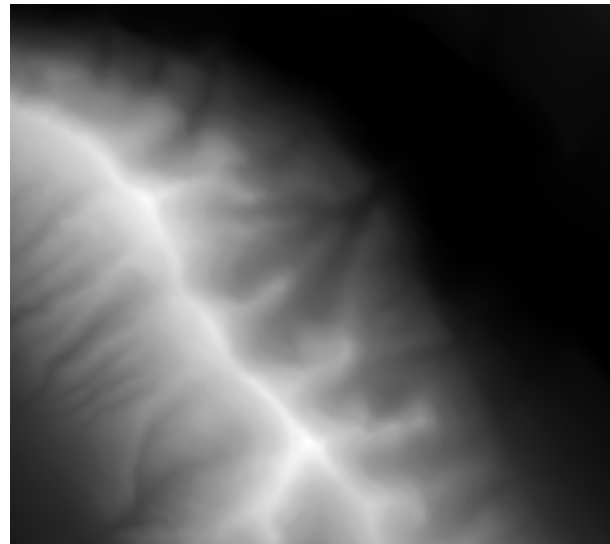


Fig. 18. DEM of Mt. Rundle (Banff, Alberta) generated by interpolating between isocontour lines.



Fig. 19. Digital isocontours of Taiwan.

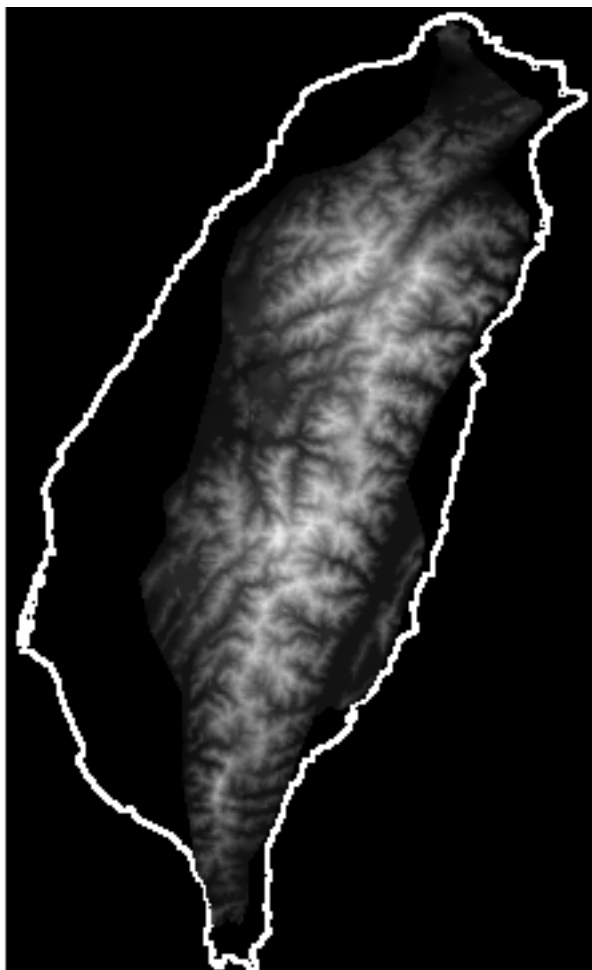


Fig. 20. DEM of Taiwan obtained by interpolating contours in Fig. 19. Coastline inserted for reference.

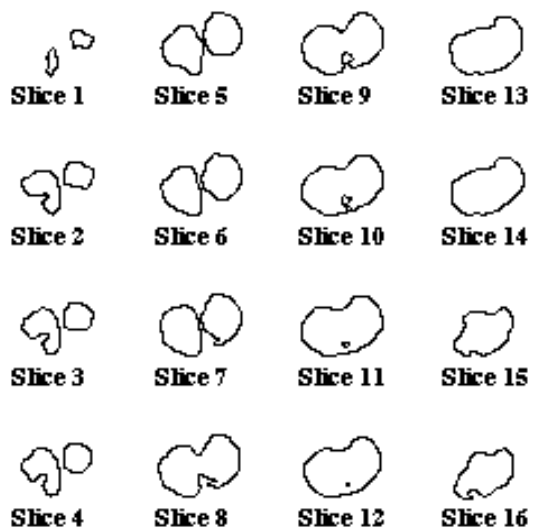


Fig. 21. Interpolated left ventricular contours.

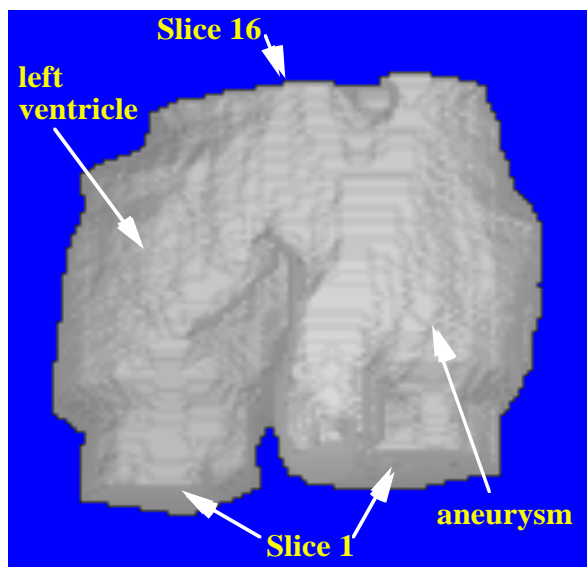


Fig. 22. 3D surface rendering of interpolated contours.

1. Contour intervals are independent of each other and can therefore be processed in parallel.
2. Recursive decomposition of each contour interval into subintervals means each contour interval can be interpolated with $\sim \log_2 n$ operations (where n = width of the contour interval). This also means that the number of parallel operations increases from 1 to 2^n for each successive subinterval, at which point the contour interval is fully interpolated.
3. The algorithm is inherently parallel since for an $N \times N$ image an average of $2N$ points are interpolated simultaneously (ie. as a family of contour/isoline points).

5. Conclusion

Computational advantages of morphological interpolation have been presented and discussed. Other (application) advantages might include the use of contours as a highly compact/encoded representation of terrain (for an on-board data base for flight simulation). Future work will include experimentation with new/different structuring elements to distribute and balance 4-connected pixel propagation. Application to overlapping and unnested nonoverlapping contours will also be investigated. Efforts to find a transform which will generate a true (spineless) "mid"-axis transform (rather than medial axis) may be useful. One approach to producing smoother junctures (i.e. minimize contour "ghosts") would be to propagate, in a weighted sense, the gradient of the interpolated values from one contour interval into the next.

References

1. P. Yoeli. Computer executed production of a regular grid of height points from digital contours. *The American Cartographer*, 13(3), 219-229, 1986.
2. D. Legates and C. Willmott. Interpolation of point values from isoline maps. *An American Cartographer*, 13(4), 308-323, October, 1986.
3. J. Nelson, et al. Enhanced contour-to-grid interpolation procedures used in producing high-fidelity digital terrain models. Technical Papers 1988 American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing Annual Convention, 1, 172-176.
4. R. Rinehart and E. Coleman. Digital elevation models produced from digital line graphs. Technical Papers 1988 American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing, 2, 291- 299, 1988.
5. E. Mortensen, B. Morse, and W. Barrett. Adaptive Boundary Detection Using "Live-Wire" Two Dimensional Dynamic Programming, *IEEE Computers in Cardiology*, Durham, North Carolina, October, 1992.
6. T. Sederberg, and E. Greenwood. A Physically Based Approach to 2-D Shape Blending, *SIGGRAPH Proceedings*, pp. 25-34, 1992.
7. F. Verbeek, et. al. 3D Base: A Geometrical Data Base System for the Analysis and Visualisation of 3D-Shapes Obtained from Parallel Serial Sections Including Three Different Geometrical Representations, *Computerized Medical Imaging and Graphics*, Vol. 17, No. 3, pp. 151-164, 1993.
8. N. Kehtarnavaz, L. Simar and R. De Figueiredo. A syntactic/semantic technique for surface reconstruction from cross-sectional contours. *Computer Vision, Graphics, and Image Processing*, 42, 399-409, 1988.
9. S. Ganapathy and T. Dennehy. A new general triangulation method for planar contours. *Computer Graphics*, 16(3), 69-75, July, 1982.
10. H. Fuchs, Z. Kedem, and S. Uselton. Optimal surface reconstruction from planar contours. *Communications of the Association for Computing Machinery*, 20(10), 693-702, October, 1977.
11. R. Haralick and L. Shapiro. *Computer and Robot Vision*. Vol. I, Addison Wesley, 1992.
12. J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
13. Mortensen, E. and Barrett, W. Morphological Interpolation Between Contours, *SPIE Proceedings of Medical Imaging VII*, Newport Beach, February, 1993.
14. Petersen, S., Barrett, W., and Burton, P. A New Morphological Algorithm for Automated Interpolation of Height Grids from Contour Images. *SPIE/SPSE Symposium on Electronic Imaging: Science and Technology*, Santa Clara, February, 1990.
15. Blum, H. Biological shape and visual science. *Journal of Theoretical Biology*, 38(1), 205-287, 1973.
16. Zhang, T., Suen, C. A fast parallel algorithm for thinning digital patterns. *Image Processing and Computer Vision*, 27(3), 326-329, March, 1974.
17. Taylor, D. and Barrett, W. An Algorithm for Continuous Resolution Polygonalizations of a Discrete Surface. *GI '94 (This Proceedings)*.
18. S. Raya and J. Udupa, "Shape-based Interpolation of Multidimensional Objects," *IEEE Trans. on Medical Imaging*, 9(1) :32-42, 1990.