Volume Graphics (2006)
T. Möller, R. Machiraju, T. Ertl, M. Chen (Editors)

Contour-Based Surface Reconstruction using Implicit Curve Fitting, and Distance Field Filtering and Interpolation

Jeffrey Marker[1] Ilya Braude[1] Ken Museth[2] David Breen[1]

[1]Drexel University, Philadelphia, PA, USA [2] Linköping University, Norrköping, Sweden

Abstract

This paper presents a volumetric approach to reconstructing a smooth surface from a sparse set of parallel binary contours, e.g. those produced via histologic imaging. It creates a volume dataset by interpolating 2D filtered distance fields. The zero isosurface embedded in the computed volume provides the desired result. MPU implicit functions are fit to the input contours, defined as binary images, to produce smooth curves with controllable error bounds. Full 2D Euclidean distance fields are then calculated from the implicit curves. A distance-dependent Gaussian filter is applied to the distance fields to smooth their medial axis discontinuities. Monotonicity-constraining cubic splines are used to construct smooth, blending slices between the input slices. A mesh that approximates the zero isosurface is then extracted from the resulting volume. The effectiveness of the approach is demonstrated on a number of complex, multi-component contour datasets.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Imaging technology, e.g. MRI, CT and histology, is now widely used in medicine, science and engineering to study the internal structures of a variety of objects and specimens. This technology produces 3D sampled data that can be interpreted as a stack of 2D slices cutting through the studied object/specimen, with each slice normally being represented by a raster image. Frequently the process of isolating, segmenting and identifying specific structures in the slices involves a manual (or semi-automatic at best) process of delineation that produces contours around the structure of interest. While CT and MRI scans can now produce 3D datasets with isotropic sampling, i.e. the same sampling resolution in X,Y, and Z directions, this is generally not the case for histologic imaging, where a sample (e.g. a tumor) is physically sliced and a digital image is taken of the exposed face. In this imaging modality it is not uncommon to have sampling ratios of $10:1$ or more. This means that the physical distance between pixels in a slice is 1/10 the distance between the slices themselves.

It is often important to visualize the 3D structures present inside a scanned dataset. Attempting to view the stacked 2D images or contours can be difficult and error-prone. Therefore techniques are needed that take a set of parallel contours and produce smooth 3D models that interpolate those contours. The work described here addresses the general problem of contour-based surface reconstruction, but more specifically focuses on addressing the challenges that arise in histologic images of highly complex structures. Those challenges arise when generating smooth 3D models from highly anisotropic input contours that inherently contain noise from the 2D delineation/segmentation process. The final hurdle to successful reconstruction is the sheer complexity of the structures to be modeled.

We have developed a computational pipeline that produces smooth surface reconstructions from a set of parallel binary contours, which also addresses some of the unique challenges associated with histologic imaging. Our approach produces an isosurface embedded in a volume dataset by first calculating distance fields in the individual 2D slices. Blending slices are computed between the input contours via spline interpolation of associated pixels in neighboring input slices. The zero isosurface embedded in the resulting volume provides the desired reconstruction.

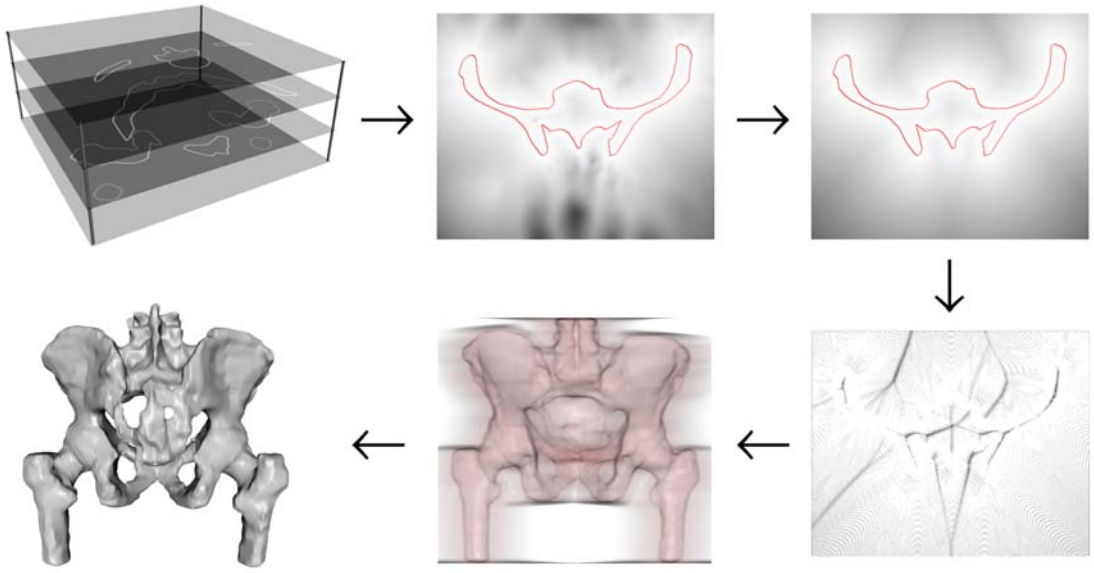

Figure 1: *Overview of the volumetric reconstruction process. Input is a set of contours represented as binary images. MPU implicit curves are fit to the contours. A Euclidean distance field is generated from the narrow band around the implicit curve. The field is filtered to remove medial axis discontinuities. The filtered fields are interpolated to produce a volume dataset. A mesh of the zero level set is extracted from the volume.*

The complete pipeline contains several stages and is presented in Figure 1. The input to the process is provided as binary images where white pixels represent the contour curves. The centers of the contour (white) pixels are interpreted as points in 2D and a Multi-level Partition of Unity (MPU) implicit curve (i.e. a 2D field whose zero level set is the curve) is approximately fit to these points. The narrow band around the MPU curve is swept out by a fast marching method to produce a 2D Euclidean distance field. The medial axis discontinuities inherent in all Euclidean distance fields are smoothed with distance-dependent Gaussian filtering. A volume dataset is produced via monotonicity-constraining spline interpolation of pixels across neighboring distance fields. The volume may be visualized either via direct volume rendering or by generating a mesh that approximates the resulting isosurface.

1.1. Contributions

Each stage of the pipeline addresses and solves a specific issue during the reconstruction process. A volumetric approach allows us to generate a reconstruction from any collection of contours, regardless of the number of contours in the slices and the complexity of their shape. As with most volumetric methods, it readily solves the branching problem as long as adjacent contours overlap. Fitting an MPU curve to the input contours smooths the noise and aliasing inherent in the delineation process and the binary images themselves. The resulting curve has user-controllable resolution and sub-pixel error bounds [Bra05]. Distance-dependent Gaussian filtering reduces or removes the medial axis discontinuities in the calculated Euclidean distance fields, that would otherwise create unwanted artifacts on the resulting surfaces, while leaving the MPU implicit curve untouched. These discontinuities have been ignored by previous volumetric approaches [Lev86, RU90, JC94]. Pixel-to-pixel spline interpolation allows us to always create smooth blendings between widely spaced input contours. Using monotonicity-constraining splines removes undesirable artifacts from the final reconstructions due to overshoot problems.

Given the capabilities of our approach, it provides the following contributions over previous work:

- easily copes with complex contour geometry and arbitrary numbers of components in each slice,
- approximates noisy, binary input contours with a smooth curve with controllable error bounds,
- removes artifacts caused by medial axis discontinuities,
- and produces smooth models from highly anisotropic input data.

1.2. Previous Work

The previous work on contour reconstruction mostly falls into one of two categories: contour stitching that is based on explicit surface representations and volumetric methods that employ implicit representations.

1.2.1. Contour Stitching

The contour stitching approach to surface reconstruction attempts to generate a surface by connecting the vertices of adjacent contours in order to produce a mesh that passes through all contours. These approaches generally need to address the correspondence (how to connect vertices between contours), tiling (how to create meshes from these edges) and branching (how to cope with slices with different numbers of contours) problems.

Keppel [Kep75] and Fuchs et al. [FKU77] described the first algorithms for creating polygonal meshes from a series of contours. The Fuchs work defines the best reconstructed surface as the one with minimal surface area. Many papers have offered incremental improvements to these seminal efforts. Several solutions to the correspondence problem have been proposed, e.g. those based on parameterization of the contours [GD78], contour decomposition [EPO91], Minimum Spanning Trees [MSS92], Angular Bisector Networks [OPC96], medial axes [KSS00] and partial curve matching algorithms [BST00]. Boissonnat [Boi88] utilizes Delaunay triangulation to cope with branching surfaces. Geiger [Gei93] proposed a geometric closeness measure to improve on this approach. Bajaj et al. [BCL96] provide a unified approach to solving the correspondence, tiling and branching problems by imposing three constraints on the surface when deriving the reconstruction rules. Johnstone et al. [JS95] describe a method for creating Bezier surfaces from contours with cylindrical topology. Fujimura and Kuo [FK99] use isotopic deformations to create non-self-intersecting surfaces from nested contours. Hormann et al. [HSS03] smoothly stitch together nested contours using Hermite interpolation.

1.2.2. Volumetric Methods

Levin [Lev86] presents the seminal volumetric approach to surface reconstruction from a series of parallel contours. Given a distance field for each contour, the 2D fields are stacked and interpolated in the z-direction with cubic B-splines. The reconstructed surface is extracted from the resulting volume as the zero isosurface, and in general will only be as smooth as the distance field, *i.e.* C^0. Raya and Udupa [RU90] extend Levin's approach to time-varying datasets. Jones and Chen [JC94] suggest that Voronoi diagrams be used to minimize the computation needed for calculating the 2D distance fields. Barrett et al. [BMT94] recursively apply morphological operators (dilation and erosion) to contour images in order to interpolate intermediate gray level values. Cohen-Or et al. [COL96, COLS96] introduce the concept, without supporting results, of creating a 3D object from contours by morphing one contour into the next using warp-guided distance field interpolation. Chai et al. [CMN98] present a gradient-controlled partial differential equation method for producing C^1 continuous surfaces from *nested* contours. Nilsson et al. [NBM05] utilize 2D level set morphing with cross-contour velocity continuity to sweep out smooth surfaces from contour images.

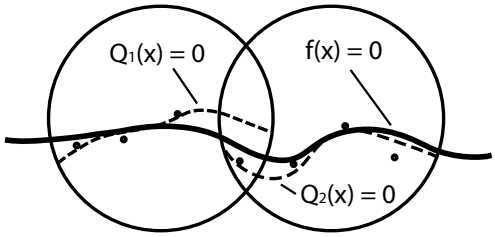

Figure 2: *Two local approximations (dashed) are blended to form the global MPU function (solid).*

2. Volumetric Surface Reconstruction

Our volumetric surface reconstruction approach consists of several stages. They include contour smoothing, distance calculations, distance field filtering, field interpolation, and mesh extraction.

2.1. Contour Smoothing

The first step in our contour reconstruction approach is the generation of smooth 2D curves that closely approximate the binary input contours. A 2D distance field is then generated from the curve. A 2D distance field consists of a 2D array of distance values where each entry contains the signed shortest distance to the closed contour from that location. Traditional distance field generation methods have calculated distances between pixel centers in input images. These methods limit the number of distance values that can exist in the immediate proximity of the contour, producing aliased results. We utilize an implicit model to approximate the contours and to generate high-resolution distance fields. An implicit model provides two important benefits. First, distances are calculated to the implicit function that represents the contour, instead of to the centers of the pixels. Second, implicit functions allow us to approximate the contour to a desired level of accuracy and smoothness [Bra05].

We use a Multi-level Partition of Unity (MPU) implicit model [OBA*03] to define the implicit curve because of its robustness, controllability and flexibility. When using MPU implicits, contour pixel coordinates are interpreted as points in $\mathbb{R}^2$, i.e. a point set. The MPU function operates on the point set and reconstructs a curve that approximately fits to the input data. The function is composed of overlapping local functions that are blended together, summing to one (partition of unity). A partition of unity is a set of non-negative compactly supported functions ω_i where $\sum_i \omega_i \equiv 1$, on a bounded Euclidean domain Φ. The global function is then

$$f(x) = \sum_i \omega_i(x) Q_i(x), \tag{1}$$

where $Q_i(x)$ is a local approximation function, see Figure 2. Each ω_i is generated by

$$\omega_i(x) = w_i(x) / \sum_{j=1}^{n} w_j(x), \tag{2}$$

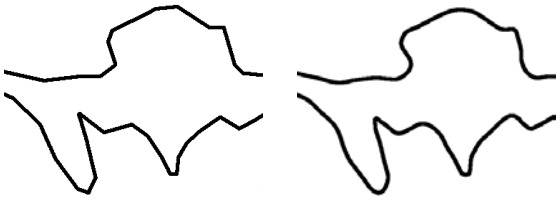

Figure 3: *(left) Input contour represented as a binary image. (right) Resulting MPU implicit curve.*

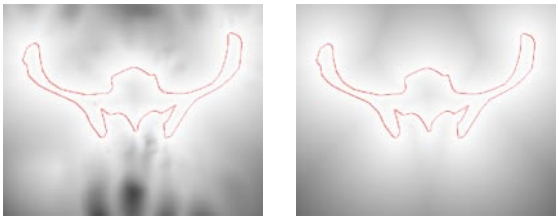

Figure 4: *(left) MPU implicit function fit to contour data. (right) Euclidean distance field extended out from the narrow band of the MPU field surrounding the implicit curve.*

where $\{w_i\}$ is a set of nonnegative compactly supported weight functions such that $\Phi \subset \cup_i \text{supp}(w_i)$. In the current MPU implicits implementation, each weight function $w_i(x)$ is a quadratic B-spline.

MPU implicits use an adaptive quadtree-based subdivision scheme in order to selectively refine areas of higher detail. Two parameters control the subdivision process. The support radius R for the weight functions is adjusted until it contains N_{min} data points. Increasing the N_{min} parameter results in fewer local approximations and increased smoothing. The subdivision process is controlled by a tolerance value (*tol*). Lowering the tolerance value increases the level of subdivision and forces a tighter fit to the input data.

The MPU fitting process requires both a point set and a normal associated with each point. Normals are produced by creating a binary mask from the contour, with boundary and interior pixels colored white and the remaining pixels set to black. Gaussian filtering is applied to the mask. The gradient of the filtered image is calculated at each contour pixel, normalized, and negated to point outwards to approximate the normal at that point [YCK92]. Figure 3 presents a portion of a contour represented as a binary image and the smooth implicit curve produced by the MPU-based fitting process.

2.2. 2D Distance Calculations

The MPU function produces a signed field value that provides inside/outside information. The values *inside* are positive, and the values *outside* are negative. Unfortunately these field values only approximate Euclidean distance near the contour. See Figure 4 (left). Therefore, we use MPU implicits to generate distance values only in a narrow band around the original data points. The narrow band of signed distances is then extended out to produce a complete 2D Euclidean distance field using a fast marching method with a correctness criterion [Mau03, Set96]. See Figure 4 (right). Future work will replace this method with a more efficient fast sweeping method [Zha04] in order to shorten computation times.

2.3. 2D Distance Field Filtering

Euclidean distance fields are C^0 continuous, because they contain medial axis discontinuities. A gradient discontinuity occurs at locations in the field that are equidistant to more than one point on the field's zero level set. The discontinuities in the fields lead to undesirable artifacts, e.g. folds and creases, on the isosurfaces extracted from generated volume datasets. While the discontinuities exist in the interior of the 2D distance fields, they produce discontinuities in the full interpolated 3D field that cross the zero isosurface; thus becoming evident during mesh extraction. In order to remove these artifacts from the distance fields and subsequent surface reconstructions, we filter (i.e. smooth) the individual 2D distance fields generated from the contours before performing the spline interpolation that creates the volume dataset. Applying a Gaussian filter over the whole 2D field would modify the MPU implicit curve. We therefore have developed a distance-dependent Gaussian filter that keeps the MPU implicit curve intact, while smoothing the medial axis discontinuities found a distance away from the curve.

In the continuous domain, a Gaussian filter is controlled by changing σ, the standard deviation of the Gaussian distribution [GW02]. Larger values of σ produce increasingly "blurred" results. In the discrete domain, a stencil radius must be introduced within which the Gaussian function is evaluated. In general, pixels beyond 3σ from the currently processed pixel contribute so little to the result that they can effectively be ignored; hence the smoothing is controlled by varying sigma, and defining the stencil radius as $3 * \sigma$.

When smoothing the individual distance fields, each pixel is filtered according to its distance value. No filtering is done within a few pixels of the implicit curve (zero level set). At a prescribed distance filtering begins with a small σ value. σ is then ramped up to a constant value as distance increases. Distance-dependent Gaussian smoothing is controlled by four parameters:

- d_{min}: No filtering is performed below this distance.
- d_{max}: The distance value at which to apply maximum filtering.
- $\sigma_{min} \rightarrow \sigma_{max}$: Minimum and maximum sigma values.

The following uses a shifted and scaled cosine function to smoothly increase σ_{min} at d_{min} to σ_{max} at d_{max},

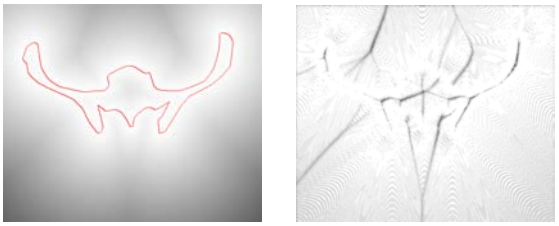

Figure 5: *(left) Filtered Euclidean distance map that removes the medial-axis discontinuity. (right) A difference image calculated from Figure 4 (right) and Figure 5 (left) that highlights the pixels that have been modified by filtering.*

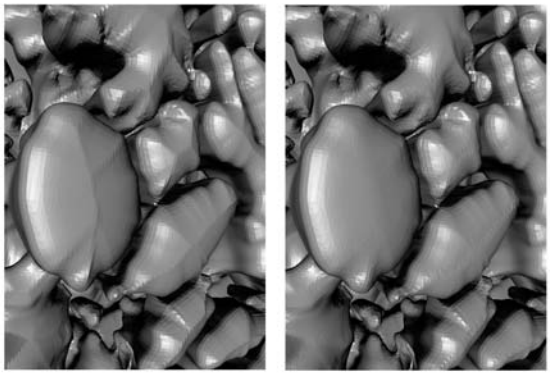

Figure 6: *Surface artifacts removed by filtering the medial-axis discontinuities in the 2D distance fields. (left) Before (right) After.*

$$\sigma = \begin{cases} 0 & d <= d_{min} \\ \left(\left(\left(-cos\left(\frac{(d-d_{min})}{(d_{max}-d_{min})} * \pi\right)+1\right)/2\right) * \right. & \\ \left.\left(\sigma_{max}-\sigma_{min}\right)\right)+\sigma_{min} & d_{min} < d < d_{max} \\ \sigma_{max} & d >= d_{max} \end{cases} \tag{3}$$

The values of d_{min} and σ_{min} must be chosen with care. From our experience σ_{min} values below 0.5 introduce sampling noise into the distance fields. Additionally the values of d_{min} and σ_{min} must be set to ensure that the filtering stencil does not include distance values with differing signs. With $\sigma_{min} = 0.5$, the stencil will have a radius of 2. Therefore choosing a d_{min} of 3 ensures that the stencil does not include distance values on both sides of the zero level set. Given these constraints, the following parameters were used in all examples shown in this paper: $d_{min} = 4$, $d_{max} = 15$, $\sigma_{min} = 0.5$, $\sigma_{max} = 3$. We found a σ_{max} of 3 provided sufficient smoothing of the medial axis. Setting d_{max} to 15 prevents the stencil radius (9) from crossing the zero level set.

Figure 5 presents a distance-dependent smoothed field. The before-and-after difference between the fields as shown in Figure 4 (right) and Figure 5 (left) is difficult to see, so a difference image, which is produced by subtracting one image from the other, is included to highlight the discontinuities that have been removed by filtering. Figure 6 (left) contains the unwanted creases produced on a reconstructed surface by the medial axis discontinuities present in the 2D distance fields. Figure 6 (right) demonstrates that the creases can be minimized by 2D distance field filtering. Note that the outlines of the 3D shapes do not change after filtering.

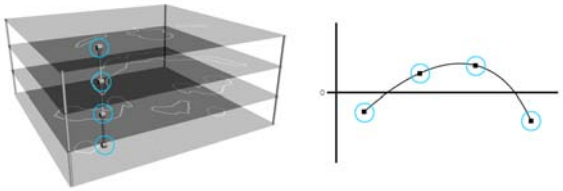

Figure 7: *A column of pixel values are interpolated with a spline that is evaluated to produce the pixel values for intermediate, blending slices.*

2.4. Field Interpolation

Once the filtered distance fields have been computed a 3D (volumetric) representation of the desired reconstructed geometry can be produced via 1D, pixel-by-pixel interpolation of the 2D images. See Figure 7. For each column in the volume, a 1D cubic spline is constructed to interpolate distance values as a function of their vertical (z) location in the volume. Once a spline is constructed for a column of input pixels, it is evaluated at different z locations to generate the values for that pixel in the intermediate slices of the volume. Doing this for all pixels creates new slices between the input data that blend the contour curves.

We have experimented with three types of interpolating cubic splines: Catmull-Rom, natural cubic, and Hermite cubic with monotonicity constraints. Catmull-Rom splines utilize Hermite basis functions, interpolate their control points and provide local control over their shape [Far02]. While they are fast to compute, they are only C^1 continuous. The C^2 discontinuity was evident in some of the reconstructions made with these splines. Additionally, they suffer from an overshoot problem near control points that have rapid changes in their data values (See Figure 8.), resulting in artifacts such as the lip shown in Figure 9. Natural cubic splines provide C^2 continuity by sacrificing local control, a feature not necessarily needed for our application. They unfortunately also create unwanted artifacts due to overshoot, and are more expensive to compute than Catmull-Rom splines.

Hyman [Hym83] incorporates monotonicity constraints into Hermite cubic splines by enforcing slope constraints at the spline's data values. This guarantees that the spline will be locally monotone, but sacrifices a guarantee of C^2 continuity, as seen in Figure 8. Ultimately, it was deemed more important to remove the overshoot artifacts, as seen in Figure 9, than the more subtle artifacts that may be produced by C^2 discontinuities.

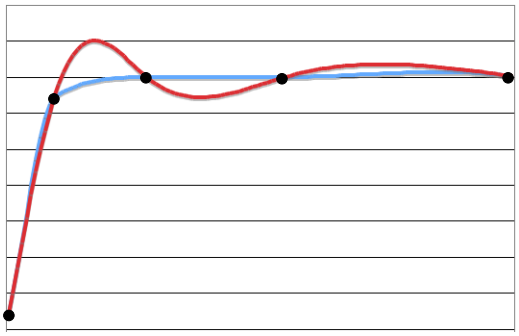

Figure 8: *Fitting a cubic spline without (red) and with (blue) monotonicity constraints to five data points.*

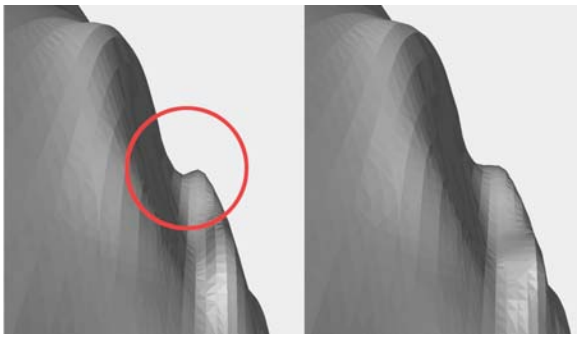

Figure 9: *(left) Undesirable surface artifact produced by the overshoot of interpolating splines. (right) Using a monotonicity constrained spline removes the artifact.*

2.5. Mesh Extraction

In the final stage of the reconstruction process a polygonal mesh representing the reconstructed surface is extracted from the volume produced in the field interpolation stage. We currently use the Marching Cubes algorithm [LC87] followed by mesh simplification [GH97, SZL92] to produce the final reconstructed surface. Future plans include utilizing techniques capable of extracting an adaptive mesh directly from the volume [GCBB01].

3. Results

Our volumetric reconstruction approach was applied to a number of datasets including contours extracted from scans of a human pelvis and jawbone, as well as segmentations of breast cancer tumors. The pelvis and jawbone datasets were provided in a polyline format and were rasterized to produce binary images. The pelvis contours were converted into 2,000 $\times$ 2,000 resolution images so that the MPU curve fitting process could properly capture important features in the data. The resulting MPU implicit field was down-scaled to 500 $\times$ 500 before the fast marching stage that calculates the full Euclidean distance field. The jawbone dataset was converted into 500 $\times$ 500 binary images and was processed

	# of slices	Input resolution	xy:z	Output resolution
P	34	2,000×2,000	14:1	500×500×458
J	48	500×500	8:1	500×500×393
T0	9	489×483	10:1	489×483×101
T1	4	870×616	63:1	435×308×191

Table 1: *Dataset Information. P - pelvis (Figure 11), J - jawbone (Figure 10), T0 - tumor (Figure 12), T1 - tumor (Figure 13)*

	N_{min}	tol
P	100	15
J	250	10
T0	300	10
T1	300	10

Table 2: *MPU parameters used during implicit curve fitting.*

	MPU	Dist	Fltr	Intrp	Mesh	Total
P	2012	272	98	71	170	2623
J	199	144	132	73	137	685
T0	811	228	165	104	288	1596
T1	76	12	18	38	119	263

Table 3: *Computation times (in secs) on an Apple 2.0 GHz G5 with 3 GB of RAM. The reconstruction stages are MPU implicit curve fitting, Euclidean distance calculation, distance field filtering, inter-slice spline interpolation, and mesh extraction and decimation.*

at this resolution for the remainder of the computational pipeline. The tumor datasets were provided as segmented images (See Figures 12 and 13), and the boundary pixels of each region were extracted to produce the input contours. Additional information about the input datasets and resulting volume datasets is listed in Table 1. The parameters used during the implicit curve fitting stage are listed in Table 2.

The results from the reconstruction process are presented in Figures 10, 11, 12 and 13. The computation times for each stage of the reconstructions are given in Table 3. The resulting surfaces demonstrate that our approach faithfully reconstructs the objects defined by the contours, producing smooth, high-resolution models. The tumor datasets are potentially challenging, given the number and complexity of their structures, as well as the small number of slices. The computation times for our approach (pelvis - 43.7 minutes, jawbone - 11.4 minutes, tumor0 - 26.6 minutes, tumor1 - 4.5 minutes), as listed in Table 3, are somewhat long, but not excessive, given the complexity of the input and the quality of the output.

Currently the main limitation of our approach is that contours must overlap in order to be connected between successive slices. This shortcoming may be addressed by utilizing warping transformations based on user-defined correspon-

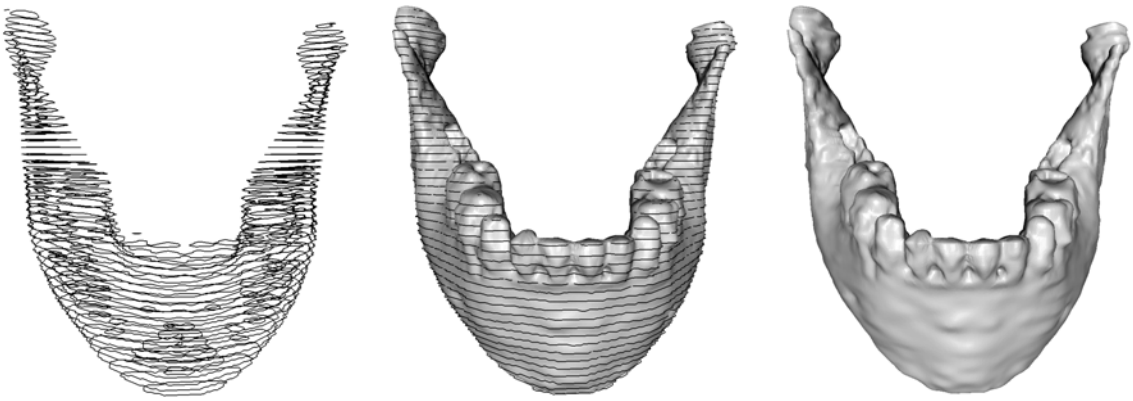

Figure 10: *48 jawbone contours. Reconstruction with and without contours displayed.*

dences [COL96, COLS96] when calculating the inter-slice interpolations.

4. Conclusion

We have presented a volumetric approach to reconstructing a smooth surface from a sparse set of parallel contours. It creates a volume dataset by interpolating 2D filtered distance fields. The zero isosurface embedded in the computed volume provides the desired result. MPU implicit functions are approximately fit to the input contours, defined as binary images, to produce smooth curves with controllable error bounds. Full 2D Euclidean distance fields are then calculated from the implicit curves. A distance-dependent Gaussian filter is applied to the distance fields to smooth their medial axis discontinuities. Monotonicity-constraining cubic splines are used to construct smooth, blending slices between the input slices. A mesh that approximates the zero isosurface is then extracted from the resulting volume. We have demonstrated its effectiveness on a number of complex, multi-component contour datasets.

5. Acknowledgments

Special thanks go to Yutaka Ohtake for making his MPU Implicits software publicly available. Additional thanks go to Kai Hormann, Salvatore Spinello, and Anders Sandholm for their contributions to the early stages of this work, as well as to Ola Nilsson for the spline interpolation software and Gunnar Johansson for the direct volume renderings. The pelvis and jawbone datasets were provided by G. Barequet of the Technion. The tumor datasets were provided by F. Garcia and S. Petushi of Drexel's Advanced Pathology Imaging Lab. This work was financially supported by National Science Foundation grant ACI-0083287, a Drexel Synergy grant, and Swedish Research Council grant 617-2004-5017.

This investigation was partially funded under a U.S. Army Medical Research Acquisition Activity; Cooperative Agreement W81XWH-04-1-0419. The content of the information herein does not necessarily reflect the position or the policy of the U.S. Government or the U.S. Army and no official endorsement should be inferred. The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick, MD 21702-5014 is the awarding and administering acquisition office.

References

[BCL96] BAJAJ C., COYLE E., LIN K.-N.: Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing 58* (1996), 524–543.

[BMT94] BARRETT W., MORTENSEN E., TAYLOR D.: An image space algorithm for morphological contour interpolation. In *Proc. Graphics Interface* (1994), pp. 16–24.

[Boi88] BOISSONNAT J.-D.: Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing 44*, 1 (1988), 1–29.

[Bra05] BRAUDE I.: *Smooth 3D Surface Reconstruction from Contours of Biological Data with MPU Implicits*. Master's thesis, Drexel University, Philadelphia, PA, August 2005.

[BST00] BAREQUET G., SHAPIRO D., TAL A.: Multilevel sensitive reconstruction of polyhedral surfaces from parallel slices. *The Visual Computer 16*, 2 (2000), 116–133.

[CMN98] CHAI J., MIYOSHI T., NAKAMAE E.: Contour interpolation and surface reconstruction of smooth terrain models. In *Proc. IEEE Visualization* (1998), pp. 27–33.

[COL96] COHEN-OR D., LEVIN D.: Guided multidimensional reconstruction from cross-sections. In *Advanced Topics in Multivariate Approximation*, Fontanella F., Jetter K., Laurent P.-J., (Eds.). World Scientific Publishing Co., 1996, pp. 1–9.

[COLS96] COHEN-OR D., LEVIN D., SOLOMOVICI A.: Contour blending using warp-guided distance field interpolation. In *Proc. IEEE Visualization* (1996), pp. 165–172.

[EPO91] EKOULE A., PEYRIN F., ODET C.: A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Transactions on Graphics 10*, 2 (1991), 182–199.

[Far02] FARIN G.: *Curves and Surfaces for CAGD*, 5th. ed. Morgan Kaufmann, San Francisco, 2002.

[FK99] FUJIMURA K., KUO E.: Shape reconstruction from contours using isotopic deformation. *Graphical Models and Image Processing 61*, 3 (1999), 127–147.

[FKU77] FUCHS H., KEDEM Z., USELTON S.: Optimal surface reconstruction from planar contours. *Communications of the ACM 20*, 10 (1977), 693–702.

[GCBB01] GAVRILIU M., CARRANZA J., BREEN D., BARR A.: Fast extraction of adaptive multiresolution meshes with guaranteed properties from volumetric data. In *Proc. IEEE Visualization* (2001), pp. 295–302.

[GD78] GANAPATHY S., DENNEHY T.: A new general triangulation method for planar contours. In *Proc. SIGGRAPH '78* (1978), pp. 69–75.

[Gei93] GEIGER B.: *Three-Dimensional Modeling of Human Organs and Its Application to Diagnosis and Surgical Planning*. Tech. Rep. 2105, INRIA, 1993.

[GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *Proc. SIGGRAPH* (1997), pp. 209–216.

[GW02] GONZALEZ R., WOODS R.: *Digital Image Processing*, 2nd. ed. Prentice Hall, Upper Saddle River, NJ, 2002.

[HSS03] HORMANN K., SPINELLO S., SCHRÖDER P.: C^1-continuous terrain reconstruction from sparse contours. In *Proc. Vision, Modeling, and Visualization 2003* (2003), pp. 289–297.

[Hym83] HYMAN J.: Accurate monotonicity preserving cubic interpolation. *SIAM Journal on Scientific and Statistical Computing 4*, 4 (1983), 645–654.

[JC94] JONES M., CHEN M.: A new approach to the construction of surfaces from contour data. *Computer Graphics Forum 13*, 3 (1994), 75–84.

[JS95] JOHNSTONE J., SLOAN K.: Tensor product surfaces guided by minimal surface area triangulations. In *Proc. IEEE Visualization* (1995), pp. 254–261.

[Kep75] KEPPEL E.: Approximating complex surface by triangulation of contour lines. *IBM Journal of Research and Development 19* (1975), 2–11.

[KSS00] KLEIN R., SCHILLING A., STRASSER W.: Reconstruction and simplification of surfaces from contours. *Graphical Models 62*, 6 (2000), 429–443.

[LC87] LORENSEN W., CLINE H.: Marching Cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH* (July 1987), pp. 163–169.

[Lev86] LEVIN D.: Multidimensional reconstruction by set-valued approximation. *IMA Journal of Numerical Analysis 6* (1986), 173–184.

[Mau03] MAUCH S.: *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, Pasadena, California, 2003.

[MSS92] MEYERS D., SKINNER S., SLOAN K.: Surfaces from contours. *ACM Transactions on Graphics 11*, 3 (1992), 228–258.

[NBM05] NILSSON O., BREEN D., MUSETH K.: Surface reconstruction via contour metamorphosis: An Eulerian approach with Lagrangian particle tracking. In *Proc. IEEE Visualization* (2005), pp. 407–414.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.: Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 463–470.

[OPC96] OLIVA J. M., PERRIN M., COQUILLART S.: 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoi diagram. *Computer Graphics Forum 15*, 3 (1996), 397–408.

[RU90] RAYA S., UDUPA J.: Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging 9*, 1 (1990), 32–42.

[Set96] SETHIAN J.: A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Science* (1996), vol. 93 of 4, pp. 1591–1595.

[SZL92] SCHROEDER W., ZARGE J., LORENSEN W.: Decimation of triangle meshes. In *Proc. SIGGRAPH* (1992), pp. 65–70.

[YCK92] YAGEL R., COHEN D., KAUFMAN A.: Normal estimation in 3D discrete space. *The Visual Computer 8*, 5-6 (1992), 278–291.

[Zha04] ZHAO H.-K.: Fast sweeping method for Eikonal equations. *Mathematics of Computation 74* (2004), 603–627.

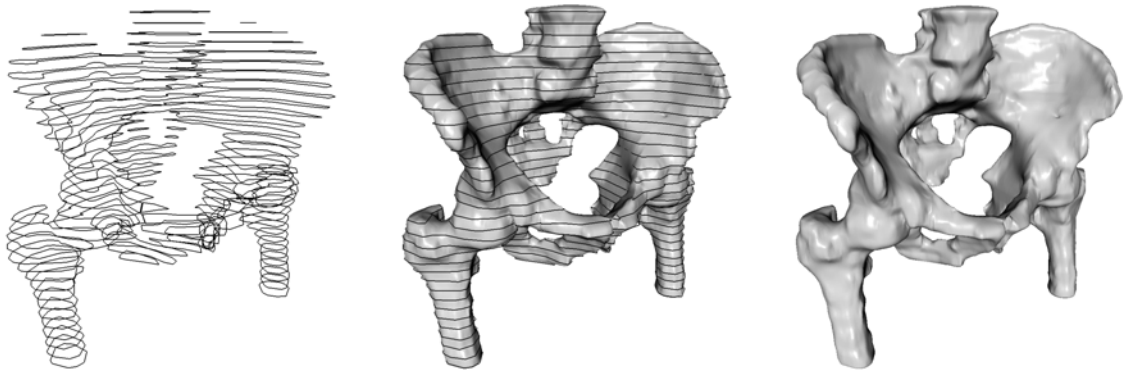

Figure 11: *34 pelvis contours. Reconstruction with and without contours displayed.*

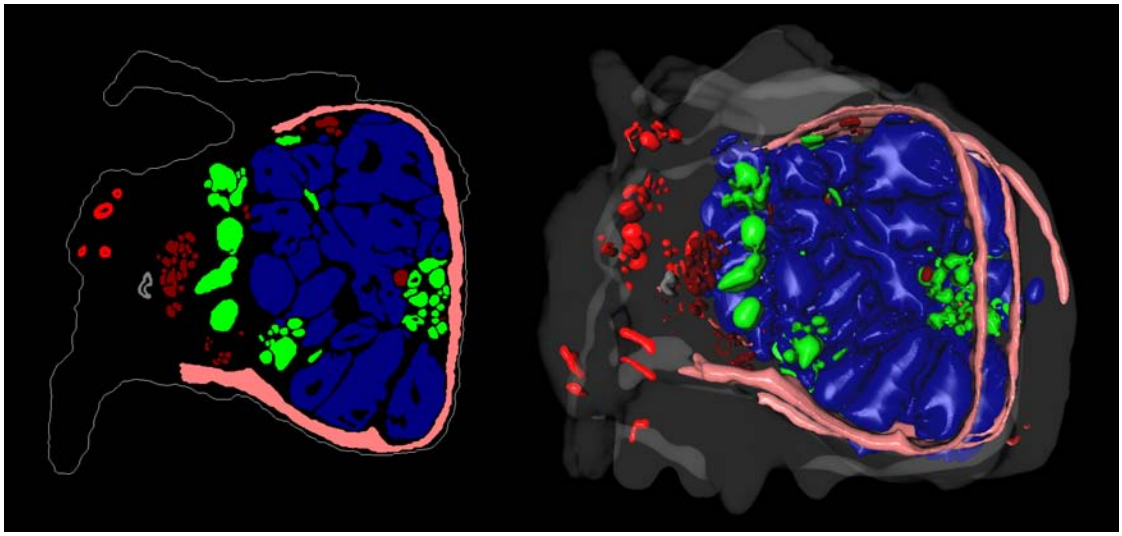

Figure 12: *(left) Single slice from a breast cancer tumor segmentation. Each color represents a particular kind of cell/structure, e.g. the blue region contains invasive cancer cells. (right) The full tumor model reconstructed from 9 segmentation slices.*

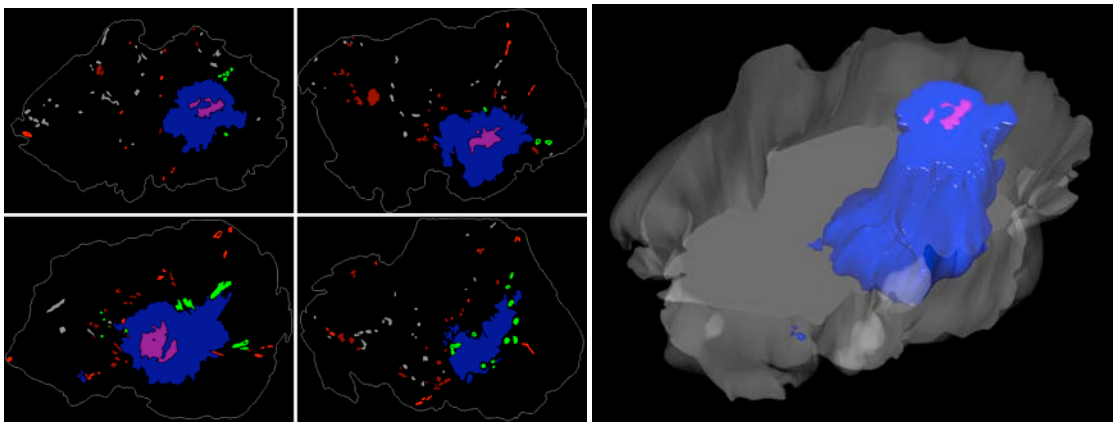

Figure 13: *A breast cancer tumor model (right) constructed from four histology-based segmentations (left). The model only includes regions with necrotic (dead) (purple) and invasive cancer (blue) cells, and the outer tumor membrane.*