REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
28-02-2009	Final Report	1-Jun-2005 - 31-May-2008

4. TITLE AND SUBTITLE

Analysis of Implicit Modeling of Complex Geometric Environments

5a. CONTRACT NUMBER
W911NF-05-1-0301

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS

Gregory M. Nielson

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

Arizona State University
Office of Research & Sponsored Projects Administration
Arizona State University
Tempe, AZ 85287 -3503

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
48402-MA.1

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; federal purpose rights

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

1. Developed methods for extracting mathematical surface description of urban/terrain implicitly defined from scattered point cloud data obtained from scanners and other detecting devices

2. Developed methods for parameterizing surfaces representing urban/terrain environments which are implicitly modeled.

15. SUBJECT TERMS

Urban Terrain Modeling, Implicit Mathematical Methods, Adaptive, Progressive, Efficient

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gregory Nielson
U	U	U	SAR		19b. TELEPHONE NUMBER 480-965-2785

Report Title

ABSTRACT

1. Developed methods for extracting mathematical surface description of urban/terrain implicitly defined from scattered point cloud data obtained from scanners and other detecting devices

2. Developed methods for parameterizing surfaces representing urban/terrain environments which are implicitly modeled.

3. Developed a custom software environment for simulating scanner output producing point cloud data sets.

4. Developed and analyzed adaptive, implicit mathematical methods for modeling data scanned from complex geometric environment.

5. Developed, tested and currently improving some very ingenious methods for detecting corners, edges and other features inferred by point cloud data sets. These are extremely effective methods that are based upon Lp norms and nonlinear optimization algorithms.

6. Developed Modeling and Visualization Techniques for Comparative Visualization for Wave-based and Geometric Acoustics

---

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

1. E. Deines, M. Bertram, J. Mohring, J. Jegorovs, F. Michel, H. Hagen & G. Nielson, Comparative Visualization for Wave-based and Geometric Acoustics, IEEE Transactions on Visualization and Computer Graphics, Volume 12, Issue 5, September/October 2006, pp. 1173-1179.

2. G. Nielson, H. Hagen & K. Lee, Implicit fitting of point cloud data using radial Hermite basis function, Computing, Volume 79, No. 3, 2007, pp. 301-307.

3. G. Nielson, L. Zhang, K. Lee & A. Huang, Parameterizing Marching Cubes Isosurfaces with Natural Neighbor Coordinates, In: Advances in Geometric Modeling and Processing 2008, LNCS 4975, pp. 315-328. Springer, 2008

4. G. Nielson, L. Zhang, K. Lee & A. Huang, Spherical Parameterization of Marching Cubes Isosurfaces based upon Nearest Neighbor Coordinates, Journal of Computer Science and Technology, 24(1): 30-38, Jan. 2009

5. G. Nielson, Normalized Implicit Radial Models for Scatterred Point Cloud Data without Normal Vectors, Computing, Volume 81, No. 3, 2009

Number of Papers published in peer-reviewed journals: 5.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

1. E. Deines, F. Michel, M. Bertram, H. Hagen, G. M. Nielson, Visualizing the Phonon Map, Eurographics/IEEE VGTC Symposium on Visualization (2006), pp. 1-8

2. G. Nielson, & K. Lee, Adaptive, Implicit Modeling of Urban Terrain Point Cloud Data, Proceedings of Advanced Computing and Applications, ACOMP 2008, Ho Chi Minh, Vietnam, March 12-14, 2008, pp. 235-246.

3. G. Nielson, Dual Marching Tetrahedra: Contouring in the Tetrahedronal Environment, Proceedings of International Symposium on Visual Computing, ISVC 2008, Spinger-Verlag, Part I, LNCS 5358, pp. 183-194, 2008.

4. G. Nielson & K. Lee, Urban Terrain Data Modeling with Adaptive Marching Tetrahedra and Feature Detecting Techniques, The International Conference on Information Science, Technology and Applications, ISTA 2009, Kuwait, March 2009

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 4

(d) Manuscripts

Number of Manuscripts: 0.00

Number of Inventions:

Graduate Students

NAME	PERCENT_SUPPORTED
Ryan Holmes	0.25
FTE Equivalent:	**0.25**
Total Number:	**1**

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Greg Nielson	0.50	No
FTE Equivalent:	**0.50**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: .....

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:·····

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:·····

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale): .....

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering: .....

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense .....

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: .....

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Accomplishments
ARO Final Progress Report, March 31, 2008
Proposal Number 48402MA, Agreement Number: W911NF0510301
PI: Gregory M. Nielson (nielson@asu.edu)

Outline/Synopsis

1. **Developed methods for extracting mathematical surface description of urban/terrain implicitly defined from scattered point cloud data obtained from scanners and other detecting devices**

2. **Developed methods for parameterizing surfaces representing urban/terrain environments which are implicitly modeled.**

3. **Developed a custom software environment for simulating scanner output producing point cloud data sets.**

4. **Developed and analyzed adaptive, implicit mathematical methods for modeling data scanned from complex geometric environment.**

5. **Developed, tested and currently improving some very ingenious methods for detecting corners, edges and other features inferred by point cloud data sets. These are extremely effective methods that are based upon Lp norms and nonlinear optimization algorithms.**

6. **Developed Modeling and Visualization Techniques for Comparative Visualization for Wave-based and Geometric Acoustics**

1. Developed methods for extracting mathematical surface description of urban/terrain implicitly defined from scattered point cloud data obtained from scanners and other detecting devices

We have developed a new and effective method of extracting level sets from implicitly defined models of urban/terrain data which we call dual marching tetrahedra (DMT) method. The DMT can be viewed as a generalization of the classical cuberille method of Chen et al. to a tetrahedronal. The cuberille method produces a rendering of quadrilaterals comprising a surface that separates voxels

deemed to be contained in an object of interest from those voxels not in the object. A cuberille is a region of 3D space partitioned into cubes. A tetrahedronal is a region of 3D space decomposed into tetrahedra. The DMT method generalizes the cubille method from cubes to tetrahedra and corrects a fundamental problem of the original cuberille method where separating surfaces are not necessarily manifolds. For binary segmented data, we propose a method for computing the location of vertices this is based upon the use of a minimal discrete norm curvature criterion. For applications where dependent function values are given at grid points, two alternative methods for computing vertex positions are discussed and compared. Examples are drawn from a variety of applications, including the Yes/No/Don't_Know data sets resulting from inconclusive segmentation processes, Well-Log data sets and the main application of implicitly modeling urban/terrain data.

Yes/No/Don't_Know Data Sets

MRI and CAT scan data sets can be viewed as samples of a function defined over a rectilinear lattice. Often segmentation algorithms are invoked to determine which of the lattice points belong to a certain specified type and which do not. Quite often, segmentation algorithms require the user to specify the values for certain parameters. Even if domain experts are used to "tune" these algorithms, the results can be, in some applications, "inconclusive". This means that, for some lattice grid points, the algorithm reports that these lattice points definitely belong to the object, for others it reports that these lattice points definitely do not belong to the object. But for some points the algorithm cannot definitely report one way or the other and so these lattice points could or could not be part of the object of interest. The technique, we present, for handling this type of data is illustrated in Figure 1. The left image illustrates with the black circles the lattice points known to be part of the object and the white circles indicate lattice points which are definitely not part of the object. The gray circles indicate the "Don't_Know" lattice points where the segmentation algorithm is inconclusive. The gray points are simply removed and the convex hull of the remaining white and black lattice points are collectively triangulated or, in the 3D case, tetrahedralized into a tetrahedronal.

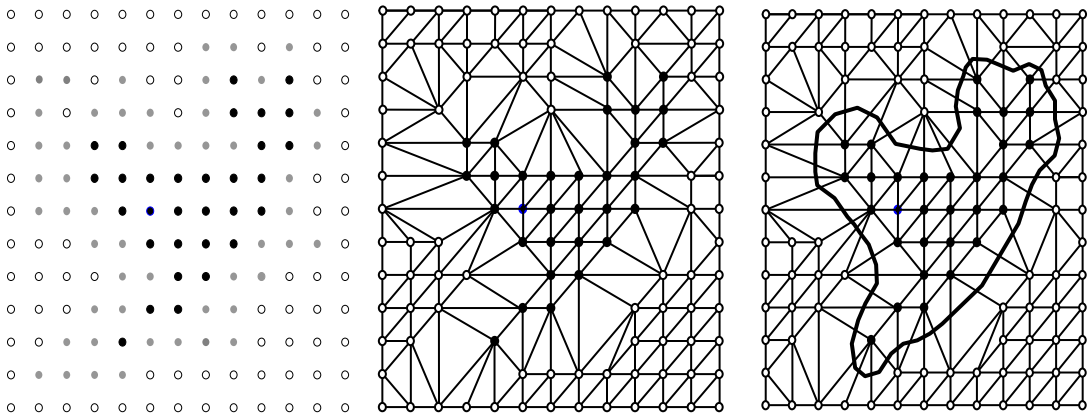


Figure 1. In the left image the white circles indicate lattice (grid) points that are definitely not contained in an object of interest, the black points are ones that are definitely belonging to the object and the grey points are undetermined. In the middle image, the grey points have been removed and the convex hull of the remaining white and black points are triangulated. In the right image a separating polygon is obtained by the 2-D version of the DMT method

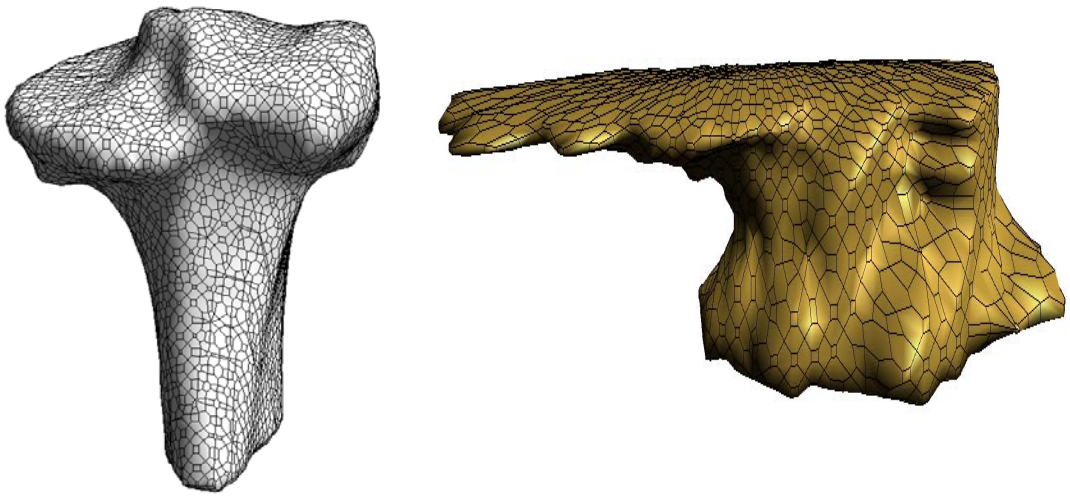


Figure 2. The results of the DMT method applied to a yes/no/don't know data set resulting from a segmentation algorithm applied to rectilinear data obtained from the scan of a Chimp tibia bone (only the top portion is used). The right image illustrated the application of the DMT to a Well-Log data set yielding a bounded region where copper exists at beyond trace levels

In Figure 3, we show an example which uses the well known “Blunt Fin” data set. Here, the IbC method of computing vertex positions is used. In addition, we include the isosurface produced by the Marching Tetrahedra (MT) method. We use the variation that produces a quadrilateral (rather than two triangles) in the case where function values at two grid vertices are above the threshold and two are below. With this version, the surfaces produced by the MT method and the present tetrahedronal method are formal mathematical dual polygon mesh surfaces of each other. Each vertex of one, uniquely associates with a face of the other. This is illustrated in the blow-up shown at the bottom, right of Figure 3 where the (white) edges of the tetrahedronal method are displayed along with the (black) edges of the surface produced by the MT method.

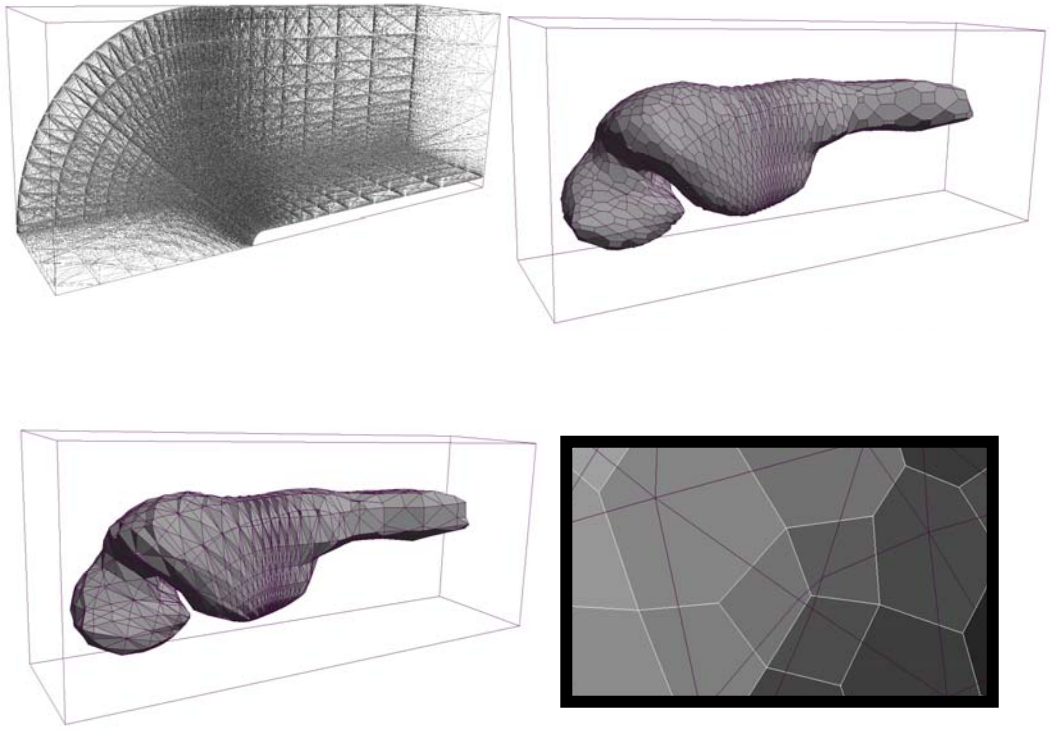


Figure 3. The grid of the “Blunt Fin” data set is shown in upper left. The upper right is the DMT using the IbC method of vertex selection and the lower left is the MT. The lower right illustrates the duality of the MT and the DMT

2. Developed methods for parameterizing surfaces representing

The triangular mesh surfaces (TMS) which result form the Marching Cubes (MC) algorithm have some unique and special properties not shared by general TMS. We exploit some of these properties in the development of some new, effective and efficient methods for parameterizing these surfaces. The parameterization consists of a planar triangulation which is isomorphic (maps one-to-one) to the triangular mesh. The parameterization is computed as the solution of a sparse linear system of equations which is based upon the fact that locally the MC surfaces are functions (height-fields). The coefficients of the linear system utilize natural neighbor coordinates (NNC) which depend upon Dirchlet tessellations. While the use of NNC for general TMS can be somewhat computationally expensive and is often done procedurally, for the present case of MC surfaces, we are able to obtain simple and explicit formulas which lead to efficient computational algorithms.

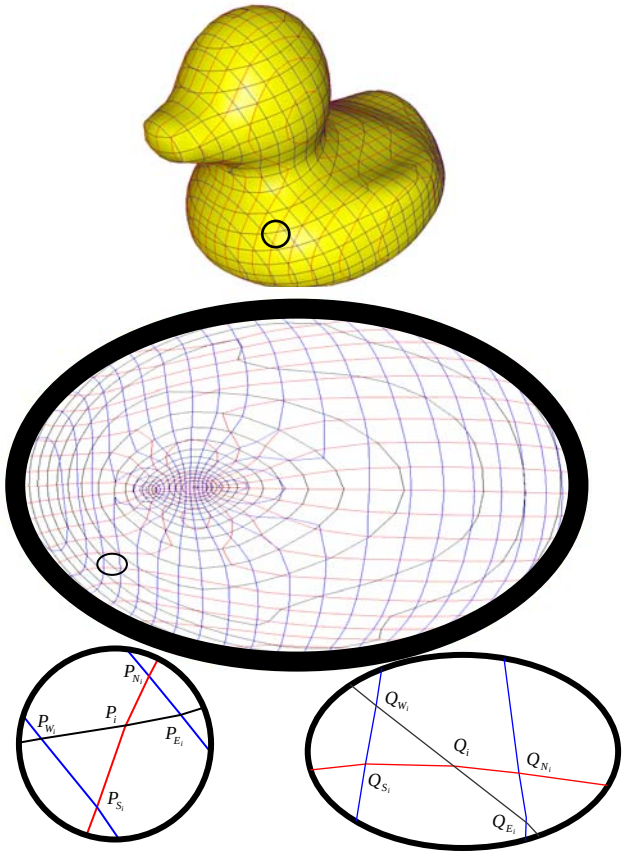


Figure 4. Center image shows the tiling of the 4*-Network which will lead to a parameterization of the MC triangular mesh surface.

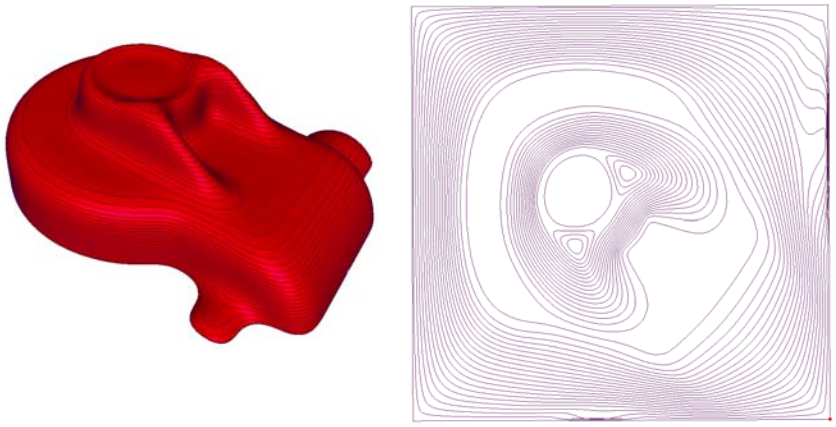


Figure 5. The left image show the MC isosurface of the geometry of a casting for the fabrication of a mechanical part. The right image illustrate the parameterization with one of the collection of 4*-network curves.

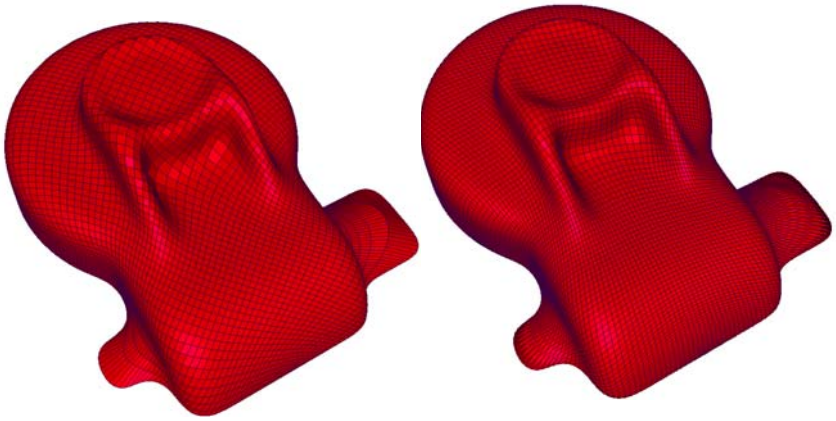


Figure 6. The images illustrate the use of the parameterization for re-sampling the surface. This not only illustrates a practical use (tool path control) of the parameterization, but also a method for graphically assessing the quality of a particular parameterization.

Illustrated in Figure 7 is a comparison of the new NNC method and the method of Floater, based upon mean value coordinates. The two parameterizations (one for the head and the other for the body) have a common boundary consisting of a 4*-network curve in the vicinity of the neck. The parameterization has been sampled over a uniform 64x64 grid. The mean value coordinate approach leads to larger quadrilaterals near the center and most deeply embedded portion of the parameterization.

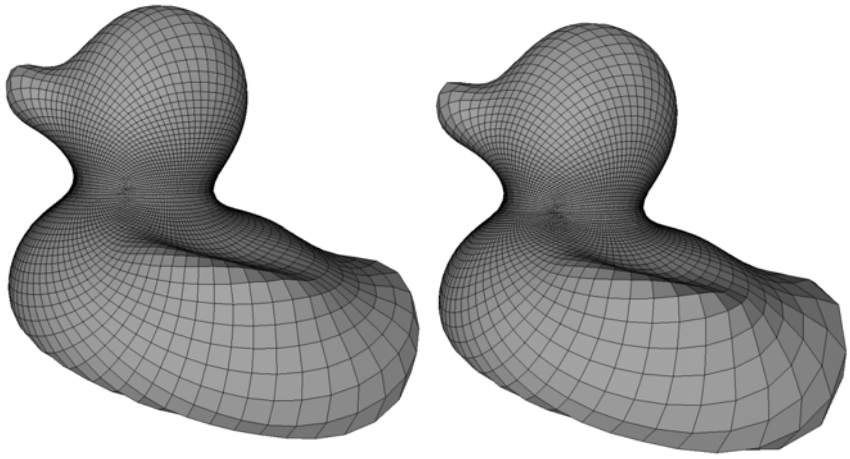


Figure 7. A parameterization is used to re-sample a surface. On the left, the new NNC parameterization is used and on the right the mean value coordinates (see [5]) are used.

And now some comments on our new method of adaptively approximating a Marching Cubes surface.

We represent the parameterization as a mapping from the sphere to the TMS as

$$\left(X(\theta,\phi), Y(\theta,\phi), Z(\theta,\phi)\right)$$

where $0 \le \theta \le 2\pi$, $-\pi/2 \le \phi \le \pi/2$ are longitude and latitude respectively. The parameter points are denoted as $Q_i = (\theta_i, \phi_i)$ so that we have the association

$$\begin{aligned} P_i &= \left(X(Q_i), Y(Q_i), Z(Q_i)\right) \\ &= \left(X(\theta_i,\phi_i), Y(\theta_i,\phi_i), Z(\theta_i,\phi_i)\right). \end{aligned}$$

The parameterization transforms many of the problems of geometry processing of the TMS to the more conventional problems of approximating and analyzing bivariate functions. With a parameterization, conventional techniques can be applied simultaneously to all three component functions. In the example of Figure 8 and Figure 9

we illustrate the process of adaptively approximating a TMS. The process starts with an optimal approximation defined over tessellation of the sphere utilizing only a small number of triangles. (Say, 4 points of tetrahedron.) The triangle with the largest error is selected for refinement. In order to avoid poorly shaped domain triangles, triangles are refined by splitting the longest edge. In order to avoid the cracking caused by "T" junctions, the process propagates to adjoining triangles as is illustrated in the images of Figure 8. This overall strategy is efficient in that it allows the complexity of the approximation to conform to the complexity of the problem. This is further illustrated with the example of Figure 9.

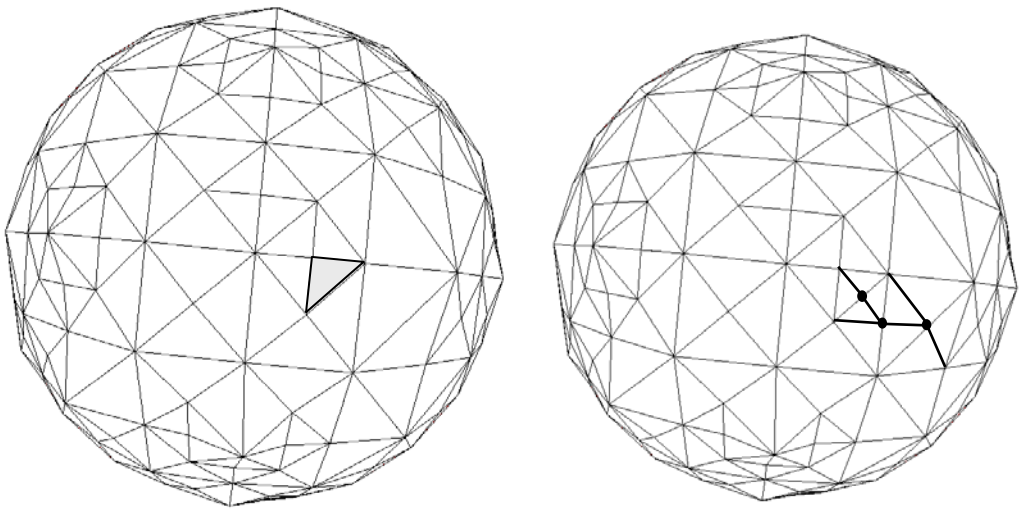


Figure 8. The propagation aspect of longest edge subdivision strategy is illustrated with the shaded triangle which gives rise to an additional three vertices and the four original triangles being replaced with 10 additional triangles.

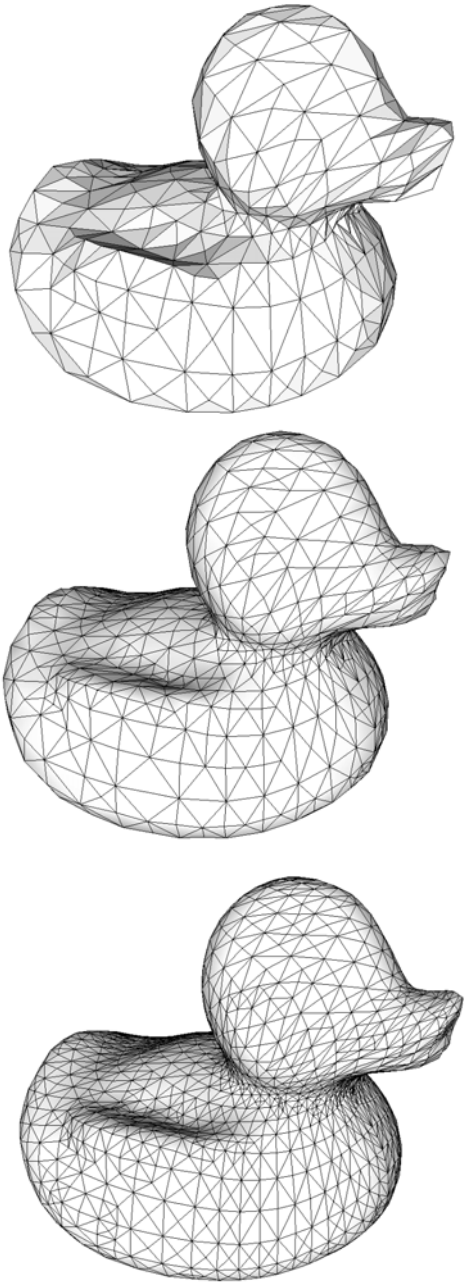


Figure 9. Adaptive Resampling with 500, 1000 and 2000 triangles. Note the efficiency with more triangles in the more complex shaped regions.

The term “flattening” has been used in the medical imaging field to convey the concept of mapping the surface of an object of interest to a planar domain. For example, with functional magnetic resonance (fMR) data it is possible to observe neural activity deep

within the folds or sulci of the brain cortical surface using brain flattening techniques. There is also the potential use of flattening techniques in virtual colonoscopy where "the approach favored by pathologists, which involves cutting open the tube represented by the colon, and laying it out flat for comprehensive inspection" is simulated. We illustrate flattening with an example that utilizes a volume data set that is available from the Surgical Planning Laboratory at Brigham and Women's Hospital, www.spl.harvard.edu. It is data that has been segmented from a MRI scan representing a brain which has had a tumor removed. It consists of a binary array of size $124 \times 256 \times 256$ with a value of $F_{ijk} = 1$ indicating the presence of brain matter and a value $F_{ijk} = 0$ representing anything other than brain material. The MC algorithm can be used to compute a triangular mesh surface at threshold (say) 0.5 that will separate all lattice points in the volume from those outside the volume. This surface utilizes the midpoints of edges and it rather blocky and difficult to perceive. Using some recently developed techniques, we can obtain a smoother surface that has the same topology (edges) as the midpoint surface. but the actual positions are chosen so as to optimize (subject to constraints) an energy/cost functional used to measure global smoothness. The top, left image of Figure 10 shows the midpoint surface for a down-sampled volume data set of size $62 \times 128 \times 128$. Even though our algorithms have no problems at the higher resolution, the graphs illustrating the parameterization are too cluttered for this context and so a lower resolution model is used. Figure 11 shows a sequence of convex combinations of two TMS with the same number of points and topology (edge connectivity).

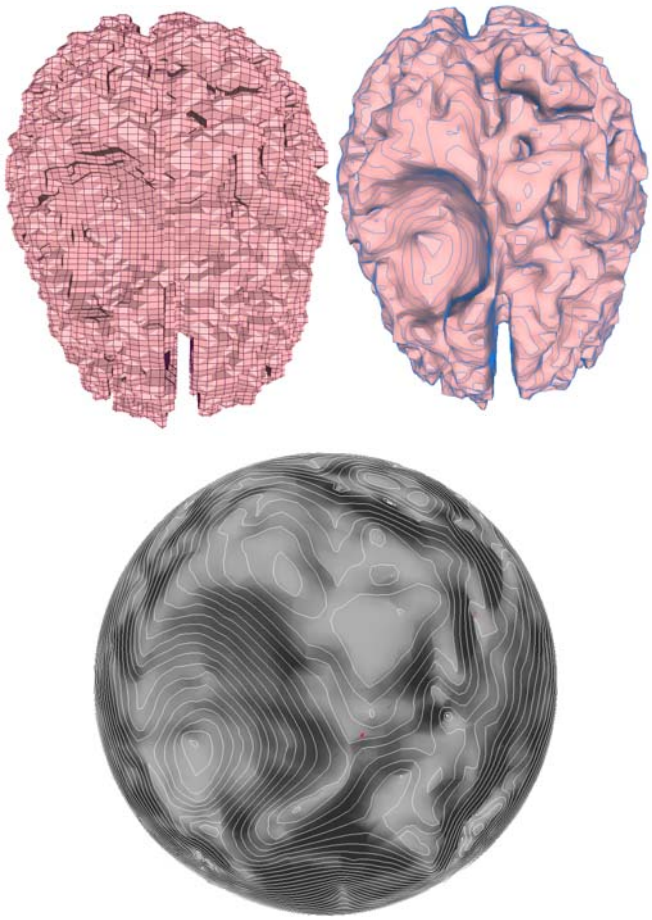


Figure 10. The top left image is the midpoint separating surface for the segmented data. The top right image is the smooth shroud). A tumor has been removed. The bottom image is the spherical parameterization with the illumination inherited from the rendering of the MC isosurface. The “real” way to view this is interactively!

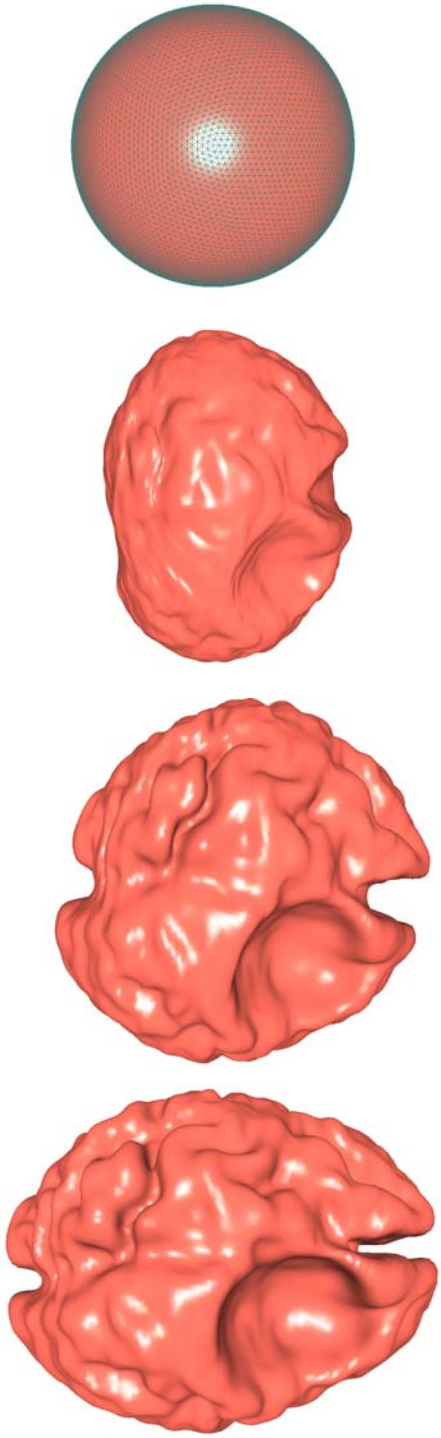


Figure 11. Sequence of convex combinations (morphs) between a sphere and the Harvard brain taken at values .4, .7 1.0.

Texture mapping is one of the predominant applications of parameterization techniques. In Figure 12, we illustrate the use of our parametric techniques for mapping a spherical texture to a genus zero TMS resulting from the MC algorithm.

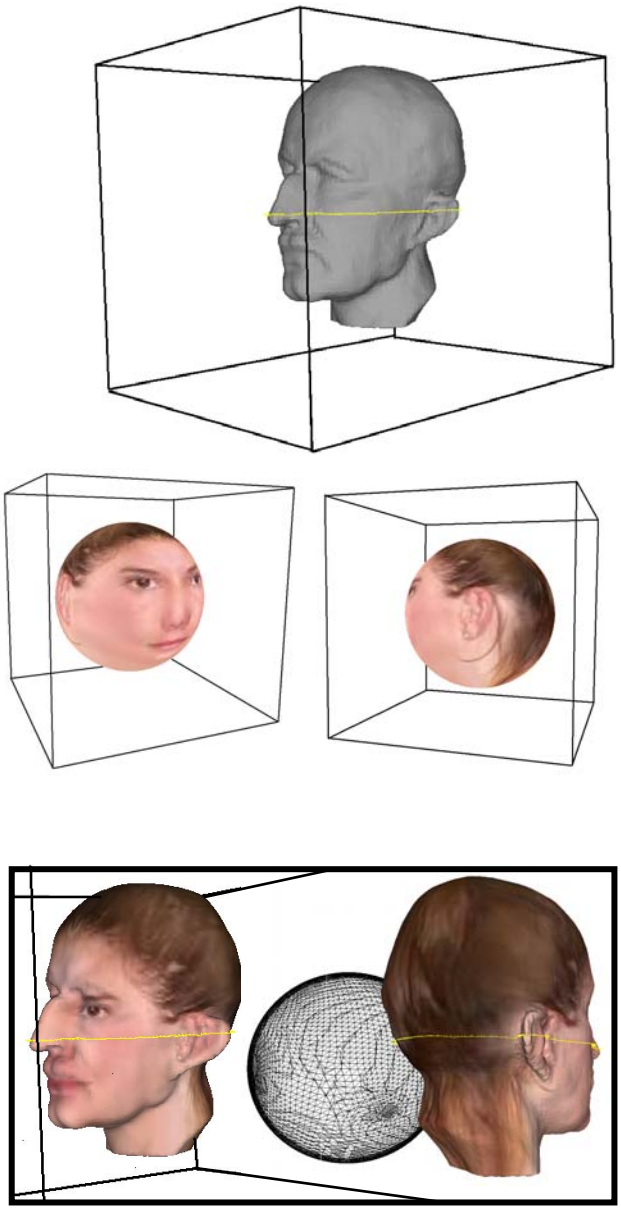


Figure 12. Top image illustrates the isosurface representing a bust of Max Planck. The y-loop for the initial parameterization is marked in yellow. The middle, spherical image is used as a texture map. The bottom image shows two views of the texture applied to the MC isosurface along with the triangulated sphere which defines and represents the parametric map from the sphere to the TMS.

3. Developed a custom software environment for simulating scanner output producing point cloud samples.

For the development and analysis of adaptive, implicit methods for mathematical fitting of scanned data from complex geometric environments we need a variety of different data sets including: both large and small data sets, data sets which exhibited considerable variation in sample densities, data sets with different error characteristics, data sets that infer sharp features, ect. Since it would be rather difficult to obtain all of these data sets from other sources with precisely the characteristics we need, we decided to invest the time and effort into developing a software environment for simulating scanner output.

Input to this software system is a polygon mesh surface. The user can establish the position, orientation and parameters (resolution, error distribution statistics/characteristics, ect.) of a collection of scanners. The output is point cloud of data grouped by the scanner which produced it. The scanner parameters are output along with the point cloud.

The program runs in an interactive mode allowing the users to freely position and orient any number of scanners (or scanner positions/orientations). Each scanner can have its own individual characteristics. The user can activate (scan) any subset of the scanners and have the point could output to storage. The scanner data (positions, orientations, resolutions, error properties, ect.) can be saved for use later.

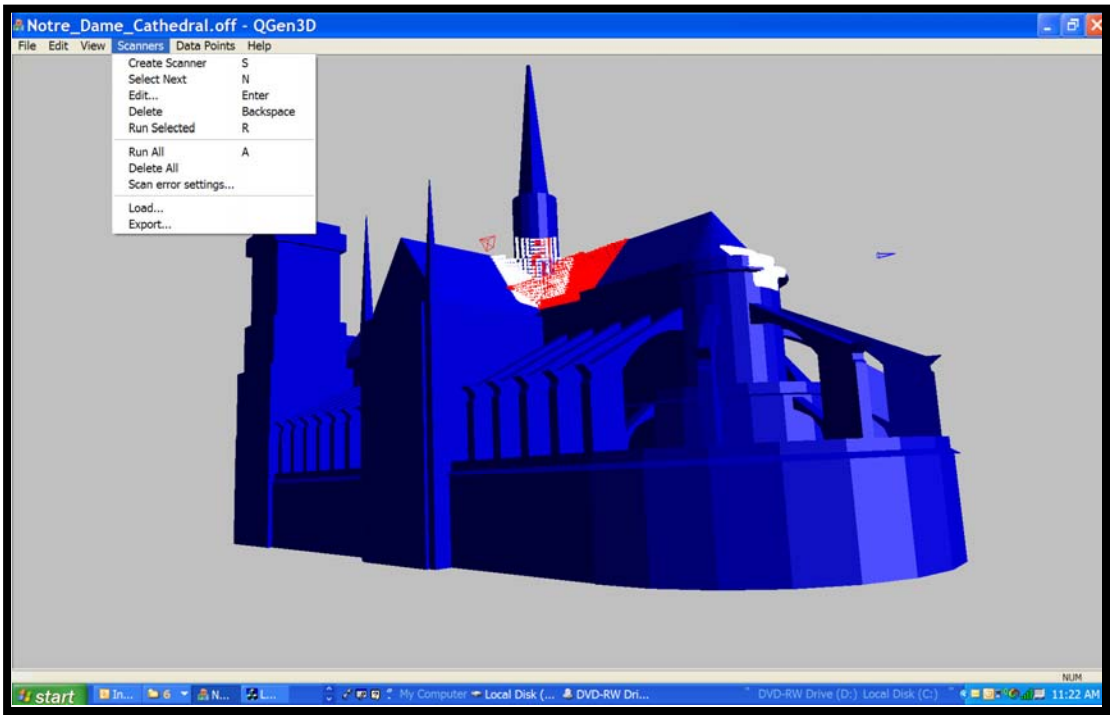


Figure 13. Typical screen image of Qgen3D.exe, an interactive program that simulates the process of acquiring scanned data. Any number of scanners can be created and positioned/oriented in the scene, each with their own characteristics (resolution, error distributions, ect.) In this example we only show a very small number of scanners, the icons illustrating the position and orientation of scanners are small in this image, but still discernable. The pull down menu indicates some of the operations possible for use in defining scanners and their properties.

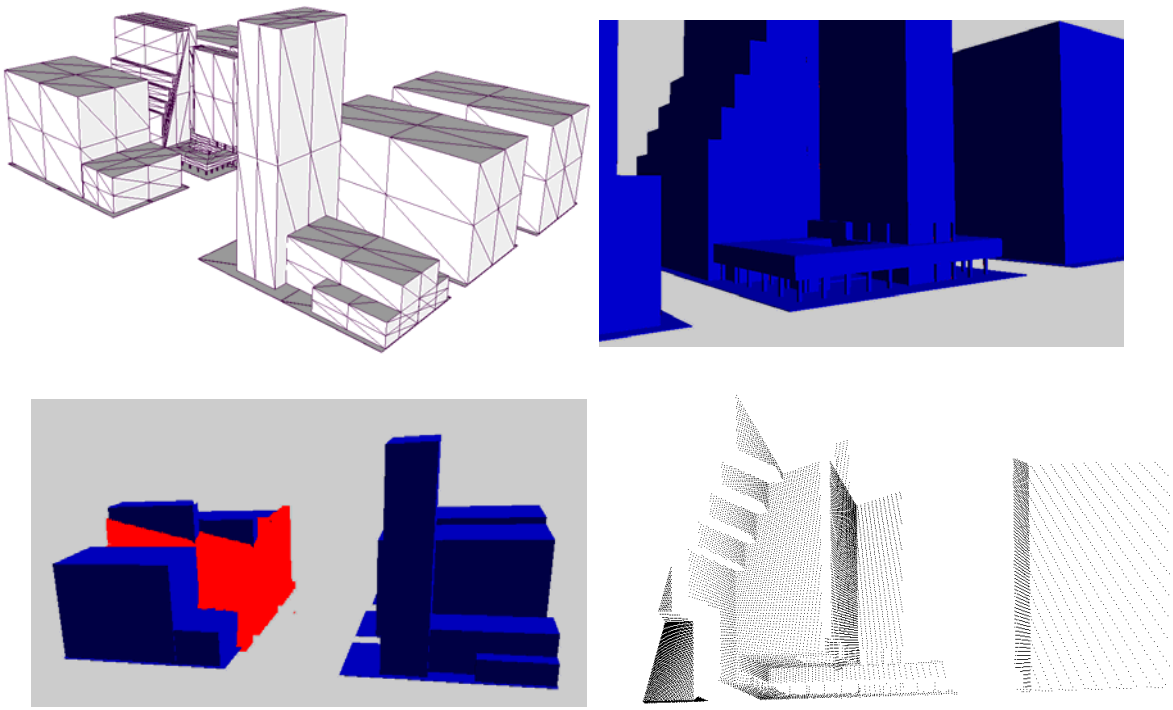


Figure 14. Another example illustrating the use of Qgen3D.exe, an interactive program that simulates scanners. The upper left shows the input consisting of a polygon mesh surface. The upper right shows the detail of a region covered by one user-specified scanner and the lower left (in red) shows the point cloud of the scan overlaid on the geometry of the urban scene (Lever House). The point cloud resulting form this single scan is shown in the lower right. Typically, several scanner positions and orientations are used to produce a single point cloud of scatted data for modeling an entire scene.

4. Developed and analyzed adaptive, implicit mathematical methods for modeling data scanned from complex geometric environment.

Given a point cloud of data (x_i, y_i, z_i), the methods we are analyzing, determine the parameters of an implicit model F(x, y, z) such that the surface {(x, y, z) : F (x, y, z) = 0} is an approximation to the surface implied by the point cloud.

The methods are adaptive, and based upon piecewise trivariate quadratic polynomials. An initial tetrahedral grid is determined so as to encase the point cloud. The parameters of the piecewise implicit model are associated with the vertices and edges of the tetrahedral grid. These values (10 per tetrahedron (4 vertices + 6 edges) uniquely determine the implicit piecewise quadratic modeling function. If the model satisfies the error criterion, then the process is finished, if not, then the domain mesh (tetrahedral grid) is subdivided by subdividing the tetrahedron with the largest error (a variety of methods of measuring error have been analyzed). In order to avoid the "cracking problem" certain adjacent tetrahedra must also be subdivided.

Attributes that have been analyzed throughout this development include:

1) Error measurement techniques
2) Normal vector estimation
3) Subdivision strategies
4) Smoothness constraints (across cell boundaries)
5) Parameter specification and control for modeling sharp features
6) Isosurface/dual contour extraction techniques (preserving features, ect.)

We do not go into the reporting the details of this analysis here, except to point out one of the most important key features of this type of implicit modeling, which is the ability to efficiently and effectively model sharp features implied by the data. The ability to do this is a direct consequence of the choice for the implicit field function. A simple example is shown in Figure 15 below. In this case we have a piecewise, implicit quadratic polynomial which can have a hyperbola (and it degenerate case of two intersecting lines) as its level set. It is really rather remarkable how well this process works. This entire point cloud is fit (just about perfectly!) with only a small number of implicit curves.

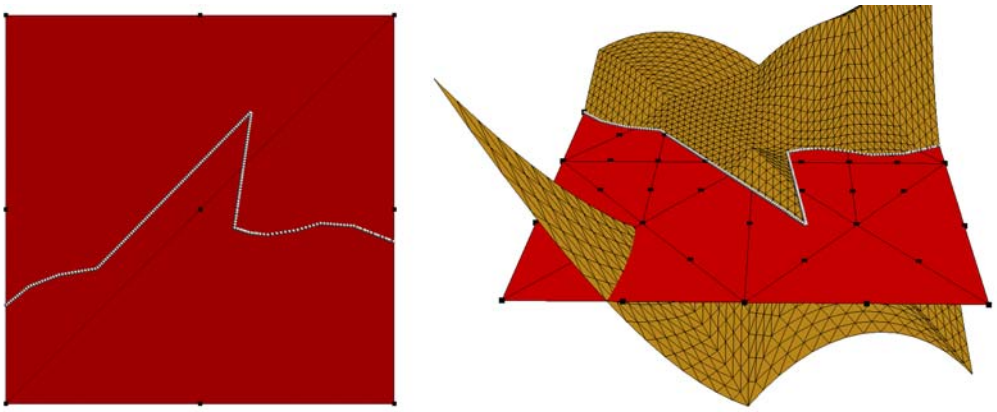
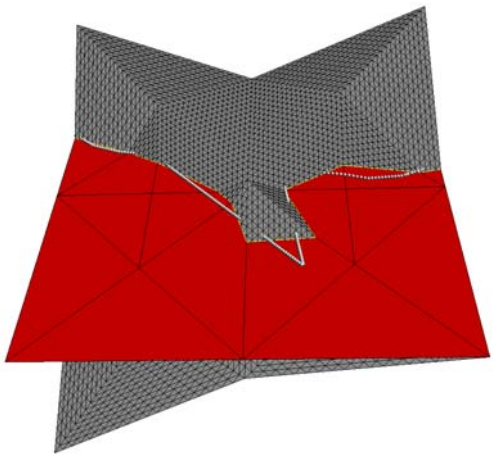


Figure 15. The data points are shown in the left image and the piecewise quadratic implicit fit is shown in the right image. The main purpose of this example is to illustrate the ability of implicit quadratic models to capture corners implied by the data and the fact that these sharp features do not necessarily have to be located on cell boundaries. The efficiency of this type of modeling can also be noted by the relatively small number of domain cells required to obtain a very accurate fit. This example also points out one potentially undesirable attribute of this type of model and this is the possible appearance of extraneous contours. But this is not a real problem as these extraneous contours are easily removed by using the context of data in cells and continuity of contours.

Figure 16. Piecewise linear implicit fit to the same data as in Figure 3. We include this example to point out that it would be rather difficult to model a corner with a piecewise linear implicit modeling function. In order for a piecewise model to fit a corner, the corner would have to be on the boundary between domain cells. This means we would have to know the location of the corner in advance.



We now show two 3D examples which illustrate the power of this class of implicit methods to model point cloud data. They are rather different from each other. The first is more traditional and consist of a mechanical part. The second illustrates the utility of

this class of methods for modeling urban terrain data consisting of natural terrain and artifacts.

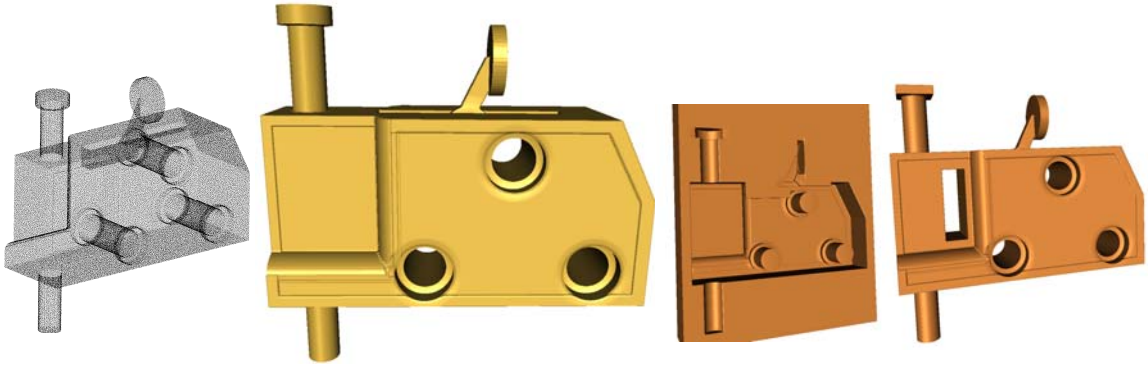


Figure 17. From left to right we have: 1) Point Cloud, 2) Level Set of Implicit Model, 3) Mold made with Boolean operations applied to field function and 4) Some additional modifications made with Boolean operations (This is to "prove" we really have an implicit model.) A notch has been cut out and a square top has been put on the latch pin.

Note in the example above that this type of implicit model has the capability of capturing features (corners and edges) implied by the point cloud. Since the implicit model is piecewise quadratic, the fitting process we use automatically identifies and approximates these features.

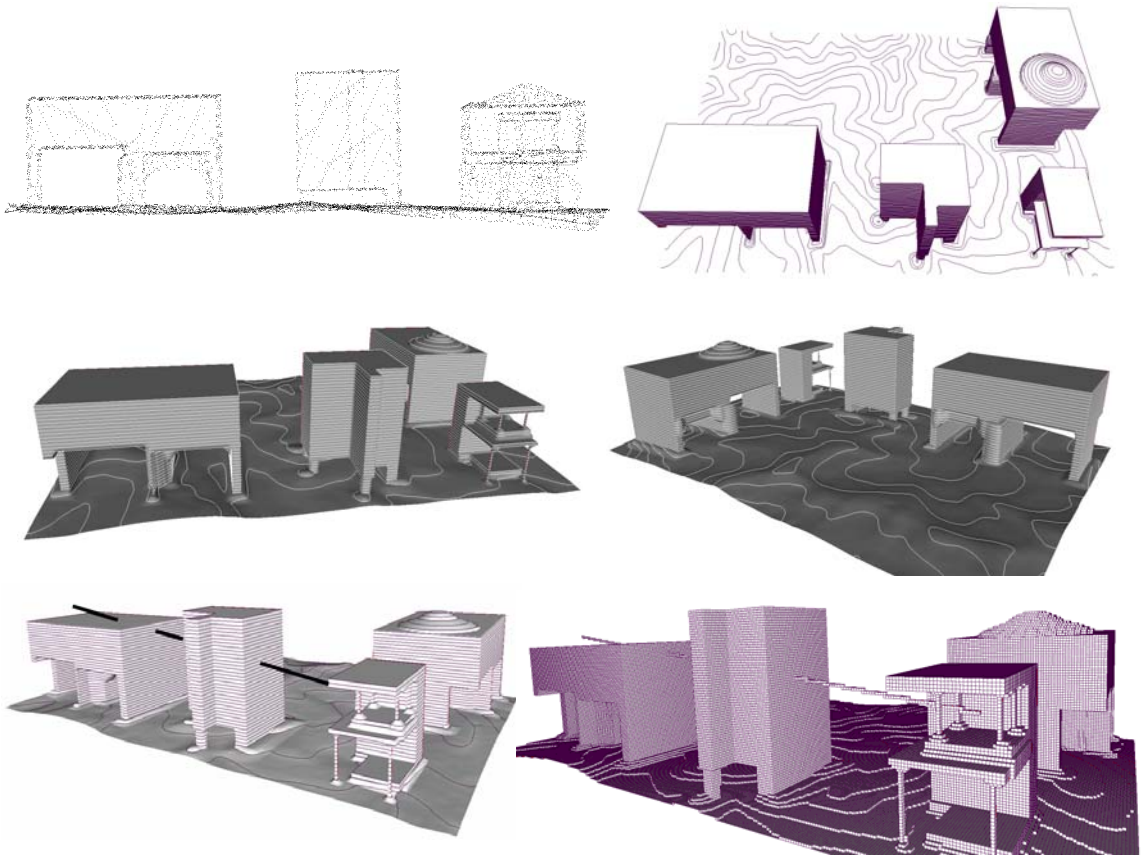


Figure 18. Upper left shows 1% of the point cloud data set of an urban terrain data set consisting of artifacts located over topographically terrain. Upper right is contour plot of an implicit model computed by the adaptive, piecewise quadratic techniques analyzed for this project. This contour plot clearly shows the sharp features preserved by the modeling process. The middle images are two different views of the dual isosurface of the implicit model. The bottom two images relate to line-of-sight computation facilitated by the implicit model. Intersections are computed as roots of quadratic equations (left) or spatial enumeration techniques (right) based upon discrete (Bresenham's algorithm) lines.

5. Developed, tested and currently improving some very ingenious methods for detecting corners, edges and other features inferred by point cloud data sets. These are extremely effective methods that are based upon Lp norms and nonlinear optimization algorithms.

In order to appreciate the difficulty of the problem that we are considering and the value of the methods we are analyzing and developing, consider finding the "corner" implied by the data set { (8.1, 3.9), (4.1, 1.9), (3.9, 7.8), (1.8, 1.1), (1.1, 1.9), (3.1, 6.1), (5.9, 3.1), (9.9, 5.1), (0.4, 1.1), (1.4, 3.9), (5.1, 9.9) } without using graphical techniques. You may compute all you want, but you are not allowed to graph the data and use your visual perception to detect the "corner". Using discrete L_p norms, nonlinear optimization techniques and techniques related to the adaptive implicit fitting discussed above, we have recently discovered a general approach and class of methods for analyzing the data yielding the positions of these corners. Unlike most of the work in computer vision which relates to this problem, our methods directly extend to any number of dimensions and are based upon geometric modeling concepts. We have written some demonstration programs for 2D and 3D which allow a user to test the early versions of our techniques on some examples. The images of Figures 19 and 20 show typical screen shots of these programs in action.

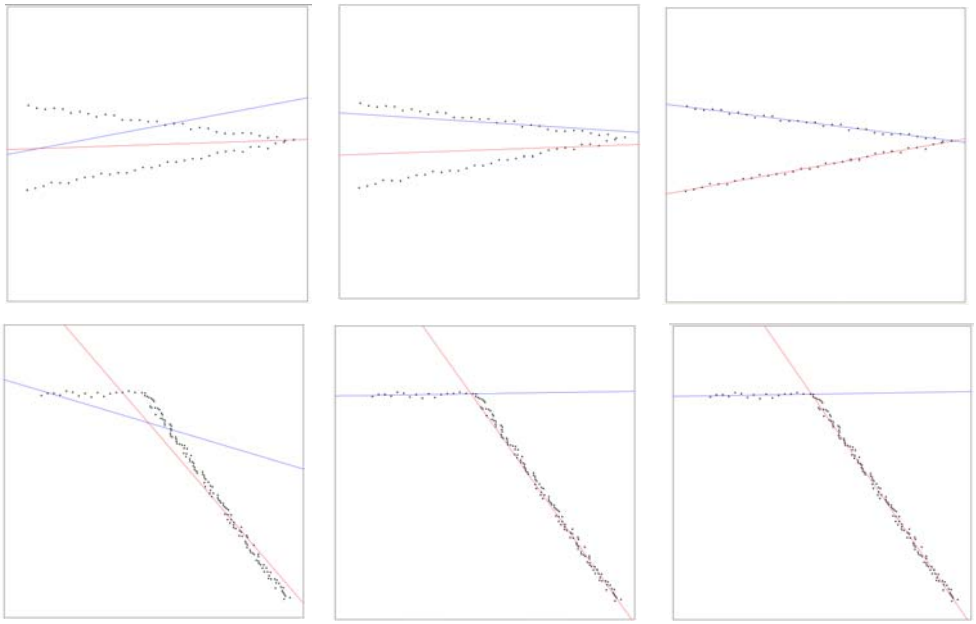


Figure 19. Here we have (from left to right) the first, second and final iterations of a new method of determining corners implied by scattered point cloud data. The method is based upon discrete Lp norms and nonlinear optimization and the adaptive, implicit methods of fitting point cloud data mention above. A demonstration program, Conner2D.exe, is available at www.public.asu.edu/~nielson/ARO/

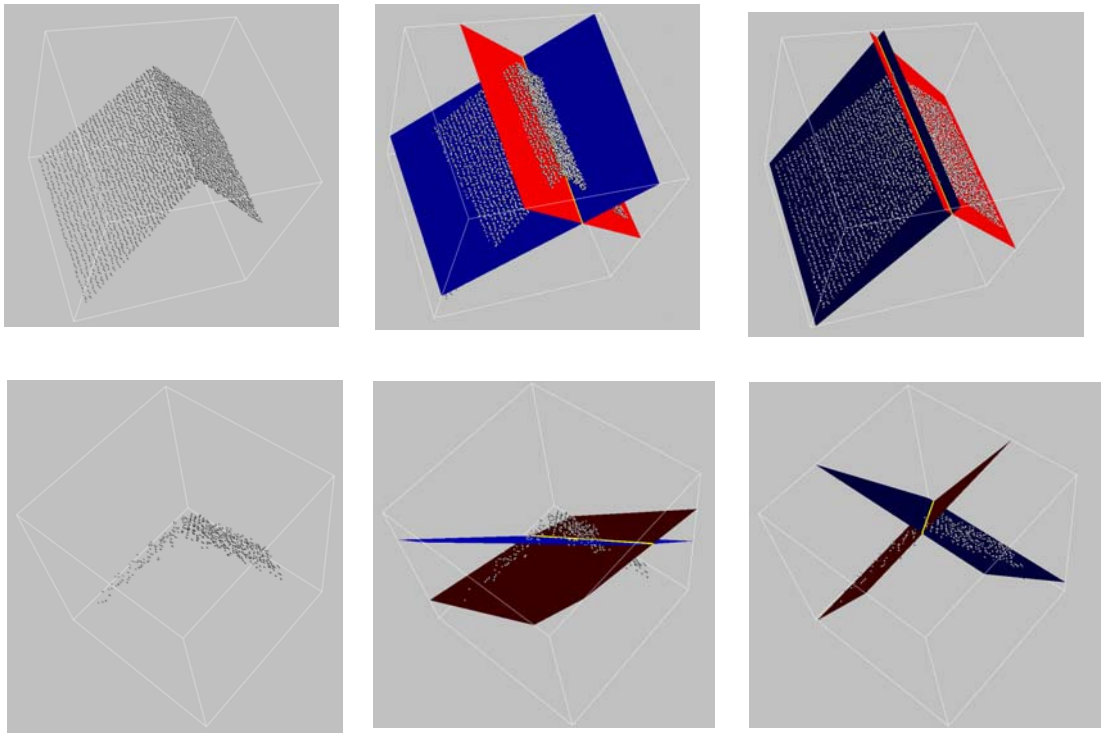


Figure 20. From left to right, we have the data, an intermediate approximation and the final edge determined by the new method of analyzing scattered data for determining features. A demonstration program, Edge3D.exe, is available at www.public.asu.edu/~nielson/ARO

Even though the demonstration program Edge3D.exe only covers the case of 3D edges (where two planes intersect), our new techniques will also find 3D corners (where three planes intersect). This is illustrated in the images of Figure 21. The top example is an "outside" corner and the bottom example is a "notch" corner.

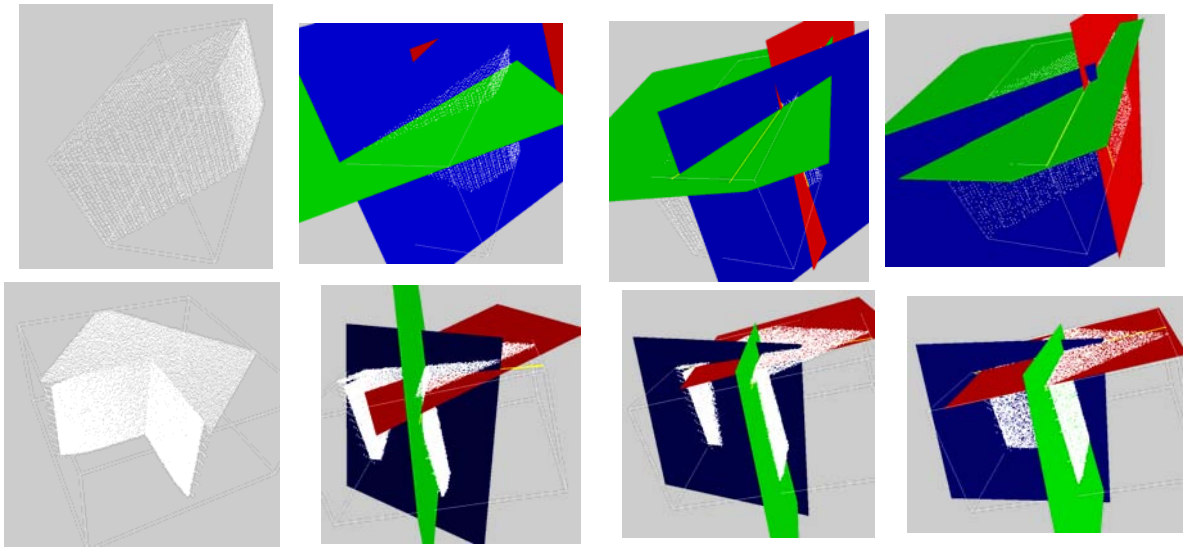


Figure 21. The data for an "outside" corner is shown in the left image of the top row. The first, second and final iterates are shown in the next three images (left to right). The three different planes are red, green and blue. The intersections lines between these planes (the edges of the corner) are illustrated with yellow lines. The second row utilizes a point cloud that infers a "notch" corner. It is really rather amazing that it is possible to have an automatic analysis tool that can "find" these corners.

There are some intriguing implications and interesting applications to ponder here. In principle, the number of lines or planes could be unlimited. Six planes could be used to find the boundary of box like objects or 10, 000 planes could be used to find a model of a complete building!

We could also consider generalizing in another direction. There is nothing intrinsic in this process that limits the fitting primitive to be lines (2D) or planes (3D). Hyperplanes of any dimension could be used. This leads to the extremely intriguing possibility of creating some totally different tools and techniques for finding information is large multidimensional data sets. (For example, consider an extremely large (maybe millions or even billions) sample set of many (hundreds or thousands) of dimensions representing some financial records.) This is a very hot topic in the area of homeland security and infovis where it is now generally accepted that "visualization" tools are not going to be the solution for finding "features" in extremely large and high dimensional data sets. The "smart money" is on the notion that there will be some type of mathematical analysis that will find the "features" which are then presented (possibly visually) to an analyst for subsequent analysis. The features which are called corners and edges in 3D would certainly be of potential interest in this more general setting.

6. Developed Modeling and Visualization Techniques for Comparative Visualization for Wave-based and Geometric Acoustics

We have developed some tools for doing comparative visualization of the acoustic simulation results obtained by two different approaches that were combinedinto a single simulation algorithm. The first method solves thewave equation on a volume grid based on finite elements. The secondmethod, phonon tracing, is a geometric approach that we havepreviously developed for interactive simulation, visualization andmodeling of room acoustics. Geometric approaches of this kindare more efficient than FEM in the high and medium frequencyrange. For low frequencies they fail to represent diffraction, which on the other hand can be simulated properly by means of FEM. When combining both methods we need to calibrate them properlyand estimate in which frequency range they provide comparable results. For this purpose we choose the visualization of interference patterns, since these depend not only on diffraction, but also exhibit phase-dependent amplification and neutralization effects.

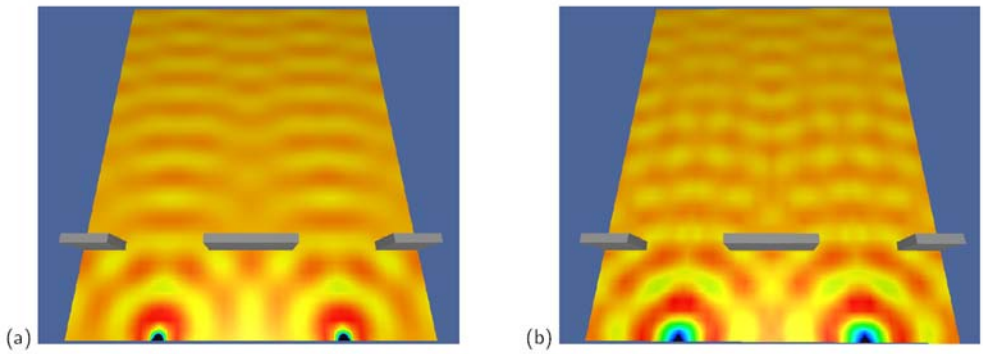


Figure 22. Interference pattern. FEM simulation (a) and phonon tracing (b) for the wave number k = 12.

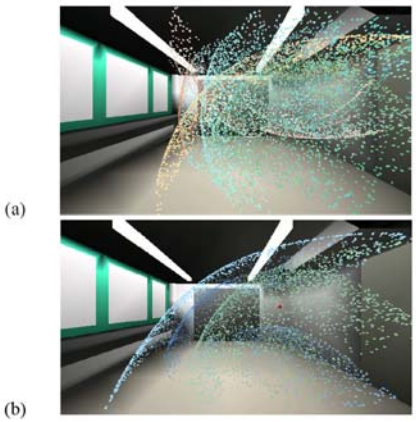


Figure 234. Visualizattion of sound wave propagation (a) different wave fronts. (b) reflections on the bottom.