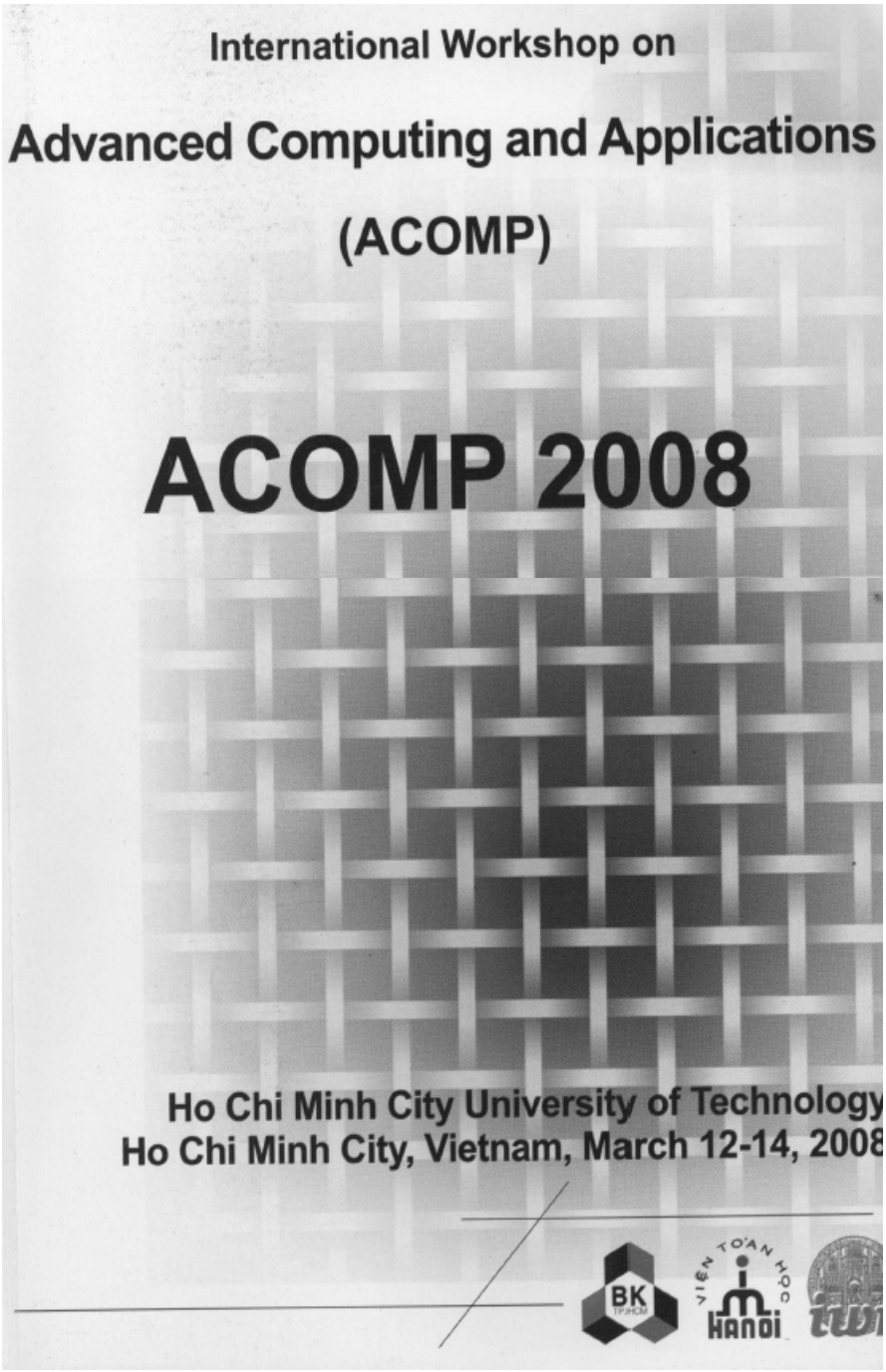


G. M. Nielson and K. Lee, Adaptive, Implicit Modeling of Urban Terrain Point Cloud Data, *Proceedings of Advanced Computing and Applications, ACOMP 2008*, Ho Chi Minh City, March 12-14, 2008, pp. 235-246.

Adaptive, Implicit Modeling of Urban Terrain Point Cloud Data

Gregory M. Nielson[1] and Kun Lee[2]
[1] School of Computing Informatics, Arizona State University, Tempe, AZ, USA
[2] Handong Global University, Pohang, Kyungbuk, South Korea

Abstract. We describe some work in progress on a project focused on the mathematical and geometric modeling of scanner data (e.g. LIDAR) which has been obtained from urban terrains. The methods that are being developed are based upon the concepts of adaptively refined implicit models. There are special and particular advantages to using implicit models for this type of geometry including the ease of performing Boolean operations (union and intersection) and creating multiresolution models. The field functions for the implict models are selected on the basis of efficient compact models that can replicate complicated geometry and also faithfully reproduce sharp detailed features and artifacts. The requestor and funding agency of this research is the United States of America Army through the US Army Research Office.

Keywords: Modeling, LIDAR, Geometric Modeling, Urban Terrain, Point Cloud, Scattered Data, Implicit Modeling,

1 Introduction, Motivation and Background

We present new methods of modeling scattered point cloud data obtained from scanners applied to urban terrain. This type of geometry is indicated and depicted in the images of Fig. 1. These scenes consist of combinations of artifacts and natural terrain. There are buildings with connecting walkways, overhangs, underpasses, bridges, mountains, rivers, water towers, statues and a myriad of different geometric objects. It is well known (see [10], [25]) that this type of geometry can not be represented solely with a digital elevation map (DEM) or with a normal triangulated irregular network (TIN) due to the fact that the geometry is not a height field (single valued function) relative to a planar domain. In the approach described here, an implicitly defined surface, $S = \{(x,y,z): F(x,y,z) = 0\}$ is determined that approximates the point cloud (x_i, y_i, z_i), i=1, . . . N, which is produced by the scanners. The surface geometry is defined as a level set of a trivariate function $F(x,y,z)$ called the *field* function. In contrast to the conventional explicit methods which usually are in the form of a list of triangles, we define, manipulate and store the geometry of the scene by means of the field function, F. There are distinct advantages to implicitly defined geometry over explicitly defined geometry. These advantages include i) The ease of performing unions and intersection operations with extremely simple operations on the field function; ii) The ability to obtain multi-resolution models with simple operations applied to the field function and iii) The ease of incorporating corners and

sharp features in the surface model without pre-described discontinuities or other properties of the field function.

Fig. 1. Examples of scenes involving urban terrain consisting of natural terrain along with artifacts.

1.1 Brief Overview and background on point cloud fitting

For this project, we are interested in the following general problem of point cloud fitting. Given a collection of points in 3D space (x_i, y_i, z_i), i=1, . . . N, taken from a 3D surface, the problem is to develop mathematical methods for respresenting and approximating the underlying surface. We should point out that the problem of point cloud fitting should be distinguished from that of scattered data modeling [8, 9, 18]. Even though many of the basic techniques and tools from CAGD (Computer Aided Geometric Design) and multivariate approximation theory apply to both problems, they are basically different. The problem of scattered data modeling is concerned with methods of producing a bivariate function $F(x,y)$ such that $F(x_i, y_i) \sim z_i$. That is, for the traditional scattered data modeling there is the assumption that the data consists of samples taken from the surface graph of a bivariate function. One fundamental connection between the two problems is through some type of parameterization of the point cloud [4,5] whether this be implicit or explicit, but this relationship is not well and completely understood today. The term "scattered data" was coined by Schumaker in his 1976 paper [24] and there was a great deal of interest and published research on (mainly) bivariate problems in the 70s and 80s. For applications to terrain modeling see [6, 7, 10, 12, 13, 22]. With the advent of scientific visualization along with volume visualization in the 90s, there was growing interest in trivariate scattered data modeling [18, 20] and interest in this area continues to grow. In many respects, the problem of point cloud fitting is more difficult because it is less understood, but the widespread and strong need for practical and effective methods make this an important problem. Today, there is widespread interest in the problem and many algorithms, methods and techniques are being proposed [11, 17, 26]

A number of methods involve the signed distance function, $D(P)$, which is a trivariate function defined to be zero on the surface S, negative interior to S and positive outside of S. The surface is extracted from $D(P)$ as a triangular mesh surface approximation to the zero level isosurface. Typically, $D(P)$ is sampled on a 3D rectilinear grid and a method like the marching cubes algorithm [21] is used. Once it is decided what the metric or the definition of distance from a surface to a point cloud is to be, it is usually not too difficult to develop algorithms for the efficient computation of the distance function. The particular difficulty here is getting the sign right; that is, to be able to efficiently and effectively determine when a point is inside the surface or outside. Typical of the methods based upon distance functions is that of [11], where the sign is based upon local least squares estimates of the normal vector of the surface and a consistent orientation (in or out) is sought with the Riemannian map estimate. One of the drawbacks to this method is the heuristics of the signed distance function calculation may lead to gaps in the surface and the difficulty of choosing the proper resolution for the marching cubes voxel grid can have detrimental effects on the success of the method.

Another important concept involved in point cloud fitting problems is the Delaunay tetrahedrization [19] and its dual, the Dirichlet tessellation and Voronoi diagram. Methods that utilize these concepts include [2, 3]. The method based upon alpha shapes of [3] is a typical and early example. Here the first step is the Delaunay tetrahedrization. The second step is to apply the alpha-erasure to remove tetrahedral, triangles and edges whose minimum surrounding sphere is not contained in the alpha –erasure sphere. The result is called the alpha-shape. In the third step, triangles for the final surface are selected so that a sphere of radius alpha containing the triangle does not contain any other point cloud points. The main negative aspect of this approach is the choice of a suitable value of alpha. Too big of a choice leads to poor approximations not utilizing many of the points of the point cloud and too small a choice leads to gaps and fragmented surfaces. Even though the methods of [2] guarantee for sufficiently dense samplings a surface that is homeomorphic and geometrically close to the point clouds resulting from sampling, in practice, it can be the case that point density varies considerably limiting the success for these methods for certain applications. In [1], a postprocessing, topological clean up phase based upon linear programming is suggested "that can (to some extent) reconstruct non-smooth or undersampled surfaces." We mention another potential drawback to these types of methods for certain applications. The resulting triangular mesh surface has vertices that are points of the original point cloud. For noisy data or overlapping data resulting from imperfect registration of scanned data, this may be undesirable. Rather than interpolated the point cloud (or a subset), it is potentially more desirable to approximate it for some applications. In addition it would be desirable to take advantage of the fact the many point cloud data sets have varying density.

The discussion above motivates and leads to the present method which is described in the next section.

1.2 Adaptive/Progressive/Multiresolution Field Functions

It is clear that a single, globally defined field function can not capture the complex geometry of urban/terrain data. Also, it is well known that piecewise defined field functions over uniform lattices are not efficient because the complexity of the modeling function does not necessarily conform to the regions of complexity of the geometric data. Adaptive models which refine in regions of increased detail and complexity are much more efficient for this application. This leads to the two main aspects that must be analyzed: 1) What is the form of the modeling field function and 2) what is the refinement strategy?

The refinement strategy:

For the methods described here, two separate adaptive refinement strategies: 1) the longest edge tetrahedral refinement (see [15] and [23] and Fig. 4) and 2) the red/green tetrahedral refinement (see [16]) are used. The left portion of Fig. 2 shows the red and green type of splits for two dimensional problems. The response to a need for refinement results in a red split followed by green splits of the neighbors. In order to avoid poorly shaped element domains, a green split is never followed by a another green split; rather a green split is replaced by a red split and the necessary neighbor green splits.

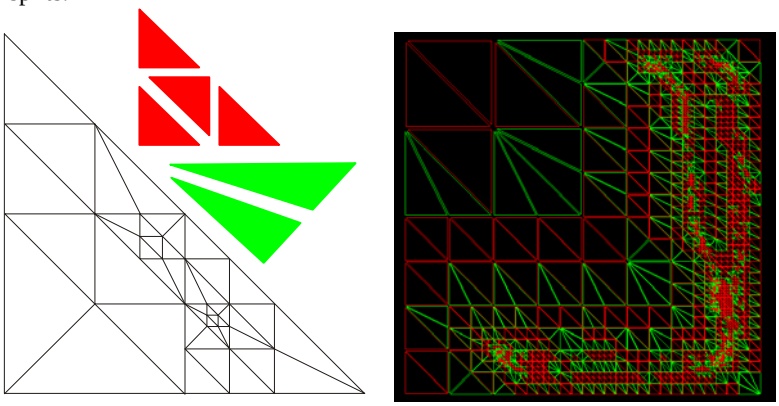


Fig. 2. The red/green refinement scheme for two dimensions.

Our application requires the 3D version of the red/green strategy. The situation for 3D is somewhat more complicated. Rather than just one “green” split, as is the case for 2D, there are several (10) required green splits in 3D. A sampling of the “green” splits is shown in Figure 3. Also we show an example of a cube that has been successively refined using this strategy.

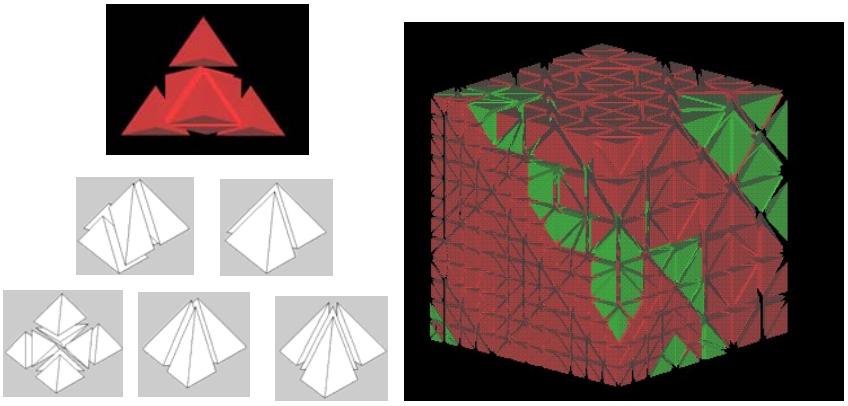


Fig. 3. The red/green refinement scheme for three dimensions. There is only one "red" split, but several "green" splits. Only a sampling of the "green" splits are shown. A typical example of a tetrahedral decomposition based upon this strategy is shown on the right.

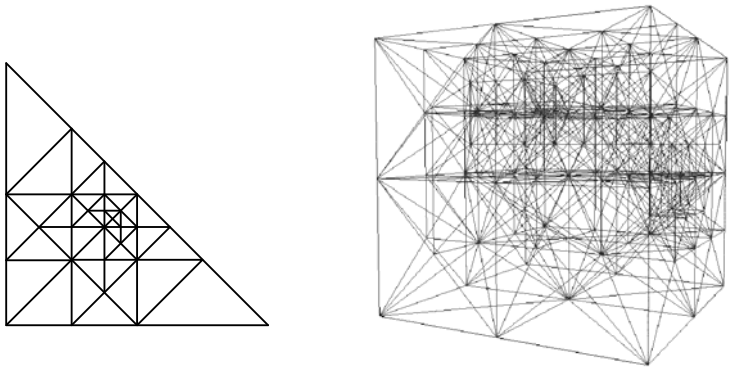


Fig. 4. Illustrating the "longest edge" adaptive refinement strategy.

1.3 Form of the piecewise defined trivaritate field function

The field function is defined in a piecewise manner as a trivariate quadratic over each tetrahedron in the domain decomposition (achieved through the adaptive refinement process). For a given tetrahedron T_{ijkl} with vertices V_i, V_j, V_k and V_l the field function is defined as

$$F(x,y,z) = F(b_i,b_j,b_k,b_l) = F_i b_i + F_j b_j + F_k b_k + F_l b_l + F_{ij} b_i b_j + F_{ik} b_i b_k + F_{uk} b_i b_l + F_{jk} b_j b_k + F_{jl} b_j b_l + F_{kl} b_k b_l$$

where $F_\alpha = F(V_\alpha)$ and $F_{\beta\delta} = F\left(\frac{V_\beta + V_\delta}{2}\right)$ and

$$(x, y, z) = b_i V + b_j V_j + b_k V_{ki} + b_l V_l \qquad b_i + b_j + b_k + b_l = 1$$

The use of the barycentric coordinates b_i, b_j, b_k, b_l is particularly useful for this application.

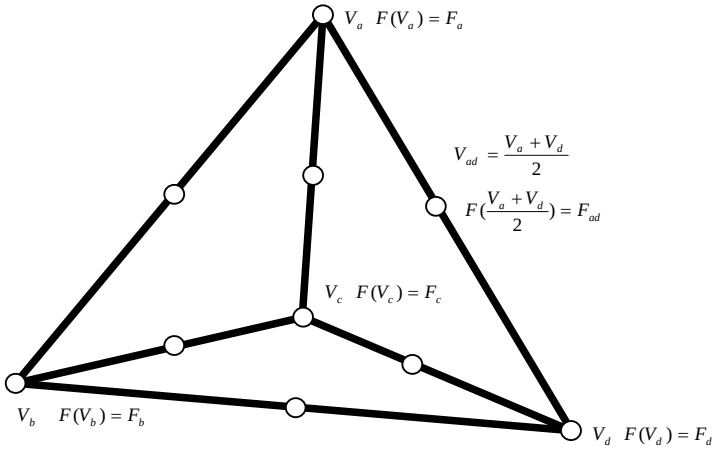


Fig. 5. The notation used for the control points of the basis functions for the piecewise defined quadratic, implicit fitting function.

Normal vector estimates, ∇F_i, at each data point P_i are taken as the gradient of a local, least squares fitting plane. The closest M points $\{Q_k\}, k = 1, \cdots, M$ to P_i are determined and the estimate of the normal vector is taken to be the normalized eigenvector associated with the smallest eigenvalue of the 3x3 matrix

$$\sum_{k=1}^{M} \tilde{w}_k (Q_k - P_i)^t (Q_k - P_i)$$

The weights $\tilde{w}_k$ are chosen on the basis of inverse distance of P_i to Q_k. The inside/outside choice is determined by knowing the position and/or the direction of the scanner. The overall error and fitting criterion is the quantity

$$E\{F_a, F_{bc}\} = \sum_i w_i F^2(V_i) + \sum_i \omega_i \left\| \nabla F(V_i) - \nabla F_i \right\|^2$$

Where the weights w_i and ω_i are chosen to reflect the relative importance or accuracy of the data points as compared to the normal vector estimates.

The task of minimizing $E\{F_a, F_{bc}\}$ is a linear least squares fitting problem which can be solved by conventional iterative techniques. Computation of the necessary quantities is facilitated by the following equations which relate derivatives taken with respect to barycentric coordinates to those in terms of conventional Cartesian coordinates.

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial b_i}\frac{\partial b_i}{\partial x} + \frac{\partial F}{\partial b_j}\frac{\partial b_j}{\partial x} + \frac{\partial F}{\partial b_k}\frac{\partial b_k}{\partial x} + \frac{\partial F}{\partial b_l}\frac{\partial b_l}{\partial x}$$

$$\nabla F(b_i, b_j, b_k, b_l) = \nabla F(x, y, z) = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right)$$

$$\begin{aligned}\frac{\partial F}{\partial x} &= F_i 2b_i \frac{\partial b_i}{\partial x} + F_j 2b_j \frac{\partial b_j}{\partial x} + F_k 2b_k \frac{\partial b_k}{\partial x} + F_l 2b_l \frac{\partial b_l}{\partial x} \\ &+ F_{ij}\left[2b_j \frac{\partial b_i}{\partial x} + 2b_i \frac{\partial b_j}{\partial x}\right] + F_{jk}\left[2b_k \frac{\partial b_j}{\partial x} + 2b_j \frac{\partial b_k}{\partial x}\right] + F_{ik}\left[2b_k \frac{\partial b_i}{\partial x} + 2b_i \frac{\partial b_k}{\partial x}\right] \\ &+ F_{jl}\left[2b_l \frac{\partial b_j}{\partial x} + 2b_j \frac{\partial b_l}{\partial x}\right] + F_{il}\left[2b_l \frac{\partial b_i}{\partial x} + 2b_i \frac{\partial b_l}{\partial x}\right] + F_{kl}\left[2b_l \frac{\partial b_k}{\partial x} + 2b_k \frac{\partial b_l}{\partial x}\right]\end{aligned}$$

Similar formulas hod for $\partial F / \partial y$ and $\partial F / \partial y$.

A piece-wise defined trivariate quadratic will only lead to a C^0 field function, but, in addition, we impose "near" C^1 conditions which are included in the linear systems used for optimal fitting.

2 Examples

2.1 The Ability of the Implicit Model to Capture Features

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 the cell boundaries of the piecewise defined modeling function. This is only a 2D example but it is sufficient to illustrate the desired points and yet easier to perceive than a 3D example. The data points are shown in the left image of Fig. 6 and the piecewise quadratic implicit model is shown in the right image. 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 to the corner feature. This example also points out one potential drawback to the use of implicit models and this is the extraneous contour in the front left region of the domain. But this is not a real problem as these extraneous contours can easily be removed by context of the data relative to a triangle (tetrahedron) cell and the continuity proximity features.

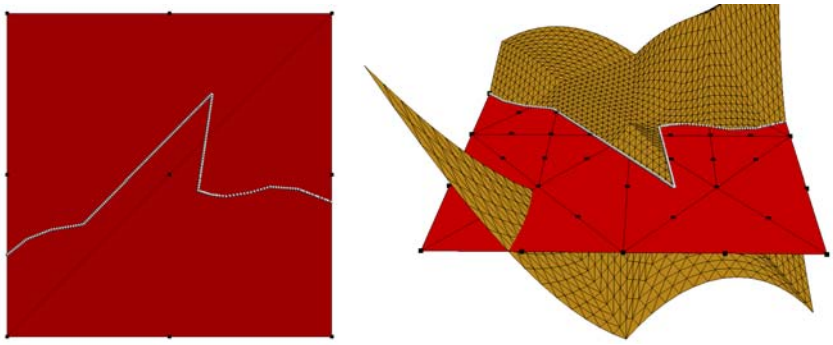


Fig. 6. The data is shown in the left image and the implicit, piecewise quadratic with a level set that approximates the data is shown on the right. For the entire domain the implicit function has a total of 51 scalar parameters which define it. There are 14 F_i control points at the vertices of the triangle mesh and 37 F_{jk} control points associated with edges for a total of 51 parameters which define the implicit fitting function. Only 12 and 21 (respectively) have to me maintained for the support of the contour. The important point of this example is to point out that the proper implicit model can capture features that do not necessarily lie on domain cell boundaries.

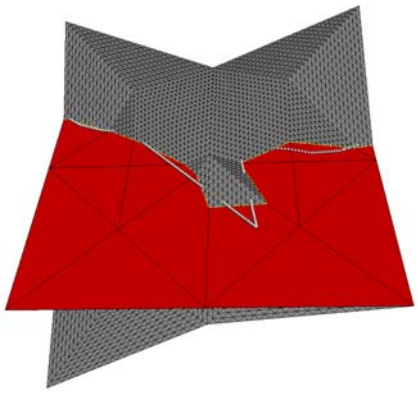


Fig. 7. Piecewise linear implicit fit to the same data as in Figure N. We include this example to point out that it would be rather difficult to model a corner with a piecewise linear implicit modeling function. In order for a piecewise linear model to fit a corner, the corner would have to be on the boundary between domain cells. This means we would have to know the location of the corner in advance.

2.2 Artifact Example

For this example, an artifact consisting of a patio door lock has been scanned. This object has many features of corners and edges. As we illustrated in the previous section, the implicit, piecewise quadratic modeling function has the ability to fit these types of features inferred by the point cloud of data.

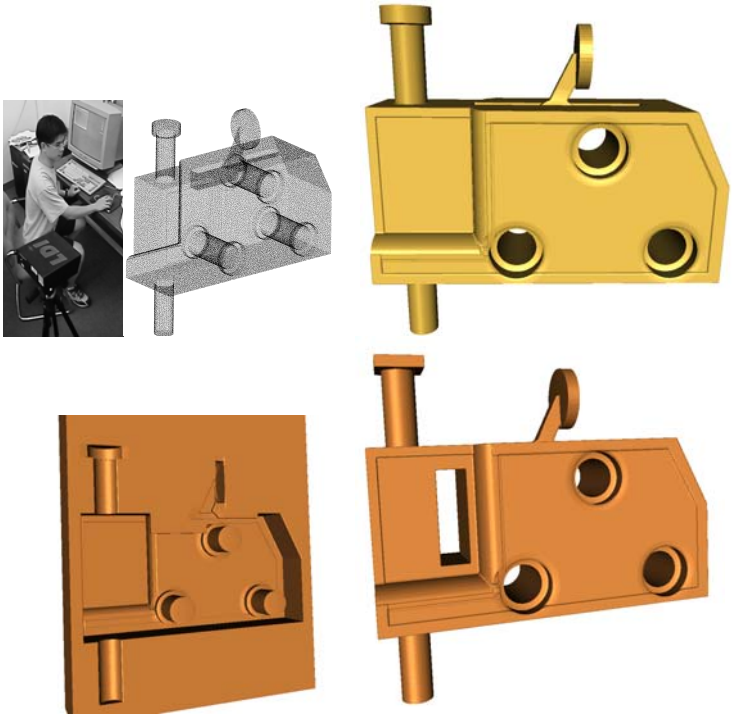


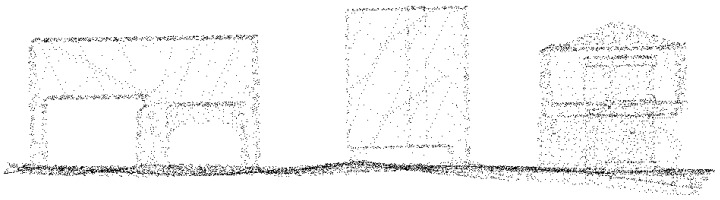
Fig. 8. In the upper left is shown the scanning process using the LDI scanner and the resulting point cloud. The upper right image is the level set of the implicit model where it can be noted that the sharp features and detail inferred by the point cloud are captured by the implicit modeling process. The lower two images illustrate some of the positive features of an implicit model. In the lower left image a simple Boolean operation of intersection is used to make a mold (left portion) of the artifact. In the lower right image a modified object is made with Boolean operatio0ns. A notch has been cut out and a square top has been put on the latch pin. For this model, 1943, F_i, control points and 5701, F_{jk}, control points are needed to represent the implicit functions whose contour is this object.

In general, the Boolean operations of union and intersection are useful for scene analysis. They can be used to eliminate objects allowing the consideration of hypothetical situations and they can be used to relocate objects in a scene. Boolean operations for explicitly defined geometry such as TMS or TINs require extensive, tedious and error prone computations. Boolean operations for implicitly defined

geometry is efficiently realized with simple operations applied to the associated field functions. If an object A is defined by the field function F_A such that $A = \{(x,y,z): F_A(x,y,z) \geq 0\}$ and if B is another three dimensional point set with the field function F_B, then it is easy to see that the *union* is defined by $A \cup B = \{(x,y,z): Max(F_A(x,y,z), F_B(x,y,z))\}$ and the *intersection* $F_{A \cap B} = Min(F_A, F_B)$. Boolean operations are illustrated in the lower two images of Fig. 8.

2.3 Urban Terrain Example

In Fig. 9, we show results of our methods applied to data which is obtained from an urban terrain scene where we have normal geographic terrain along with artifacts (buildings) in the same scene. This example utilizes the "longest edge" refinement strategy. The data set consists of 2.1 million data points. The implicit fit for the support of the contour has 25,123 control points F_i and 101,487 contol points F_{jk}. The fit has an RMS error of 0.013%. The time to compute this model is 43 minutes on a 2.1 GHz PC.



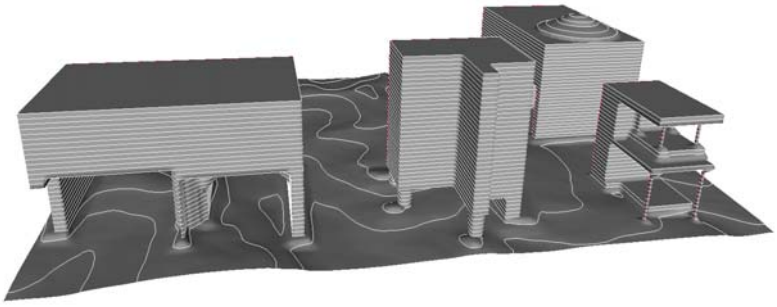


Fig. 9. Upper image shows approximately 1% of the point cloud data set of an urban terrain data set consisting of artifacts located over topographically terrain. The middle and bottom images are two different views of the contour plot of an implicit model computed by the adaptive, piecewise quadratic technique. This contour plots clearly show the sharp features preserved by the modeling process.

Acknowledgments. We wish to acknowledge the support of the United States Army Research Office under contract W911NF-05-1-0301. Also, we wish to thank Adam Huang, Ryan Homes, Gary Graf, Kalpesh Shah, Yosei Sugiyama, Liyan Zhang, Steve Sylvester, Eric Olander, Dave Holliday and Sean Williams for their contributions to this project.

References

1. Adamy, U., J. Giesen, M. John,: New Techniques for Topologically Correct Surface Reconstruction. Proceedings of Visualization ’00 (2000) 372-380
2. Amenta, N., Choi, S., Dey, T. K., Leekha, N.: A Simple Algorithm for Homeomorphic Surface Reconstruction, Proceedings of the 16th Annual ACM Symposium on Computational Geometry (SCG ’00) (2000) 213-222
3. Edelsbrunner, H., Mücke, E.: Three-Dimensional Alpha Shapes, ACM Transactions on Graphics 13, No. 1 (1994) 43-72
4. Floater, M. S., Reimers, M.: Meshless Parameterization and Surface Reconstruction, Computer Aided Geometric Design 18 (2001) 77-92
5. Floater, M. S.: Parameterization of Triangulations and Unorganized Points, In: Tutorials on Multiresolution in Geometric Modelling, A. Iske, E. Quak and M. S. Floater (eds.), Springer (2002) 287-316
6. de Floriani, L., Marzano, P., Pupp E.: Hierarchical Terrain Models: Survey and Formalization, In: Proceedings SAC ’94, Phoenix (1994) 323-327
7. Fowler, R., Topographic Lidar. In: Maune, D.F.: Digital Elevation Model Technologies and Applications: The DEM Users Manual, Bethesda, Maryland: American Society for Photogrammetry and Remote Sensing (2001) 207-236
8. Franke, R., Nielson, G. M.: Smooth Interpolation of Large Sets of Scattered Data, International Journal of Numerical Methods in Engineering 15 (1980) 1691-1704

9. Franke, R., Nielson, G. M.: Scattered Data Interpolation and Applications: A tutorial and survey, In: Hagen, H., Roller, D., (eds) Geometric Modelling: Methods and Their Applications, Spring (1990) 131-160
10. Hodgson, M.E., Jensen, J.R., Schmidt, L., Schill, S. and Davis, B.: An evaluation of LIDAR- and IFSAR-Derived Digital Elevation Models in Leaf-on Conditions with USGS Level 1 and Level 2 DEMs. Remote Sensing of the Environment 84 2 (2003) 295-308
11. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface Reconstruction from Unorganized Points, In: Proceedings of SIGGRAPH 1992, ACM Press/ACM SIGGRAPH, New York. Computer Graphics Proceedings, Annual Conference Series, ACM (1991) 71-78
12. Kelley, A., G. Nielson & M. Malin: Terrain simulation using a model of stream erosion, SIGGRAPH, Computer Graphics, Vol. 22, No. 4 (1988) 263-268
13. Lee, H.S. and Younan, N.H.: DTM Extraction of Lidar Returns via Adaptive Processing. IEEE Transactions on Geoscience and Remote Sensing 41, 9 (2003)2063-2069
14. Maas, H.-G. and Vosselman, G.: Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data, ISPRS Journal of Photogrammetry & Remote Sensing 54 (1999) 153-163
15. Maubach, J. M.: Local Bisection Refinement for N-Simplicial Grids Generated by Reflection, SIAM Journal of Scientific Computing, 16 1 (1995) 210-227
16. Molino, N., Bridson, R., Teran, J., Fedkiw, J. R.: A Crystalline, Red Green Strategy for Meshing Highly Deformable Objects with Tetrahedra", Proceedings, 12th International Meshing Roundtable, Sandia National Laboratories (2003) 103-114
17. Mueller, H.: Surface Reconstruction – An introduction, In: Scientific Visualization (Hagen, H., Nielson, G., Post, F. eds.), IEEE Computer Society Press (1999) 239-242
18. Nielson, G. M.: Scattered Data Modelling, Computer Graphics and Applications 13 (1993) 60-70
19. Nielson, G. M.: Tools for Triangulations and Tetrahedrizations, Scientific Visualization, G. Nielson, H. Hagen and H. Mueller, eds., Academic Press (1997) 429-526
20. Nielson, G. M.: Volume Modeling, In: Volume Graphics, M. Chen, A. E. Kaufman and R. Yagel, eds., Springer (2000) 29-48
21. Nielson, G. M.: On Marching Cubes, IEEE Transactions on Visualization and Computer Graphics 9 (2003) 283-297
22. Petzold, B., Reiss, P., Stössel, W.: Laser Scanning—Surveying and Mapping Agencies are using a new Technique for the Derivation of Digital Terrain Models, ISPRS Journal of Photogrammetry & Remote Sensing 54 2/3 (1999) 95-104
23. Rivara, M. C.: Local Modification of Meshes for Adaptive and/or Multigrid Finite Element Methods, Journal of Computaiton and Applied Mathematics 36 (1991) 79-89
24. Schumaker, L.: Fitting Surfaces to Scattered Data, In Approximation Theory II, G. G. Lorentz, C. K. Chui and L. L. Schumaker, eds., (1976) 203-268
25. Weibel, R.: Models and Experiments for Adaptive Computer-Assisted Terrain Generalization. Cartography and Geographic Information Systems 19 3 (1992) 133-153
26. Zhao, H.-K., Osher, S.: Visualization, Analysis and Shape Reconstruction of Unorganized Data Sets, In: Geometric Level Set Methods in Imaging, Vision and Graphics, Springer (2002) 19-43