

## References

- [ 1] G. Nielson, CAGD Top 10: What to Watch, IEEE CG&A, Vol. 13, No. 1, Jan. 1993, pp. 35-37
- [ 2] G. Nielson, Scattered Data Modeling, IEEE CG&A, Vol. 13, N. 1, Jan. 1993, pp. 60-70
- [ 3] Data source: Mandalay Scientific, Inc., 1098 East Thornhill Place, Frederick, MD 21702, USA
- [ 4] G. Guy and G. Medioni, Inference of Surfaces, Edges and Junctions from Sparse 3D Points, IEEE Int'l Symposium on Computer Vision, Florida, Nov. 1995
- [ 5] W. Dahmen, C. Micchelli, and H. Seidel, Blossoming Begets B-Spline Bases Built Better by B-Patches, Mathematics of Computation, 1(1), 97-115, Jul. 1992
- [ 6] P. Fong and H. Seidel, An Implementation of Triangular B-Spline Surfaces over Arbitrary Triangulations, CAGD, 10, 267-275, 1993
- [ 7] R. Pfeifle and H. Seidel, Fitting Triangular B-Splines to Functional Scattered Data, Graphics Interface, 26-33, 1995
- [ 8] R. Pfeifle and H. Seidel, Spherical Triangular B-Splines with Applications to Data Fitting, Eurographics, 89-96, 1995
- [ 9] S. Auverbach et al., Approximation and Geometric Modeling with Simplex B-Splines Associated with Irregular Triangles, CAGD, 8 (1991), 67-87
- [10] G. Farin, From Conics to NURBS: A Tutorial and Survey, IEEE CG&A, 12, 1992, 78-86
- [11] L. Piegl, On NURBS: A Survey, IEEE CG&A, 11, 1991, 55-71
- [12] L. Piegl and W. Tiller, The NURBS Book, Springer-Verlag, 1995
- [13] H. Qin and D. Terzopoulos, Triangular NURBS and their Dynamic Generalizations, personal communications.
- [14] Y. Shirai, 3-Dimensional Computer Vision, Chapters 3, 7 (Edge Detection), Springer-Verlag, 1987
- [15] Y. Leclerc and S. Zucker, The local structure of image discontinuities in one dimension, IEEE Trans. on PAMI 9, 1987, 341-355
- [16] Y. Leclerc, Capturing the Local Structure of Image Discontinuities in Two Dimensions, IEEE CVPR, Jun. 1985
- [17] A. Blake and A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, 1987
- [18] D. Terzopoulos, Regularization of inverse visual problems involving discontinuities, T-PAMI 8, 1986, 413-424
- [19] P. Alfeld, M. Neamtu and L. Schumaker, Bernstein-Bezier Polynomials on Spheres and Sphere-like Surfaces, CAGD, to appear.
- [20] P. Alfeld, M. Neamtu and L. Schumaker, Fitting Scattered Data on Sphere-like Surfaces using Spherical Splines, CAGD, to appear.
- [21] W. Hohenberger and T. Reuding, Smoothing Rational B-spline Curves Using the Weights in an Optimization Procedure, CAGD, Dec. 1995, 837-848
- [22] G. Dobson, W. Waggenspack, and H. Lamousin, Feature Based Models for Anatomical Data Fitting, CAD, vol. 27, no. 2, pp 139-146, 1995
- [23] H. Qin and D. Terzopoulos, Dynamic NURBS Swung Surfaces for Physics-based Shape Design, CAD, vol. 27, no. 2, pp 111-127, 1995
- [24] P. Laurentgengoux, M. Mekhilef, Optimization of a NURBS Representation, CAD, vol. 25, NOV, 1993, 699-710
- [25] V. Theodoracatos and D. Calkins, A 3-D Vision System Model for Automatic Object Surface Sensing, (using NURBS), Int'l J. of Comp. Vision, vol. 11, no. 1, Aug. 1993, 75-99
- [26] L. Creswell et al., Mathematical Modeling of the Heart using MRI, (using NURBS), IEEE Trans. Medical Imaging, vol. 11, no. 4, Dec. 1992, 581-589
- [27] W. Lorensen and H. Cline, Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm, Siggraph, Jul. 1987
- [28] M. Garcia, Fast Approximation of Range Images by Triangular Meshes Generated through Adaptive Randomized Sampling", IEEE Conf. Robotics and Automation, Nagoya, Japan, May 21-27, 1995, 2043-2048
- [29] C. Grimm and J. Hughes, Modeling Surfaces of Arbitrary Topology using Manifolds, SIGGRAPH, 359-368, 1995
- [30] R. Franke, Smooth Interpolation of Scattered Data by Local Thin-Plate Splines, Comp. Math. Appls., 8, 273-281, 1982

## 5 Summary and Future Work

In this paper we have proposed a new scheme for modeling and visualizing sparse noisy scattered data that may contain unspecified discontinuity edges and junctions. At first, we use a voting technique to infer dense surface/edge/junction potential information from the noisy sparse scattered data. The voting procedures make use of several general perceptual grouping constraints, such as smoothness, co-surfacity, proximity, and curvature. The voting is robust to noise, efficient, and can handle arbitrary topology. Then we drop a quadratic triangular NURBS deformable surface coupled with active edges in the inferred potential fields. After some energy minimization iterations by adjusting the control points, the discontinuity edges and junctions are automatically aligned and detected, and then preserved by setting the knot configurations in constructing the final  $C^1$  smooth surface. For tasks with high precision requirements, fine-tuning and fairing the surface by adjusting the weights are also necessary. Our work demonstrates the advantages of triangular NURBS over rectangular NURBS: lowest degree (quadratic for  $C^1$ ), arbitrary triangulation (no trimming), no pole artifact for genus-0 shapes, etc.

In this paper we only describe open surface and spherical surface modeling from scattered data. To represent surfaces with arbitrary topological types (i.e., objects with several holes), a few primal NURBS surfaces are needed and the  $G^1$  condition must be met if the transition between the primal surfaces is smooth. Many conventional approaches abut adjacent primal surface by solving continuity constraint equations. Grim and Hughes have recently extended the idea of continuous blending in B-splines to manifolds in modeling arbitrary topology [29]. In this scheme, the adjacent primal surfaces are overlapped a bit and blended together along the boundaries, similar to the overlap/blend of basis functions inside each primal B-spline surface. Franke also reported similar overlap/blend scheme in the smooth thin-plate spline interpolation of scattered data [30]. We are now extending TriNURBS to arbitrary topology: first trace out the topology information (a dense triangular mesh) from the potential volume fields using the Marching-cube algorithm, then perform mesh reduction and surface-fitting/edge-alignment, finally construct a  $G^1$  overlap/blend TriNURBS surface with preserved discontinuities, which is an adaptive and multiresolution representation. It seems that this scheme is also valid for non-manifolds.

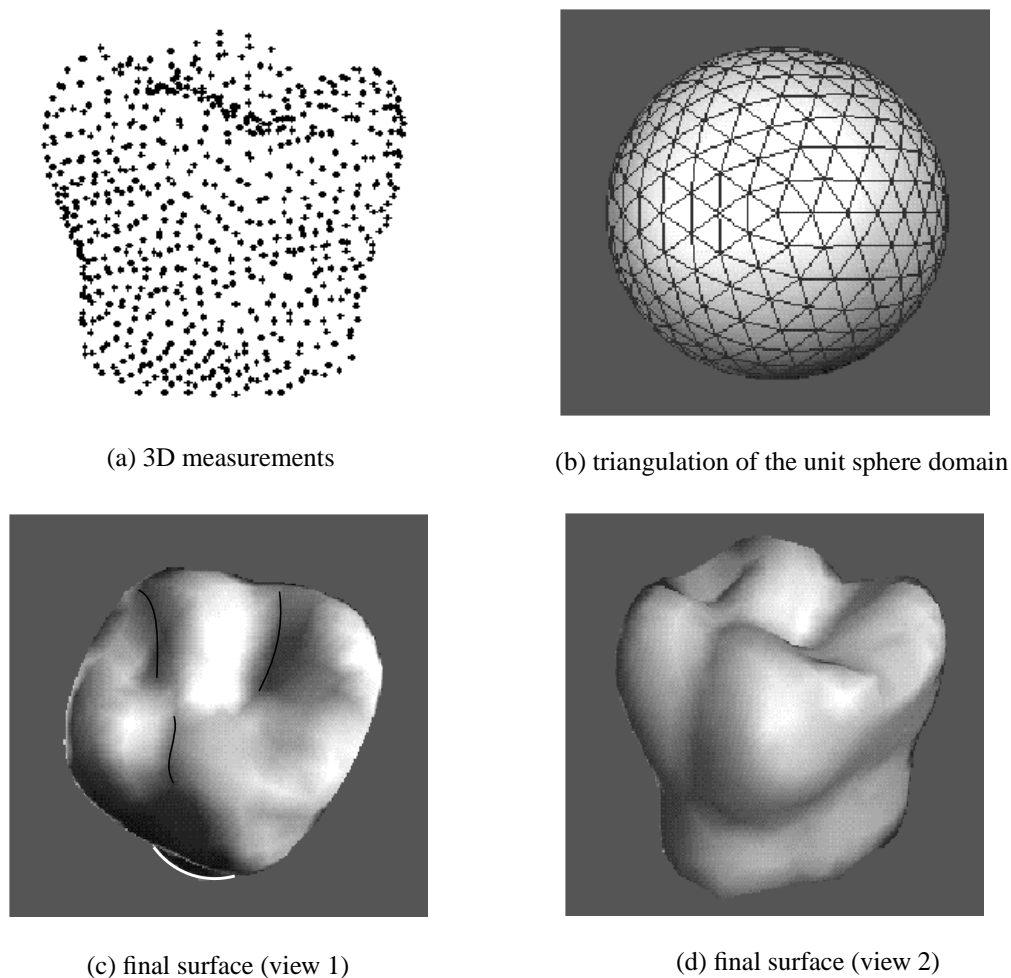


Figure 8. A tooth

Figure 8 (a) shows 650 points measured on a tooth (a closed shape). The domain unit sphere tessellation is shown in (b), and the initial surface is specified by a sphere enclosing the data set. A top view of the final  $C^1$  surface with detected and preserved edges is given in (c). Another view is given in (d). In dental CAD/CAM, preserving the sharp edges on a tooth is very important, otherwise an upper tooth cannot bite tightly with the lower tooth. On the other hand, the sides of the tooth must be smooth, or else the machined crown cannot be put on a specific patient's tooth. Our winged B-snakes represented with TriNURBS seem very promising for smooth surfaces with embedded discontinuities. We are to merge the triangles in smooth areas for model simplification; also we are to work on more complicated objects, such as mechanical parts and medical MRI/CT data of human organs.

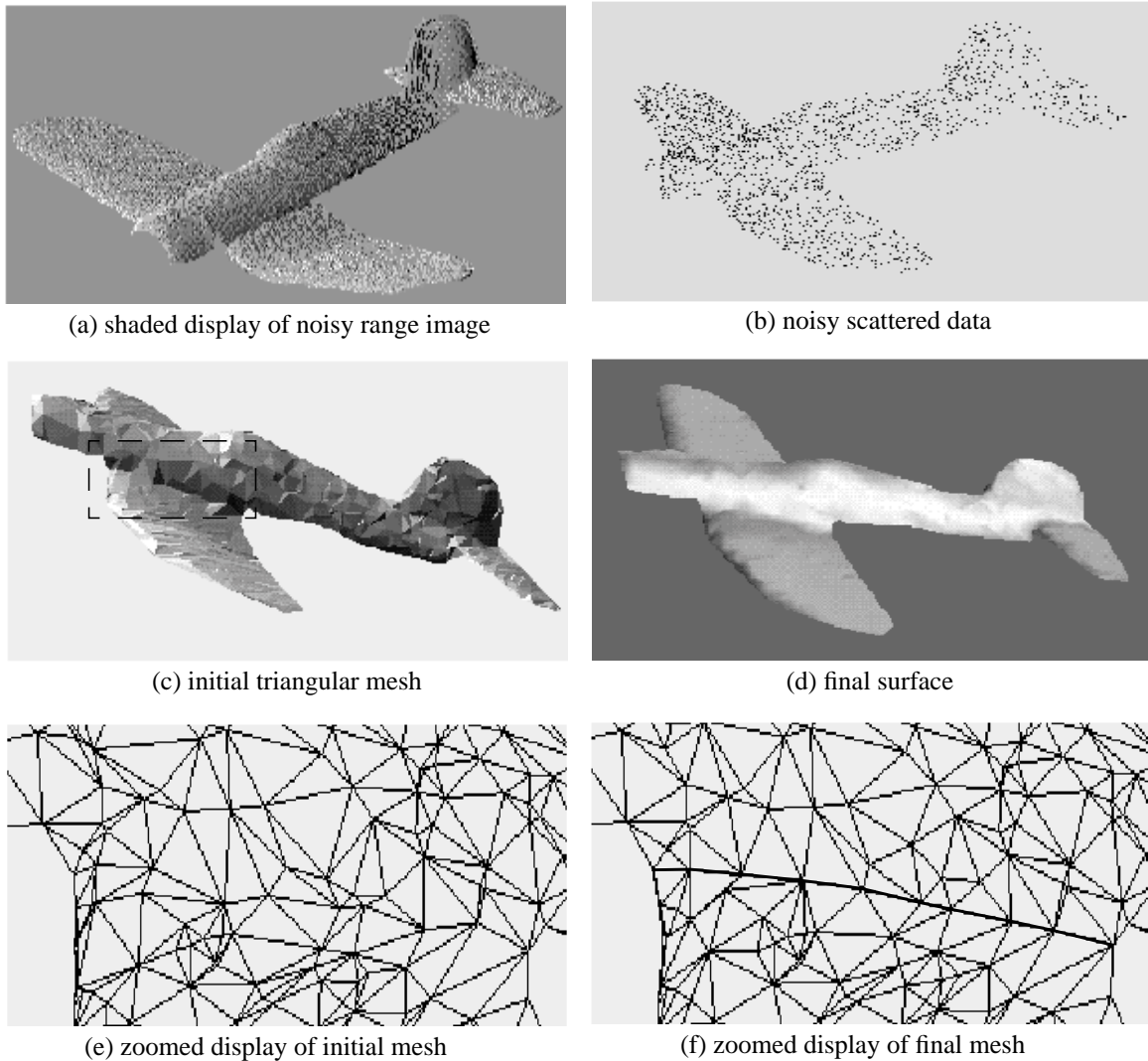
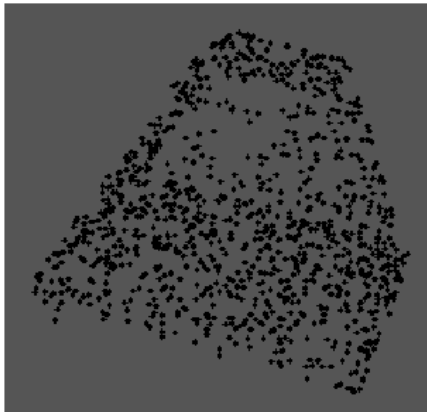
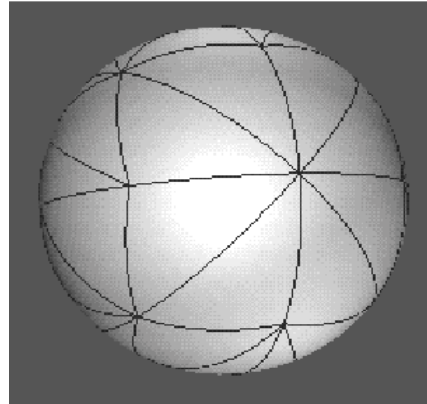


Figure 7. An airplane

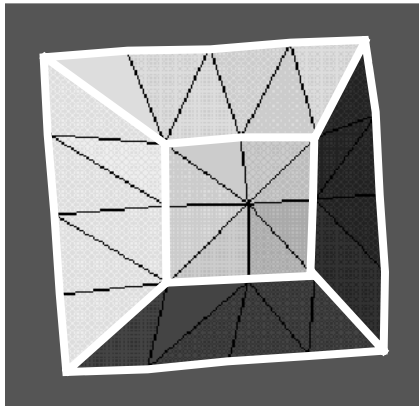
Figure 7 (a) is the shaded display of the range image of an airplane, and (b) shows 900 scattered samples. We apply a Delaunay triangulation algorithm [28] to the scattered data and produce the initial triangular mesh (c), in which some triangle edges severely misalign at the boundary between the wing and the body as seen from the zoomed display in (e). After surface fitting and edge alignment, the edges of the spline triangles align with discontinuity curves in the data and can be detected as shown in (f). Finally, fine-tuning and fairing is performed to give the result in (d). Note that the whole surface is a single quadratic TriNURBS mesh. On contrast, if TP-NURBS are used, we have to trim a rectangular bi-quadratic (quartic) surface into the airplane shape, or stitch-up several surface pieces; both methods are very tedious.



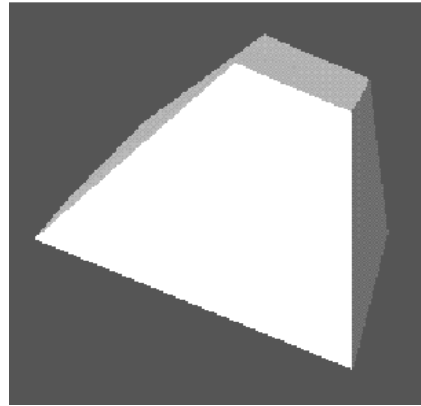
(a) scattered data



(b) triangulation of the unit sphere domain



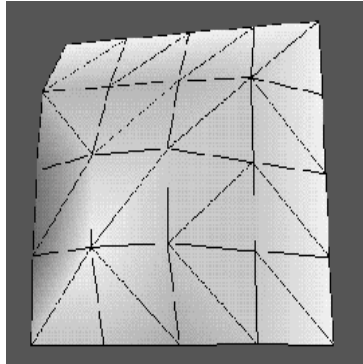
(c)  $C^1$  surface with preserved edges



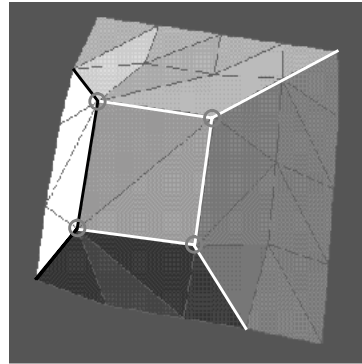
(d) fine-tuning and fairing by adjusting weights

Figure 6. spherical data

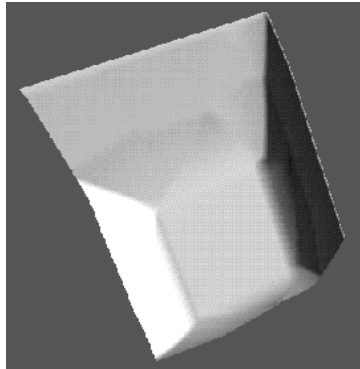
To test the triangular NURBS defined over a unit sphere, we add some samples at the bottom of the pyramid to make it a closed surface, as shown in Figure 6 (a). A triangular tessellation of the unit sphere is given in (b) by subdividing and flipping the diagonals on the faces of a unit cube. The initial surface is specified by a sphere enclosing the data set. After the surface fitting and edge alignment, the  $C^1$  result with preserved edges and junctions is shown in (c). Then, (d) gives the final result after fine-tuning and fairing by adjusting the weights. Every triangle is treated equally, and no pole artifact regions exist at all. This demonstrates the major advantage of TriNURBS over TP-NURBS for modeling spherical data. Previous work using TP-NURBS had to tolerate the pole degeneracy, or use two or more pieces of surfaces and then glue them together.



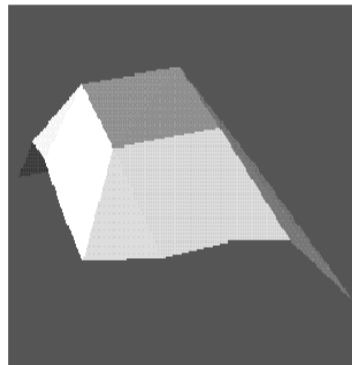
(f) initial surface with misaligned edges



(g) aligned and detected edges/junctions



(h)  $C^1$  surface with preserved edges



(i) fine-tuning and fairing by adjusting weights

Figure 5. A pyramid (cont.)

An initial triangular mesh is needed to start the surface deformation and edge alignment. A dense triangular mesh can be traced out from the volumetric surface potential field by the “marching cube” algorithm [27]. Different from the standard marching cube algorithm, the surface to be traced is now the minimum-potential surface or zero-derivative surface of the potential field. For this experiment, however, we just regularly split the bounding box of the data points into rectangles and then split each rectangle into two triangles along a diagonal. Which diagonal is chosen depends on which yields smaller surface fitting energy within this rectangle. Such an initial surface is shown in Figure 5 (f) with misaligned edges and junctions.

Due to the model’s local control property, a variable’s change has effect only within a single triangle and all its adjacent triangles, so it is reasonable to justify that a local minimization may not sacrifice the global optimality. In our implementation, we adjust only a triangle and all its neighboring triangles at a time, and then move to the next triangle. To do a minimization with all variables would be very slow and sometimes produces worse results. We use the Levenberg-Marquardt algorithm with numerically estimated gradients. Figure 5 (g) shows aligned/detected edges and junctions after adjusting the control points. Conventional approaches will subdivide the triangles into smaller ones due to misalignment to reduce the fitting error and thus results in too many tiny pieces along edges. (h) is the  $C^1$  surface with edges and junctions preserved by pulling apart or setting collinear knots then adjusting the control points again. Finally, (i) gives the result after the fine-tuning and fairing by adjusting the weights. We can see that the edges in (i) are sharper than (h), which indicates the effectiveness of adjusting the weights for final-stage subtle improvement. The three phases for reconstruction from the potential fields run about 10 minutes on an SGI/Indigo.

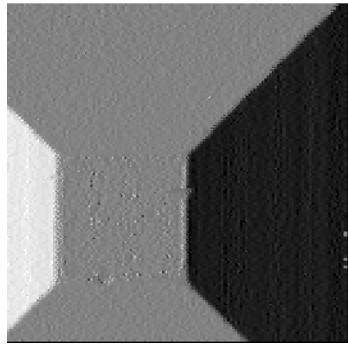
The energy for fine-tuning and fairing is defined as

$$E = E_{surface-fairness} + c \cdot E_{surface-fit}$$

where the coefficient  $c$  is chosen such that the two terms are normalized to 1 : 5 before the minimization starts. Since the surface-fitting has been good enough in phase 3 where the smoothness/fitting ratio is 1 : 20, we can now use smaller ratio (1 : 5). To minimize this energy, only small changes of weights are necessary, and we did not see negative weights occur in our experiments. Initially all weights are 1's, then later they stay between 0.5 ~ 10. We did not use a negative weight penalty term such as  $\{min(0, w_{i,b})\}^2$ .

## 4 Experimental Results

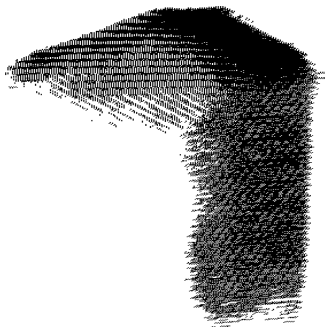
Figure 5 (a) is the shaded display of a dense range image of a pyramid, (b) is the randomly sampled data (400 points) for use in our experiment. (c), (d), (e) display the three volumetric potential fields for surfaces, edges and junctions, respectively (only voxels with potentials below a threshold are displayed). This phase runs about 10 minutes on a SUN Sparc 10, using three 50x50x50 arrays.



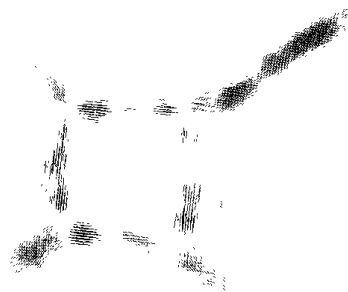
(a) shaded display of laser-scanned range image



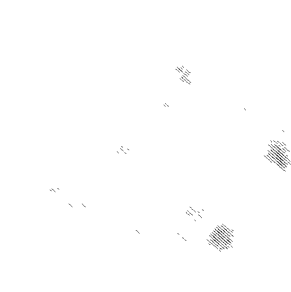
(b) scattered data



(c) potential field for surfaces



(d) potential field for edges



(e) potential field for junctions

Figure 5. A pyramid

locally adaptive thresholding works well. Then we check the change of angles between adjacent edge segments along the detected edges, and those vertices with sharp changes are marked as corners, and those with more than two neighbors are marked as junctions. To build the  $C^1$  smooth surface while preserving the edges and junctions, we pull apart the knots in the continuous regions, and allow knots to be duplicate or collinear/co-circular to respect edges/junctions (at first all three knots are duplicate at each vertex). With the new knot configuration, we adjust the control points once more.

### 3.4 Fine-tuning and Fairing by Adjusting the Weights

In the above fitting and alignment procedures, we never adjust the weights. The reasons are:

(1) *Convex Hull:* We want to exploit the descriptive power of control points as much as possible, since they provide convex hull (first-order approximation) of the final shape. There exist much redundancy in the weights (scaling of weights does not change the shape at all), so leaking information to the weights are not desired. In CAD/CAM industry, the designers usually first specify shapes with the control net, then fine-tune the detailed regions by manipulating the weights; they also display the curvature distribution map, and perform surface fairing by adjusting the weights. Next we shall define the surface fairness similar to the surface smoothness in Section 3.2 but use the second-order derivatives (with respect to  $xx$ ,  $xy$ ,  $yy$ ). Variation of curvature (third-order) might be a good choice for CAD/CAM model reconstruction from high-precision data sensed by coordinate measuring machines, but we did not test it.

(2) *Cost and Convergence:* We tested adjusting the weights at the same time with control points, the deformation and alignment slow down from 10 minutes to 30 minutes; more memory is also needed for the Hessian matrix in the minimization routine. Due to the  $c_{t,b} \cdot w_{t,b}$  multiplication terms, the minimization becomes non-linear. After some experimental tests, we found that if the initial guess is very good, simultaneous adjustment results in a little improvement (about 1~5% further reduction of the total energy), and that if the initial guess is not good, the residual energy may even be larger! Including knots in the minimization makes the situation even worse since the knots are buried in the quadratic basis functions. In a word, it seems that adjusting control points and weights simultaneously is not always better, or is not worth the cost, even if it can bring slight improvement. Due to these considerations, we decided to adjust only control points in the surface fitting phase, and finally adjust weights only for fine-tuning and fairing. These observations about TriNURBS also agree with previous experiments with TP-NURBS [21-26].

in terms of the first-order right-hand derivatives:

$$E_{surface-smooth} = \sum_{t, X} \iint_{uv} \left( \frac{\partial X^2}{\partial u} + \frac{\partial X^2}{\partial v} \right) dudv$$

where the summation is over all triangles and  $(x,y,z)$ , and the integration is over the barycentric coordinates  $(u,v)$  within a triangle. Since the goal of this phase is mainly surface fitting, we just use the above membrane energy. The more costly thin-plate energy based on curvature or second-order derivatives will be used in the final subtle modification and fairing phase. Note that the derivatives above are right-hand derivatives, so they exist even at discontinuous edges, where the left- and right-hand derivatives are unequal, and thus the smoothness constraint is enforced even at these discontinuity edges. In previous work, the smoothness constraint at and near discontinuity edges is released or reduced, so the surface reconstruction near edges become sensitive to noise and unstable. We always enforce the smoothness constraint.

### ***(2) Surface Fitting Energy***

Inside the surface potential field, the surface patches (wings) will flap to reach the local minimum. If the triangle edges cross-over the actual discontinuity edges, the surface energy will be large; when the edges move to align with the actual edges, the surface energy will be reduced. However, we found that the surface energy just makes the edges move a little bit but is not strong enough to pull them to exactly align with the actual discontinuity edges. This is why we shall introduce an explicit edge alignment energy to make the edges become “active” by themselves.

### ***(3) Edge and Junction Alignment Energy***

Inside the edge potential field, the triangle edges (snakes) will slide to the local minimum so that the triangle edges will align with the actual discontinuity edges, instead of crossing over them. Also, the triangle vertices will move to the local minimum in the junction potential field, making the vertices align with the inferred junctions. Such alignment will significantly reduce the total energy or cooperatively improve the surface fitting precision, and we thus do not have to subdivide the triangles into many tiny ones to obtain a good fitting due to the misaligned edges and junctions as in most previous methods.

## **3.3 Building $C^1$ Surfaces with Preserved Edges/Junctions**

After the surface patches have fitted to the data, and the edges and junctions have aligned with actual ones, the detection of discontinuity edges and junctions from the surface is straightforward. We simply need to check the value/derivative differences along the boundary between every pair of adjacent triangles. We normalize the differences by the total area of the pair of triangles. Such

(3) **Estimating the saliency measures for surfaces, edges and junctions:**  $l_{max} - l_{mid}$  can be used as the likelihood of a surface passing through a location X, since if  $l_{max} - l_{mid}$  is large,  $l_{max}$  will be large and  $l_{mid}$  is small, also  $l_{min}$  is small (since  $l_{min} < l_{mid}$ ). Thus, there is only one strong vote group here, i.e., the consistency of votes at this location is high. In other words, the probability of a real surface passing through this location is high.

$l_{mid} - l_{min}$  can be used as an edge saliency measure, since if  $l_{mid} - l_{min}$  is large, we conclude that  $l_{mid}$  is large (so  $l_{max}$  is also large), and  $l_{min}$  is small. Both  $l_{max}$  and  $l_{mid}$  being large means that there are two groups of strong votes at this location, hence this location is an intersection (discontinuity) edge between two surfaces.

$l_{min}$  can be used as the junction saliency measure, since if  $l_{min}$  is large, we can say that  $l_{max}$ ,  $l_{mid}$ ,  $l_{min}$  are all large, which means that there are three or more groups of strong votes at this location, and therefore, this location is more likely to be a junction point of three or more surfaces.

(4) **Estimating potential fields:** Just negate the above three saliency measures, we obtain the potential fields for surfaces, edges and junctions. The minimum potential locations of the three potential fields indicate the existence of surfaces, edges and junctions, respectively. Normal vector estimations are also available at each location.

The above voting procedures are done over each voxel in a 3D grids, and they can work for any number of surfaces of arbitrary topology. Actually they even work for non-manifolds. The voting complexity is  $O(n^3k)$  in general, where  $n$  is the side size of the volume grid, and  $k$  is the number of data points.

## 3.2 Patch Deformation and Edge Alignment

We define the energy  $E$  for the winged B-snakes as follows:

$$\begin{aligned} E &= E_{surface} + E_{edges} + E_{junctions} \\ &= E_{surface-smooth} + w_1 \cdot E_{surface-fit} + w_2 \cdot E_{edge-align} + w_3 \cdot E_{junction-align} \end{aligned}$$

where  $w_1$ ,  $w_2$ , and  $w_3$  are coefficients to balance the effects of the three types of energy. And we shall adjust the control points to minimize the total energy. The three coefficients are chosen by normalizing the above four terms to be 1 : 20 : 5 : 5 before the start of minimization.

### (1) Surface Smoothness Energy

Although each triangular patch is a quadratic polynomial which is always continuous, we still need the smoothness energy to minimize the mesh roughness. The smoothness energy is defined

The goal of the first phase is to infer dense probability measures from the scattered data. The idea is to make use of some human perceptual grouping principles, such as, proximity, co-surfacity, low curvature. Such general constraints are encoded in the majority voting scheme below. There are four sub-steps:

**(1) Voting for the surface normal vector at each data site:** Given a data point P, what is the most likely surface normal vector at another data point X? A plane is assumed to pass through P and X, and thus any orientation perpendicular to the line PX can equally likely be the normal vector for this plane. The length of the normal vector decays with distance PX, and a Gaussian decay function is used. Hence, at each data site, all nearby data points (including itself) will vote for the normal vector at this site. We might imagine each data point as a particle with electronic charge, and the final electromagnetic field vector (strength and direction) at a location depends on all nearby particles. A scatter matrix (or covariance matrix, correlation matrix) is computed at each data site, then the eigenvector corresponding to the largest eigenvalue estimates the most likely normal vector of the surface at this site. Geometrically, such eigen-analysis is equivalent to fitting an ellipsoid and finding the three principal axes. Note that such majority voting scheme is quite insensitive to noise and outliers; also it is more reasonable than nearest-neighbor local fitting, Gaussian interpolation, or other conventional methods. The disadvantage of local fitting is clearly indicated in Figure 4. Gaussian interpolation cannot preserve discontinuities.

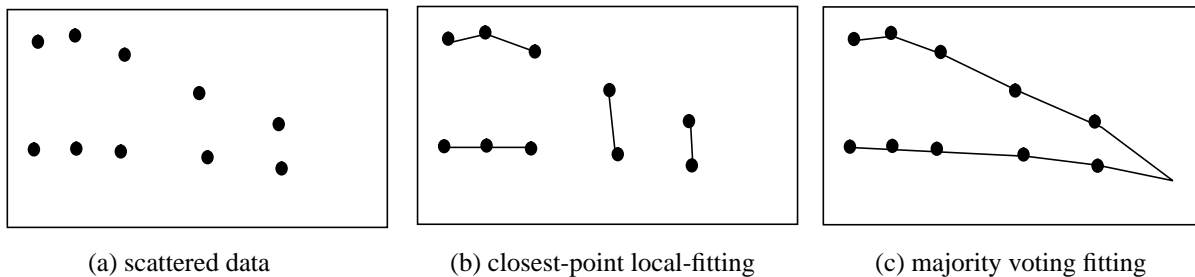


Figure 4. Local fitting vs. perceptual grouping

**(2) Voting at any location in space:** now that we have inferred the normal vector at each data site, we can vote for the likelihood of a surface passing through any location in the space and estimate the most likely normal vector there. For any location X and a data site P with a normal vector, an osculating circle at P can be determined to pass through X. The normal vector of the circle at X is taken as the estimated normal vector at location X, and the length of the normal vector is the Gaussian decay function due to distance PX, times the Gaussian decay function due to the curvature of the circle. After vector vote accumulation from all data sites, a scatter matrix can be computed for this location X, also the three eigenvalues ( $l_{max}$ ,  $l_{mid}$ ,  $l_{min}$ ) and the three eigenvectors.

$$X = \sum_{t=0}^{T-1} \sum_{b=0}^5 c_{t,b} \cdot B_{t,b}(u)$$

where  $X$  is a point  $(x,y,z)$  in 3D space,  $c_{t,b}$ 's are the scaling weights to the basis functions acting as control points (adjacent triangles may share 3D control points to guarantee  $C^0$ , or share their  $xy$  components but not  $z$  component to allow a step edge). If a weight  $w_{t,b}$  is associated to each control point, the above triangular B-spline surface is extended to triangular NURBS surface:

$$X = \frac{\sum_{t=0}^{T-1} \sum_{b=0}^5 c_{t,b} w_{t,b} \cdot B_{t,b}(u)}{\sum_{t=0}^{T-1} \sum_{b=0}^5 w_{t,b} \cdot B_{t,b}(u)}$$

Note that if all the weights are equal, the triangular NURBS specialize to triangular B-splines; furthermore, if the three pulled-apart knots collapse to be duplicate at each vertex, then the overlap/blend effect among adjacent basis functions disappear, and thus triangular B-splines degenerate to triangular Bezier-splines, so the automatic continuity property is lost.

Compared with triangular B-splines, TriNURBS have one more degree of freedom for each control point: the weight. If a weight is zero, the control point has no effect on the surface; if the weight is positive, then control point with larger weight will pull the surface more strongly and thus the surface mimics the control net more closely. Negative weights will push away the surface from the control points and the convex hull property will not hold; also negative weights will probably result in zero denominators and cause unstable computations. Therefore, we do not use negative weights.

If the above domain is a triangulation of the surface of a unit sphere, we define a closed TriNURBS surface. The only difference is that the summation of three spherical barycentric coordinates are not 1 anymore (it is usually greater than 1), and that three or four knots on a same great circle will produce discontinuity. Such spherical representation is very useful in geographical and many other applications [8,19,20]. Since the TriNURBS are defined over arbitrary triangulations of the sphere, there are no pole artifact as in the TP-NURBS, also the continuity are automatically guaranteed. The simplest case is the tetrahedron tessellation of the unit sphere, in which as few as four triangles will suffice to exactly model a sphere. On contrast, to cover a sphere with rectangles, at least six rectangles (like the faces of a cube) are required to obtain the continuity, and a set of constraint equations have to be maintained.

### 3 Triangular NURBS Surface Reconstruction

#### 3.1 Building Potential Fields

## 2 Triangular NURBS Surfaces

Triangular NURBS (TriNURBS) overcome the shortcomings of tensor-product NURBS (TP-NURBS) but still preserve the good properties. TriNURBS are defined over 2D domain triangulations, and  $C^{k-1}$  continuity can be achieved by  $k$ -th degree polynomials defined over the domain triangles. In our work, we consider  $C^1$  surfaces using quadratic polynomials ( $k=2$ ).

Given an arbitrary proper triangulation of the 2D domain (proper means neighboring triangles meet either at a vertex or along a whole edge), two additional points are added near each vertex of the triangle  $[v_i, v_j, v_k]$  to provide nine knots for each triangle (two adjacent triangles share six knots). Then from the nine knots, five knots (including the original vertices  $v_i, v_j, v_k$ , plus two additional knots) are chosen to form a knot set. Six different knot sets are chosen as in Figure 3, and over each set, a basis function is defined.

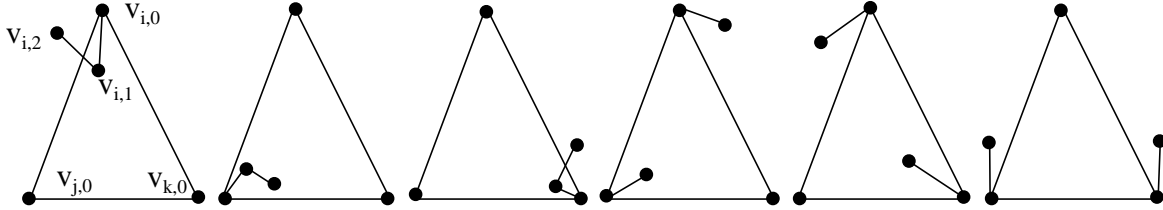


Figure 3. Six knot configurations, each defining a basis function over the convex hull of the five knots.

A basis function  $B$  defined by the 5-knot set  $K$  can be defined recursively as follows:

$$B(u|K) = a_0 B(u|K \setminus v_0) + a_1 B(u|K \setminus v_1) + a_2 B(u|K \setminus v_2)$$

where  $u$  is any point in the 2-D domain,  $a_0, a_1, a_2$  are related to the barycentric coordinates of  $u$  w.r.t. any three knots  $v_0, v_1, v_2$ ;  $K \setminus v_i$  is the 5-knot set  $K$  minus  $v_i$ , i.e., a 4-knot set. As the  $K \setminus v_i$  reduces to only 3 knots, a zeroth-degree B-spline basis function is obtained, which is a flat unit-height triangle:

$$B(u|v_0, v_1, v_2) = \begin{cases} 1 & \text{if } u \in [v_0, v_1, v_2) \\ 0 & \text{otherwise} \end{cases}$$

It can be shown that three collinear knots will result in a crease edge while four collinear knots will produce a step edge. If the nine knots are pulled apart and no collinear knots exist at all, each B-spline basis function automatically becomes  $C^1$ . Linear combination of the six basis functions give rise to a local patch, and over the whole domain triangulation ( $T$  triangles) a surface is obtained:

dicted surface can be estimated. Space curves and 3-D junctions can be easily derived from the same map. We then negate the saliency measures to obtain the potential fields, so that the local-minimum voxels correspond to the most probable positions for surfaces, edges and junctions. Note that this phase imposes no restriction on the number of objects, genus (topology), number of discontinuities, and the algorithm is non-iterative and efficient. In the second phase, a deformable triangular NURBS surface coupled with active edges is dropped into the potential fields. We call the model “winged B-snakes”. After adjusting the control points by some iterations of energy minimization, the surface (wings) will flap to fit the data, and the edges (snakes) will slide to align with the actual edges in the data. Then in the third phase, values and derivatives along each edge are checked, so that discontinuities will be detected and preserved in constructing the surface. Finally the surface can be fine-tuned and faired by adjusting the weights. The flowchart of the whole system is shown in Figure 2. An introduction to triangular NURBS is given in Section 2. In Section 3, we explain the principles of our surface reconstruction, with some experimental results given in Section 4. In the last section, we summarize the work and discuss our future research.

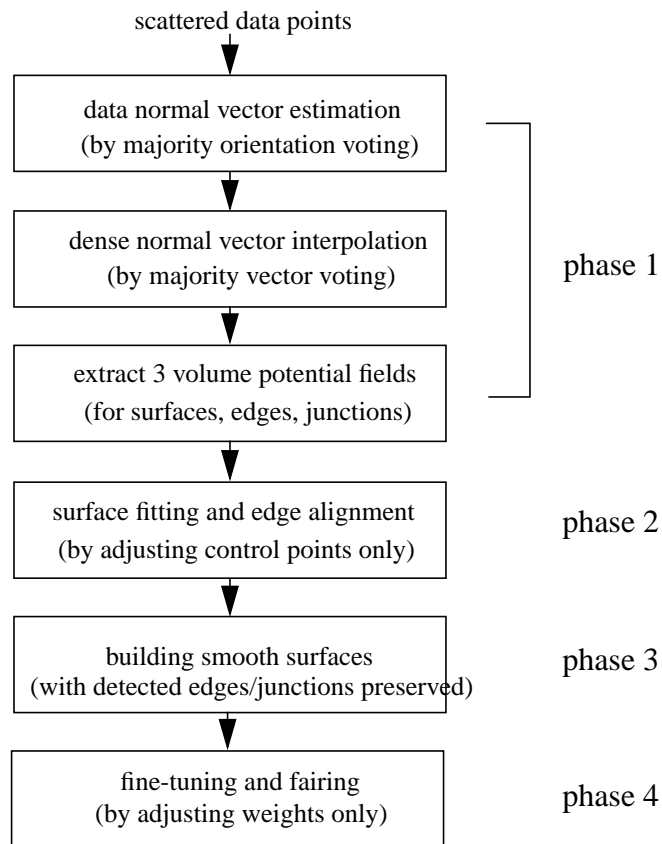


Figure 2. Flow chart of the system

$y=kx+b$ , and the height at the left- and right-hand sides of the line is  $h$  and  $H$ , respectively. Then this disc template is fitted to the data by adjusting  $(k, b, h, H)$ . After convergence, significant difference of  $h$  and  $H$  indicates the existence of a step edge, whose position and orientation are determined by the line  $y=kx+b$ . Leclerc and Zucker [15,16] fitted polynomial curves to the left- and right-hand profiles of a point on the surface, then the values and derivatives of the left- and right-hand profiles at this point are compared to decide if this point is a continuous point or not.

Blake and Zisserman [17], and Terzopoulos [18] were the first to study deformable surfaces coupled with edge detection. The common idea of their schemes is the *adaptive* enforcement of the internal smoothness constraint for deformable surfaces. At likely discontinuous points, the smoothness constraint is turned off so that discontinuity can occur on these points. The implementations are, however, distinct. Blake and Zisserman's approach is mathematically difficult and slow due to the required minimization which involves many continuous variables and also the "on/off" binary variables. Terzopoulos started with very smooth surfaces, then detected discontinuities on the fitted smooth surfaces by seeking zero-crossings of significant bending moments, and a new fitting was done after releasing the smoothness constraint at these points. The key problem of Terzopoulos's approach is in the search for new discontinuities from the fitted surface, since the obtained surface is smoothed already and the discontinuity information might have been lost or changed. After their original work, many related improvements have been done by other researchers following the adaptive smoothness scheme. The differences between those improved approaches are only in when and how the smoothness constraint is totally released or gradually reduced.

### 1.3 Overview of Our Work

We address the problem of surface modeling of sparse noisy scattered data with unknown discontinuities. Our approach consists of four phases. In the first phase, we use Guy and Medioni's perceptual grouping method to derive dense information from sparse data [4]. The idea is to locally enforce the general constraints, which are *co-surfacity*, *proximity*, and *constancy of curvature*. These constraints are encoded into a 3D vector mask. Such a mask, when aligned with an input data site, associates a preferred direction and strength to every voxel in a large volume of space around the input site. By aligning the field with each input site, we produce, at each voxel location, a collection of vector votes. This voting information is then compressed into the second order moments of the vector collection, graphically represented by an ellipsoid, or equivalently, by three orthogonal 3-D vectors. These three vectors are then combined and interpreted as a volumetric saliency map corresponding to surfaces. Such a map holds high values for locations in space which are strong candidates for surfaces. Also, at each such location, a normal to the pre-

From this example, we see that we humans can effectively perceive smooth shapes and detect discontinuities from noisy sparse scattered data without any prior knowledge about the data. This indicates that there are some very general principles in human perception for inferring general shapes and detecting discontinuities. If we can convert such principles into computer programs, we can automatically detect discontinuities and model the smooth surfaces from noisy sparse scattered data, and thus relieve us from the laborious manual work. Such manual work may become infeasible for multidimensional data. Our group [4] have implemented such a perceptual-grouping-based system for computer vision research and a dense triangular mesh and discontinuity curves were automatically extracted. In this paper, we follow the same approach, but upgrade the triangular mesh to triangular NURBS surfaces.

Triangular NURBS (TriNURBS) are rational generalizations of the triangular B-splines [5-9], and overcome the shortcomings of tensor-product NURBS (TP-NURBS) but retain the good properties of local control, automatic continuity, convex hull, affine and projective invariance, completeness, etc. TP-NURBS [10-12] are currently the industrial standard for surface representation, but they have a rectangular topology and thus require tedious trimming techniques to handle pole artifacts and arbitrary boundaries; also, to obtain  $C^1$  continuity, the TP-NURBS should be quadratic in terms of both parameters, so the total degree is four (quartic), whereas TriNURBS can model arbitrary topology and a total degree of as low as two (quadratic) can maintain the  $C^1$  continuity; duplicate knots in TP-NURBS will produce a discontinuity curve across the whole surface, while triangular NURBS can produce discontinuity edges between any desired adjacent knots inside the surface, therefore, a single TP-NURBS mesh cannot model the data in Figure 1, whereas a single TriNURBS mesh can. We do not know of any published work on triangular NURBS reconstruction. Qin and Terzopoulos are working on physics-based triangular NURBS design and reconstruction, but have only reported results on triangular B-splines [13].

## **1.2 Coupling Between Discontinuity Detection and Surface Fitting**

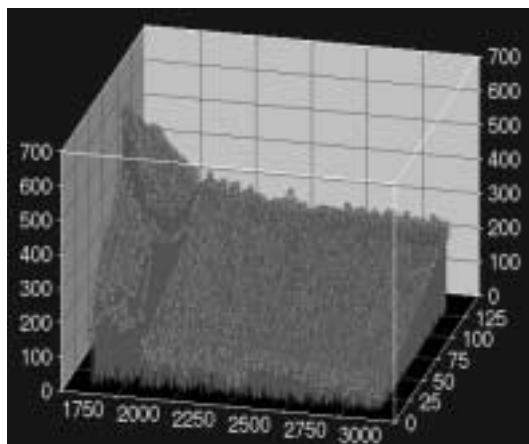
Discontinuity edges are the profile curves along which the surface values or orientations cannot be uniquely defined, such as a crack or a crease. On one hand, we would like to have a faithful surface fitting to the scattered data in order to compute and compare surface properties to detect unknown discontinuities; on the other hand, however, we would like to know the locations of the discontinuities before obtaining a good surface fitting, otherwise the unknown discontinuities will be smoothed out in the surface fitting. This states that the surface fitting and discontinuity detection are coupled processes and should be handled concurrently.

Hueckel [14] used a simple surface model for step edge detection: a local disc is divided by a line

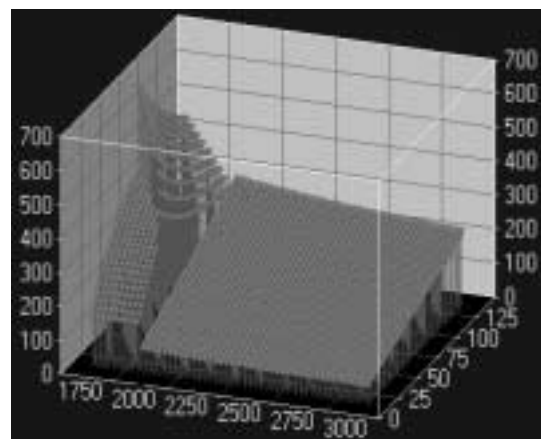
# 1 Introduction

## 1.1 Visualization of Scattered Data with Discontinuities

The essence of visualization is not to provide a naive display, but to assist people to gain insight into the data and discover important trends, structures, relationships, etc. Nowadays visualization tools have become indispensable in medical imaging, product design/inspection, reverse engineering, geographical, meteorological, oceanographic, and atmospheric research, just name a few. Most previous work in visualization assume that the input data are on regular grids (e.g., CT/MRI images and volumes), but for many other applications, the sampled data are scattered and hence we need to interpolate the scattered data into regular grids in order to use the existing softwares. Scattered data modeling was listed as one of the top ten research topics in CAD/CAGD [1,2]. Furthermore, most research papers and commercial softwares on scattered data modeling deal with smooth surfaces only; however, there are many discontinuity phenomena around us, such as steep cliffs, seafloor faults, electric shocks, seismic velocity fields, creases and corners on shapes, etc. Fig. 1 (a) displays the data collected from an electrostatically controlled flow valve of an anesthesiology device, in which Z axis measures the flow of anesthetic gas through the valve, and X and Y axes are the differential pressure across the valve and the applied electrostatic control voltage, respectively [3]. We can see that there are discontinuities in the data, so fitting a smooth surface will lose the discontinuities and fail to reveal the important property of this device. For such data, the data analysts have to manually mark the discontinuity boundaries and then select three different simple functions to fit the data as a mixed model, as indicated in (b).



(a) scattered data with discontinuities



(b) 3 mixed models fitted to the data

Figure 1. Scattered data visualization

# Triangular NURBS Surface Modeling of Scattered Data

Song Han and Gerard Medioni  
Institute for Robotics and Intelligent Systems  
University of Southern California  
Los Angeles, CA 90089-0273  
han@iris.usc.edu, medioni@iris.usc.edu

## Abstract

Scattered data modeling is useful in many scientific fields and industrial applications to reveal the properties and relationships in empirically acquired data sets. If the data contain discontinuities, the data analysts must manually mark the segmenting boundaries before using the current software packages. Here, we automatically detect discontinuities from noisy sparse scattered data and use triangular NURBS surfaces to model and visualize the data. We use Guy and Medioni's global voting method to interpolate from sparse data three dense potential fields for surfaces, edges, and junctions. The global voting interpolants encode several human perceptual grouping principles such as *co-surfacity*, *proximity*, and *constancy of curvature*. The inferred potential fields are stored in three volumetric grids, giving each voxel the *probability* of being a surface point, an edge point, and a junction point. Then we use a new model called "winged B-snakes", which are deformable triangular NURBS surfaces embedded with active curves, to fit the surfaces and align the edges and junctions. Finally, a smooth  $C^1$  surface which preserves discontinuity edges and junctions is constructed. Fine-tuning and surface fairing is automatically done by adjusting the weights. We present experimental results on both functional and spherical data sets.

**Keywords:** Scattered Data, Surface Reconstruction, Discontinuity Detection, Perceptual Grouping, Triangular NURBS, Membrane/Thin-Plate Energy, Marching Cubes, Iso-surface extraction