

1	Introduction	9
1.1	3-D Descriptions from Intensity Images	9
1.2	Object Recognition	9
1.3	Analysis of Range Data	10
2	Inference of Segmented, Volumetric Shape from Intensity Images	11
2.1	Introduction	11
2.1.1	Motivation	11
2.1.2	Summary of previous work	12
2.2	Overview of our approach	12
2.2.1	Properties of Generalized Cylinders	13
2.2.2	GC axis is straight	14
2.2.3	GC axis is curved	14
2.3	Inferring volumetric primitives	15
2.3.1	Extracting groups and labelling contours	15
2.3.2	Inferring volumetric parts for groups with true edges	16
2.3.3	Inferring volumetric parts for groups with limb edges	20
2.3.4	Inferring volumetric parts for groups with a mixture of both edges	21
2.4	Description of more complex objects	23
2.5	Conclusion	24
2.6	References	24
3	Structure and Motion from a Sparse Set of Views	27
3.1	Introduction	27
3.2	Problem Formulation	28
3.3	Feature extraction and matching	29
3.4	Structure and motion estimation	30
3.4.1	Image clustering	31
3.4.2	Local structure and motion estimation	31
3.4.3	Propagation of structure estimation	31
3.4.4	Merging local parameter estimates into a global framework	32
3.5	Experimental Results	32
3.6	Conclusion and Discussion	34
3.7	References	35

4	Generic Shape Learning and Recognition	37
4.1	Introduction	37
4.2	Previous Work	38
4.2.1	Recognition in Image Understanding	38
4.2.2	The Case-Based Reasoning paradigm	40
4.3	Symbolic Hierarchical Generic Shape Description	41
4.3.1	Hierarchical Shape Decomposition	41
4.3.2	Symbolic Description of Parts	42
4.3.3	Extension to 3-D Objects Description	44
4.4	Dynamic Database Organization	44
4.4.1	Hierarchical Indexing of Shape Descriptions	45
4.4.2	Dynamic Evolution of the Database	47
4.4.3	Discussion	47
4.5	Shape Retrieval	47
4.5.1	Exact Matching vs. Partial Matching	47
4.5.2	Dissimilarity Between Shapes	48
4.5.3	An Incremental Partial-Matching Retrieval Algorithm	50
4.5.4	Complexity Study	52
4.5.5	Summary	54
4.6	Experimental Results	54
4.6.1	2-D Shape Hierarchical Descriptions	54
4.6.2	Shape Recognition	57
4.7	Conclusion and Perspectives	62
5	Pose Estimation of Multi-Part Curved Objects	66
5.1	Introduction	66
5.2	Representations	67
5.2.1	Image Objects	67
5.2.2	Model Objects	68
5.3	Description and Matching	69
5.4	Pose Estimation	70
5.4.1	Coordinate Systems and Pose Parameters	70
5.4.2	Establishing Correspondences for Single-Part Objects	71

5.4.3	Establishing Correspondences for Multi-Parts Objects	73
5.4.4	Solving for the Pose Parameters	74
5.5	Conclusion.	76
6	Inference of Surfaces, Curves, and Junctions from Sparse Data	79
6.1	Introduction.	79
6.2	Previous work.	80
6.3	Overview of our approach	80
6.4	The design of the fields.	82
6.4.1	The Diabolo field.	82
6.4.2	The 3-D curve segment field.	83
6.4.3	The 3-D point field	84
6.5	Implementation of the algorithm	84
6.5.1	Vector convolution and vote aggregation	85
6.5.2	Representation at each voxel	85
6.5.3	Vote interpretation.	86
6.5.4	High-level feature extraction	87
6.6	Complexity	91
6.7	Results	91
6.7.1	Oriented input	91
6.7.2	Non-oriented input.	91
6.7.3	Real data	92
6.8	Conclusion.	93
6.9	References.	94
7	Surface Approximation and Segmentation of Objects with Unknown Topology	97
7.1	Introduction.	97
7.2	Goal.	98
7.3	Overview of the approach in 2-D.	99
7.4	The 3-D case	100
7.4.1	Bézier Connecting surface.	102
7.4.2	Point correspondence	103
7.4.3	Cross-border tangent	103
7.4.4	Connecting surface construction	104

7.4.5	Determining the outward direction of the surface normal	104
7.4.6	Fitting connecting surface to data points nearby.	104
7.5	Initial guess	105
7.5.1	2-D pattern initialization	105
7.5.2	3-D object initialization	105
7.5.3	Connecting surface initialization	106
7.6	Experimental Results	107
7.7	Conclusion and future work	110
8	Triangular NURBS Surface Modeling of Scattered Data	116
8.1	Introduction.	116
8.2	Triangular NURBS Surfaces.	117
8.3	Surface Inference	118
8.3.1	Grouping Stage	118
8.3.2	Patch Deformation and Edge Alignment.	119
8.3.3	Building C1 Surfaces with Edges/Junctions	120
8.3.4	Fine-tuning/Fairing by Adjusting Weights	121
8.4	Results	121
8.5	Conclusions	124
8.6	Acknowledgment	124
8.7	References.	125
9	List of Publications	128
10	Professional Personnel	129
10.1	Personnel	129
10.2	Ph. D. Graduates	129

Figure 2.1 Description of approach	13
Figure 2.2 GC with straight axis	13
Figure 2.3 Examples of GCs - with straight and curved axes	14
Figure 2.4 Three views of a monitor (a), matched groups (b), labelled edges (c)	15
Figure 2.5 Examples of surfaces generated from groups with edges labelled “true”	16
Figure 2.6 Computation of the GC axis for GCs with true edges. GC axis is (a)straight (b) curved.	17
Figure 2.7 Computed GC axis and cross section for the PC monitor example	17
Figure 2.8 Hypothesizing cross sections	18
Figure 2.9 Inferred volumetric shape of the PC monitor.	19
Figure 2.10 Inferred volumetric description for a tool	19
Figure 2.11 Segmented parts of the tool and the computed symbolic description	20
Figure 2.12 Inferred volumetric description of a bottle	20
Figure 2.13 Parts of bottle(above) and viewpoints with texture mapped on surface (below).	21
Figure 2.14 Examples of GCs which generate groups with a combination of true and limb edges.	22
Figure 2.15 Approximating the cross section when no cross section is seen	22
Figure 2.16 Processing the wooden block images.	23
Figure 2.17 Decomposition of the Renault Piece	25
Figure 3.1 Overview of our approach	28
Figure 3.2 Synthetic images.	32
Figure 3.3 Two views of the computed data and the actual data. Computed data are represented as dots (left) and actual data are connected by lines (right)	33
Figure 3.4 Real image	33
Figure 3.5 Three views of the recovered structure showing the top and front (left), the back (center), and the bottom (right) part of the Renault piece.	34
Figure 4.1 Decomposition of an “f106” shape (from [51]).	41
Figure 4.2 Directed connection graph obtained from the decomposition of the “f106” shape shown in Figure 4.1 (hierarchy of depth 3).	42
Figure 4.3 Local geometrical characterization of parts: (a) symmetry; (b) curvature; (c) termination.	43
Figure 4.4 Relative parameters for part description: (a) size of the part (normalized length of axis); (b) relative angle between the part axis and its superpart axis; (c) type of connection.	44
Figure 4.5 I-Structure derived from the “f106” shape symbolic description.	45
Figure 4.6 Example of hierarchical index, discriminating shapes on the structure of their description and on local characterization of parts.	46
Figure 4.7 Example of comparison structure instantiated between two hierarchical shape descriptions. Only a few links are shown to preserve readability.	49
Figure 4.8 Decomposition of an “f16” shape (from [51])	56
Figure 4.9 Symbolic skeleton for the “F16” shape shown in Figure 4.8	56
Figure 4.10 I-Structure associated with the “F16” shape shown in Figure 4.8.	56
Figure 4.11 Case built for an “F4” shape (a) and corresponding symbolic skeleton (b).	58
Figure 4.12 Occluded shapes.	58

Figure 4.13 View-point variation simulation: symbolic skeleton of the altered shape.	
59	
Figure 4.14 Average values of c_1 against n for engines (2), (3), (4).	61
Figure 4.15 Average number of distances computed between parts with engines (1), (2), (3) and (4).	62
Figure 4.16 Average total number of operations computed between parts (distances+compatibility tests) with engines (1), (2), (3) and (4).	62
Figure 5.1 Descriptions (2-D) extracted from an image.	68
Figure 5.2 Object-centered part coordinate system for SHGC (a); for PRGC (b).	69
Figure 5.3 Representation of one of the model objects used in the experiments. Left: side view of the object also showing axes. Right: scaling functions of the object's parts (SHGC function is shown first).	69
Figure 5.4 Sequence of transformations applied to the model object OM so that its image coincides with OI.	71
Figure 5.5 Finding correspondences between image cross-sections and model axis positions using the similarity of the invariant scaling ratios.	72
Figure 5.6 Axes points correspondences between image and model SHGCs	72
Figure 5.7 Image-model SHGC axis correspondences	73
Figure 5.8 Using the SHGC axes correspondences to find PRGC axes correspondences.	74
Figure 5.9 An example of image-model PRGC axes correspondence (the corresponding points have tangents parallel to the SHGC axes).	74
Figure 5.10 Overlay of the model objects with their estimated poses on the edges of the intensity image of Figure 5.1 . (a); a top view of the objects (b).	76
Figure 5.11 Other results. Top: an image of a previous object (back object of Figure 5.1) and its estimated pose. Bottom: another object.	76
Figure 6.1 Input data and symbolic representation	79
Figure 6.2 Flowchart of our system	81
Figure 6.3 Design of the field	82
Figure 6.4 The Diabolo field	83
Figure 6.5 The general shape of the curve segment field.	84
Figure 6.6 The point field	85
Figure 6.7 The three important voting ellipsoids.	86
Figure 6.8 Curve saliency	88
Figure 6.9 Different possible labels of voxel faces.	89
Figure 6.10 Surface saliency	90
Figure 6.11 3 Planes.	91
Figure 6.12 Final description of the 3 planes.	92
Figure 6.13 Recovering a complex non-constant curvature sheet surface.	92
Figure 6.14 Dealing with a genus 1 object	93
Figure 6.16 The egg	93
Figure 6.15 Wooden block	94
Figure 7.1 Two examples of decomposition in 2-D.	100
Figure 7.3 Flowchart of our approach	100
Figure 7.2 Further refinement on the 2-D and 3-D fitting result.	101
Figure 7.4 Connecting surface between two objects.	102

Figure 7.5	A example of the cross-border tangent.	102
Figure 7.6	The connecting surface boundary on the object is a closed curve C. We tune the control points on the (u,v) parameter space (that is, Pp1, Pp2, Pp3, Pp4, and Pp5) to adapt the connecting surface (and the boundary curve C also) to the data points nearby connecting this pair of objects.	102
Figure 7.7	Initial Snake in 2-D	105
Figure 7.8	Example of initial surface for a 3D object.	105
Figure 7.9	Boundary curve in 3-D and parameter spaces	106
Figure 7.10	An illustrative example showing how to initialize the boundary curves of the connecting surface.	107
Figure 7.11	Segmentation and surface fit for a complex synthetic object.. . . .	108
Figure 7.12	Boundary curves in parameter space.	109
Figure 7.13	Experiment on a Genus 1 object (teapot).	109
Figure 7.14	Add a connecting surface when the target is divided into two elements due to the insufficient number of control points for the fitting surface. . .	110
Figure 8.1	Flow chart of the grouping stage	119
Figure 8.2	Open pyramid	123
Figure 8.3	Closed pyramid.	124
Figure 8.4	An airplane (open surface).	125
Figure 8.5	A tooth (closed shape)	126

8 Triangular NURBS Surface Modeling of Scattered Data

8.1 Introduction

Most research papers and software packages on scattered data modeling deal with smooth surfaces only [138]-[140]; however, discontinuities abound: steep cliffs, seafloor faults, electric shocks, seismic velocity fields, creases and corners on shapes, etc. Marking discontinuities manually is very tedious and may become infeasible for multidimensional data. Guy and Medioni [140] have implemented an automatic algorithm for discontinuity detection and surface fitting, which produces dense triangular meshes and discontinuity curves automatically from scattered data. In this paper, we follow the same approach, but upgrade the triangular meshes to triangular NURBS surfaces.

Triangular NURBS (TriNURBS) are rational generalizations of the triangular B-splines [141,142,143,144,145], and overcome the shortcomings of tensor-product NURBS (TP-NURBS) [146,147,148] but retain the good properties of local control, automatic continuity, convex hull, affine and projective invariance, completeness, etc. TP-NURBS 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.

Our approach starts with a grouping stage to infer dense potential information from the sparse data. In the second stage, 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 a few iterations of energy minimization, the surface (wings) flap to fit the data, and the edges (snakes) slide to align with the actual edges in the data. Then in the third stage, values and derivatives along each edge are checked, so that discontinuities can be detected and preserved in constructing the surface. Finally the surface are fine-tuned and faired by adjusting the weights. 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.

8.2 Triangular NURBS Surfaces

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 triangulation of the 2D domain, 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; and over each set, a basis function is defined.

A basis function B over 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:

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

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:

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

where X is a point (x,y,z) in 3D space, $c_{t,b}$'s are the scaling factors 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, and the automatic continuity property is lost.

If the above domain is a triangulation of the surface of a unit sphere, a closed TriNURBS surface is defined. 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 a discontinuity. Such a spherical representation is very useful in geographical and other applications [141][155][156]. Since the TriNURBS are defined over arbitrary triangulations of the sphere, there are no pole artifact as in the TP-NURBS, and the continuity is automatically guaranteed. The simplest case is the tetrahedron tessellation of the unit sphere, in which as few as four triangles suffice to exactly model a sphere. By contrast, to cover a sphere with rectangles, at least six rectangles (the faces of a cube) are required to obtain the continuity, and a set of constraint equations have to be maintained.

8.3 Surface Inference

8.3.1 Grouping Stage

The goal of the first stage is to infer dense probability measures from the scattered data, as described in the flow chart in Figure 8.1 . The input can be points, segments, or patches. In our work, we only use points. The details about the grouping method can be found in [140].

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 by the covariance matrix, graphically represented by an ellipsoid, or equivalently, by three eigen-vectors. The eigen-values I_{max} , I_{mid} , I_{min} are interpreted as three saliency measures for surfaces, edges and junctions, and the eigen-vectors are used to estimate the surface normals.

In more details, $I_{max} - I_{mid}$ is used as the saliency of a surface passing through a location, since if $I_{max} - I_{mid}$ is large, I_{max} will be large and I_{mid} is small, also I_{min} is small (since $I_{min} < I_{mid}$). 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. Similarly, $I_{mid} - I_{min}$ is used as an edge saliency measure, and I_{min} is used as junction saliency measure.

Note that *this grouping methodology imposes no restriction on the number of objects, genus (topology), number of discontinuities, and the algorithm is non-iterative and efficient*. By negating 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.

The above voting procedures are performed for each voxel in a 3D grid, and they can work for any number of surfaces of arbitrary topology. They even work for non-

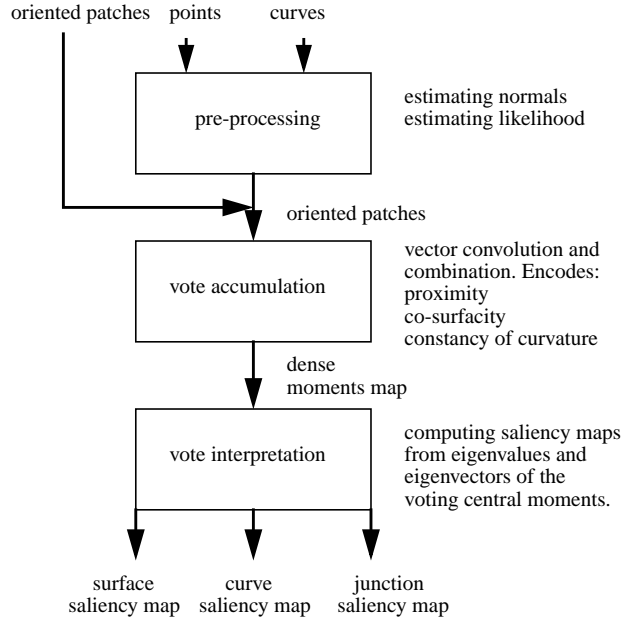


Figure 8.1 Flow chart of the grouping stage

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.

8.3.2 Patch Deformation and Edge Alignment

From the surface potential field, a triangular mesh can be traced out by the “marching cube” algorithm [163]. Different from the standard marching cube algorithm, the surface to be traced out is now the minimum-potential surface or zero-derivative surface of the potential field, instead of the iso-surface of the potential values. Similarly, the discontinuity curves and junctions can be traced out by marching methods also. However, the mesh is quite dense with each facet being a triangle; the edges and junctions are traced out from separate fields and are not integrated into the surfaces.

Our goal is to obtain a sparse curved surface representation, with the discontinuity edges and junctions preserved in the surfaces. We drop a triangular NURBS surface into the potential fields, and deform the surface to reach minimum energy. Different from previous work, we couple the deformable surface model with the active edge model, so that a fitted surface together with aligned edges will be obtained, and the result gives a compact integrated representation from the three potential fields. Since each surface patch can deform and its edges can slide, we call our model “winged B-snakes”.

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

where w_1 , w_2 , and w_3 are coefficients to balance the effects of the three types of energy, and are chosen by normalizing the above four terms to be 1 : 20 : 5 : 5 before the start of minimization. They bring the four terms to comparable orders of magnitude, and the minimization results are stable with their variations. We adjust the control points to minimize the total energy.

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

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

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 stage is mainly surface fitting, we simply use the above membrane energy. The more costly thin-plate energy based on curvature or second-order derivatives is used in the final subtle modification and fairing stage.

(2) Surface Fitting Energy

Inside the surface potential field, the surface patches (wings) flap to reach the local minimum. If the triangle edges cross-over the actual discontinuity edges, the surface energy becomes large; when the edges move to align with the actual edges, the surface energy is reduced. However, we found that the surface energy 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) can slide to the local minimum so that the triangle edges may align with the actual discontinuity edges, instead of crossing over them. Also, the triangle vertices can move to the local minimum in the junction potential field, making the vertices align with the inferred junctions. Such alignment significantly reduces the total energy or cooperatively improves 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.

8.3.3 Building C^1 Surfaces with Edges/Junctions

After the surface patches have been 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 locally adaptive thresh-

olding works well. We then 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.

8.3.4 Fine-tuning/Fairing by Adjusting Weights

In the above fitting and alignment procedures, we never adjust the weights. The reasons are that we want to exploit the descriptive power of control points as much as possible; there exist much redundancy in the weights (scaling of weights does not change the shape at all), so leaking information to the weights is not desired. Also, adjusting too many parameters is difficult for the minimization routines. We tested adjusting the weights at the same time as moving the 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 $c_{t,b} w_{t,b}$ 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 summary, 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.

We use the thin-plate energy as the surface fairness measure, which is based on the second-order derivatives. The energy for fine-tuning and fairing is defined as

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

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 done in stage 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 observe 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 $\{\min(0, w_{t,b})\}^2$.

8.4 Results

Figure 8.2 (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 stage runs for about 10 minutes on a SUN Sparc 10, using three 50x50x50 arrays.

To obtain an initial surface, we 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 one yields smaller surface fitting energy within this rectangle. Such an initial surface is shown in Figure 8.2 (f) with misaligned edges and junctions.

Due to the model's local control property, a control point only affects a triangle and its surrounding 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. We use the Levenberg-Marquardt algorithm with numerically estimated gradients. Figure 8.2 (g) shows aligned/detected edges and junctions after adjusting the control points. (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 stages for reconstruction from the potential fields run for about 10 minutes on an SGI/Indigo.

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 8.3 (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.

Figure 8.4 (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. By contrast, if TP-NURBS were used, we would have to trim a rectangular bi-quadratic (quartic) surface into the airplane shape, or stitch-up several surface pieces; both methods are tedious.

Figure 8.5 (a) shows the intensity image of a plaster tooth and (b) shows 650 points measured on it. We regularly subdivide a sphere into 720 triangles as the initial surface. A top view of the final C^1 surface with detected and preserved edges is

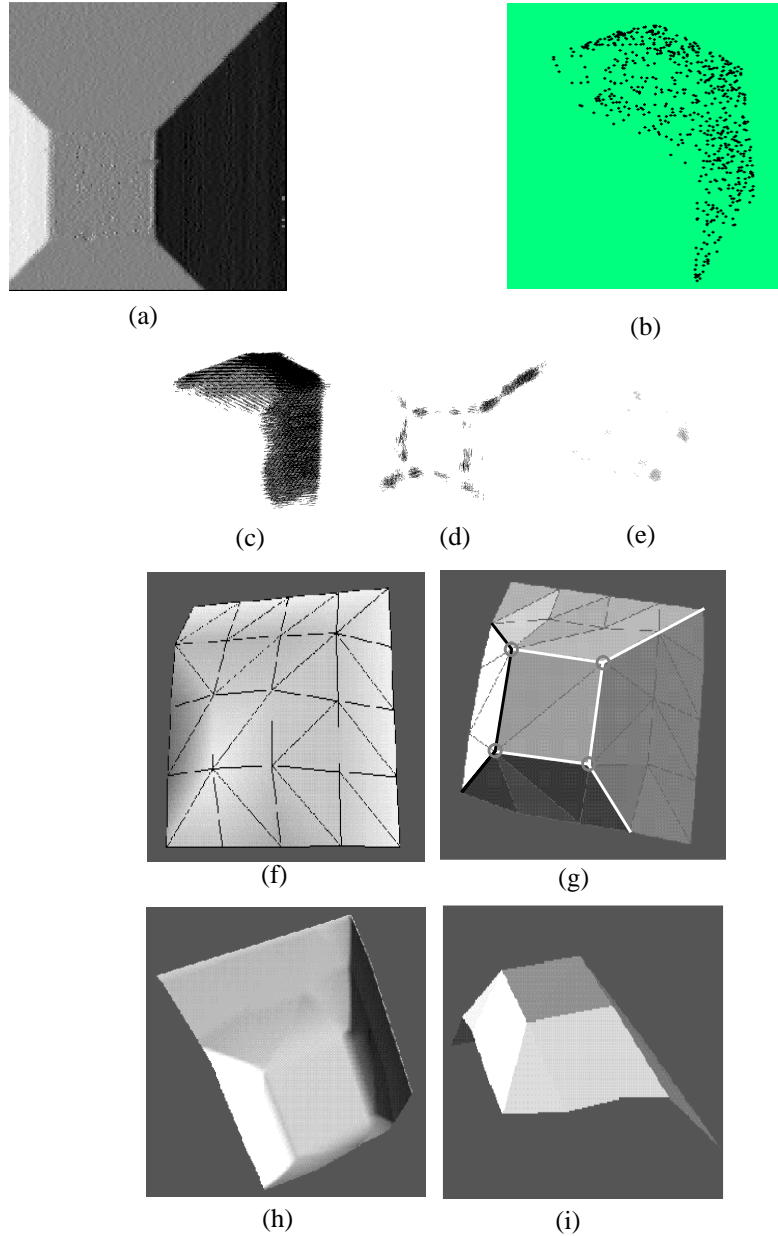


Figure 8.2 Open pyramid

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 align 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 working on merging the triangles in smooth areas for model simplification; also we are starting to work on more complicated objects, such as mechanical parts and medical CT/MRI data of human brains and organs.

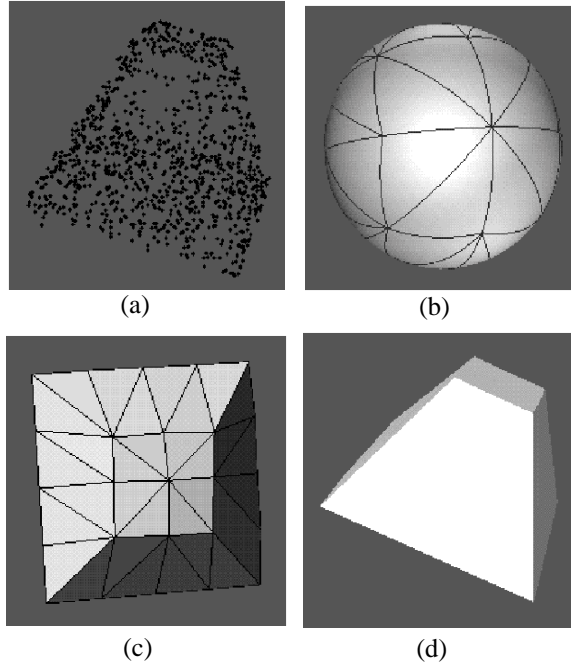


Figure 8.3 Closed pyramid

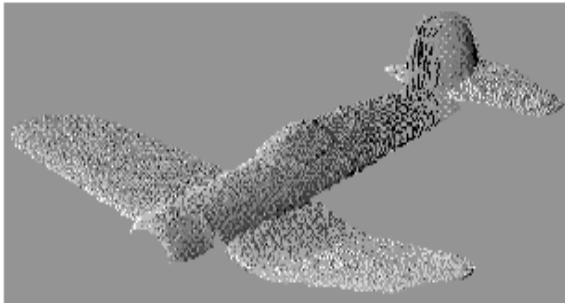
8.5 Conclusions

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 vector voting technique to infer dense surface/edge/junction potential information. 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 further adjusting the weights are also necessary.

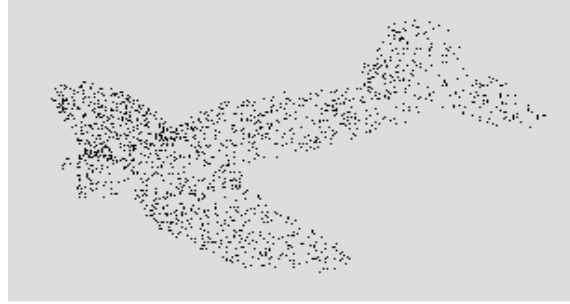
In this paper we only describe open surface and spherical surface modeling from scattered data. We are in the process of extending TriNURBS to arbitrary topology: first trace out the topology information (a dense triangular mesh) from the potential volume fields using the Marching-cube algorithm [163], then perform mesh reduction and surface-fitting/edge-alignment, finally construct a G^1 overlap/blend TriNURBS surface with preserved discontinuities [165], which is an adaptive and multiresolution representation.

8.6 Acknowledgment

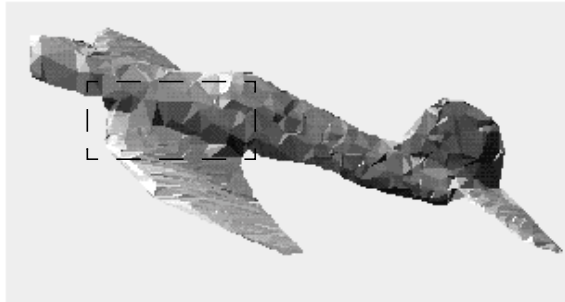
We thank Dr. R. Pfeifle and Prof. H.-P. Seidel for giving us their source codes for the triangular B-spline evaluation. The 3D displays use the *geomview* package by Univ. of Minnesota.



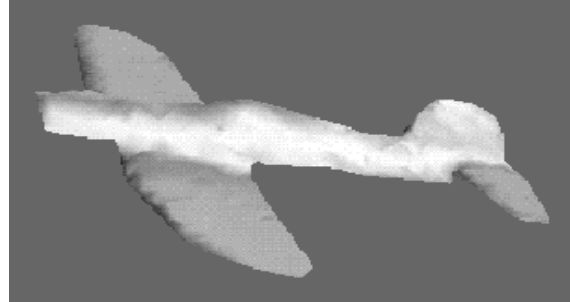
(a) shaded display of the range image of an airplane



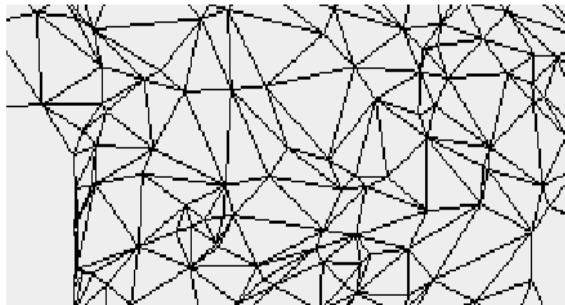
(b) noisy scattered data



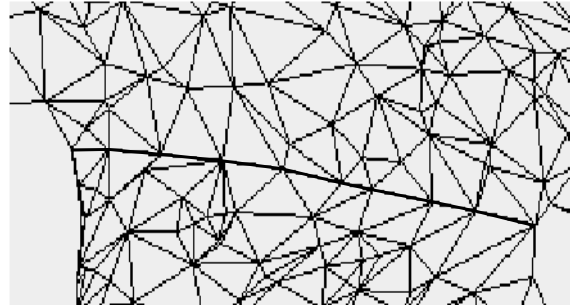
(c) initial triangular mesh



(d) final surface after fine-tuning and fairing.



(e) zoomed display of initial mesh.



(f) zoomed display of the final surface.

Figure 8.4 An airplane (open surface)

8.7 References

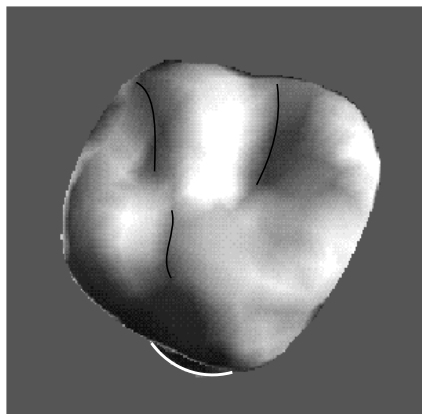
- [138]G. Nielson, CAGD Top 10: What to Watch, IEEE CG&A, Vol. 13, No. 1, Jan. 1993, pp. 35-37
- [139]G. Nielson, Scattered Data Modeling, IEEE CG&A, Vol. 13, N. 1, Jan. 1993, pp. 60-70
- [140]G. Guy and G. Medioni, Inference of Surfaces, Edges and Junctions from Sparse 3D Points, IEEE Int'l Symposium on Computer Vision, Florida, Nov. 1995
- [141]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
- [142]P. Fong and H. Seidel, An Implementation of Triangular B-Spline Surfaces over Arbitrary Triangulations, CAGD, 10, 267-275, 1993



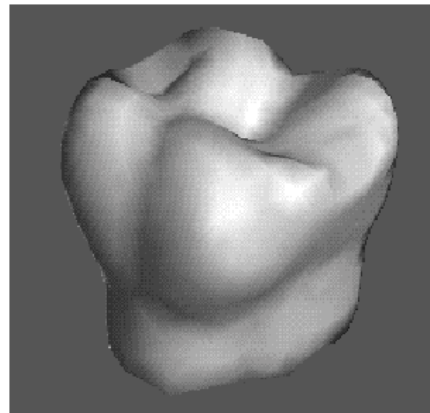
(a) intensity image of a plaster tooth



(b) scattered points measured on (a)



(c) result (view 1)



(c) result (view 2)

Figure 8.5 A tooth (closed shape)

- [143]R. Pfeifle and H. Seidel, Fitting Triangular B-Splines to Functional Scattered Data, Graphics Interface, 26-33, 1995
- [144]R. Pfeifle and H. Seidel, Spherical Triangular B-Splines with Applications to Data Fitting, Eurographics, 89-96, 1995
- [145]S. Auverbach et al., Approximation and Geometric Modeling with Simplex B-Splines Associated with Irregular Triangles, CAGD, 8 (1991), 67-87
- [146]G. Farin, From Conics to NURBS: A Tutorial and Survey, IEEE CG&A, 12, 1992, 78-86
- [147]L. Piegl, On NURBS: A Survey, IEEE CG&A, 11, 1991, 55-71
- [148]L. Piegl and W. Tiller, The NURBS Book, Springer-Verlag, 1995
- [149]H. Qin and D. Terzopoulos, Triangular NURBS and their Dynamic Generalizations, personal communications.
- [150]Y. Shirai, 3-Dimensional Computer Vision, Chapters 3, 7 (Edge Detection), Springer-Verlag, 1987

- [151]Y. Leclerc and S. Zucker, The local structure of image discontinuities in one dimension, IEEE Trans. on PAMI 9, 1987, 341-355
- [152]Y. Leclerc, Capturing the Local Structure of Image Discontinuities in Two Dimensions, IEEE CVPR, Jun. 1985
- [153]A. Blake and A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, 1987
- [154]D. Terzopoulos, Regularization of inverse visual problems involving discontinuities, T-PAMI 8, 1986, 413-424
- [155]P. Alfeld, M. Neamtu and L. Schumaker, Bernstein-Bezier Polynomials on Spheres and Sphere-like Surfaces, CAGD, to appear.
- [156]P. Alfeld, M. Neamtu and L. Schumaker, Fitting Scattered Data on Sphere-like Surfaces using Spherical Splines, CAGD, to appear.
- [157]W. Hohenberger and T. Reuding, Smoothing Rational B-spline Curves Using the Weights in an Optimization Procedure, CAGD, Dec. 1995, 837-848
- [158]G. Dobson, W. Waggenspack, and H. Lamousin, Feature Based Models for Anatomical Data Fitting, CAD, vol. 27, no. 2, pp 139-146, 1995
- [159]H. Qin and D. Terzopoulos, Dynamic NURBS Swung Surfaces for Physics-based Shape Design, CAD, vol. 27, no. 2, pp 111-127, 1995
- [160]P. Laurentgengoux, M. Mekhilef, Optimization of a NURBS Representation, CAD, vol. 25, NOV, 1993, 699-710
- [161]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
- [162]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
- [163]W. Lorensen and H. Cline, Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm, Siggraph, Jul. 1987
- [164]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
- [165]C. Grimm and J. Hughes, Modeling Surfaces of Arbitrary Topology using Manifolds, SIGGRAPH, 359-368, 1995

9 List of Publications

- M. Lee, G. Medioni, R. Deriche, "Structure and Motion from a Sparse Set of Views," in *Proceedings IEEE Symposium on Computer Vision*, Coral Gable, FL, November 1995.
- A. Francoi and G. Medioni, "Generic Shape Learning and Recognition," in *Proceedings of the International Workshop on Object Representations in Computer Vision*, Cambridge, England, 1996.
- S. Han and G. Medioni, "Deformable Surface Reconstruction Coupled with Discontinuity Edge Detection," in *Proc. ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.
- S. Han and G. Medioni, "Triangular NURBS Surface Modeling of Scattered Data," *IEEE Visualization*, San Francisco, California, USA, October 1996.
- S. Han and G. Medioni, "Spherical Winged B-Snakes," *IEEE Intl. Conf. on Image Processing*, Lauzanne, Switzerland, September 1996.
- S. Han and G. Medioni, "Reconstructing Free-form Surfaces from Sparse Data," *Intl. Conf. on Pattern Recognition*, Vienna, Austria, August 1996.
- P. Havaldar and G. Medioni, "Inference of Segmented, Volumetric Shape from Intensity Images," in *Proc. ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.
- P. Havaldar and G. Medioni, "Inference of Segmented, Volumetric Shape from Intensity Images," in *Proc. IEEE Computer Vision and Pattern Recognition*, San Francisco, CA, June 1996.
- M. Lee and G. Médioni, "Structure and Motion from a Sparse Set of Views," in *Proc. ARPA Image Understanding Workshop*, Palm Springs, CA, February, 1996.
- C. Liao and G. Medioni, "Surface Approximation and Segmentation of Objects with Unknown Topology," in *Proc. ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.
- M. Zerroug and R. Nevatia, "Pose Estimation of Multi-Part Curved Objects," in *Proc. ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.
- P. Havaldar and M. Lee, "View Sythesis From Unregistered Images," to appear in *Graphics Interface*, Toronto, Canada, May 1996.
- M. Zerroug and R. Nevatia, "Volumetric descriptions from a single intensity image," to appear in *International Journal of Computer Vision*.

10 Professional Personnel

10.1 Personnel

Professional personnel included Dr. R. Nevatia, Dr. G. Medioni, Dr. K. Price, A. Huertas, G. Guy, S. Han, P. Havaladar, M. Lee, C. Liao, M. Zerroug.

10.2 Ph. D. Graduates

In this contract period we have had 2 Ph. D. graduates:

Image Understanding Research in Complex Environments

Grant No F49620-95-1-0457

Annual Technical Report

August 1, 1995 to July 31, 1996

Contractor: University of Southern California

Start Date: 6/1/95

Principal Investigator: Ram Nevatia

(213) 740-6427

Program Manager: Abraham Waksman

(202) 767-5025

R. Nevatia , G. Medioni and K. Price (Editors)

Institute for Robotics and Intelligent Systems

School of Engineering

University of Southern California

Los Angeles, CA 90089-0273

August 1996

Sponsored by Defense Advanced Research Projects Agency and by Air Force Office of Scientific Research Under Grant No F49620-95-1-0457.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.