A Surface Reconstruction Method for Highly Noisy Point Clouds [⋆]

DanFeng Lu[1], HongKai Zhao[2], Ming Jiang[1], ShuLin Zhou[1], and Tie Zhou[1]

[1] LMAM, School of Mathematical Sciences, Peking Univ.
[2] Department of Mathematics, University of California Irvine. zhao@math.uci.edu

Abstract. In this paper we propose a surface reconstruction method for highly noisy and non-uniform data based on minimal surface model and tensor voting method. To deal with ill-posedness, noise and/or other uncertainties in the data we processes the raw data first using tensor voting before we do surface reconstruction. The tensor voting procedure allows more global and robust communications among the data to extract coherent geometric features and saliency independent of the surface reconstruction. These extracted information will be used to preprocess the data and to guide the final surface reconstruction. Numerically the level set method is used for surface reconstruction. Our method can handle complicated topology as well as highly noisy and/or non-uniform data set. Moreover, improvements of efficiency in implementing the tensor voting method are also proposed. We demonstrate the ability of our method using synthetic and real data.

1 Introduction

Surface reconstruction is to retrieve the original surface from the partial information of that surface. The partial information can include points, pieces of curves and surfaces. In our paper, we mainly consider reconstruction from unorganized point clouds. Surface reconstruction is an important task in many applications such as computer vision, computer graphics, medical imaging, computer aided design and scientific computing.

The main difficulties of surface reconstruction from point clouds include unknown connection or ordering information among the data points, unknown topology of the original surface, and noise and/or non-uniformity in the data. Based on different representations of reconstructed surfaces, most previous reconstruction approaches can be classified as parametric or non-parametric (implicit surfaces). One parametric approach is NURBS (Non-Uniform Rational

[⋆] H. Zhao is partially supported by ONR, DARPA and Sloan Fellowship. M. Jiang is partially supported by the National Basic Research Program of China under Grant 2003CB716101, National Science Foundation of China under Grants 60325101, 60272018 and 60372024, and Engineering Research Institute, Peking University. S. Zhou and T. Zhou are partially supported by the National Basic Research Program of China under Grant 2003CB716101, National Science Foundation of China under Grant 60372024, and Engineering Research Institute, Peking University.

B-Spline) [1] in which the reconstructed surface is smooth, and the data set can be non-uniform. However, this method require a nice parameterization of the surface and possible patching of different pieces for the reconstruction, which can be difficult for an arbitrary data set. Also, it is difficult to treat noisy data. Another popular computational geometry algorithm is based on Delaunay triangulations and Voronoi diagrams to construct triangulated surfaces [2, 3, 4, 5, 6]. For this kind of method, it is challenging to find the right connections among all data points in three and higher dimensions, especially for noisy and highly non-uniform data. Implicit surface methods try to find an implicit function such that a particular level set of this function fits the data best and is extracted as the reconstructed surface [7, 8, 9, 10, 11, 12, 13, 14, 15]. Implicit methods usually have topological flexibility, a simple data structure and depth/volumetric information. However it is a challenge to deal with open surfaces.

To deal with noisy data a variational formulation is usually used and is composed of both a fitting term for the data and a regularization term for the reconstructed surface. There are two issues for this approach: (1) all data points, even outliers, are treated equally and can affect the final reconstruction; (2) there is a lack of effective communications among all data points and the balance of the fitting term and the regularization term is usually local during the reconstruction/evolution which can cause the evolving surface trapped into local minimum easily. For highly noisy data, these approaches will likely to fail.

Tensor voting method, proposed by Medioni et al. [16], is a nice feature extraction algorithm. By designing an appropriate voting procedure among all data points a tensor field and an associated saliency field can be constructed. Coherent geometric information can be extracted from the tensor field and the saliency field. However, using tensor voting method alone is difficult to reconstruct a smooth and well-represented surface.

In this paper, we propose a surface reconstruction method combining the minimal surface model [17, 18] and the tensor voting method for highly noisy and/or non-uniform data. We use tensor voting method to preprocess the noisy data as well as to provide coherent information for the minimal surface model. We show that our model can handle significant noise in the data.

2 New Surface Reconstruction Model

2.1 Minimal Surface Model

In [13] the following weighted minimal surface model is proposed: let $\mathcal{S}$ denote the data set which can include points, pieces of curves and surfaces. Define

$$d(x) = \mathrm{dist}(x, \mathcal{S}) \tag{1}$$

to be the distance function to $\mathcal{S}$. Then define the surface energy functional as:

$$E(\Gamma) = \left[\int_{\Gamma} d^p(x) ds \right]^{1/p} . \tag{2}$$

Here Γ is an arbitrary surface and ds is the surface area. Thinking of $d(x)$ as a potential function for $\mathcal{S}$, this energy is the L^p norm of potential on Γ. The purpose is to try to find a local minimizer of the energy functional that behaves like a minimal surface or an elastic membrane attached to the data set.

Level set method [19] is used to evolve an initial guess to the steady state. Define the corresponding level set function to be $\phi(x,t)$. The energy functional can be reformulated as:

$$E(\phi) = \left[\int d^p(x)\delta(\phi(x))|\nabla\phi(x)|dx\right]^{1/p} \quad , \tag{3}$$

where the integration domain can be any open set (e.g., the computation domain) that contains the zero level set of ϕ.

The gradient flow for the level set function $\phi(x,t)$ ([13, 20]) is:

$$\frac{\partial\phi}{\partial t}=|\nabla\phi|\left[\int d^p(x)\delta(\phi)|\nabla\phi|dx\right]^{1/p-1}\times d^{p-1}(x)\left[\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}+\frac{1}{p}d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right]. \tag{4}$$

By neglecting a scaling factor we can simplify (4) as:

$$\frac{\partial\phi}{\partial t} = |\nabla\phi|\left[\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|} + \frac{1}{p}d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right] \quad . \tag{5}$$

The term $\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}$ corresponds to the attraction by the distance field and the term $d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$ corresponds to a minimal surface regularization weighted by the distance function, where $\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$ is the mean curvature of the surface. The parameter $1/p$ balance the potential force and surface tension. Since the nonlinear regularization due to surface tension has a desirable scaling $d(x)$, the membrane is more flexible close to the data and is more rigid away from the data. Fast implementations were discussed in [14]. The minimal surface model can handle complicated topologies and construct a surface that is smoother than triangulated surface in three dimensions. It can deal with noisy and non-uniform data to some extent by balancing the attraction of the data (fitting) and the surface area regularization. However, the minimal surface model can not deal with highly noisy data because (1) The distance field is the distance to all data set. If there are many outliers, the evolution surface will be attracted by all these points and get stuck. (2) There is a lack of global communications or denoising for the noisy data set. The surface regularization (the curvature) term is very local and is only related to the evolution surface not to the data set.

2.2 Tensor Voting Method

Tensor voting method [16] allows more effective and robust communications among the data to extract coherent geometric features and saliency. A second order symmetric tensor is used to store geometric information, orientation information and saliency. The tensor can be visualized as an ellipse in 2D, and an ellipsoid in 3D. The shape of the tensor defines the geometric information (point, curve, or surface element), and its size represents the saliency. In 3D, a surface is represented by a tensor in the shape of an elongated ellipsoid (stick tensor) with its major axis along the surface normal. A curve is represented by a tensor in the shape of a flat ellipsoid (plate tensor) that is perpendicular to the curve's tangent. An isolated point has no orientation preference and is represented by a tensor in the shape of a spherical ellipsoid (ball tensor). The tensor field is generated by a voting procedure.

We give a brief review of the basic idea behind tensor voting in 2D. Suppose there exists a smooth curve connecting the origin O and a point P and suppose that the normal to the curve at O is known. Then what is the most likely normal direction at P? Fig. 1 illustrates the situation. It can be argued [16] that the osculating circle connecting O and P is the most likely connection since it keeps the curvature constant along the hypothesized circular arc. So the most likely normal is given by the normal to the circular arc at P (thick black arrow in Fig. 1). This normal at P is oriented such that its inner product with the normal at O is nonnegative. The length of this normal, which represents the voting strength, is inversely proportional to the arc length s and curvature k. So the decay function of vote strength is defined as:

$$DF(s, \kappa, \sigma) = e^{-\frac{s^2 + c\kappa^2}{\sigma^2}} \quad , \tag{6}$$

where σ controls smoothness, which also determines the effective neighborhood size [21]. c is a constant which controls the decay with high curvature, and about its value we refer readers to [22]. We here set $c = 3.57$. If we vote for all locations of P, we get a 2D stick voting field from O.

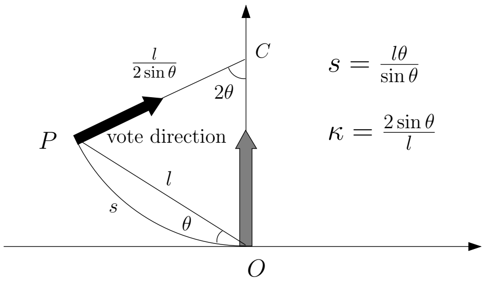


Fig. 1. Voting between two points

We denote the stick vote at P receiving from O as $[v_x \; v_y]^T$. Other points cast votes to P as well. So at P we can get a symmetric positive tensor by summing all votes received:

$$S = \begin{bmatrix} \sum v_x^2 & \sum v_x v_y \\ \sum v_y v_x & \sum v_y^2 \end{bmatrix} \quad . \tag{7}$$

Let the two eigenvalues of S be $\lambda_1 \geq \lambda_2 \geq 0$ and two corresponding eigenvectors be $\hat{e}_1$ and $\hat{e}_2$. S can be rewritten as

$$S = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + \lambda_2(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) \ . \tag{8}$$

$\hat{e}_1\hat{e}_1^T$ is called a 2D stick tensor $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$ is called a 2D ball tensor. The stick saliency field $\lambda_1 - \lambda_2$ indicates the saliency of curve. The larger the difference the more likely P is on a curve whose normal is $\hat{e}_1$. Here we give an example of stick saliency field in 2D for eight points from a circle in Fig. 2. The saliency field gives a good indication of the circle.

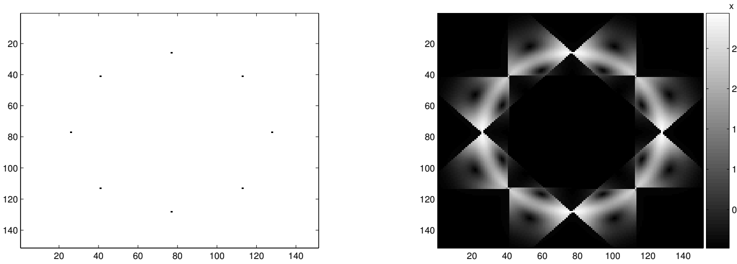


Fig. 2. The stick saliency field for eight points from a circle

If the input is discrete points, each point has a ball tensor initially. A tensor field is generated by a tensor voting procedure. For every pair of points, for example O voting on P in the previous example, since no direction information is available initially, a discrete set of uniformly distributed directions are used as possible normals at O and vote on P with the weight function in (6). By summing votes from all directions at all other points we can get a tensor at each data point and we can further generate a tensor field.

In 3D a tensor field can be decomposed as

$$S = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + (\lambda_2 - \lambda_3)(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) + \lambda_3(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T) \ , \tag{9}$$

where $\hat{e}_1\hat{e}_1^T$ is a 3D stick tensor, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$ is a 3D plate tensor, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T$ is a 3D ball tensor. The stick saliency field $\lambda_1 - \lambda_2$ represents the saliency of surface with normal $\hat{e}_1$, the field of $\lambda_2 - \lambda_3$ represents the saliency of curve with tangent direction orthogonal to both $\hat{e}_1$ and $\hat{e}_2$, and λ_3 represents the saliency of junction or isolated point. We denote the stick saliency field $\lambda_1 - \lambda_2$ as $s(x)$ which will play an important role in our new model.

In summary, we use the following tensor voting procedure for our data points:
1. Tensor calculus. Generate the initial stick tensor information at every original data point.
2. Voting process. Every data point propagates its tensor information to neighboring grid points and generate a tensor field in the computation domain.

3. Feature extraction. Extract the geometric features and saliency field at each grid point.

Although tensor voting method can reveal coherent geometric features and saliency for data points, to use the method directly for surface reconstruction has the following disadvantages: (1) The tensor information is not very accurate or sharp, especially in the case of complicated topology and/or geometry. Surface regularization after reconstruction is needed [23]. (2) To extract the surface from the saliency field directly the algorithm is quite complex, and need to tune parameters of threshold empirically [16, 23].

2.3 Our New Model

The above discussions show that the stick saliency field from tensor voting among all data points contains more global and robust information for our surface reconstruction. The strength of the saliency field gives a good likelihood indication of surface at each point. Therefore we incorporate the stick saliency field and combine it with the distance field for surface reconstruction. Let

$$k(x) = 1 - \frac{s(x)}{M}, \quad M = \max\{s(x)\} \tag{10}$$

be the normalized stick saliency field. Then we define our new evolution as:

$$\frac{\partial \phi}{\partial t} = \alpha \nabla d(x) \cdot \nabla \phi + \beta \nabla k(x) \cdot \nabla \phi + \gamma d(x) |\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \ . \tag{11}$$

The term $\nabla d(x) \cdot \nabla \phi$ corresponds to the attraction of the data set through the distance field. The term $\nabla k(x) \cdot \nabla \phi$ corresponds to attraction of saliency field. These two terms advect the surface closer to the data set as well as to high saliency region. The term $d(x)|\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$ corresponds to a weighted surface tension which regularizes the reconstructed surface. Tuning the parameters α, β can balance the effect of two fields, and the value of γ affect the smoothness of the reconstructed surface. For highly non-uniform data, the saliency field can provide more useful information than the distance field.

For highly noisy data sets, we first use tensor voting to remove outliers, i.e., those points that are not likely on the surface. After we get the normalized stick saliency field we remove those points whose saliency value is smaller than a threshold. This procedure can be repeated if necessary. This preprocessing step allows us to clean up the data substantially even for very noisy data, which will be shown later by examples. After this step we redo the tensor voting procedure for the remaining data set and generate a new tensor and saliency field. Then we use the above model for surface reconstruction.

We do not advocate of only using saliency field to propagate the surface. In some situations with simple topological structure and surface details, we can solely use saliency field, e.g., Fig. 5(c). However, the global tensor voting process usually results in a quite smeared saliency field, i.e., the gradient of the saliency field is not sharp, especially if the data set is sparse or has noise or complicated topological or geometric structures. Moreover, this makes the evolution slow too.

3 Numerical Implementation

3.1 The Level Set Method for Surface Evolution

Since we do not know a priori the topology of the final surface, we use level set method for surface evolution according to (11). This equation is of the same type of the minimal surface model used in [17]. The two convection terms $\nabla k(x) \cdot \nabla \phi(x)$ and $\nabla k(x) \cdot \nabla \phi(x)$ are treated in the same way. We refer readers to [17] for implementation details. Here is our implementation procedure:
1. If the original data is noisy, we first use tensor voting method to remove outliers in the data set.
2. Get distance function $d(x)$ and normalized saliency field $k(x)$.
3. Start with an initial guess of Γ and evolve it to steady state using (11).

The distance field is computed using the fast sweeping method which is of $O(N+M)$ complexity, where N is the number of grid point and M is the number of data point [13, 17]. Local level set method [24] is used to cut down the computation cost.

To further accelerate the computation, we can neglect the curvature term and just use the two convection terms initially to evolve the surface as suggested in [18]. This allows us to remove more strict CFL condition due to the curvature term. When the evolution is near steady state, we can put in the curvature term to make the final surface smoother.

3.2 Some Improvements in Implementing the Tensor Voting Method

Generating the Initial Stick Tensor. In Sect. 2.2, we give out the original method of generating the initial stick tensor. However using stick tensor to simulate ball tensor or plate tensor is time-consuming and is not very accurate.

We consider the communication of two data points $P,\ Q$ in Fig. 3. Without any prior information, the most likely relationship of these two points is that they are on the straight line connecting them. So they give each other a plate tensor. Let $\hat{e}_{PQ} = (t_x,\ t_y,\ t_z)$ be the unit vector pointing from P to Q. Taking into account the decay with the increasing distance, we define the plate tensor as:

$$e^{\frac{l^2}{\sigma^2}}(I - \hat{e}_{PQ}\hat{e}_{PQ}^T) = e^{\frac{l^2}{\sigma^2}} \cdot \begin{bmatrix} 1-t_x^2 & -t_xt_y & -t_xt_z \\ -t_xt_y & 1-t_y^2 & -t_yt_z \\ -t_xt_z & -t_yt_z & 1-t_z^2 \end{bmatrix} \ , \tag{12}$$

where l is the distance between $P,\ Q$. When l is bigger than a threshold we can ignore their communication. For every data point, summing up the contribution from all neighbors gives it the initial tensor.

Suppose the number of data points is N, and every point has M neighbors. Then if we use k stick tensor to represent a ball tensor, the original method needs $O(kNM)$ operations and the value of k can be large to represent all directions well. While our new algorithm needs $O(NM)$ operations.

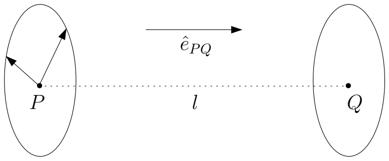


Fig. 3. Communication between two data points

Voting Process. In the second step of tensor voting, every data point propagates its tensor information to its neighboring grid points like in Fig. 1. But every time a data point votes on a grid point, we need to compute the weight $DF(s, \kappa, \sigma)$. It is a time-consuming process. In fact, to reduce the computation cost, we can first establish an index table of the weight function, then when we need to compute the contribution from a point to another point we can refer to the table which significantly speed up the computation. Moreover, the higher resolution the index table has, the better we approximate the true weight function. In our implementation, we use an index table whose resolution is three times of that of our computation grids. As showed in Fig. 4, suppose we know the coordinates of point P, Q, and know the unit stick vector $\hat{e}_P$ at P, we look up the index table to get the value of $DF(s, \kappa, \sigma)$. We demonstrate this in 2D. First, we can determine the unit stick vector $\hat{e}_Q$ at Q voted from P by the formula:

$$\hat{e}_Q = \hat{e}_P - 2\frac{\overrightarrow{QP}}{\| \overrightarrow{QP} \|}\left(\frac{\overrightarrow{QP} \cdot \hat{e}_P}{\| \overrightarrow{QP} \|}\right) . \tag{13}$$

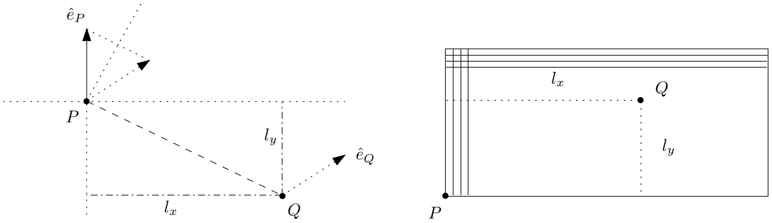


Fig. 4. Use the index table instead of computing $DF(s, \kappa, \sigma)$

Then we find the corresponding weight from the index table by l_x and l_y. Times this weight with the stick saliency value at P gives the stick saliency value at Q voted from P. Here,

$$l_x = \| \overrightarrow{QP} \| \sqrt{1-c^2}, \quad l_y = \| \overrightarrow{QP} \| \ |c|, \quad c = \frac{\overrightarrow{QP} \cdot \hat{e}_P}{\| \overrightarrow{QP} \|} . \tag{14}$$

The establishment of the index table is relatively simple, we only need to store the contributions from a unit stick tensor to its neighboring points.

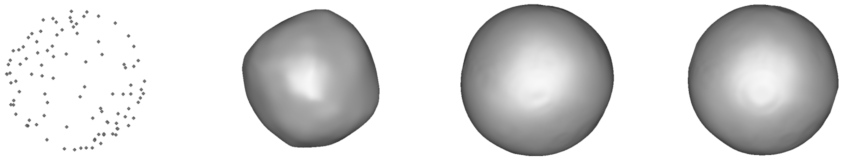
Tune the Range of θ. In the original tensor voting method, a point vote on another point only if $\theta \leq \pi/4$ (see Fig. 1). However, for our surface reconstruction purpose it seems that two nearby points on a smooth surface (relative to the grid size) are not likely to form a large angle. In our experiments for real data, we get better results if we restrict θ in a smaller interval. In our experiments, we set $\theta \leq \pi/12$. In the following, we use $\theta = \psi$ to mean the range of θ is $[0,\ \psi]$.

4 Experimental Results

In this section we present experimental results for our method. All results are displayed using OpenDX.

The computations are carried out on a CPU of AMD Athlon XP 3200+ and 1GB memory. All the reconstructions are on a $61 \times 61 \times 61$ grid. We use both synthetic and real data to show the ability of handling highly noisy and non-uniform data.

In Fig. 5 we show the case of non-uniform data. The data set is 100 random points on a sphere. We can see from the result that the reconstructed surface (d) using our new model is better than (b) reconstructed from minimal surface model or (c) which only uses stick saliency field for reconstruction.



(a) 100 random points on a ball

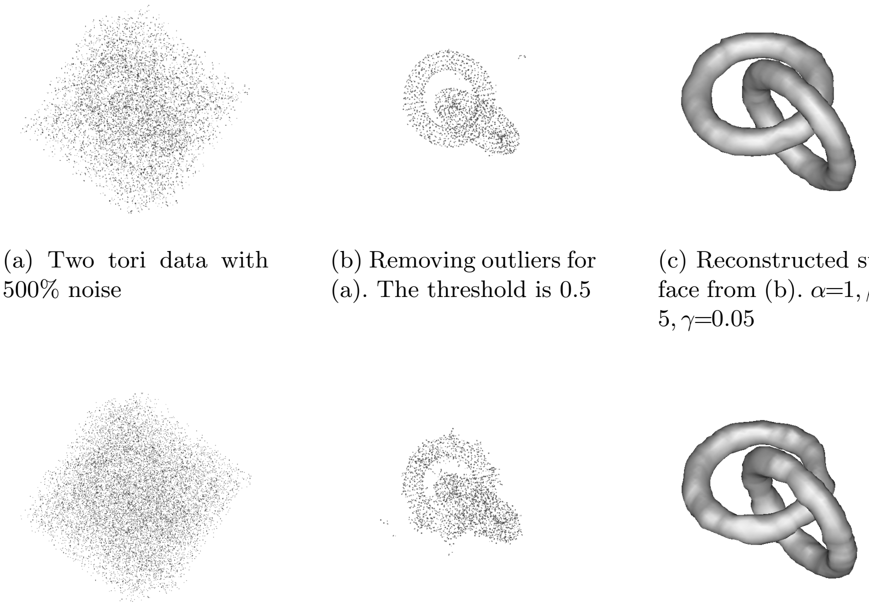
(b) Reconstruction using minimal surface model. $\alpha = 1,\ \beta = 0,\ \gamma = 0.5$

(c) Reconstruction only using stick saliency field. $\alpha = 0,\ \beta = 10,\ \gamma = 0.5$

(d) Reconstruction using the new model. $\alpha = 1,\ \beta = 10,\ \gamma = 0.5$

Fig. 5. Surface reconstruction from non-uniform data

In Fig. 6, Fig. 7, we demonstrate the ability of dealing with highly noisy data for the new model. We add points randomly in the box that contain the original data. The noise ratio is the ratio between the number of added points and the number of original data points. The original data sets for the two tori and the bunny have 1200 and 35947 points respectively. Some experimental data are displayed in Table 1 from which we can see that when the number of data points is large, the most expensive step is tensor voting.



(a) Two tori data with 500% noise

(b) Removing outliers for (a). The threshold is 0.5

(c) Reconstructed surface from (b). α=1, β= 5, γ=0.05

(d) Two tori data with 1000% noise

(e) Removing outliers for (d). The threshold is 0.5

(f) Reconstructed surface from (e). α=1, β= 11, γ=0.05

Fig. 6. Surface reconstruction from noisy two tori data. $\sigma = 3,\ \theta = \pi/18$

References

[1] D. F. Rogers, *An introduction to NURBS: with historical perspective.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

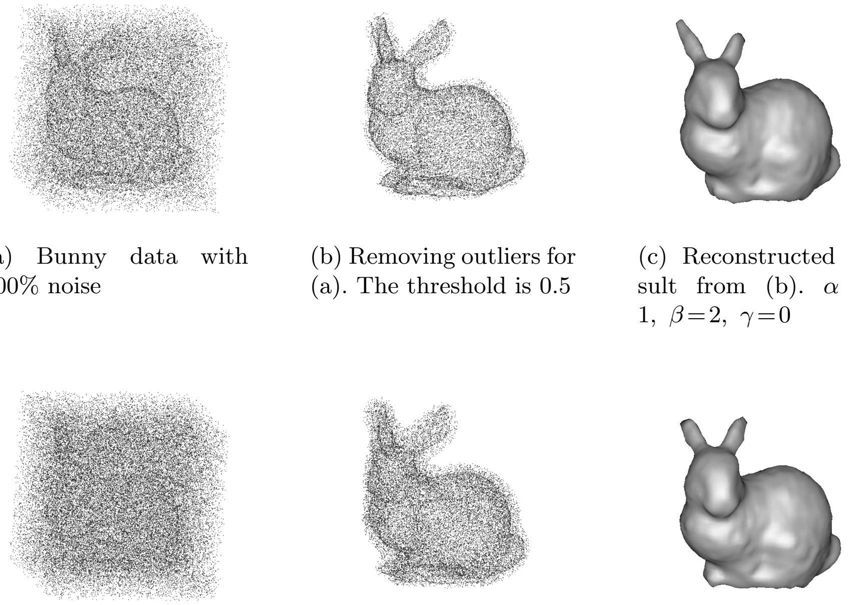
[2] N. Amenta and M. Bern, "Surface reconstruction by voronoi filtering," *Discrete and Comput. Geometry*, vol. 22, pp. 481–504, 1999.

[3] H. Edelsbrunner, "Shape reconstruction with delaunay complex," in *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics*, (London, UK), pp. 119–132, Springer-Verlag, 1998.

[4] J. D. Boissonnat, "Geometric structures for three dimensional shape reconstruction," *ACM Trans. Graphics 3*, pp. 266–286, 1984.

[5] N. Amenta, M. Bern, and D. Eppstein, "The crust and the -skeleton: Combinational curve reconstruction," *14th ACM Symposium on Computational Geometry*, 1998.

[6] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 415–421, ACM Press, 1998.



(a) Bunny data with 500% noise

(b) Removing outliers for (a). The threshold is 0.5

(c) Reconstructed result from (b). $\alpha=1,\ \beta=2,\ \gamma=0$

(d) Bunny data with 1200% noise

(e) Removing outliers for (d). The threshold is 0.6

(f) Reconstructed result from (e). $\alpha=1,\ \beta=2,\ \gamma=0$

Fig. 7. Surface reconstruction from noisy bunny data. $\sigma = 3,\ \theta = \pi/18$

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 71–78, ACM Press, 1992.

[8] C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3d scans," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 109–118, ACM Press, 1995.

[9] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 303–312, ACM Press, 1996.

[10] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, "Implicit surface-based geometric fusion," *Comput. Vis. Image Underst.*, vol. 69, no. 3, pp. 273–291, 1998.

[11] J. Bloomenthal and B. Wyvill, eds., *Introduction to Implicit Surfaces.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[12] R. Whitaker, "A level set approach to 3D reconstruction from range data," *International journal of Computer Vision*, 1997.

Table 1. Experimental data

	2 tori		bunny	
Reconstructed result	Fig. 6(c)	Fig. 6(f)	Fig. 7(c)	Fig. 7(f)
Noisy ratio	500%	1000%	500%	1200%
Total points	7200	13200	215682	467311
Points after removing noise	1601	2413	62807	119232
Time for tensor voting at twice	8.11s	14.313s	1606.015s	6514.141s
Evolution time	87.516s	119.5s	57.875s	132.875s

[13] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang, "Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method," *Computer Vision and Image Understanding*, vol. 80, pp. 295–319, 2000.

[14] H. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, (Washington, DC, USA), p. 194, IEEE Computer Society, 2001.

[15] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 67–76, ACM Press, 2001.

[16] G. Medioni, M.-S. Lee, and C.-K. Tang, *A computational framework for segmentation and grouping.* Elsevier, 2000.

[17] H. Zhao, "Fast sweeping method for Eikonal equations," *Mathematics of Computation*, 2004.

[18] H. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction and deformation using the level set method," *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver*, July, 2001.

[19] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces.* Springer, New York, 2002.

[20] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, pp. 179–195, 1996.

[21] J. Jia and C.-K. Tang, "Inference of segmented color and texture description by tensor voting," vol. 26, pp. 771–786, June 2004.

[22] W.-S. Tong, C.-K. Tang, P. Mordohai, and G. Medioni, "First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 594–611, 2004.

[23] C.-K. Tang and G. Medioni, "Inference of integrated surface, curve, and junction descriptions from sparse 3d data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1206–1223, 1998.

[24] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, "A PDE based fast local level set method," *J. Comput. Phys.*, vol. 155, pp. 410–438, 1999.