# MULTI-RESOLUTION SURFACE RECONSTRUCTION

*HE Mingyi*[1], *XIONG Bangshu*[1,2], *YU Huajing*[1]

[1] Department of Electronic & Information Engineering
Northwestern Polytechnical University, Xi'an, 710072, China
[2] Department of Electronic Engineering
Nanchang Institute of Aeronautical Technology, Nanchang, 330034, China
E-mail: myhe@nwpu.edu.cn

## ABSTRACT

This paper presents a novel surface reconstruction algorithm, which can directly generate multi-resolution meshes from the unorganized cloud points. Local normal change is used to determine whether local surface is flat or rough. According to the user-specified normal change threshold, a min-max box containing all sample points is split into octree cells with different sizes. If the local surface is flat, a portion of the isosurface will be extracted in a large cell, otherwise in a small one. Thus the proposed algorithm can reduce the number of triangles in mesh compared with conventional Marching Cubes (MC) algorithm. The datasets of Terracotta Warrior and Horses in Qin Dynasty and the Kettle in Song Dynasty are used to verify the efficiency of our algorithm. The experimental results show that this algorithm can greatly decrease the number of triangles with the increase of the normal change threshold and maintain the details of surface.

## 1. INTRODUCTION

With the developing of 3D information acquiring equipment, especially 3D laser scanner, 3D sampled points of the object can be obtained accurately and rapidly. How to generate concise triangle mesh from sampled points has recently attracted much attention in image processing, computer graphics and related fields. Hoppe *et al*. [1] presented an algorithm in which surface is represented by the zero set of a signed distance function. The zero set is then contoured by a continuous piecewise-linear surface using the marching cubes algorithm. The approach of Curless and Levoy [2] is fine-tuned for laser range data, and therefore is well suited for handling very large data sets. Their algorithm was successfully used in the digital Michelangelo project [3]. Kobbelt *et al.*'s method [4] reduced alias artifacts at sharp features on the extracted surfaces by calculating and storing directed distance in *x*, *y*, and *z* direction, instead of using a scalar distance value for each grid point of a uniform spatial grid. Yong and Yuen [5] used a new mesh evaluation criterion to optimize meshes extracted by MC algorithm, which generates high quality meshes from unorganized cloud points with time and space efficiency.

In the above algorithms, small and uniform cubes were used to recover the details of the surface to be reconstructed. However, the generated mesh usually consists of a large number of triangles, which results in many problems in practice, such as model storing, transmitting, and real time displaying. In this paper, we propose a multi-resolution surface reconstruction (MRSR) algorithm in order to reduce the number of triangles as well as maintain adequate surface details. The algorithm can automatically extract isosurface in different sized cubes with respect to a given threshold of normal change, and can generate multi-resolution meshes from cloud points by giving a series of thresholds. The number of triangles in the model will decrease with the increase of the normal change threshold.

## 2. LOCAL NORMAL CHANGE AND GAUSSIAN CURVATURE

Suppose that a min-max box of cloud points is divided into many different cubes, and a subset of sampled points $\mathbf{P} = \{p_1, p_2, \cdots, p_m\}$ is contained in a cube, which intersects with surface of the object. Then, the local normal change $\theta$ is defined as the max-angle between normal vectors of sample points in the cube, and $\theta$ is calculated by

$$\theta = \max\left(\angle \mathbf{n}(p_i)\mathbf{n}(p_j)\right), \left(i \neq j, 1 \leq i, j \leq m\right) \tag{1}$$

0-7803-8554-3/04/$20.00 ©2004 IEEE.

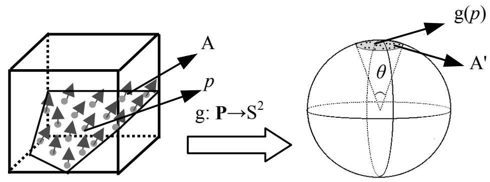


**Fig. 1.** Gaussian translation of points in a cube.

where $\mathbf{n}(p_i)$ and $\mathbf{n}(p_j)$ are the unit normal vectors of points $p_i$ and $p_j$ , respectively. If the starting point of unit normal vector $\mathbf{n}(p_i)$ is translated to origin of coordinates, its end point will falls on the unit sphere $S^2$ (Fig. 1), which is referred to Gaussian map g: $\mathbf{P} \to S^2$. Where, $A$ is the area of an intersectant region $D$ between the cube and the surface $S$, $p$ is one point of the set $\mathbf{P}$, $A'$ is the area of Gaussian transform of $D$ in the sphere $S^2$, g($p$) is Gaussian transform of point $p$, and $\theta$ is the local normal change of the cube containing region $D$.

If the size of a cube is very small, we can view $D$ as the local neighborhood of point $p$. Based on the knowledge of differential geometry, the Gaussian curvature $K(p)$ of the point $p$ can be described as follows

$$K(p) = \lim_{D \to p} \frac{A'}{A} \tag{2}$$

Here, $A' = 2\pi\left(1 - \cos\frac{\theta}{2}\right)$ approximates to the surface area of intersectant region between the Gaussian sphere and a taper. By substituting this term for $A'$ in Equation (2), we can rewrite this equation as follows

$$K(p) = \lim_{D \to p}\left(\frac{2\pi}{A}\left(1 - \cos\frac{\theta}{2}\right)\right) \tag{3}$$

Suppose that $D$ is sufficiently small, we can further rewrite the above equation as

$$K(p) \approx \frac{2\pi}{A}\left(1 - \cos\frac{\theta}{2}\right) \tag{4}$$

Equation (4) indicates that the local normal change $\theta$ can approximately represent the Gaussian curvature. Thus the normal change $\theta$ can be used to evaluate curved degree of local surface contained in this cube.

## 3. PROPOSED ALGORITHM

The proposed MRSR algorithm comprises three steps: 1) related information computation, 2) min-max box partition and isosurface extraction, and 3) crack patching.

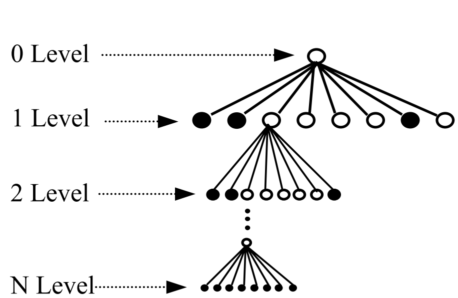


**Fig. 2.** Min-max box division based on octree.

### 3.1. Related information computation

The related information computation of sample points includes *k*-nearest neighbor searching, normal vector calculating and normal orientation adjusting. The spatial partitioning strategy is applied to finding *k*-nearest neighbors of cloud points so as to improve the efficiency of searching. After getting *k*-nearest neighbors of each point, the unit normal vector of all points are determined by principal component analysis. The normal vectors of all points are oriented consistently by traversing the minimal spanning tree (MST) of their *Riemannian Graph*.

### 3.2. Min-max box partition and isosurface extraction

The size of min-max box of all sampled points is defined by $L_0 = \max\left((x_{\max} - x_{\min}), (y_{\max} - y_{\min}), (z_{\max} - z_{\min})\right)$, i.e. the edge length of the cube containing all points is $L_0$. According to the user-specified normal change threshold $\theta_{\max}$, the min-max box is recursively split into octree cells with different sizes by a depth-first manner, which is shown in Fig. 2. In this figure, the black and the white dots represent valid and invalid leaf nodes of the octree, respectively. Each valid leaf node in the octree does contain a part of isosurface, and each invalid leaf node doesn't. The depth of the octree is determined from the length of the box and the density of the dataset. If the dataset $\mathbf{X}$ is a $\rho$-dense and noiseless sampling of surface, the depth of the octree will be the largest integer greater than or equal to the logarithm of the ratio of $L_0$ to $\rho$. $\rho$ is derived from the dataset $\mathbf{X}$ by

$$\rho = \max(d(x_i, \mathbf{X})), x_i \in \mathbf{X}$$

where $d(x_i, \mathbf{X})$ is the minimum distance between point $x_i$ and the point set $\mathbf{X}$.

In the subdividing process of min-max box, the threshold $\theta_{\max}$ is used to determine whether current cell in the octree is further subdivided or not. If the normal

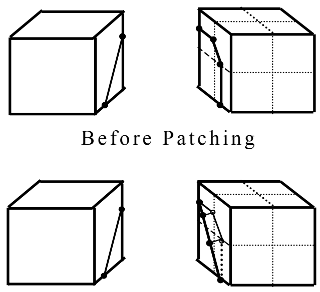


**Fig.3.** A schematic diagram of crack patching.

change $\theta$ of the current cell is larger than $\theta_{\max}$, the current cell will be subdivided eight child cells, otherwise, a piece of isosurface is extracted in this cell by using the zero set of a signed distance function [1].

Our algorithm differs from the MC algorithm in that the isosurface is extracted in cells with different size. The number of triangles in the isosurface, generated by our algorithm, is reduced when the threshold $\theta_{\max}$ increases.

### 3.3. Crack patching

If the triangulation is performed in the above section, cracks will be generated at the interfaces of cells with different dimensions. Fig. 3 shows one simple scenario where the ($n$-1)th level cell interfaces with the $n$th level cell of the octree. There are two edges at the interface that do not quite match, which causes the crack.

Shekhar *et al.*'s method [6] is used for crack patching. In this method, edges of the smaller cells are forced to lie along those of the larger neighbouring cells. The points that make the high-resolution edge are replaced with their perpendicular projections on the low-resolution edge. Patching is accomplished by stretching the high-resolution to match with low-resolution edge, and stops while the breadth-first traversal of nodes stops.

## 4. EXPERIMENTAL RESULTS

Experiments within different datasets have been carried out to verify our method. All the experiments are run on a PIII 800/256MB memory computer. We introduce the mean root square error $D_{\text{surface}}$ to evaluate the geometric distortion of the reconstructed model. $D_{\text{surface}}$ is defined as

$$D_{\text{surface}} = \sqrt{\frac{1}{n+m}\left(\sum_{x_i \in \mathrm{X}} d^2(x_i, \mathrm{M}) + \sum_{y_j \in \mathrm{M}} d^2(y_j, \mathbf{TP})\right)}$$

where M and **TP** represent the reconstructed mesh and the tangent plane set of original points, respectively;

| Dataset | MC | MRSR | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0° | 10° | 20° | 30° | 40° | 50° |
| WARRIORS | 75.2 | 63.9 | 58.3 | 48.8 | 42.2 | 37.3 | 34.6 |
| Kettle | 4.9 | 5.8 | 5.3 | 4.3 | 3.6 | 3.0 | 2.7 |

**Table 1.** Comparison of the execution times (sec) of the proposed MRSR algorithm at different threshold $\theta_{\max}$ with the MC algorithm.

$d(x_i, \mathrm{M})$ is the minimum distance between a point $x_i \in \mathbf{X}$ and the mesh M ; $d(y_j, \mathbf{TP})$ is the minimum distance between a vertex $y_j \in \mathrm{M}$ and **TP** ; $n$ is the number of points of dataset **X** ; $m$ is the number of vertices of the reconstructed mesh M .

This paper has shown the main procedures of the reconstruction of the Terracotta Warriors and Horses (WARRIORS) in Qin Dynasty and the Kettle in Song Dynasty. The datasets in Fig. 4(a) and Fig. 5(a) are obtained with a 3D laser scanner. Fig. 4(b)~(d) and Fig. 5(b)~(d) show the multi-resolution models at different threshold $\theta_{\max}$ of the WARRIORS and the Kettle dataset, respectively.

The proposed MRSR algorithm is compared with the Marching Cubes algorithm. Table 1 shows the execution times of the MC algorithm and the MRSR algorithm at different threshold $\theta_{\max}$. It can be suggested in this table that the running times of our MRSR algorithm decreases with increasing of the threshold $\theta_{\max}$. The reconstructed result by the MC algorithm is the same as that by the MRSR algorithm at $\theta_{\max} = 0^\circ$ . Fig. 6 shows the relationship between face deleting ratio and the threshold $\theta_{\max}$, and Fig. 7 shows that between $D_{\text{surface}}$ and $\theta_{\max}$.

## 5. CONCLUSION

In this paper, a multi-resolution surface reconstruction algorithm is proposed and the relationship between Gaussian curvature and the local normal change is given. The proposed algorithm can be used to directly generate the multiresolution models of cloud points at different user-specified normal change threshold. And it can also used to extract the isosurface in cubes with different dimensions at a threshold. Experiments with the datasets of Terracotta Warrior and Horses in Qin Dynasty and the Kettle in Song Dynasty show that the number of faces will be greatly reduced when the normal change threshold is increased, while the details of surface are maintained well. The MRSR algorithm can generate the mesh and simplify the mesh at the same time, which greatly improved the modeling efficiency.

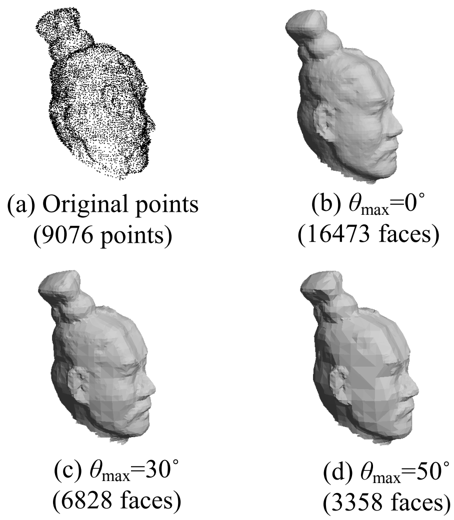


**Fig. 4.** Terracotta Warriors and Horses

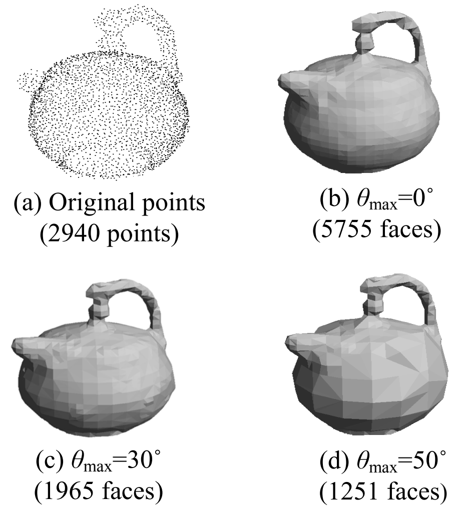


**Fig. 5.** Kettle

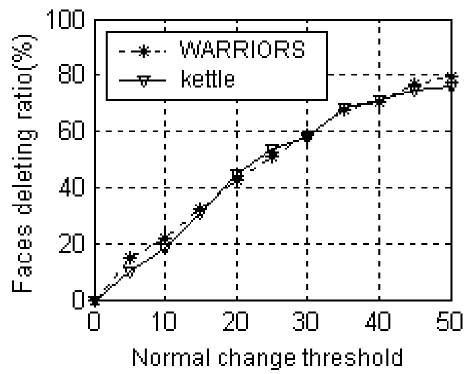


**Fig. 6.** Faces deleting ratio as a function of $\theta_{max}$

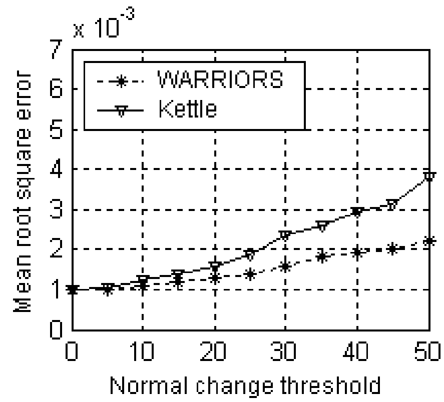


**Fig. 7.** $D_{surface}$ as a function of $\theta_{max}$


## 6. ACKNOWLEDGEMENTS

This work is partly supported by National Science Foundation of China (grant number: 60172040) and National HiTech Program (grant number: 2001AA131022). The second author is also supported by Opening Foundation of Jiangxi Test & Control Center, China (grant number. 2002-14).


## 7. REFERENCES


[1] H.Hoppe, T.DeRose, T.Duchamp, J.McDonald and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH'92 Proceedings*, pp. 71-78,July. 1992.

[2] B.Curless and M.Levoy, "A volumetric method for building complex models from range images," *SIGGRAPH'96 Proceedings*, pp. 303-312, Aug.1996

[3] M.Levoy, K.Pulli, B.Culess, et al. "The digital Michelangelo project: 3D scanning of large statues," *SIGGRAPH'00 Proceedings,* pp.131-144, Aug. 2000.

[4] L.P.Kobbelt, M.Botsch, U.Schwanecke and H.P.Seidel, "Feature sensitive surface extraction from volume data," In: *Proceedings of 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 57-66, Aug. 2001.

[5] Y.J.Liu, and M.F.Yuen, "Optimized triangle mesh reconstruction from unstructured points," *The Visual Computer*, vol. 19, no. 1, pp. 23-37, Jan. 2003.

[6] Shekhar, R., Fayyad, E., Yagel, R., et al. "Octree-based decimation of marching cubes surfaces". In: *Proceedings of the IEEE Visualization'96,* pp. 335-342, Nov. 1996.