J-054

# Extraction of Boundary Points from 3D Scattered Points

Zeyang Qiu† Yoshinao AOKI‡

## 1. Introduction

Reverse engineering includes three main aspects, i.e. digitization, reprocessing and reconstruction. Digitization studies the measuring problems of real object. Currently, three kinds of measuring tools are used commonly, i.e. three coordinates measuring machine (CMM), 3D digitizer and contour lines scanner equipment, such as industrial computer tomography (ICT). Reprocessing emphasizes on measurement data process. Little work has done on this side. Reconstruction is on converting of data points to CAD model used in CAM. Relatively, much work has been done and the earliest one can be traced back to 70 decades of 20 centuries.

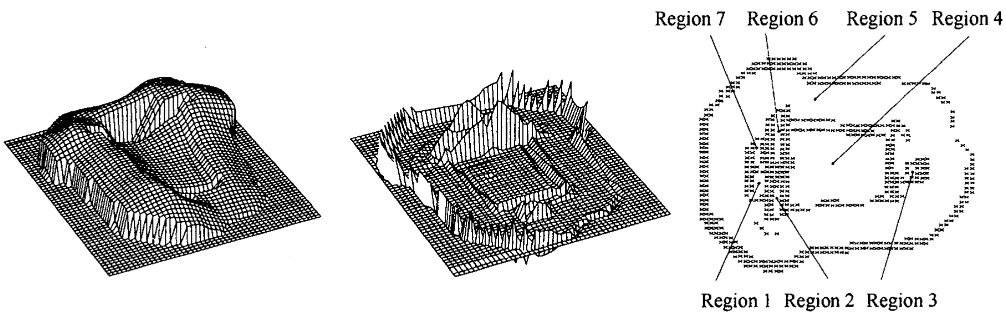
In the procedure of reconstruction, usually needs to know the boundary points or curves of data points in advance. When using CMM to measure, we can get boundary points by special measurement, however, there is no effective ways for other measuring tools, and generally obtained it by interactive way. B. Sarkar and C. H. Meng[1] draw on the experience of edge extraction in image process, and regarded z coordinates of each point as the gray-level intensity of pixel. Then using 6-point Laplacian Gaussian operator to detect a turbine-blade die wax model, which was scanned for 3000( $60\times 50$ ) points on a Sheffield Cordax RS-30 CMM, and got the results as Fig 1. This method has certain limitations, because Laplacian operator is sensitive to noise and the actual measured surface isn't in rectangular array. Recently, Li Jiang-Xiong[2] used projection method, that is, projected points to a 2D plane and separated their border grids in 2D plane firstly, then return to 3D to perform edge revision, energy optimization and finally get the boundary curves. The defect of two ways is they can only deal with strict single value surface, i.e. it could not be overlapped when projected to a 2D plane. Therefore, they are not fit for extraction of ordinary surface boundary points.

The idea of this paper is to depart from surface's local feature, that is, based on the condition it approximates to a plane, then transform part of the points to the position approach to x-o-y coordinate plane and perform boundary point extraction. In section 2, we will introduce the uncertainty of boundary points and their extraction plans. Section 3 discusses the case of 3D spaces. Finally, we summarize the idea and analyze its adaptability.

## 2. Extraction of the boundary points in 2D

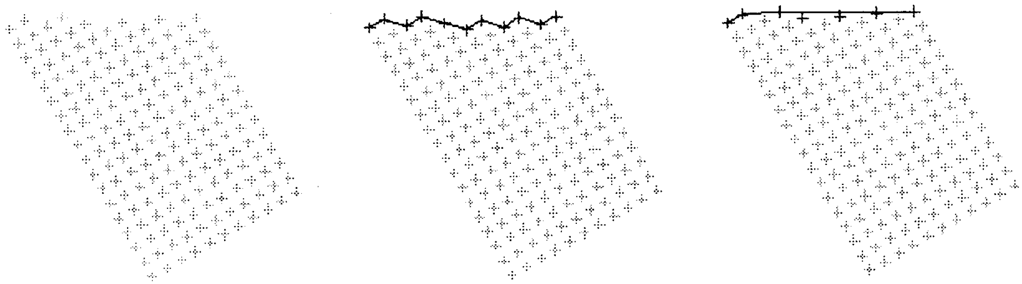
### 2.1 Uncertainty of the boundary points

It isn't a hard problem, in theory, to extract the boundary points from discrete ones, but it is very difficult or impossible to get completely and exactly ones. it has uncertainty if the boundary points (in dark color) are on the real borderline or not. See Fig 2. Fig 2a is an unorganized collection of points, which has two distinct explanations on its acclivitous edge, reference to Fig 2b and 2c. Actually, it is far more than these two kinds. Therefore, it's impossible to obtain the boundary points that




(a) 3D plot of scanned points on turbine-blade model (b) 3D plot of scanned points after use of Laplacian Gaussian operator (c) Detected regions in turbine-blade die

Fig 1. Edges of scanned points on turbine-blade model and its regions



(a) An original point set (b) One explanation of the bevel edge (c) Another explanation of the bevel edge

Fig2. Uncertainty of the boundary points

† IEEE

‡（社）電子情報通信学会, IEICE

coincide with its border, but only its approximation.

**2.2 Extractive plans of the boundary points**

Based on boundary points' complexity and the edge detection idea of Tan Zheng *et al*[3], we conceived three plans to dicide whether a 2D point is a boundary point or not, see Fig 3. For the convenience of description, here gives related definitions.

2.2.1 Definitions

***Point set***: A collection of all scattered data points.

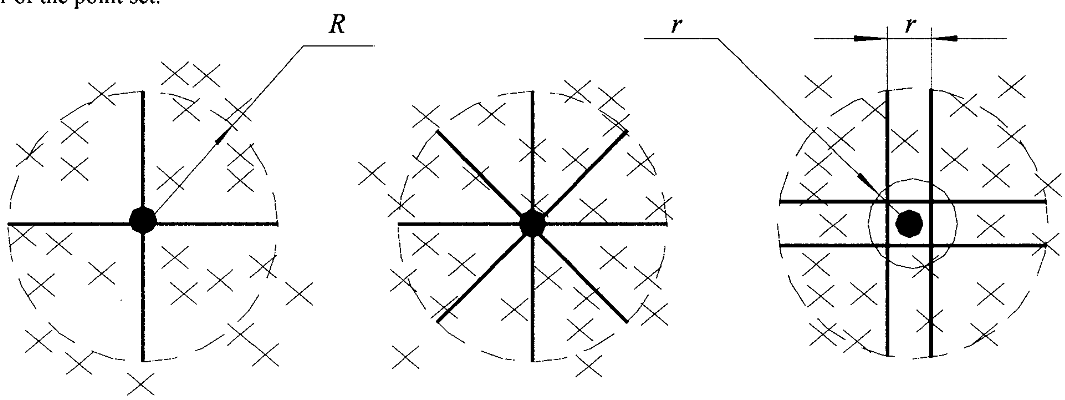
***Current point***: A point in the point set that is processing, which showed with "●" in Fig 3.

***Neighborhood***: It's the part around the current point, which indicated within a hidden circle.

***Local points***: It's all discrete points within the neighborhood.

***Region***: A neighborhood is divided into several parts, which is partitioned with bold lines in Fig 3, and each one is called a region.

***Boundary points***: They're points, saying strictly, on the boundary lines or curves of the surface, but here they're points on the border of the point set.




(a) Quadrantal plan (b) Octal plan (c) "井" shape plan

Fig 3. Extractive plans of the boundary points

2.2.2 Analyses and instructions about the plans

In Fig 3a, the neighborhood is divided into four regions, where if there is one region is empty, i.e. there is no point in the region; the current point is regarded as a boundary one. Because this plan can only detect the points which are in four quadrants, it is unable to deal with the case that empty area goes across two quadrants, see Fig 4. So the neighborhood is divided further, i.e. separated into eight sectors, reference to Fig 3b. In this case, if there are two adjacent regions are blank, the current point is a boundary one. Although the plan in Fig 3b is better than that one in Fig 3a, it is not enough, that is, in case of a point is on the quadrantal lines, it would be equivocal. Therefore, in Fig 3c, we divide the neighborhood into another eight regions just like the form "井", while its judging condition is the same with that one in Fig 3b.

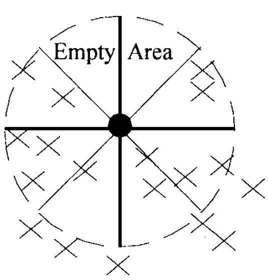



Fig 4. Empty area occupies two quadrants

**Instruction:**

1) $R$ is the radius of a neighborhood. It should be less than the minimum diameter of inner holes; otherwise, it is impossible to recognize boundary points on the hole.

2) $r$ is the width of the hollow showed with "●" in Fig 3c, while its value equals to the minimum distance from the current point to the all local points.

**2.3 Procedures of extraction**

Step 1: Select two arbitrary points from point set; compute the distance between them and as the value of R.

Step 2: designate an extraction plan.

Step 3: take out a point from the point set and regard it as a current point, then search for its local points and decide whether it is a boundary point or not, if it is then set its flag to 1.

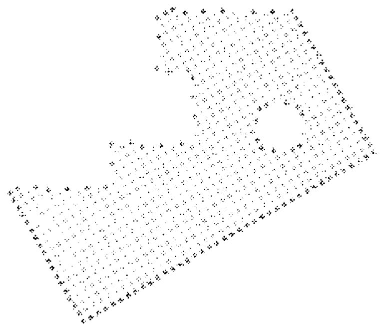
Step 4: Repeat Step 3 till all points in original point set are processed.

**2.4 Test results and their analyses**

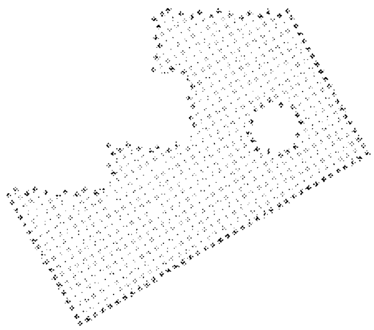
Fig 5 shows the results using three plans to extract boundary points, which indicated in dark color. From Fig 5, we could find that part of the boundary points couldn't be extracted when using the quadrantal plan. While using others, only a few boundary points are left. Even though so we couldn't say that other two plans are better than the quadrantal one. As the front mentioned that boundary points have uncertainty, and the extractive results are only its boundary's approximation. If we judge them from the smoothness, the quadrantal plan may be better.

## 3. Extracting the boundary points of measuring data in 3D

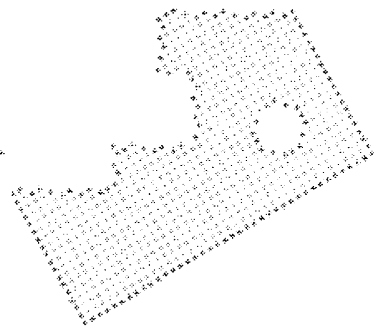
Just similar to the extraction in 2D plane, the extraction in 3D spaces is also to take a current point out continuously from point set and look for its local points, but there are differences in two aspects. One is that using a cylinder when searching for its local points in 3D spaces; the other is that the local points obtained are on a surface, but aren't really on a plane. In this case, if we want to use the algorithm in 2D plane, we not only have to give the hidden circle in Fig 3 a thickness, but to perform the following transformation to local points also.



(a) Using quadrantal plan



(b) Using octal plan



(c) Using "井" shape plan

Fig 5. Test results in 2D plane

Supposing that the current point is $O$, showed with "●" in Fig 3, and the other points in the neighborhood are $\boldsymbol{P}_i$ $i = 0,1,\cdots,s$ .Then the least-square plane of these $s+2$ ① points is expressed in formula (1).

$$(\boldsymbol{P}_i - \boldsymbol{C}) \times \boldsymbol{n} = \boldsymbol{0}, \qquad i = 0,1,\cdots,s+1 \qquad (1)$$

Where,

$C$ is the centroid of points $\boldsymbol{P}_i$ $i = 0,1,\cdots,s+1$. It can be got from formula (2),

$n$ is the unit eigenvector associate with the minimum eigenvalue of matrix $M$ which can be defined by formula (3).

$$\boldsymbol{C} = \frac{1}{s+2}\sum_{i=0}^{s+1}\boldsymbol{P}_i \qquad (2)$$

$$M = \boldsymbol{D}^T \cdot \boldsymbol{D} \qquad (3)$$

In which $D$ is a $(s+2)\times 3$ matrix which defined by points $\boldsymbol{P}_i(x_i, y_i, z_i)$ $i = 0,1,\cdots,s+1$ and $\boldsymbol{C}(x,y,z)$. That is,

$$\boldsymbol{D} = \begin{bmatrix} x_0 - x & y_0 - y & z_0 - z \\ x_1 - x & y_1 - y & z_1 - z \\ \vdots & \vdots & \vdots \\ x_{s+1} - x & y_{s+1} - y & z_{s+1} - z \end{bmatrix}$$

With direction vector $n$ and the original $O$, the local points could be transformed to the location where the vector $n$ coincides with the axis $z$. That is, all local points approach to x-o-y plane and can use the related algorithm in 2D. Fig 6 are the boundary points, which indicated by dark color, extracted with the algorithm mentioned above to the measurement data of facemask.

## 4. Conclusion

This paper synthesized the ideas of neighborhood [1,3] and extracting boundary points in 2D plane [2], and presented a new approach for 2D boundary points' extraction. Based on the fact that local surface approaches to a plane, the method is extended to 3D spaces, and forms a 3D way, which can abstract boundary points in 3D spaces directly. Because the way in 3D needn't to project all points to a plane and only changed the position of local points, it has no strict single requirement to surface's shape, i.e. it can adapt to any shape of complicated surfaces.

① For the convenience of calculation and writing, here let the point $\boldsymbol{O}$ to $\boldsymbol{P}_{s+1}$.

## References


1) Sarkar B and Menq C H. Smooth-Surface Approximation and Reverse Engineering. Computer Aided Design, 1991,23(9):623-628

2) Li JiangXiong. A Method for Extract Boundary of Complex Surface in Reverse Engineering. Machinery Design & Manufacture, 2000,29(2):26-28 (in Chinese)

3) Tan Zheng and Wang Lisheng. Detection of Edge of Surface Mesh Data. Journal of Computer-Aided Design & Computer Graphics, 2000,12(8):580-584 (in Chinese)

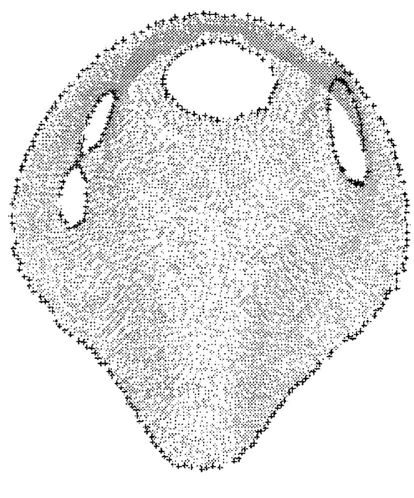



Fig 6. A facemask for absorbing oxygen

## Remark

The research was supported by Natural Science Funds of Gansu Province (ZS031-A25-008-Z) and 'Qing Lan' Talent Engineering Funds of Lanzhou Jiaotong University.