Project #1 Report

ECE-S 682:

Digital Image Processing

Prof. Oleh Tretiak

Zexi Liu zl54@drexel.edu
Zhongchuan Zhang zz57@drexel.edu
April 28th, 2010

Electrical and Computer Engineering
Drexel University

1. Introduction

The goal of the project is to extract the patterns (drawings on the surface) from 3D surface data sets (See Fig. 1) automatically. And find the boundaries (ordered edges) of these patterns.

In this project, an algorithm will be designed in MATLAB to evaluate the images and find the differences or thresholds between the patterns and the backgrounds automatically without user interference. Then the patterns will be extracted from the image, in other words, background will be eliminated. In the end, the boundaries of the discovered patterns will be collected by a traverse algorithm.

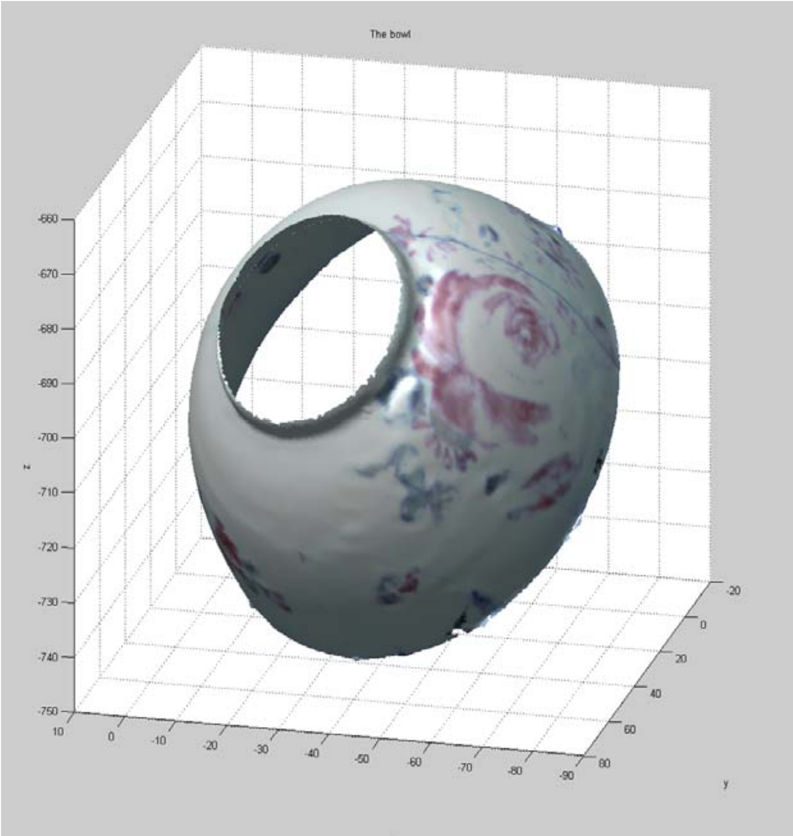

Fig. 1. 3D surface data of a bowl (from a 3D scanner)

Note that the data used in this project is in obj format as following:

```
# Studio
g Polygonal_Model_1
# Number of geometric vertices: 25991
v 4.2222338 -26.9709826 -715.4676914 117 105 136
v 4.4319513 -27.1825176 -715.5084610 110 89 121
v 4.6404568 -27.3899417 -715.4764533 114 86 116
...
# Number of texture vertices: 0
```

```
g Polygonal_Model_1 Triangles_0
# Number of triangles: 51724
f 8235 8720 24023
f 22614 8235 24023
f 8699 8703 24027
...
```

Basically, obj is a geometry definition file format developed by Wavefront Technologies. It is a simple data format that represents 3D geometry alone, namely, the coordinates of each vertex, the RGB color of each vertex, and the faces that make each polygon defined as a list of vertices.

Simply speaking, the bowl shown in Fig. 1 actually consists thousands of triangles and their color information. As a result, our first goal is to evaluate the color of each triangle and find those who belong to the background and those who belong to the drawings.

2. Color Evaluation

Since the color information embedded in obj file is in RGB format, the best way to describe the color distribution is using a histogram chart. Instead of a single pixel in 2D image, the basic element in 3D image is a triangle. After evaluating RGB components of every triangle of the bowl shown in Fig. 1, a color histogram is obtained and shown in Fig. 2.

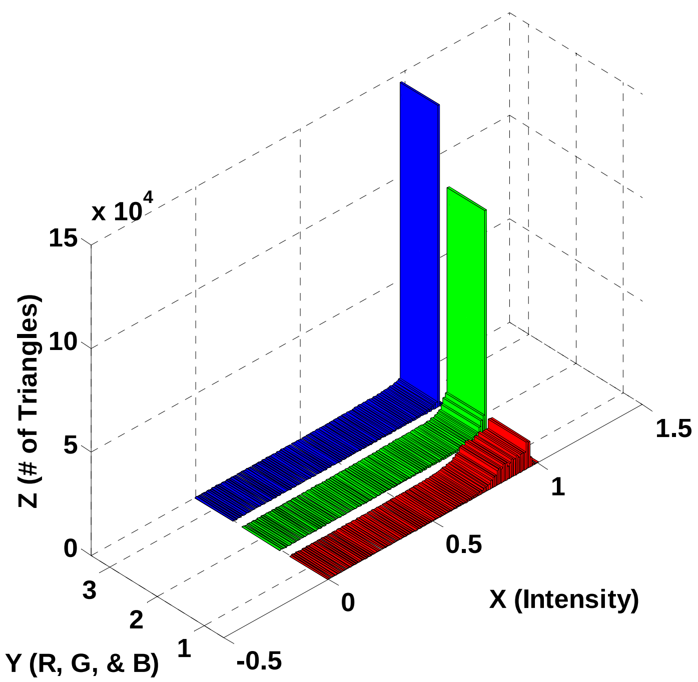

Fig. 2. RGB histogram for the bowl

The blue (right), green (middle), and red (right) bars represent RGB components (Y axis). X axis shows the intensity of each color. Z axis shows the number of triangles whose color is in that specific intensity. Based on Fig. 2, we could observe that most triangles have a very high blue intensity and green intensity. These triangles have the background color. In order to extract the patterns (delete the background), a threshold has been set up. After we delete all the triangles whose blue intensity is greater than 0.975 and green intensity is greater than 0.975, the patterns are obtained and shown in Fig. 3.

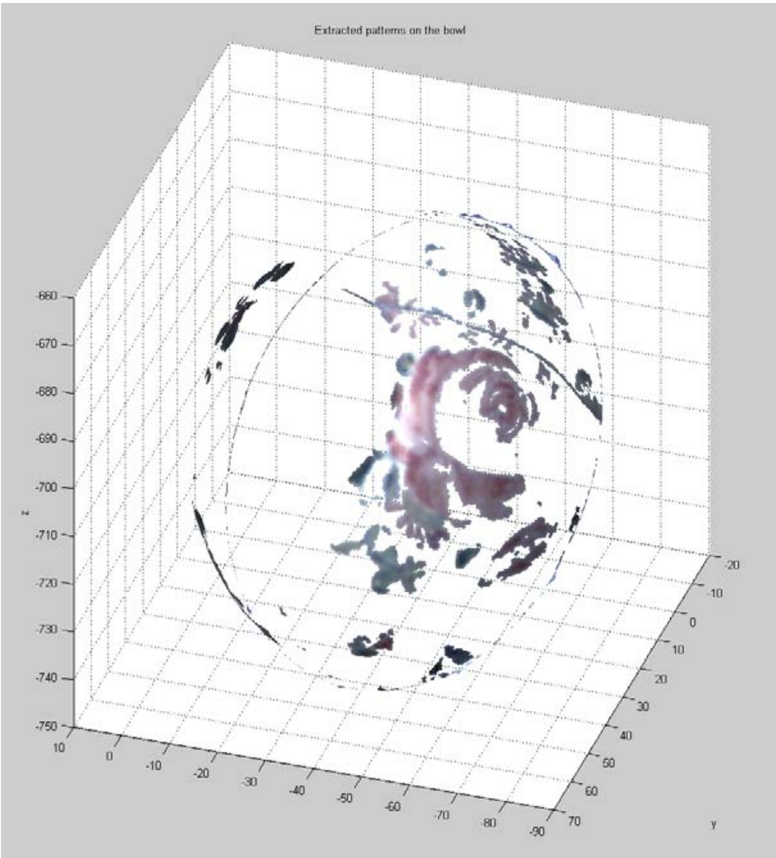

Fig. 3. Extracted patterns on the bowl

Fig. 3 shows the extracted patterns on the bowl. Although we already have the triangles which compose different drawing on the bowl (those disconnected regions in Fig. 3), there still exist two problems. First, different drawing regions are only disconnected visually. These triangles are not grouped and separated yet. Second, sometime we are more interested in the boundaries of these drawings. In the following section, a traverse algorithm is designed to find all the triangles that have at least one unique edge (The edge which is not shared by two triangles. Also, these edges are exactly the boundaries of the drawings).

3. Discover Boundary Edge

For each and every triangle, three of its edges have been evaluated to see if one of them is not shared with other triangles. If an edge in a triangle is not shared with other triangles, it is a unique edge which appear only once in our 3D context. After the first unique edge is detected, the entire boundary of one specific drawing could be discovered by simply find the next connected unique edge. And after ‘traveling’ through one boundary, we move to the next disconnected region and discover its boundary until all the regions have been evaluated.

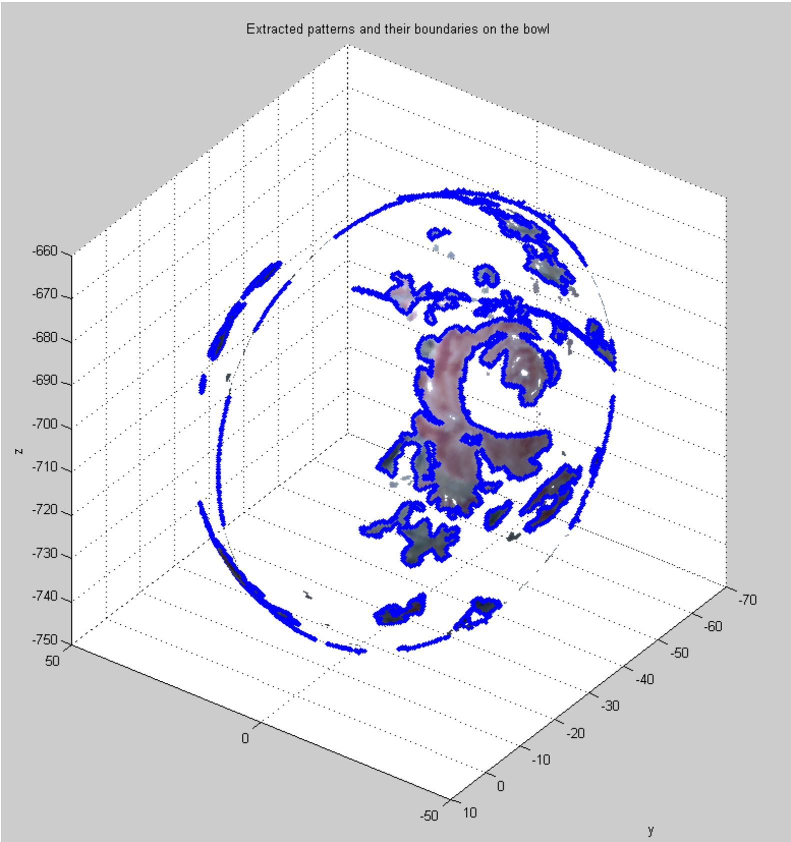

Fig. 4. Discovered Patterns with their Boundaries (blue curves)

Fig. 5 shows the discovered patterns along with their boundary curves. These curves actually consist of unique triangle edges. After we ignore the triangles inside each region, we could obtain only the boundaries (See. Fig .6). Every boundary is a closed 3D curve and all of these curves are stored separately.

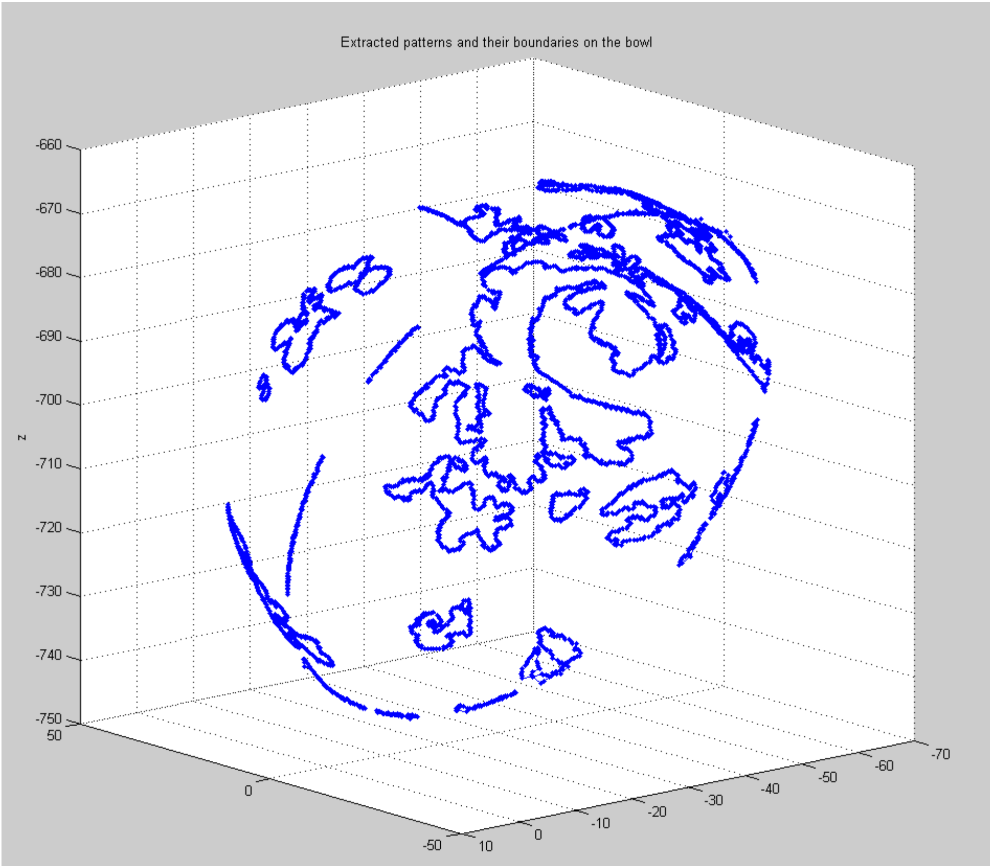

Fig. 5. Discovered Boundaries only (blue curves)

4. Conclusions

For a give 3D data set (in obj format, with color information), this project managed to extract the patterns (drawings/markings on the surface) automatically. The extracted patterns are disconnected regions on the surface. Also, an algorithm has been designed to find the boundaries of these disconnected regions. In the end after eliminating the data inside the boundary, we obtain only the edges of the extracted patterns. Basically, this is a 3D edge detection project based on obj format.