BROWN UNIVERSITY
Department of Computer Science
Master's Project
CS-97-M12

"Sketching Curved Three-Dimensional Surfaces"

by

Bing Chin

Sketching Curved Three-Dimensional Surfaces

Bing Chin

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Department of Computer Science at Brown University

April 1997

[illegible]

Professor John F. Hughes
Advisor

Sketching Curved 3D Surfaces

Bing Chin

May 12, 1997

1 Introduction

How to interpret a 2D sketch as a 3D object is a very important issue in computer graphics. Unfortunately, there are two severe factors which affect the correctness of interpretation. One is that a single 2D sketch can be interpreted *"correctly"* in multiple ways; the second is that it is impossible to interpret a 2D sketch as a 3D object without any prior hypotheses. One way to approach this problem is to give a set of gestures for certain objects, e.g., "three orthogonal lines meeting at one point" represents a cube.

In this project, I interpret any closed contour drawn by a user as a 3D surface whose boundary agrees with the contour. The number and shape of the curves in the contour are restriction free. The details of this project will be given in the following sections. Section 2 describes how this project works. Section 3 describes the approaching used in developing the program. Section 4, describes the underlying mathematics. Finally, Section 5, some ideas for improving this project are discussed.

This project could not have been done without the advice of Prof. John F. Hughes. Thanks to his help and tons of sparkling ideas. Also, I want to thank Lee Markosian, who gave me the *MESH* program which formed the basis for the topological work in this project.

2 How Does It Work?

This project uses an *Open Inventor* interface. The command "surf" will open a view window on the screen. The user uses the mouse to draw the boundary of a surface. A boundary is a closed contour which contains several curves. A curve will be drawn by three actions:*press button*(start point), *drag button*(curve body), and *release button*(end point). There is no restriction on the curve drawing. When a sequence of curves forms a closed contour, the boundary is complete and "surf" will compute the interpreted surface in 3D and display it for you.

3 Approach Highlight

The underlying structure of the program is the following:

1. Get user-input of curves.
2. If a boundary is complete, optimize it.
3. Interpret the boundary as a 3D curve.
4. Use this 3D curve to compute some points of an interpolating surface.
5. Use these points to build a surface mesh.
6. Render the mesh.

3.1 User input

The user draws a series of curves with the mouse. After each curve is finished, we check whether a boundary is finished. If it is finished, we proceed to the next step.

To check whether a boundary is finished, we look for unmarked endpoints of curves(two endpoints "match" if they are sufficiently close in screen space.)

3.2 Optimizing the boundary

To optimize the boundary, we do the following:

1. Fixing a start edge, we sort boundary edges in counterclockwise order.
2. We choose enough points to make a good approximation of the boundary.
3. Distribute these points uniformly by arc-length.

3.3 3D Interpretation of boundary

Given a 2D representation of the boundary, we must build a 3D representation. We first classify each curve as lying in one of the coordinate planes. The classification rule is following:

Given two parameters $0 \leq \alpha_{xz} \leq \alpha_{yz} \leq 90$. Let $\vec{e_i} = e_i$ end point - e_i start point; let $\vec{x}$ be the unit vector of the x-axis of the screen plane. Let $\theta_i =$ the absolute value of the angle between $\vec{e_i}$ and $\vec{x}$. Then

1. if $0 \leq \theta_i \leq \alpha_{xz}$, e_i lies on xy-plane.
2. if $\alpha_{xz} \leq \theta_i \leq \alpha_{yz}$, e_i lies on xz-plane.
3. if $\alpha_{yz} \leq \theta_i \leq 90$, e_i lies on yz-plane.

Start from the start point of the boundary. Compute the embedding of each curve in sequence, by projecting it onto its coordinate plane. The final point of the last edge and initial point of the first edge will generally not agree, so we compute the difference between them and distribute it uniformly along the curve.

3.4 Compute the points

The surface described by the boundary will be regarded as the minimal surface with the given boundary. Each point on the surface will be computed via the Green's function. The details are given below.

3.5 Build the mesh

To generate a surface mesh in R^3, we build a mesh on the unit disk in R^2, and then map this mesh to 3D by the harmonic map computed below.

4 Mathematical Framework

How do we fill in the interior of the surface? A natural way is to find the minimal surface with the given boundary.

The mathematical model is: given the unit disk $D^2 = \{(r,\theta)|0 \le r \le 1, 0 \le \theta \le 2\pi\} \in R^2$ and a piecewise lipchitz function

$$g(\theta) = g_i(\theta) \quad \theta_{i-1} \le \theta \le \theta_i$$

on the boundary ∂D^2, find the map: $f : D^2 \longrightarrow R^3$ such that f is a minimal surface such that f agrees with g on the boundary. i.e $f|_{\partial D^2} = g$.

Let

$$\begin{aligned} f(r,\theta) &= (f_1(r,\theta), f_2(r,\theta), f_3(r,\theta)), \\ f(1,\theta) &= (g_{1i}(\theta), g_{2i}(\theta), g_{3i}(\theta)), if \theta_{i-1} \le \theta \le \theta_i \end{aligned}$$

In order to know that $f(D^2)$ is a minimal surface, we need to know that f is a harmonic map. i.e., $\Delta f = 0$. Therefore, all the components of f are harmonic functions.

Consider the Laplacian equation:

$$\begin{aligned} \Delta f_k &= 0 \\ f_k(1,\theta) &= g_{ki}(\theta),\ if\ \ \theta_{i-1} \le \theta \le \theta_i \end{aligned}$$

Let $G(x,y)$ be the Green's function of the region D^2. $G(x,y)$ can be expressed explicitly by

$$\frac{\partial G(x,y)}{\partial \nu} = \frac{1-|y|^2}{2\pi}|x-y|^{-2}, \tag{1}$$

where ν is the outward unit normal vector and $x, y \in D^2$. Then the harmonic function $f_k(x)$ can be written as:

$$\begin{aligned} f_k(y) &= \int_{\partial D^2} u \frac{\partial G}{\partial \nu} ds \\ &= \frac{1-|y|^2}{2\pi} \int_{\partial D^2} \frac{g_k(x)}{|x-y|^2} ds_x \end{aligned}$$

If we express this equation in polar coordinates, then $ds_x = d\theta$. The expression becomes

$$\begin{aligned} f_k(r,\theta) &= \int_{\partial D^2} u \frac{\partial G}{\partial \nu} ds & (2) \\ &= \frac{1-r^2}{2\pi} \sum_{i=0}^{N-1} \int_{\theta_{i-1}}^{\theta_i} \frac{g_{ki}(\phi)}{(r\cos\theta - \cos\phi)^2 + (r\sin\theta - \sin\phi)^2} d\phi & (3) \end{aligned}$$

5 Future Work

This project can be improved in the following aspects. First, the boundary could be smoothed by using wavelets or other smoothing techniques before we actually store it. Second, the mesh can be obtained by fewer points; then interpolating subdivision can be applied to obtain a smooth surface. Third, editing, such as change the shape of the surface by dragging a point, should be allowed. The fourth, interpret a frame boundary in a 3D object.

References

[1] R.C. Zeleznik, K.P. Herndon, and J.F. Hughes *SKETCH: An Interface for Sketching 3D Scenes. Computer Graphics (SIGGRAPH' 96 Proceedings)* Pages 163-170. New Orleans, Aug. 1996

[2] D. Gilbarg and N.S. Trudinger *Elliptic Partial Differential Equations of Second Order. 2nd Edition, Springer-Verlag 1983*

[3] R. Osserman *A Survey of minimal surfaces. 2nd Edition, Dover Publication, New York 1986*

[4] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweizer and W. Stuetzle *Piecewise smooth surface reconstruction. Computer Graphics (SIGGRAPH' 94 Proceedings)* Pages 295-302. 1994