An Interactive Surface Design Tool*

Yan Zhou and Yuan Zhao
Department of Computer Science,
Michigan Technological University,
Houghton, MI 49931–1295
Email: `[yzhou|yzhao]@mtu.edu`

Abstract

Teaching surface design in computer graphics, visualization and computer-aided design courses usually requires involved mathematics that students feel very difficult to grasp. This poster discusses a system that can illustrate the fundamentals of surface design with an interactive environment in which students can design, modify and manipulate surfaces. In this way, students would interactively learn the fundamentals of surface design rather than through understanding and memorizing mathematical formulae.

1 Problem and Motivation

Curve and surface design has been a very important skill for building models in computer graphics and computer-aided design. However, the involved mathematics is frequently cited as an obstacle for students to understand the subject. Fortunately, although the derivations of important properties may be very mathematical and complicate, students do not have to know much mathematics to design and manipulate NURBS type surfaces. Therefore, our goal is putting these fundamental properties together into an interactive environment for students to learn important concepts.

While a real commercial CAD or modeling system such as Pro-Engineer, IDEAS and Softimage could be used, these systems are very expensive and only a fraction of their features would be used in classroom. Moreover, they are not designed as pedagogical tools. On the other hand, there are only a handful pedagogical tools available and none of them focuses on surface design. To address these problems, we feel that a free and highly portable interactive system that can illustrate important and fundamental concepts is required.

2 Related Work

Although surface design is important in graphics, CAD and computer aided geometric design [2, 1], there are only a handful pedagogical aids. Rockwood and Chambers' multimedia tutorial [3] runs on Microsoft Windows with an emphasis more or less on the curve part. Yen's *Knotty* [4] is a video program explaining important features of B-Spline and NURBS curves and surfaces. Neither provides an interactive environment for students to carry out experiments. A fully interactive system is being developed by Zhao *et al.* [5] in which students can interactively create, edit and modify one or more curves on screen.

3 Approach and Uniqueness

Our system supports four types of surfaces: Bézier, rational Bézier, B-spline, and NURBS (Non-Uniform Rational B-Spline) surfaces. For each type, the surface is defined by a grid of control points $\mathbf{p}_{ij}$, where $0 \leq i \leq m$ and $0 \leq j \leq n$. Students can change the position of a control point to modify the shape of the surface. The surface is in parametric form mapping $(u, v) \in [0, 1] \times [0, 1]$ to a point $\mathbf{p}(u, v)$ on the surface. For B-Spline and NURBS surfaces, the u- and v-direction are further subdivided with knots $0 \leq u_0 \leq u_1 \leq \ldots \leq u_p \leq 1$ and $0 \leq v_0 \leq v_1 \leq v_2 \ldots \leq v_q \leq 1$. Changing the positions of one or more knots (*i.e.*, u_i or v_j) will also change the shape of the surface. The generated surface can be translated, rotated and scaled on-the-fly.

Our system provides students with a wide array of

*This work was partially supported by the National Science Foundation under grant DUE-9653244.

features for them to interactively create, edit and modify a surface. Figure 1 shows a NURBS surface and its control points and control net. The position of (u, v) is controlled by the upper-right window. As (u, v) moves in $[0, 1] \times [0, 1]$, our system can display on-the-fly the point $\mathbf{p}(u, v)$ on surface. The knots, shown on the bottom side and left side of $[0, 1] \times [0, 1]$, can also be changed to modify the shape of the surface.

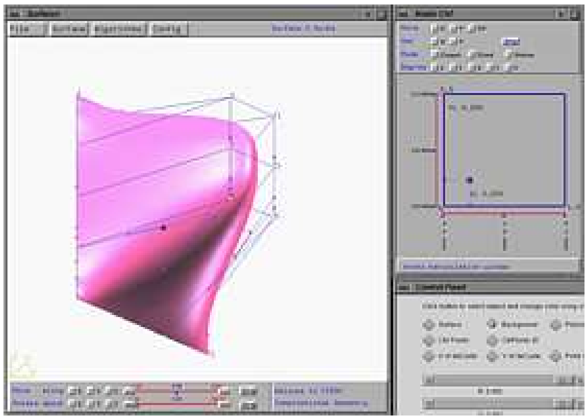

Figure 1: A NURBS Surface

The position of any control point can be changed to modify the shape of the surface. Or, a user can request a flat grid of control points and then moves the control points to generate a desired surface. The surface can be open, clamped or closed. Figure 2 shows the control net of a rational Bézier surface and four small windows. The lower-right one is used for changing the weight of a selected control point, while the other three provides the xy-, xz- and yz-coordinates of the selected control point. Modifying the position in one of these windows will change the position of the selected point and as a result changes the shape of the surface.

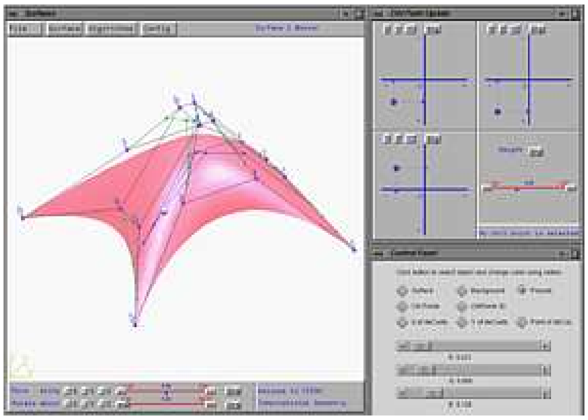

Figure 2: A Subdivided Bézier Surface

Our system can display many important properties in surface design. For example, the inner working of de Casteljau's algorithm (for Bézier and rational Bézier surfaces) and de Boor's algorithm (for B-Spine and NURBS surfaces). With this capability activated, all intermediate computations along the u- and v-direction are displayed as shown in Figure 2. As (u, v) moves in $[0, 1] \times [0, 1]$, the whole configuration moves accordingly. Therefore, students learn the way of generating a point on the surface.

For portability reasons, we only use OpenGL and GLUT. GLUT is a windowing toolkit for OpenGL and is on public domain and available on most platforms, including Windows 95/NT. OpenGL is also available on many platforms. Moreover, for those who do not have access to OpenGL, Mesa (a free OpenGL clone available on popular platforms, including Linux) is an excellent alternative. Therefore, our system is highly portable and will be free to all educators.

4 Results and Contributions

In this poster, we have briefly discussed an interactive tool for surface design. We are currently testing the existing components and at the same time adding advanced features such as surface subdivision, the generation of ruled surfaces, surfaces of revolution and other surfaces, and triangular patches. Since our system can greatly offload the burden of an instructor in teaching surface design and help students to quickly learn the fundamentals by experiments without concerning the involved mathematic, we believe this is the major contribution of our system.

References

[1] Gerald Farin, *Curves and Surfaces for CAGD*, fourth edition, Academic Press, 1997.

[2] James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed., Addison-Wesley, 1990.

[3] Alyn Rockwood and Peter Chambers, *Interactive Curves and Surfaces: A Multimedia Tutorial on CAGD*, Morgan Kaufmann, 1996.

[4] Jonathan Yen, *Knotty: A B-Spline Visualization Program, Part*, Part I and II, Morgan Kaufmann, San Francisco, 1993.

[5] Yuan Zhao, John L. Lowther and Ching-Kuang Shene, A Tool for Teaching Curves Design, this proceedings.