

Cross-Sectional Design: A Tool for Computer Graphics and Computer-Aided Design Courses

Yuan Zhao Yan Zhou John L. Lowther Ching-Kuang Shene
Department of Computer Science
Michigan Technological University
Houghton, MI 49931-1295, USA

Abstract—Teaching curve and surface design in a computer graphics or a computer-aided design course is a challenge because of the very involved mathematics, the lack of easy-to-use tools, and the differing abilities of students to visualize three-dimensional objects. This paper presents the authors' work in teaching cross-sectional surface design using a pedagogical software tool DesignMentor. With this tool, students have hands-on practice and still learn surface design theory in an intuitive way. A summary of the authors' experiences and future directions is also included.

I. INTRODUCTION

Teaching curve and surface design in a computer graphics or a computer-aided design course is a challenge because of the very involved mathematics, the lack of easy-to-use tools, and the differing abilities of students to visualize three-dimensional objects. Under the support of NSF, the authors have developed a curve and surface design tool DesignMentor to address these problems. DesignMentor provides an interactive environment for students to explore Bézier, rational Bézier, B-spline and NURBS curves and surfaces. It also supports cross-sectional design, a surface design technique that reduces the design of a surface to the design of a few curves. This paper describes the capability of DesignMentor and its use in teaching cross-sectional techniques. In what follows, Section II covers some background of NURBS curves and surfaces; Section III provides a brief introduction to our tool DesignMentor; and Section IV presents a general review of cross-sectional design. Section V to Section IX describe the construction of ruled surfaces, surfaces of revolution, and swung, swept and skinned surfaces. Section X discusses some of our findings in teaching cross-sectional design using DesignMentor; Section XI indicates possible future enhancements of DesignMentor; and Section XII has our conclusions.

II. BACKGROUND

This section reviews some technical aspects of NURBS curves and surfaces, shape editing, and advanced features that

Please send correspondence to the fourth author.

are important to cross-sectional design. Details can be found in [5].

A NURBS curve of degree p is defined by $n + 1$ control points $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$, with \mathbf{p}_i associated with weight $w_i \geq 0$, and a knot vector of $m + 1$ knots $0 \leq u_0 \leq u_1 \leq \dots \leq u_m = 1$, where the fundamental equality $m = n + p + 1$ must hold. The equation of this NURBS curve is

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i\mathbf{p}_i}{\sum_{i=0}^n N_{i,p}(u)w_i}$$

where $N_{i,p}(u)$ is the i th B-spline basis function of degree p .

A NURBS surface of degree (p, q) is defined by a grid of $(m + 1) \times (n + 1)$ control points $\mathbf{p}_{i,j}$, where $0 \leq i \leq m$ and $0 \leq j \leq n$, and knot vectors $0 \leq u_0 \leq u_1 \leq \dots \leq u_r = 1$ and $0 \leq v_0 \leq v_1 \leq \dots \leq v_s = 1$. Each control point $\mathbf{p}_{i,j}$ is associated with weight $w_{i,j} \geq 0$. The fundamental equalities must hold: $r = m + p + 1$ and $s = n + q + 1$. The equation of this NURBS curve is

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{p}_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)w_{i,j}}$$

where $N_{i,p}(u)$ and $N_{j,q}(v)$ are B-spline basis functions of degree p and q , respectively.

To modify the shape of a NURBS curve/surface, one can change the positions of control points, the values of weights, and the values of knots. Moving a control point causes a portion of the curve or the surface to move in the same direction (*i.e.*, the *local modification* scheme). Increasing (*resp.*, decreasing) the value of weight w_i pulls (*resp.*, pushes) a portion of the curve/surface toward (*resp.*, away from) control point \mathbf{p}_i . Since the effect of modifying knots is difficult to predict, it is rarely used.

There are a number of fundamental algorithms in NURBS curve/surface design. The most important one is the *knot insertion* algorithms. Knot insertion means adding a new knot into an existing knot vector *without* changing the shape of the curve/surface. *Degree elevation* means increasing the degree of a NURBS curve/surface *without* changing the shape of the curve.

III. DESIGNMENTOR

We have developed a set of course materials and software tools for a new course CS390 Introduction to Computing with Geometry [4]. A by-product of this work is the DesignMentor system for teaching curves and surfaces. The DesignMentor base system consists of two subsystems, one for curves and the other for surfaces. Both subsystems support Bézier, rational Bézier, B-spline and NURBS curves and surfaces. Advanced features such as degree elevation, knot insertion and subdivision are also supported. Both systems provide users with environments for them to experiment and explore many important concepts and skills in curve and surface design and visualize the inner working of important algorithms (e.g., de Casteljaeu's and de Boor's algorithms). Figure 1 is the user interface of both subsystems. The curve subsystem on the left displays the convex hull, moving triad (i.e., tangent, binormal and normal vectors), and curvature sphere of a point on a NURBS curve. Details of both subsystems can be found in [6] and [7]

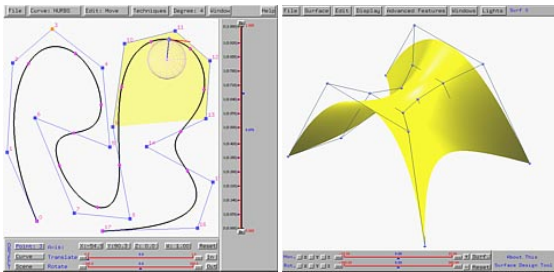


Fig. 1. DesignMentor User Interface

Under DesignMentor a user can click on the screen to create a set of control points and the defined curve appears instantly. For a B-spline or NURBS curve, a degree is also required. Then, a user can move, add or delete control points, modify weights, and change knot values for shape editing. In the surface case, a user can ask for a $(m + 1) \times (n + 1)$ planar grid, and modify the defining elements (i.e., control points, knot vectors and weights) to obtain a desired shape.

IV. CROSS-SECTIONAL DESIGN

Cross-sectional design reduces the design of a surface to the design of a few curves. This is a technique frequently used in car body and ship hull design, and in animation character construction. Many classical surfaces (e.g., ruled surfaces and surfaces of revolution) can easily be obtained with this technique. At least two curves are required to complete a cross-sectional design. The first is the *profile* curves, which describe the profile of the desired surface. The second is the *trajectory* curves. In cross-sectional design, the profile curves usually move along the trajectory curves to generate a surface.

Under DesignMentor, a user uses the curve subsystem for designing profile and trajectory curves. Once these curves become available, based on the type of the desired surface, the curve subsystem generates and exports a NURBS representation to the surface subsystem. DesignMentor supports the construction of the following surfaces: ruled surfaces, surfaces of revolution, swung surfaces, swept surfaces and skinned surfaces. The following sections present an intuition-based introduction to cross-sectional design. For mathematical details, the interested readers should consult [5], [8] and [9].

V. RULED SURFACES

The ruled surface is the simplest one that can be constructed with cross-sectional design. A *ruled surface* requires two NURBS curves, $C_1(u)$ and $C_2(u)$, where $u \in [0, 1]$, and the surface is the union of line segment $\overline{C_1(u)C_2(u)}$. More precisely, as u moves from 0 to 1, the segment $\overline{C_1(u)C_2(u)}$ generates a ruled surface.

The hyperboloid of one sheet is a ruled surface. It is created with two circles, one of which has an angle shift. Under the curve subsystem, a user can design two circles, rotate one of them, and export the ruled surface (Figure 2(a)). The result is shown in Figure 2(b).

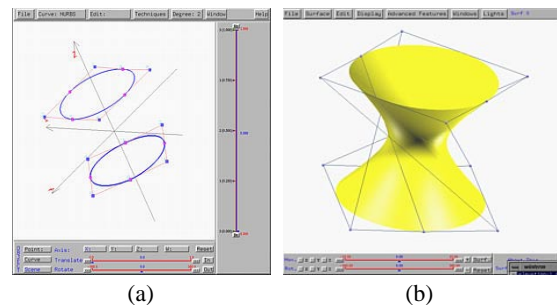


Fig. 2. Hyperboloid of One Sheet

Figure 3 shows an interesting ruled surface defined by a line segment and a NURBS curve of degree 3 twisted like a helix. The result is a helical surface.

VI. SURFACES OF REVOLUTION

Surfaces of revolution are very popular and frequently used as they can be easily manufactured. To generate a *surface of revolution*, one needs a profile curve and an axis of revolution. Under DesignMentor the profile curve is in the xz -plane and the axis of revolution is the z -axis. Figure 4 shows the generation of a torus. The profile circle is a NURBS curve of degree 2 in the xz -plane and the generated torus is a NURBS surface of degree $(2,2)$.

As long as one can design an interesting profile curve, a good-looking and useful surface of revolution can be created

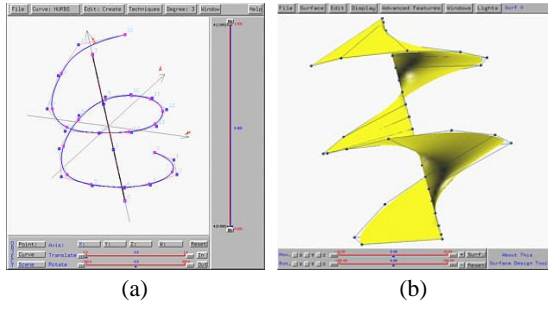


Fig. 3. A Helical Surface

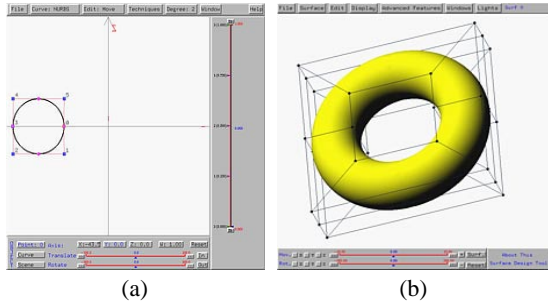


Fig. 4. A Torus

quickly. Figure 5(a) shows a NURBS profile curve of degree 3, and (b) is the generated vase-like surface.

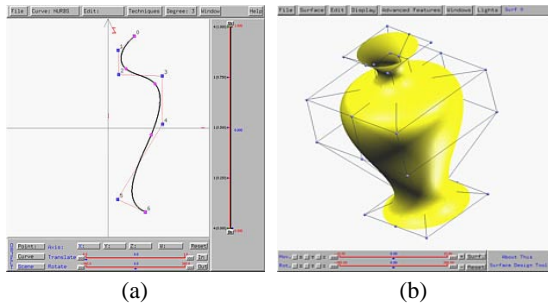


Fig. 5. A Vase

VII. SWUNG SURFACES

Swung surfaces are extensions of surfaces of revolution. To design a *swung surface*, a profile curve in the xz -plane and a trajectory curve in the xy -plane are required. Let the profile and trajectory curves be $\mathbf{P}(u) = \langle P_x(u), 0, P_z(u) \rangle$ and $\mathbf{T}(v) = \langle T_x(v), T_y(v), 0 \rangle$. The swung surface defined by $\mathbf{P}(u)$ and $\mathbf{T}(v)$ is

$$\mathbf{S}(u, v) = \langle \alpha P_x(u)T_x(v), \alpha P_x(u)T_y(v), P_z(u) \rangle$$

where α is a positive scaling factor. The definition of $\mathbf{S}(u, v)$ can be interpreted as follows. For each point on the trajectory

curve $\mathbf{T}(v)$, the xz -plane is rotated about the z -axis so that it contains $\mathbf{T}(v)$, and the profile curve $\mathbf{P}(u)$ is scaled in the direction of $\mathbf{T}(v)$. As a result, a swung surface is obtained by swinging the profile curve about the z -axis and at the same time scaled by the trajectory curve. Cross-sections parallel to the xy -plane resemble the trajectory curve, while cross-sections cut by a plane through the z -axis resemble the profile curve. If the trajectory curve is a circle, the generated swung surface reduces to a surface of revolution.

Figure 6 shows a swung surface as a generalization of a surface of revolution. The profile curve is the same as that of Figure 5; however, the trajectory curve is replaced with a squarish semi-circle. This squarish semi-circle is a NURBS curve of degree 2 with the weights of the middle two control points increased to 5, and as a result the curve is pulled toward the two corners creating a squarish effect. The difference between Figure 5(b) and Figure 6(b) is obvious.

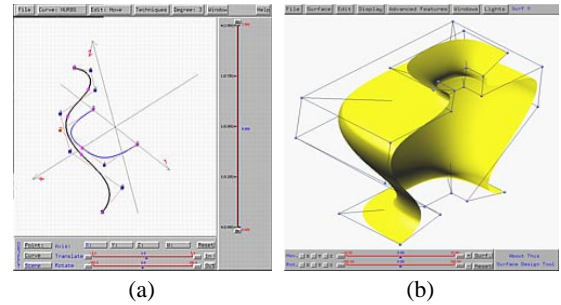


Fig. 6. A Swung Surface

Figure 7 is another example. We use the same trajectory curve as in the previous one; but, the profile curve is replaced with an up-side-down Ω . This up-side-down Ω -curve swings out an interesting surface.

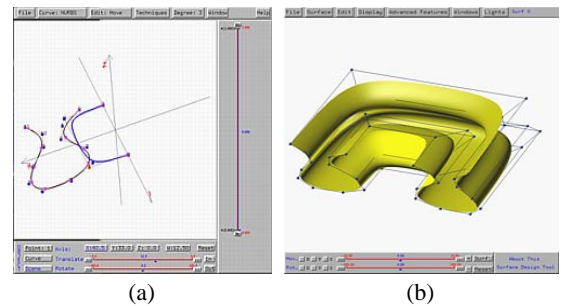


Fig. 7. Another Swung Surface

VIII. SWEEPED SURFACES

The class of swept surfaces is a very general and powerful class. A *swept surface* is generated by sliding a transformed

profile curve $\mathbf{P}(u)$ along a trajectory curve $\mathbf{T}(v)$. More precisely, for each v , the profile curve $\mathbf{P}(u)$ is transformed (*i.e.*, rotated and scaled) and translated to $\mathbf{T}(v)$. Therefore, the surface is defined as

$$\mathbf{S}(u, v) = \mathbf{T}(v) + \mathbf{M}(v) \cdot \mathbf{P}(u)$$

where $\mathbf{M}(v)$ is the transformation matrix at v . Different v can have a different transformation matrix $\mathbf{M}(v)$.

One can obtain all previously discussed surfaces by carefully choosing $\mathbf{M}(v)$. Consider a ruled surface defined by curves $\mathbf{C}_1(u)$ and $\mathbf{C}_2(u)$. One can take $\mathbf{C}_1(v)$ as the trajectory curve and the line segment $\overline{\mathbf{C}_1(0)\mathbf{C}_2(0)}$ as the profile curve. Matrix $\mathbf{M}(v)$ gives the affine transformation that maps $\overline{\mathbf{C}_1(0)\mathbf{C}_2(0)}$ to $\overline{\mathbf{C}_1(v)\mathbf{C}_2(v)}$. Therefore, ruled surfaces are swept surfaces.

Swung surfaces, and hence surfaces of revolution, are also swept surfaces. Curves $\mathbf{P}(u)$ and $\mathbf{T}(v)$ are still the profile and trajectory curves, respectively. Since $\mathbf{T}(v)$ is in the xy -plane and makes an angle with the x -axis, transformation matrix $\mathbf{M}(v)$ is the composition of a rotation about the z -axis that rotates the xz -plane to contain $\mathbf{T}(v)$, and a scaling that scales $\mathbf{P}(u)$ in the direction of $\mathbf{T}(v)$. The rotated and scaled $\mathbf{P}(u)$ is $\mathbf{M}(v) \cdot \mathbf{P}(u)$. Since $\mathbf{M}(v) \cdot \mathbf{P}(u)$ already contains $\mathbf{T}(v)$, no translation is necessary and the surface is $\mathbf{S}(u, v) = \mathbf{M}(v) \cdot \mathbf{P}(u)$. Therefore, swung surfaces are also swept surfaces.

Unfortunately, $\mathbf{S}(u, v)$ may not have a NURBS representation for general transformation matrices. DesignMentor only implements the case of $\mathbf{M}(v)$ being the identity matrix. More precisely, one can only slide the profile curve $\mathbf{P}(u)$ along the trajectory curve $\mathbf{T}(v)$ *without* rotation and scaling.

In Figure 8(a), the circle and S-curve are the profile and trajectory curves, respectively. As the circle moves without rotation and scaling along the S-curve, a tubular surface is generated (Figure 8(b)).

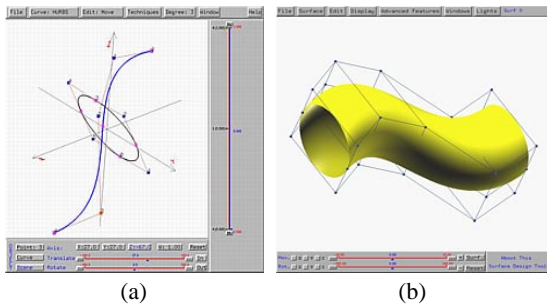


Fig. 8. A Swept Tube

In Figure 9, the profile curve is the one used in Figure 5, while the trajectory curve is a W-curve.

IX. SKINNED SURFACES

In many cases, an exact design is difficult or even impossible. Frequently, engineers design a few key curves and con-

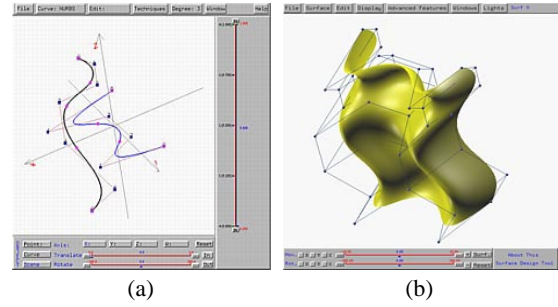


Fig. 9. Another Swept Surface

struct a surface that contains all of them. This technique was referred to as *lofting* and was widely used in ship hull, automobile and aircraft industries. Liming even wrote a book about the use of conics lofting [3]. The modern term of lofting is *skinning* and the surface obtained is a *skinned surface*.

Given a set of n profile curves $\mathbf{P}_1(u), \mathbf{P}_2(u), \dots, \mathbf{P}_n(u)$, there exists at least one skinned surface $\mathbf{S}(u, v)$ that contains all of them. The constructed surface $\mathbf{S}(u, v)$ will not only contain the profile curves, but also has them as isoparametric curves. More precisely, we have $\mathbf{S}(u, v_i) = \mathbf{P}_i(u)$ for $0 = v_1 < v_2 < \dots < v_n = 1$. Note that the order of these profile curves is important because they correspond to increasing v_i 's. Switching two profile curves yields a different skinned surface.

Since $\mathbf{S}(u, v_i) = \mathbf{P}_i(u)$, we require the surface to have the same degree of the profile curves in the u -direction. If the profile curves do not have the same degree, DesignMentor will use degree elevation and knot insertion to make the knot vectors and degrees equal. Degree 2 or 3 for the v -direction would be sufficient in practice. Thus, DesignMentor constructs skinned surfaces of degree $(p, 2)$, where p is the degree of the profile curves.

Figure 10(a) shows a series of identical circles placed in a L-shape and Figure 10(b) is the constructed skinned surface. The constructed surface can be considered as a surface that approximates the surface of sweeping a circle. It is an “approximation” rather than an actual sweeping because the surface is constructed based on a number of discrete *key* profile curves.

Skinning can be used to construct very complex surfaces (Figure 11). A set of profile curves are placed at locations that approximately describe the shape of a car. The constructed skinned surface follows the shape specified by the given profile curves.

Skinning can also be used to approximate very complex swept surfaces. As mentioned earlier, if transformation matrix $\mathbf{M}(v)$ is complex, the resulting swept surface may not have a NURBS representation. To approximate a complex swept surface, DesignMentor allows a user to create a number of instances of the profile curve, and place them along

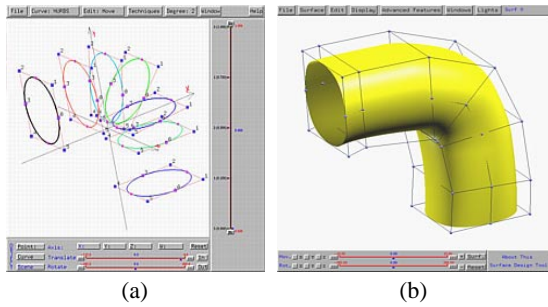


Fig. 10. A Skinned Tube

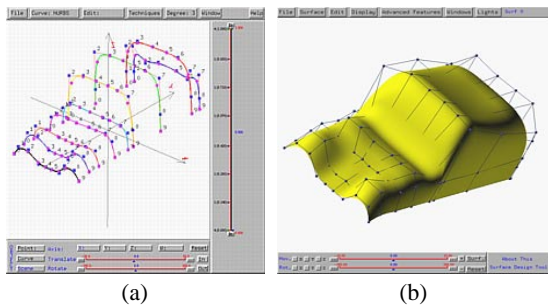


Fig. 11. A Complex Skinned Surface

the trajectory curve. Then, a user can pick an instance and transform it with translation, rotation and scaling. After fine tuning all instance curves, DesignMentor uses skinning to construct a skinned surface. If instance curves describe the desired shape well, the constructed skinned surface would be very close to the true swept surface.

X. EXPERIENCE

DesignMentor has been used twice in our course CS390 Introduction to Computing with Geometry for juniors. The goal of this course is to provide our students with a comprehensive and yet elementary introduction to the use of computers in solving geometric problems. All discussions are based on an intuitive approach. In a ten-week quarter, we were able to cover the representations of and operations on solids, curves and surfaces. With the help of DesignMentor, students can quickly learn the advanced geometric algorithms such as knot insertion, degree elevation and subdivision. We never derive any formulae, however. Students are given important formulae and asked to work out a number of simple examples, and use DesignMentor to verify the effect, because we believe that in a course for juniors it is perhaps not worth spending time in investigating the mathematical depth and elegance.

One of our findings is that students like our non-mathematical approach and realize that even mathematical discussions can be minimized, it cannot be completely

avoided. The gap between their mathematical background and the understanding of curve and surface design is closed by working out simple examples and using DesignMentor to explore and experiment.

Cross-sectional design is normally discussed with minimal mathematics by defining and explaining a particular surface type using an intuitive approach with examples. Then, students are asked to use DesignMentor (we teach our course in a computer equipped classroom) to do hands-on design examples. This learning-by-doing and hands-on approach has been very effective, because in less than 1.5 hours all students claim they know what cross-sectional design is and are confident about what they have learned. Some students even mention (in their attitudinal survey) that they do not have problems in understanding and doing cross-sectional design, even though they may have trouble in designing NURBS surfaces from scratch. Based on this observation, we believe that cross-sectional design can be taught in a computer graphics and/or computer-aided design course after students have learned about curves. In so doing, students will be able to quickly learn and create good-looking surfaces and be prepared for more complicated discussions of other types of surfaces.

XI. FUTURE DIRECTIONS

The cross-sectional design component sits on top of DesignMentor. Other components for supporting our course are under development. These include regular and irregular mesh interpolation and approximation, which is very frequently used in animation character construction, multi-sided surface patch design for smoothing sharp corners (of the intersections of walls), triangular patch design, and surface interrogation techniques that can help engineers and students to identify irregular areas of a surface (e.g., sudden curvature change). But, the most important component would be the support of the blossoming principle. With blossoms, one can treat Bézier, rational Bézier, B-spline and NURBS under a single elegant and simple theory, which, we believe, would benefit many students and engineers.

Since curve and surface design is about three-dimensional objects construction, the ability of visualizing and/or recognizing three-dimensional objects is crucial to student's success. It is well-known that not all students possess this ability and that there are differences in genders ([1] and [2]). We have also observed this in our class that some students are weaker in converting two-dimensional images on screen to three-dimensional objects. While training can improve this situation, we would prefer to address this problem from the very beginning. We plan to develop a component for displaying stereographics images and embark a new approach using virtual reality. In this way, students will be able to sense, touch and manipulate three-dimensional objects directly. Moreover, they will be able to step into the virtual

world to actually see and learn geometric properties of curves and surfaces.

XII. CONCLUSIONS

We have presented an overview of the cross-sectional design capabilities of DesignMentor, and experience of using it in a new course CS390 Introduction to Computing with Geometry. We also point out possible future development for enhancing DesignMentor.

The DesignMentor base system that supports cross-sectional design is currently available to the public for free. Interested readers can find all related information at the following URL:

<http://www.cs.mtu.edu/~shene/NSF-2/index.html>

This page contains links to a set of online course notes of our course, the manuals of DesignMentor, and a download page for retrieving DesignMentor. Currently, DesignMentor can run on SGI, SunOS, Solaris, Linux and Windows 95/98/NT platforms. It is written in C and OpenGL and uses GLUT for windowing operations. For those platforms that do not have official OpenGL support (*i.e.*, SunOS and Linux), a public domain OpenGL compatible system Mesa is used. This page also contains the recent work of the Geometric Computing Group, which include papers, workshop, posters and theses.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under grant DUE-9653244. The fourth author is also supported by the National Science Foundation under grant number CCR-9696084 (formerly CCR-9410707) and by a grant from the 1998–1999 Michigan Research Excellence Fund.

REFERENCES

- [1] R. Gorska, S. Sorby and C. Leopold, Gender Differences in Visualization Skills – An International Perspective, *The Engineering Design Graphics Journal*, Vol. 62 (1998), No. 3, pp. 9–18.
- [2] S. Hsi, M. C. Linn and J. E. Bell, The Role of Spatial Reasoning in Engineering and the Design of Spatial Instruction, *Journal of Engineering Education*, No. 1 (April), 1997, pp. 151–158.
- [3] R. A. Liming, *Practical Analytic Geometry with Applications to Aircraft*, Macmillan, 1944.
- [4] J. L. Lowther and C.-K. Shene, Geometric Computing in the Undergraduate Computer Science Curricula, *The Journal of Computing in Small Colleges*, Vol. 13 (1997), No. 2 (November), pp. 50–61.
- [5] L. Piegl and W. Tiller, *The NURBS Book*, Springer-Verlag, 1995.
- [6] Y. Zhao, J. L. Lowther and C.-K. Shene, A Tool for Teaching Curve Design, *The Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education*, February 25 – March 1, Atlanta, Georgia, 1998, pp. 97–101.
- [7] Y. Zhou, Y. Zhao, J. L. Lowther and C.-K. Shene, Teaching Surface Design Made Easy, to appear in *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*, March 24 – 28, New Orleans, 1999.
- [8] C. Woodward, Cross-Sectional Design of B-spline Surfaces, *Computers & Graphics*, Vol. 11 (1987), No. 2, pp. 193–201.
- [9] C. Woodward, Skinning Techniques for Interactive B-spline Surface Interpolation, *Computer-Aided Design*, Vol. 20 (1988), No. 8, pp. 441–451.