Collaborative Geometrical Modeling in Immersive Virtual Environments

Martin Usoh and Mel Slater[1],
Department of Computer Science,
University College London,
Gower Street,
London WC1E 6BT, UK.

Tzwetomir I. Vassilev[2],
Department of Computer Science,
QMW University of London,
Mile End Road,
London E1 4NS, UK.

Abstract. Although single person virtual environment systems can aid the individual in particular aspects of work they do not address issues of interaction and collaboration between participants and how these can enhance productivity. We present a system for geometrical modeling which permits collaboration between designers at physically separated sites to build and modify objects composed of free-form surfaces. The system is immersive therefore allowing workers not only to be in the same environment as the objects they're designing, but to come together and interact in the same extended space. In this context higher levels of interaction between workers must be used to cooperatively design and modify complex objects. We realise the importance of appropriate behaviour and provide tools which are intuitive in their use. These tools enable the designer to use natural hand and body motion to sweep out complex surfaces and to interactively deform and reshape them. These objects can in turn be seamed together and used as components for more complex and higher level structures.

Keywords. Collaborative virtual environments, distributed virtual environments, virtual reality, virtual classroom, information retrieval, geometrical modeling.

1. Introduction

This paper describes an application for multi-participant geometric design that is being constructed under the DEVRL project (Distributed Extensible Virtual Reality Laboratory). This project is a collaboration between four UK universities to establish a distributed virtual reality laboratory initially exploiting the Internet (in particular SuperJANET) as the communication channel for constructing multi-

[1]m.usoh@cs.ucl.ac.uk, m.slater@cs.ucl.ac.uk

[2]Permanent address for correspondence: ceco@ait.tu-rousse.bg

participant applications. The philosophy of the project as a whole, and its various applications are described in [1].

In [2] we described an initial system for creating and modifying free-form geometric surfaces in immersive virtual environments. This was based B-Spline surfaces, and exploited an interaction model that we refer to as "body centred interaction" (BCI) [3]. This interaction model is designed in an attempt to maximise the sense of presence of individuals in immersive environments. In this new work we have extended the sophistication of the underlying geometrical model, and have correspondingly extended the application of the BCI paradigm. A major purpose of the work is to provide the ability for shared modeling, where a (small) number of designers simultaneously inhabit a virtual space for collaborative/cooperative work. This raises a number of interesting problems for the shared environments in which there is a high degree of interaction and necessity of collaboration between participants working together on complex objects. For instance, in creating a design what should be the relationship between the participants? Should it be hierarchical whereby each designer is able to modify his/her own segment of the design but the chief designer is able to affect changes to the whole design? How will the system deal with such a structure and how can communication be facilitated between the different groups. Since the mid 1980s such issues have been debated in Computer Supported Co-operative Work (CSCW) [4]. Investigations have been conducted using video and audio links where users can interact with a viewpoint from their own space. Now in the mid 1990s issues are being considered relating to participants whose viewpoints are from within the same extended virtual space. We note, however, that such issues are most effectively investigated using systems which are able to deliver immersive capabilities.

In describing the UNISURF CAD system, Pierre Bezier distinguished three classes of surface design problem [5]:

- Objects that require great accuracy, that are of major technical importance for the product being designed, such as turbine foils or boat hulls;

- Objects that are used only as part of an assembly, for example, to separate other parts, and which do not require such great accuracy as the first;

- Objects created by stylists, where their aesthetic appearance is the most important characteristic.

In this research we restrict attention to the third type of design, only because it is recognised that with virtual environments available today, it is not feasible to create objects with the precise accuracy required for the first two. However, concepts that relate to the third type of design will invariably be relevant to the other two. Our work is also mainly focused towards immersive environments without precluding application to non-immersive desktop systems. We note, however, that given an

environment which is capable of including participants as integral and embodied parts of the virtual world their productivity can become increased since there is less of a need to learn new and often inappropriate paradigms originally developed for desktop systems. The designers can be placed in the same space as the object they are creating and are able to utilise natural movements of their body in creating the design. In addition they can relate more directly to the objects which are being modeled in terms of size and spatial characteristics.

In Section 2 we briefly describe the deformable B-Spline surface model and its capabilities. In Section 3 we discuss how these capabilities are mapped to the interaction paradigm, and in Section 4 the particular collaborative aspects. Conclusions are presented in Section 5.

2. Deformable B-Spline Surface

For designers to effectively utilise a modeling system they must be aware of the properties and behaviour of the materials used in the system. The virtual materials must exhibit constant and repeatable deformations following a set of characteristics such as rigidity, cohesiveness, elasticity and plasticity. We assume that the properties of each surface will allow it to:

- be deformed when a force or a set of forces is applied in an appropriate manner;
- be split or broken into a set of component surfaces;
- regain its previous shape after a deformation;
- seam with other surfaces given appropriate adhesive and cohesive forces.

In [2] we used a simple CAGD scheme based on B-Spline surfaces, represented by control points to create deformable surfaces. In order to change the surface the control points would have to be manipulated. As was pointed out, and as has been pointed out by others before, this does not provide for an elegant method for deforming a surface. If the surface is complex, there are too many control points to manipulate, and the relationship between the surface and the control points becomes non-intuitive. For a single bi-cubic Bezier patch we must deal with sixteen control points. Even in this simple case parts of the patch will occlude some control points and the spatial relationship between control points and the patch is difficult to understand. Given the capability of an immersive virtual environment system, however, a designer can become more aware of this relationship when in comparison to viewing through a desktop display without stereo. It nevertheless

remains impractical for the designer to use direct manipulation of the control points to achieve a desired shape since the effect on the surface of manipulating a group of control points must be well understood [6].

In line with our theory of body centred interaction, direct manipulation of the surface became a goal. However, in order to provide an added sense of "reality" to the deformation process we opted for a mixed geometrical-physical model, the underlying paradigm being that of the application of forces to a surface that would deform elastically as a result. In order to achieve this the method of minimising appropriate functionals was exploited, which is briefly described below.

Let $\mathbf{w}(u,v) = [x(u,v), y(u,v), z(u,v)]$ be a B-Spline surface parameterized by u and v, and let $\mathbf{f}(u,v)$ denote the applied sculpting forces on it. The energy functional for surfaces was employed [7] shown in Eq. (1).

$$E = \iint_{\text{surface}} [c_1 \mathbf{w}_u{}^2 + c_2 \mathbf{w}_v{}^2 + c_3 \mathbf{w}_{uu}{}^2 + c_4 \mathbf{w}_{uv}{}^2 + c_5 \mathbf{w}_{vv}{}^2 + c_6 \mathbf{w}_{u^2v}{}^2 + c_7 \mathbf{w}_{uv^2}{}^2 + c_8 \mathbf{w}_{u^2v^2}{}^2 - 2\mathbf{fw}]dudv\,, \quad (1)$$

where the suffixes mean partial derivatives in respect to the parameters u and v.

The integral (1) has the following advantages over the other functionals described in the literature [8, 9, 10, 11]:

(a) Its minimisation leads to a linear system $\mathbf{K}u\mathbf{V}\mathbf{K}v = \mathbf{C}u\mathbf{F}\mathbf{C}v^T$ where $\mathbf{V}$ is the matrix of spline control points matrix, $\mathbf{K}u$ and $\mathbf{K}v$ are stiffness matrices, $\mathbf{C}u$ and $\mathbf{C}v$ are coefficients and $\mathbf{F}$ is the force matrix. As one can see the two dimensionality of control points and forces is preserved and due to this fact the system can be solved very efficiently.

(b) Most of the functionals in the literature referenced above have two parameters to control the physical properties of the surface; resistance to stretching and to bending. Here it is also possible to control the hardness of the material thus affecting the degree of spread of the deformation across the material.

(c) The new proposed fairness norm has two stiffness matrices, one in direction of the parameter u and another one for v. This gives users the opportunity to design nonisotropic materials, i.e. with different properties in the different directions.

The following actions can be performed on the surface:

- Applying a single force or a set of forces;
- Deforming a curve embedded on the surface;
- Deforming an area of the surface;

- Moving a single point from the surface to a new position;
- Moving a surface curve to a new target curve in the space.

3. Interactive Surface Creation

One of the important aspects of our model for the enhancement of presence in immersive virtual environments (for example, [12]) is that there should be a strong correlation between sensory data (e.g., the visual input) and proprioception (correspondingly, the proprioceptive information generated by head movements). This match requires that the internal mental representation of the movement and disposition of a participant's body correspond to the sensory data generated by those body movements. So when, for example, proprioceptive feedback is generated as a result of an arm moving, then correspondingly the participant should see the arm move, and ideally hear the effect of the movement, and (even more ideally) feel any (virtual) objects with which the arm comes into contact as a result of this movement. The maximisation of this match between proprioception and sensory data is one of the dimensions that, in our model, increases the sense of presence that an individual has of being in the virtual environment.

The concept in the BCI paradigm for interaction derives from this idea. BCI requires that appropriate whole body gestures be employed for the realisation of actions in a VE. For example, it is certainly possible to move through a VE by using a 3D mouse, or by making simple hand gestures. However, this reduces the match between proprioception and sensory data - since the visual flow does not match the proprioceptive information about movements of the body. Hence the idea of "walking in place" [12] to simulate real walking was used, and experimental studies confirmed that this method does, other things being equal, increase the reported sense of presence.

Now this approach can also be used to carry out actions that are not possible in everyday life - such as creating surfaces out of nothing. The designer holds a virtual wand-like tool. As s/he sweeps out a shape in space the surface is correspondingly visually created as an interpolated surface through the space that has been swept out. The wand itself can be extended or retracted, and can be bent to various curved shapes - thus the shape of the surface to be swept out is itself variable, and under control of the designer (see Plate 1). This process is essentially transforming the kinetic energy of the designer into visible surfaces, making a relationship between body moves, and visual appearance - turning a sweeping gesture into something concrete.

Having swept out a surface the designer can also deform it. This can be achieved in various ways:

- selecting a point on the surface and applying a force.

- marking out a curve on the surface, and applying a force to the interior of the curve. This is achieved by the designer using the wand to mark out the surface, and then specifying the force and the direction in which it is applied. Only the part of the surface inside this curve will be deformed.

- directly grabbing hold of part of the surface and moving it.

At all times the designer has a virtual body representation - being able to see the trunk of the body, the left and right hands and the legs. In the current system, only the right hand is tracked, so that the left hand is immobile. However, we still make use of the left hand. Rather than the tools for creation and deformation of surfaces being available as 3D icons or "menu" items, a way of thinking appropriate to 2D interfaces, the designer always has the tools literally "to hand". Tools not in use can be placed in the left hand, and so are always quickly available during the design process. Plate 2 illustrates the deformation of a flat swept out surface after applying a force to it.

Since the geometrical representation of the surfaces is based on B-Splines, it is not difficult to seam surfaces together with a desired degree of continuity along the edge through appropriate interpolation of the control points at the joining edges. Where the number of control points on two edges differ a knot insertion algorithm can be used to equalise the number of points. This aspect of the work is being implemented at the time of writing.

3.1 System Architecture

The system is implemented on the DIVISION ProVision100 VPX (PixelPlanes) system under dVS 2.0.4. Although much faster than the ProVision200 used for our original system, this still has problems with updating the model fast enough for the kind of interaction that we would like. For example, when a force is applied we would like the surface to deform continuously (rubber band fashion) as the force is being defined. Unfortunately, the system cannot keep up with this so that it is necessary to first fix the force, and then the surface will jump to the new shape specified by the force. It is hoped that this problem will be resolved with a future version (dVS 3.1).

The dVS operating system is based on a distributed client/server architecture. It consists of a set of independent modules or actors which provide a set of services in the environment. The actors are overseen by a director which maintains the distributed model database. Each actor has a local copy of the database with

elements appropriate to it. For instance the visualisation actor would contain copies of model elements relating to visualisation and rendering such as the material properties of the surface. The collision actor in turn would contain data relating to object positions and bounding volumes. Each actor is free to modify its local database but updates to the global database are performed via the director. The system provides an application actor for user programming. Objects can be defined and modified at this user level and changes then relayed to the director for lower level processing. This, however, can be also be a source of bottlenecks in the system processing, manifesting itself in the form of reduced model update rates. In attempt to overcome this we have moved the model creation and manipulation functions from the user level to the lower visualisation level. In addition we are able to make some direct calls to the renderer level. However, we are still restricted from direct access to the polygons at the renderer level - something which is important in fast creation and manipulation of dynamic objects.

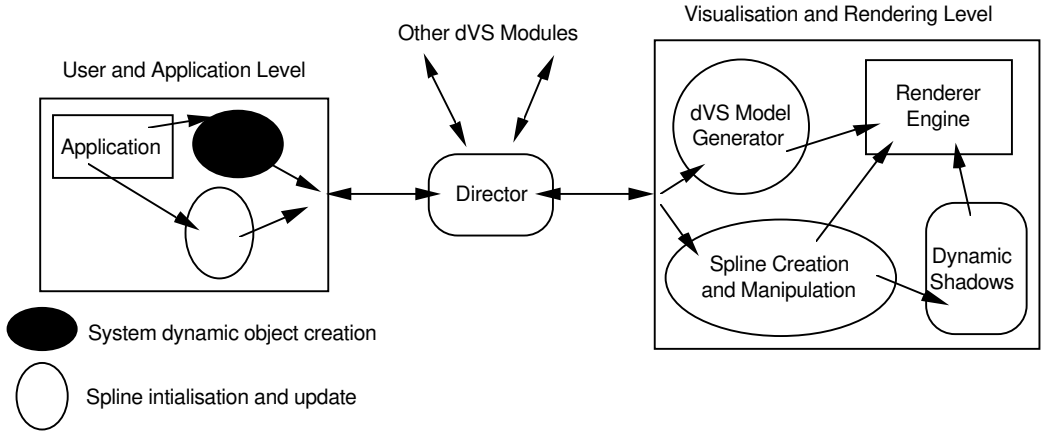

Fig. 1. The modeling architecture

Figure 1 illustrates schematically the system model of dVS with the spline module at the visualisation level. At this level it is also possible to include other modules requiring fast dynamic modifications of polygons to aid model creation. In [13], for example, we have noted the importance of dynamic shadows on presence and spatial awareness. Depth cues obtained from shadows cast by the virtual body or tool onto the sculpted surface or, one surface onto another, can enhance spatial perception and increase model awareness.

4. Collaborative Aspects

The major focus of the DEVRL project is collaboration in shared environments. The DEVRL surface design system will allow several designers to create free form

surface shapes in the same environment. The particular features that we are aiming for in this regard are that designers:

- can pass surfaces between one another;
- can merge the objects that they have created into one combined object;
- can create shapes jointly - e.g., stretching surfaces by pulling at different edges, thus allowing them to collaboratively design objects in a novel way;
- can stretch surfaces around other objects (and around each other), thus creating a basic form of "virtual clothing".

Imagine a virtual world consisting of one plane forming the ground. Our goal is that a group of designers can virtually "terraform" such a world - by deforming the ground plane itself, by creating new shapes and combining them together.

There are significant problems in designing a system for effective collaboration in this context. Designers must be able to inform each other of their intentions: for example, when a designer grabs hold of a shape currently being held by another designer, is this an attempt to take the shape away or to stretch it, or for some other purpose? A secondary channel of communication, such as voice, appears to be essential. When several designers are simultaneously attempting to deform a shape, what is the protocol that decides how exactly the shape deforms? This is made easier with the method described in Section 2, since several forces can be simultaneously applied to a surface to deform it. But the precise interaction method to be used to allow the designers to accomplish this have not been decided as yet.

5. Conclusions

At the time of writing the creation and deformation of surfaces by a single designer in the manner described in Section 3 has been implemented. Curved surface generation has been made capable in three ways:

(a) creating a surface and applying a force (direction and magnitude) to deform it;
(b) creating a surface and manipulating it by directly displacing its control points;
(c) deforming the sweeping tool and then using it to create a curved surface.

We have noted that point (b) is not the desired method of deforming parametric surfaces since it is non-intuitive. The system must provide for an elegant and direct procedure which can allow the designer to utilise real world processes which they are generally familiar with. Of course the designer can also be given appropriate 'magical' tools for enhancing the design, e.g. for smoothing and closing the model.

Using dVS's facility for shared worlds across WANs it has been possible to have two participants simultaneously in the VE and able to see each other's actions. However, at the time of writing the work on full-scale collaborative design is in progress.

References

1. Slater, M., Usoh, M., Benford, S., Snowdon, D., Brown, C., Rodden, T., Smith, G., Wilbur, S.: Distributed Extensible Virtual Reality Laboratory - A Project for Cooperation in Multi-Participant Environments, Proc. of Eurographics Workshop on Virtual Environments, Monte Carlo, Feb. 19-20 (1996). Also this volume.

2. Slater, M., Usoh, M.: Modeling in Immersive Virtual Environments: A Case for the Science of VR, International Conference on Applications of Virtual Reality, 7-9 June, (1994), Leeds, UK, and in Huw Jones, Rae Earnshaw, John Vince, Virtual Reality Applications Academic Press, (1995).

3. Slater, M., Usoh, M.: Body Centred Interaction in Immersive Virtual Environments, in N. Magnenat Thalmann and D. Thalmann (eds.) Artificial Life and Virtual Reality, John Wiley and Sons, 125-148 (1994).

4. Rodden, T.: A Survey of CSCW Systems, Interacting with Computers, December, Butterworth-Heinemann, 3(3), 319-354 (1991).

5. Bezier, P.: The Mathematical Basis of the UNISURF CAD System, Butterworth & Co., (1986).

6. Hsu, W.M., Hughes, J.F., Kaufman, H.: Direct Manipulation of Free-Form Deformations, Computer Graphics (SIGGRAPH), 26(2), July, 177-182 (1992).

7. Vassilev, T.: A Method for Fair B-spline Interpolation and Approximation by Insertion of Additional Data Points, in press (1996).

8. Celniker, G., Gossard, D.: Deformable Curve and Surface Finite Elements for Free-Form Shape Design, Proceedings of SIGGRAPH, 25(4), 257-266 (1991).

9. Celniker, G., Welch, W.: Linear Constraints for deformable B-spline surfaces, Proceedings of the 1992 Symposium on Interactive 3D Graphics, March 165-170 (1992).

10. Halstead, M., Kass, M., DeRose, T.: Efficient, fair interpolation using Catmull-Clark Surfaces, Proceedings of SIGGRAPH, Annual series, 35-44 (1993).

11. Terzopoulos, D., Qin, H.: Dynamic NURBS with Geometric Constraints for Interactive Sculpting, ACM Transactions on Graphics, 13(2), 103-136 (1994).

12. Slater, M., Usoh, M., Steed, A.: Taking Steps: The Influence of a Walking Metaphor on Presence in Virtual Reality, ACM Transactions on Computer Human Interaction (TOCHI), September (1995).

13. Slater, M., Usoh, M., Chrysanthou, Y.: The Influence of Dynamic Shadows on Presence in Immersive Virtual Environments, 2nd Eurographics Workshop on Virtual Environments, Monte Carlo, Jan. 31 - Feb. 1st, Eurographics Workshop Proceedings, ISSN 1024-0861 (1995).