A DRAWING BASED SURFACE MODELER

Roy D. McKelvey
Carnegie-Mellon University

Robert F. Woodbury
Carnegie-Mellon University

27 February 1984

ABSTRACT

A program is described which constructs three dimensional representations which correspond to two dimensional drawings input to the system. Unlike many programs of this type, the interpretation of the two dimensional drawing is constructed in parallel to the two dimensional drawing. The speed required to achieve this simultaneous creation of drawing representations results first from limiting the types of drawing styles that the system can support, and then by designing a data structure which can represent the three dimensional rendering and be accessed very efficiently. The system constructs a relief representation of the input drawing by interactively pushing and pulling a triangulated surface of a polyhedron. A specific set of operators is proposed which operate in a similar manner on both a two and three dimensional representation of the input pen strokes. These operators are derived both from the traditional syntax of drawing and from additional operators possible only in computer systems.

Un programme construisant des representations tri-dimensionnelles de dessins bi-dimensionnels realises a l'aide d'un systeme informatique, est decrit dans cet article. A la difference de beaucoup de programmes de ce type, l'interpretation tri-dimensionnelle du dessin s'effectue parallelement a l'elaboration de ce dernier. La vitesse requise pour generer cette creation simultanee de representations resulte, dans un premier temps, de la limitation des types de styles de dessin que le systeme peut supporter, et egalement, de l'elaboration d'une structure de donnees efficacement accessible, pouvant supporter un rendu tri-dimensionnel. Le systeme construit une representation en relief du dessin, en enfoncant ou extrayant de facon interactive la surface triangulee d'un polyhedre. Un jeu specifique d'operateurs operant de facon analogue sur une representation bi- ou tri-dimensionnelle de coups de crayon, est presentee ici. Ces operateurs derivent a la fois de la syntaxe traditionnelle du dessin, ainsi que d'un langage additionnel, uniquement supporte par l'outil informatique.

The authors would like to thank Douglas Cooper, who amongst many other things has developed the course called Drawing and Perception, from which this project borrows many ideas.

1 Introduction

In the spring of 1983 Carnegie Mellon University announced that it was embarking on a plan which would require each student to purchase a microcomputer upon entrance at the undergraduate level. These microcomputers would be developed specifically to meet the needs of the students and be tied into a campus wide computing network. In addition to an improved computing environment, new services and databases would be made available to numbers of students impossible with the current system of terminal rooms and time sharing computers. This announcement generated much controversy, particularly in the college in which computing had the least emphasis: the College of Fine Arts. Although the college had already embarked on an ambitious effort to integrate computing into fine arts disciplines [7], this announcement posed a serious challenge to all departments within the college to determine areas of their curriculum in which computers could make a positive contribution.

Whether a microcomputer per student would be beneficial to fine arts education is still the subject of considerable debate, but out of this debate there has arisen a strong interest among many in the college to find applications for such a pervasive computing capability. This paper describes a project based on the teaching methods of Douglas Cooper, Professor of Drawing in the Architecture department here at Carnegie-Mellon.

1.1 Kinesthetic and Tactile Drawing

The task of training undergraduate architecture students to draw with confidence and sophistication is not an easy one. The time allotted to for this training usually is no more than two semesters, in which time all presentation tools from drafting to free hand drawing are expected to be well in control. Although the majority of architecture students can be expected to possess above average design skills, they usually have little experience with the different methods of drawing available to an experienced draftsman. These students are particularly weak in free hand drawing and in their ability to comfortably articulate three dimensional sketches. The skill of teaching presentation methods to architecture students is in exposing the students to the widest variety of approaches that can be taken towards the drawing process, and in presenting them in such a way that they are clearly understood. While it is unrealistic to expect the students to master most of these skills, it is hoped that a good foundation can be laid upon which skill can develop over time.

Professor Cooper has been teaching drawing and presentation methods for the last seven years and has recently published a book outlining his particular teaching methods [4]. He has structured his course to be a survey of the various meanings that can be associated to marks produced by a drawing instrument. Rather than allowing these varieties of interpretations of marks to be learned in an evolutionary process, they are explicitly demonstrated and singled out.

Professor Cooper's method involves the students thinking of the drawing instrument as a specific tool for manipulating form. Through a series of exercises the students are encouraged to think of the pen as a renderer of air, a carver of stone, or even as a dispenser of thread. This way of drawing emphasizes the instrument as a molder of form, not as a describer of appearance. The students are encouraged to think of the drawing instrument as a tool for explicitly defining form and space, to develop a 'kinesthetic and tactile' approach to drawing. Take for example the following exercise from the Cooper text:

Reaching out

> *Exercise 9*
> *The Air Within: Section*
>
> With charcoal tone in a sectional view. Describe the air within your assigned building. Where the air is deep, draw more. Where the air is shallow, draw less. Try to bring to this exercise both a sense of depth(The Air Between) and a sense of occupancy(The Air Within).

In this exercise, as in all proposed by Cooper, the task is one dimensional; the students are using the pen to specify the spatial relationships among the objects represented by articulating the air within an object. The pen is to be viewed with respect to this function only. In other exercises, the task is to associate mark making with the specifying of mass, or the whittling of a mass by subtracting from a surface. An example of a solution to the above problem by Joseph Romano is found in figure 1.

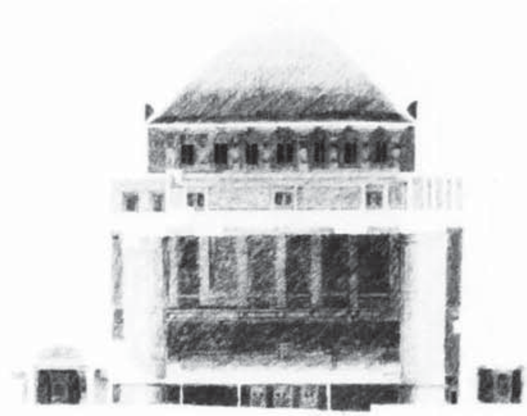

Figure 1: A student drawing from Professor Cooper's class.

1.1.1 The CFA Project

One of the primary goals of the Carnegie Mellon CFA Computing project [7] is the development of software to be integrated in the daily curriculum of fine arts students. Professor Cooper's method of teaching seems a natural candidate for computer graphic techniques: it is already broken down into clearly defined functional units and operates within a restricted sub-domain of the drawing vernacular. Just as Cooper's approach to drawing simplifies the creation and interpretation of drawings for unskilled draftsmen, it also provides a good model for simplifying the computer's task of representing three dimensional sketches. Although the range of representable drawings is narrowed considerably, the clarity of the draftsman's intention allows for a relatively simple program to effectively interpret his marks.

Since in this method of drawing the function of the drawing instrument has effectively only one interpretation, the program has just to support a variety of analogue drawing tools. Although the program aims at modeling just a small set of the tools of an expert draftsman, the set of operators is rich enough to provide an interesting graphics tool. In designing the program then, we have endeavored to define simple three dimensional operators which correspond to the different types of mark making incorporated in Cooper's system. By designing these operators to operate on some three dimensional structure we provide an interactively defined representation of drawings of the type executed by Cooper's students.

2 Design of the System

The system is designed to support two interpretations of the data provided by an input device: a simple two dimensional drawing and a corresponding three dimensional representation controlled by provided pen interpretations. The three dimensional operators are designed with as much fidelity to the two dimensional operators as possible. In particular we have endeavored to make the three dimensional operators fast and direct enough to allow immediacy comparable to those of traditional two dimensional drawing methods.

To emphasize the correspondence between the system's two and three dimensional interpretations of the input stream, we have provided two viewports; one for each representation. As the user draws, both representations update at the same rate. In this way the user can immediately detect the effects his mark making is having on the three dimensional model. Figure 2 shows an example of the display organization.

2.1 The Two Drawing Models

Implementation of the two dimensional representation is straightforward. It merely involves a direct mapping of a bit map pattern corresponding to the current drawing mode to the two dimensional viewport. The success of the system hinges upon the design of the three dimensional surface and its operators. This surface must support both high speed access and update procedures to maintain the high level of interactivity required by the system design.

2.1.1 The Manipulated Surface

In its initial state the three dimensional model is represented by a planar triangulated surface, This surface in its initial state is analogous to the 'blank page' of the two dimensional representation. The structural design of this surface is influenced by three factors: 1) The vertices represented have to accessible at a very high speed, 2) The surface has to be capable of representing a variety of formations in a suitable resolution. 3) Compatibility with the VEGA [7] data structures is essential.

For high speed access to the vertices, a regular grid seems most appropriate. Such a structure reduces the locating of surface points to simple array indexing. The use of a simple matrix to access surface points however puts severe constraints on the varieties of surfaces which can be represented. Since all surface points are rounded off to the nearest grid point, the grid representation is ill suited for representing surfaces with vertical or nearly vertical planes [6]. The grid representation makes insertion of new randomly spaced verteces impossible.

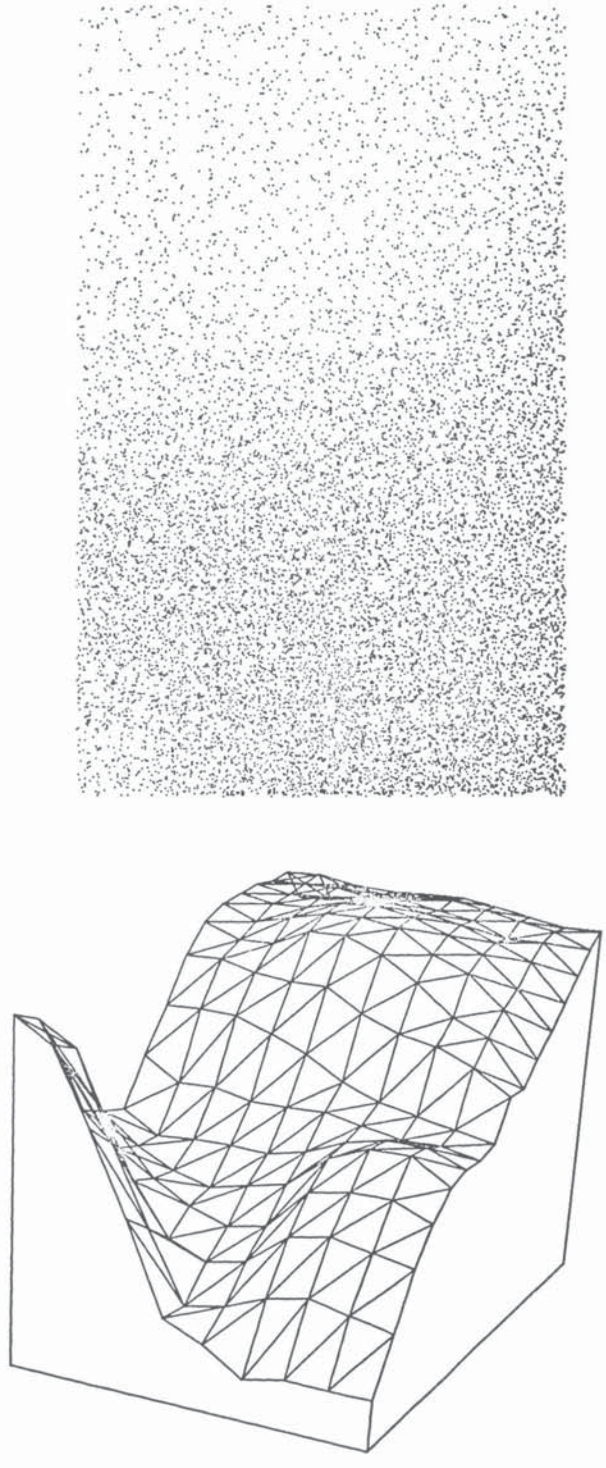

Figure 2: Example of the two viewport display.

To allow the representation of surfaces of varying complexity, some form of triangulated irregular network(TIN) [6] is often used. This surface representation allows the fixing of surface points at exact locations. While almost any surface can be represented by TINS, they support no inherent method of locating specific surface points. Use of such a representation in its raw state would clearly involve too much search to allow the speed required by our system. What we require is the best features of both these methods; a surface which can support a wide variety of configurations and which can also support a very fast hit testing scheme on its vertices.

To achieve this end we have in effect combined these two representations into one. By erecting two parallel data structures to represent the surface, the actual surface representation and its accompanying accessing mechanism can be kept separate. By modeling the surface with the winged edge data structure used in the VEGA solid modeler, and interposing a special purpose hashing grid between the surface and input device we have devised a combination of data structures which provides both the speed and generality desired for our system.

2.1.2 The Winged Edge Data Structure

The winged edge data structure represents a solid by defining it boundary as a network of bodies, faces, rings, edges and vertices [7, 2, 3, 5]. Each entity is described with respect to its topology and geometry. The topology of each of these entities is simply the description of its connectivity to other entities of the same or of different type. The geometry of an entity comprises that information which fixes it in euclidean space [7]. In the form of the winged edge data structure used by the VEGA solid modeler, there is no means of directly detecting those edges which are connected to a given vertex. Such a capability is essential to our system in both its hit testing and screen refresh procedures. To this end we have modified the winged edge structure by creating a new list of links between a vertex and its incoming edges. Thus access to the ring of verteces surrounding a given vertex is kept linear and immediate.

2.1.3 A Hashing Table Access Scheme

Given the modified winged edge data structure described above, any surface composed of polygonal facets can be represented. To provide high speed access to this surface an open hashing [1] structure is used. This hashing structure consists of a series of vertex pointers attached to the surface at regular intervals. The hash function simply takes a pen hit as input and returns the vertex pointer in the hash table which maps most closely to it. This vertex then leads the hit test to an area of the surface which is in close proximity to the actual vertex corresponding most closely to the current pen hit. This scheme reduces the search required by the hit testing mechanism to traversing a small number of vertices surrounding the vertex returned by the hashing function.

The benefits of using these data structures are many. 1) The resolution of the surface and the accessing mechanism are independent. This allows the size and dimension of the surface to be paramaterized and accessible to the user. The speed of updating the three dimensional surface (and by consequence the entire operation of the system) will be proportional to the disparity of the sizes of the two structures. 2) This scheme provides a method of instanciating the surface with a regular grid structure which maps neatly into the hash table, but which also allows the insertion of new vertices in the structure in a strictly local manner. Since a new vertex will be immediately accessible by its neighboring verteces, it can be accessed via the vertices returned by the hash function and be guaranteed to be no further away than the next hashed vertex in any given direction.

2.1.4 Insertion of New Vertices

Once the initial surface and hashing data structures are built, the insertion of new vertices in the surface is controlled by a recursive subdivision routine. This routine is both automatic and invokable by the user. When an area of the surface begins to exceed a set slope tolerance, the section automatically subdivides. This provides a higher resolution in the areas in which a higher concentration of vertices is required to define the form accurately. If the user requires an area of the representation to provide more detail than allowable by the initial resolution he may increase the resolution by invoking the sub-division routine on a specified area.

2.2 The Drawing Operators

In implementing the program we have limited the set of drawing operators to those which have immediate and unambiguous meaning. Only those operators which effect the z value of the surface verteces are included. Although this restriction is severe, there still exist a rich set of operators which can be defined. Gross effects on the surface can be invoked by the use of mass and atmospheric drawing styles. These correspond to pulling the surface out and pushing the surface in respectively. In addition we are creating a set of etching tools which operate by inscribing different shaped furrows in the surface. The development of these and other specific surface modeling tools is currently proceeding.

2.2.1 The Blending Function

In many instances it is desirable to have the current function apply to more than one vertex on the surface. To provide such a functionality we have provided a method of radiating the effects of a function application to the verteces surrounding the accessed vertex. The scope of this blending function can either be determined by the type of mark being made on the two dimensional representation, by line weght for example, or it can be explicitly entered by the user.

3 Conclusion

This paper presents a brief outline of the program design of an interactive surface manipulator. Based on the syntax of mass and atmospheric drawing techniques, the program attempts to define a set of drawing styles which can reasonably bridge the gap between two and three dimensional sketching. The program emphasizes interactivity over completeness. While the set of drawings which can be represented is limited to a bas-relief interpretation, this small domain of drawing is rich enough to provide an interesting graphics application.

References

[1] Aho, Alfred V., Hopcroft, John E., Ullman J.D.
Series in Computer Science and Information Processing: Data Structures and Algorithms.
Addison-Wesley, Reading, Massachusetts, 1983.

[2] Baumgart, B.G.
Winged edge polyhedron representation.
Stanford Artificial Intelligence Report CS-320, Stanford University, October, 1972.

[3] Braid, I.C.
The Synthesis of Solids Bounded by Many Faces.
Communication of the ACM , April, 1975.

[4] Cooper,Douglas,Mall,Raymond.
Drawing and Percieving.
Information Dynamics Inc., Silver Springs, Maryland, 1983.

[5] Eastman, Charles M.
The Design Of Assemblies.
Technical Report 11, Institute Of Building Sciences, Carnegie-Mellon Univ., October, 1980.

[6] Mirante,A.,Weingarten,N.
The Radial Sweep Algorithm for Constructing Triangulated Irregular Networks.
IEEE 2(3):11-21, May, 1982.

[7] Woodbury,.Robert.
VEGA A Geometric Modelling System.
In *Graphics Interface '83*, pages 103-109. Canadian Man-Computer Communications Society, May, 1983.