CONSTRAINT-BASED MODELING OF THREE-DIMENSIONAL SHAPES

Przemyslaw Prusinkiewicz and *Dale Streibel*

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2

ABSTRACT

This paper shows that constraint-based modeling, so far perceived primarily as a graphics technique for man-machine interaction, also provides a viable method for the modeling of complex surfaces. The idea of constraint-based modeling of three-dimensional shapes is described and illustrated by examples. Difficulties related to the practical application of this idea are discussed, and methods for overcoming them are outlined. A potential of the constraint-based approach to the modeling of shapes found in nature is indicated.

RESUME

Dans les systèmes infographiques à contraintes développés jusqu'à présent, les contraintes géométriques étaient utilisées surtout en qualité d'une technique d'interaction. Cependant, les mêmes contraintes peuvent former aussi une base pour modeler des objects à trois dimensions. Cet article présente l'idée principale du modelage à l'aide des contraintes et l'illustre avec des examples. L'application de la méthode pour modeler des surfaces complexes est mise en évidence. Les problèmes numeriques associés sont discutés, et une technique pour les attenuer par une decomposition hiérarchique du modèle est introduite. L'application potentielle de la méthode pour modeler des formes de nature est indiquée.

Keywords: constraint-based modeling, polygon meshes, free-form surfaces.

1. INTRODUCTION

Of all the constraints of Nature, the most far-reaching are imposed by space.

Peter Stevens, *Patterns in nature.*

One common computer graphics technique for representing three-dimensional objects uses polygon meshes. A mesh is defined as a set of connected, polygonally bounded planar surfaces. Polyhedra are examples of meshes, but the notion of a mesh is more general. In particular, it also includes polygonal approximations of curved surfaces.

A mesh description consists of a specification of vertices, edges and faces. Known methods of mesh description require the positions of all vertices to be explicitly specified in a system of coordinates [Foley and van Dam 1983]. This is convenient in many situations, for instance, when a mesh is rendered. However, other parameters may be more convenient to use when a mesh is modeled. For example, consider descriptions of a regular tetrahedron. In terms of edges its definition is trivial - the tetrahedron must have four edges of equal length. In contrast, the description of a regular tetrahedron in terms of vertices is by far less intuitive, since their coordinates cannot be specified without arduous calculations.

Mesh definition by specifying the lengths of edges falls into the category of constraint-based modeling. Instead of specifying vertices directly, a set of constraints, or relations between vertices, is defined. The idea of specifying geometric figures using constraints is not new to computer graphics. It was first implemented in Sketchpad [Sutherland 1963], and followed in several other interactive graphics systems [Knuth 1979, Borning 1981, Van Wyk 1982, Nelson 1985]. Due to its intuitive character, constraint-based modeling was used there primarily as the basis for man-machine interaction. The possibility of building a constraint-based system for the purpose of computer aided design was indicated by Lin, Gossard and Light [1981].

This paper presents a new application of constraint-based modeling - definition of complex three-dimensional shapes.

2. UNSTRUCTURED MODELING

Various types of constraints can be used when describing a mesh. For example, they may characterize vertices as co-linear or co-planar, specify areas of faces, fix the angles between edges and faces, etc. The mesh representation described in this paper uses distances between points as the main form of constraint. Additionally, lines can be specified as parallel to any plane of the system of coordinates (*xy*, *xz* or *yz*), and selected coordinates of vertices can be explicitly given. Explicit specification of some coordinates and directions is necessary to position a rigid object in space, so that it cannot translate nor rotate. Thus, the complete description of a mesh containing *n* vertices consists of:

Table 1. A constraint-based definition of a regular tetrahedron.

Specification	Comment
$x_A = y_A = z_A = 0$	Vertex *A* lies in the origin of the system of coordinates.
$y_B = z_B = 0$	Vertex *B* lies on the axis *x*.
$z_C = 0$	Vertex C lies on the plane *xy*.
$d(A,B) = d(B,C) = d(C,A) =$ $d(A,D) = d(B,D) = d(C,D) = L$	Distance between any two vertices is equal to a given constant *L*.
*e*1: *A–B* *e*2: *B–C* *e*3: *C–A* *e*4: *A–D* *e*5: *B–D* *e*6: *C–D*	List of edges.
*p*1: *e*1–*e*3–*e*2 *p*2: *e*1–*e*5–*e*4 *p*3: *e*2–*e*6–*e*5 *p*4: *e*3–*e*4–*e*6	List of polygons.

- A system of $3n$ equations with $3n$ unknown vertex coordinates. Each equation represents a constraint.
- A list of edges expressed in terms of vertices.
- A list of polygons expressed in terms of edges.

An example of a constraint-based definition is given in Table 1.

In order to find the coordinates of the vertices, the system of constraints must be solved. Since the equations describing distances are quadratic, only numerical methods can provide a general solution. The results presented in this paper were obtained using the Newton method [Conte 1965]. For example, Fig. 1 shows a tetrahedron resulting from the description given in Tab. 1. Another example - a cubo-octahedron - is shown in Fig. 2.

The approach to constraint-based modeling described above is called unstructured, because all constraints are combined into one large system of equations and solved simultaneously. In practice, this approach presents several difficulties. The first difficulty occurs when defining a mesh. If the constraints are not correctly chosen, the resulting mesh will not be rigid, or will contain dependent (redundant) constraints. In both cases, the Newton method will fail to provide a solution (the Jacobian is equal to zero). Proper selection of constraints is a nontrivial task, because the rigidity of an object may depend on particular values of the edge lengths. A simple example, taken from [Hain 1967], illustrates this in the two-dimensional case (Fig. 3).

Even if the set of constraints is correct, its solution may be difficult to find: for a given set of initial values, the numerical method need not converge or it may converge to a wrong solution. This second situation occurs, if more than one object satisfies the given constraints. Unfortunately, this is often the case. For example, even the simple description of a tetrahedron which has been presented in Tab. 1 allows for 8 different solutions: the base *ABC* can be placed in any of the four quadrants of the plane *xy*, and the vertex *D* can be located either above or below this plane.

The following section describes a technique for overcoming these difficulties.

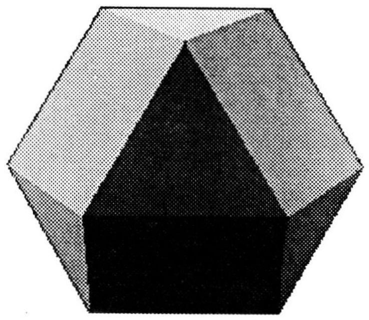
Fig. 2. A cubo-octahedron.

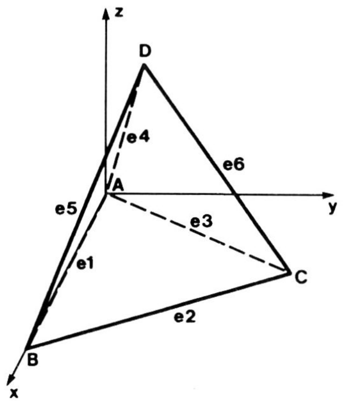

Fig. 1. A tetrahedron described by Tab. 1.

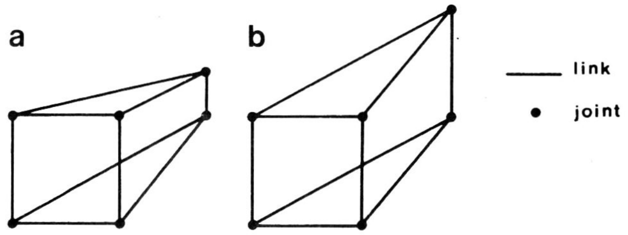

Fig. 3. Rigidity of an object may depend on particular values of distances. Planar meshes (a) and (b) differ only by the lengths of some edges; however, mesh (a) is rigid, while mesh (b) is not.

3. STRUCTURED CONSTRAINT-BASED MODELING

The correct solution can be more easily found if the mesh being defined can be thought of as the last element in a sequence of rigid submeshes. The first element in this sequence is a rigid object, called the kernel, simple enough to be properly defined and solved. The subsequent submeshes differ from each other by a few additional vertices and edges. Thus, the submesh S_{i+1} contains S_i as its proper subset. When calculating S_{i+1}, all vertices of S_i are already known, so that at each stage only a small system of equations has to be solved. Consequently, the sequence of submeshes S_i imposes a structure on the set of constraints.

As an example of the above idea, consider the construction of a pyramid from horizontal slabs. The definition of the slab is given in Fig. 4. After fixing the coordinates of the base, the construction progresses by placing consecutive slabs on top of each other, until the top point is formed (Fig. 5).

In the case of the pyramid, the length of the horizontal edges decreases from one slab to the next one by a constant value: $a_{i+1} = a_i - c,\ c > 0$. The length of the slanted edges b_i is constant. By expressing the lengths of edges using other formulas, the pyramid can be deformed, and more complex shapes can be obtained. For example, Fig. 6 shows an Eiffel-Tower-like shape resulting from decreasing the length of the horizontal edges of the mesh by a constant factor: $a_{i+1} = a_i \cdot c,\ 0 < c < 1$. Fig 7. shows a dome-like shape obtained using septagonal slabs. In this case, the lengths of both the horizontal and the slanted edges are changed according to the formulas:

$$a_{i+1} = \sqrt{a_0^2 - c^2 i^2} \qquad b_{i+1} = \alpha \sqrt{(a_i - a_{i+1})^2 + c^2}$$

Values of the constants α and c are chosen in such a way that the dome can be inscribed in a sphere. Finally, Fig. 8 shows a vase obtained by changing the length of the horizontal edges according to the function $a_{i+1} = a_i + c \cdot cos(i\omega)$, with $c,\ \omega > 0$. The length of the slanted edges is constant, and the slabs are septagonal.

Unfortunately, not every mesh can be decomposed into a sequence of rigid submeshes. Fig. 9 illustrates this in the two-dimensional case: the whole mesh is rigid, but it does not include any rigid submesh. Consequently, no kernel (other than the entire mesh) can be distinguished. Nevertheless, in many practical cases the decomposition is not only

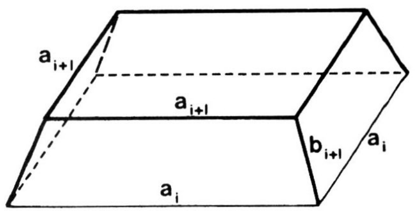

Fig. 4. Definition of a slab.

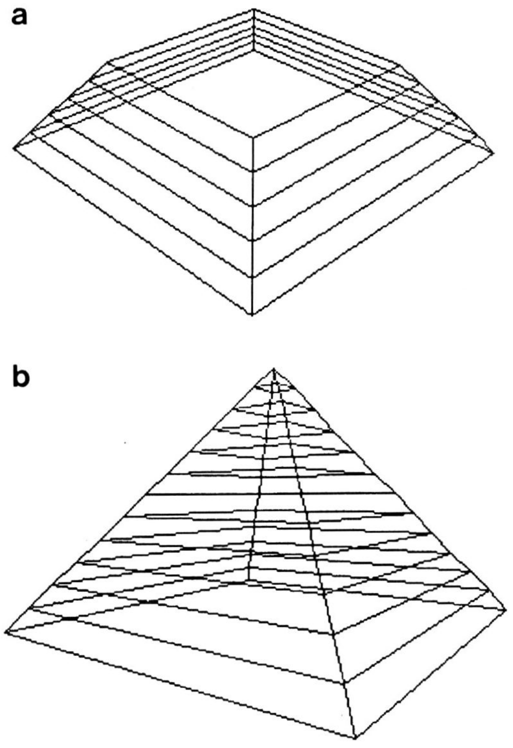

Fig. 5. Construction of a pyramid from slabs. (a) Construction in progress. (b) The final pyramid.

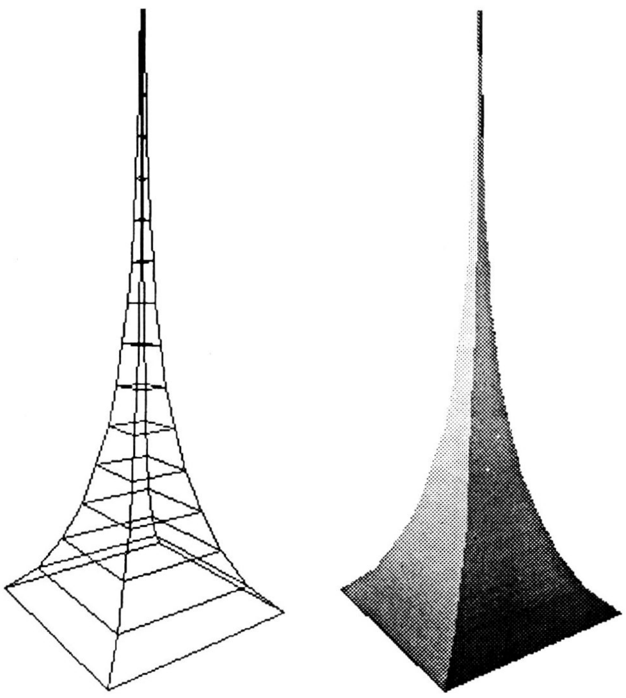

Fig. 6. The "Eiffel tower".

feasible, but it results in a straightforward way from the mesh description. This happens, when graphic modeling can be thought of as a process similar to building a real-life construction. Usually, consecutive phases of a construction correspond to rigid objects, because a construction becoming rigid only in a late phase of development would be technologically difficult to make. Presumably, this argument applies not only to man-made constructions, for example found in architecture, but also to natural objects such as crystals or living organisms, which result from a growth process.

Just as the varying length of horizontal segments determines the shape of the vase, a varying growth rate may determine a shape created by Nature. This point is best described by Stevens [1974]:

> No matter how we try, we cannot make a saddle from five equilateral triangles or a simple cup from seven... . Nature too is similarly constrained. She makes cups and saddles not as she pleases but as she must, as the distribution of material dictates... . If the perimeter of a shell grows at a faster rate than the center, the perimeter curls and wrinkles. No genes carry an image of how to place the wrinkles; no genes remember the shape of the shell; they only permit or encourage faster growth at the perimeter than at the center.

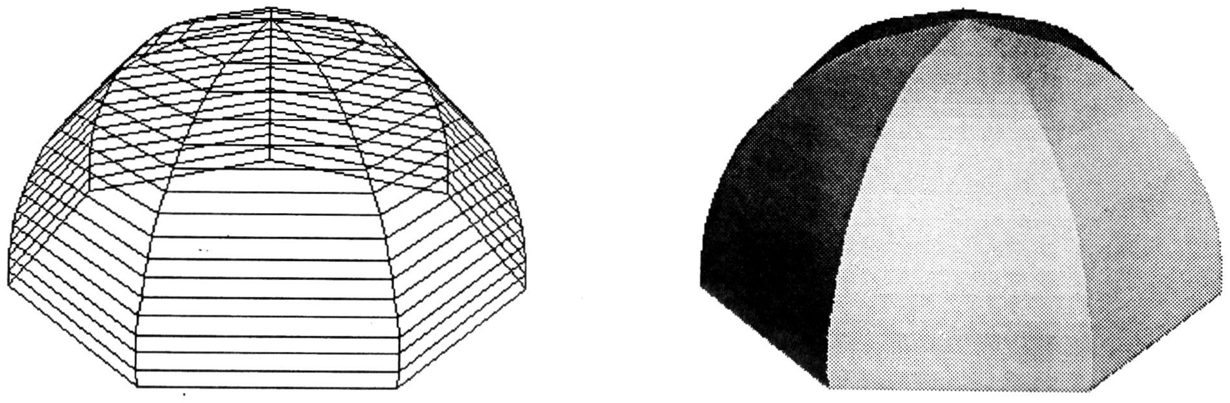

Fig. 7. A dome.

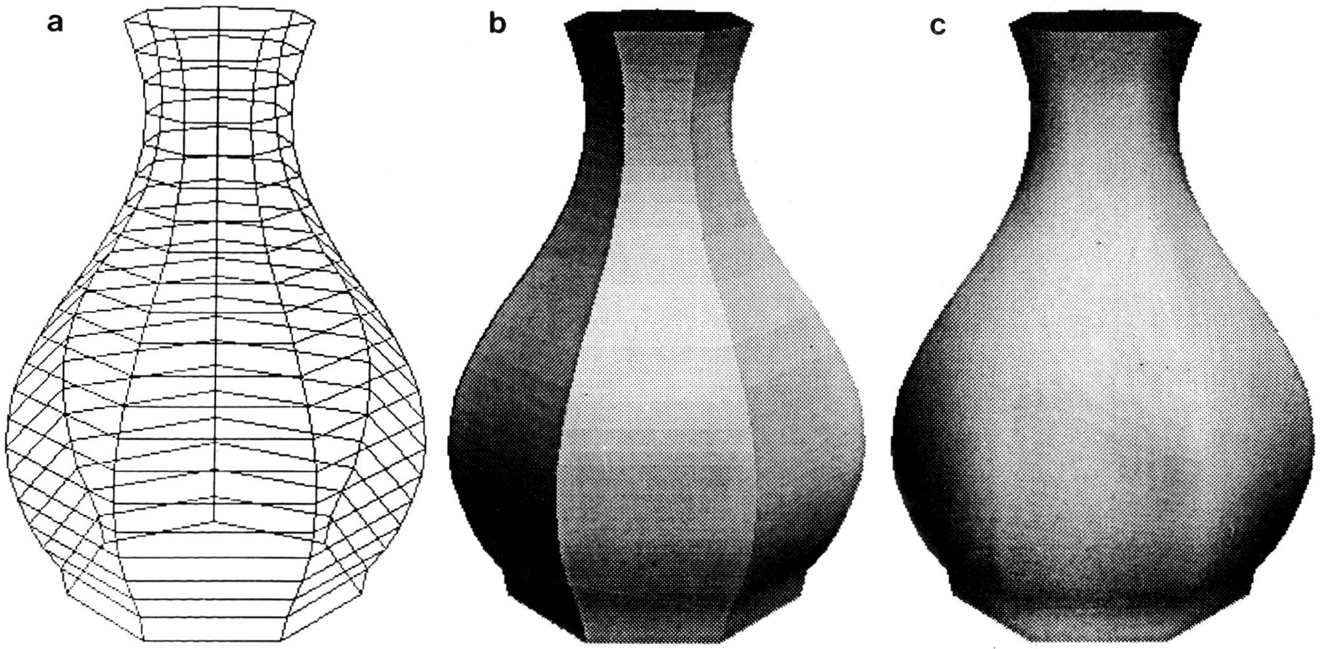

Fig. 8. A vase. Figures (b) and (c) represent two different renderings of the polygon mesh (a).

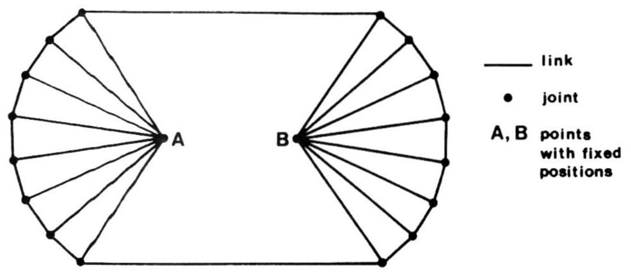

Fig. 9. Example of a planar mesh with no rigid submesh.

Since structured constraint-based graphics modeling may imitate the natural process of growth, it appears to have great potential as a method for modeling shapes found in nature. An example of a shell modeled using the constraint-based approach is shown in Fig. 10. First, a planar section of the shell is "grown" by adding consecutive trapezoidal compartments (a). This provides a basis for creating three-dimensional "top" (b) and "bottom" portions of the shell. The two parts are connected together to form the complete polygon mesh (c). The final shape is shown in Fig. (d).

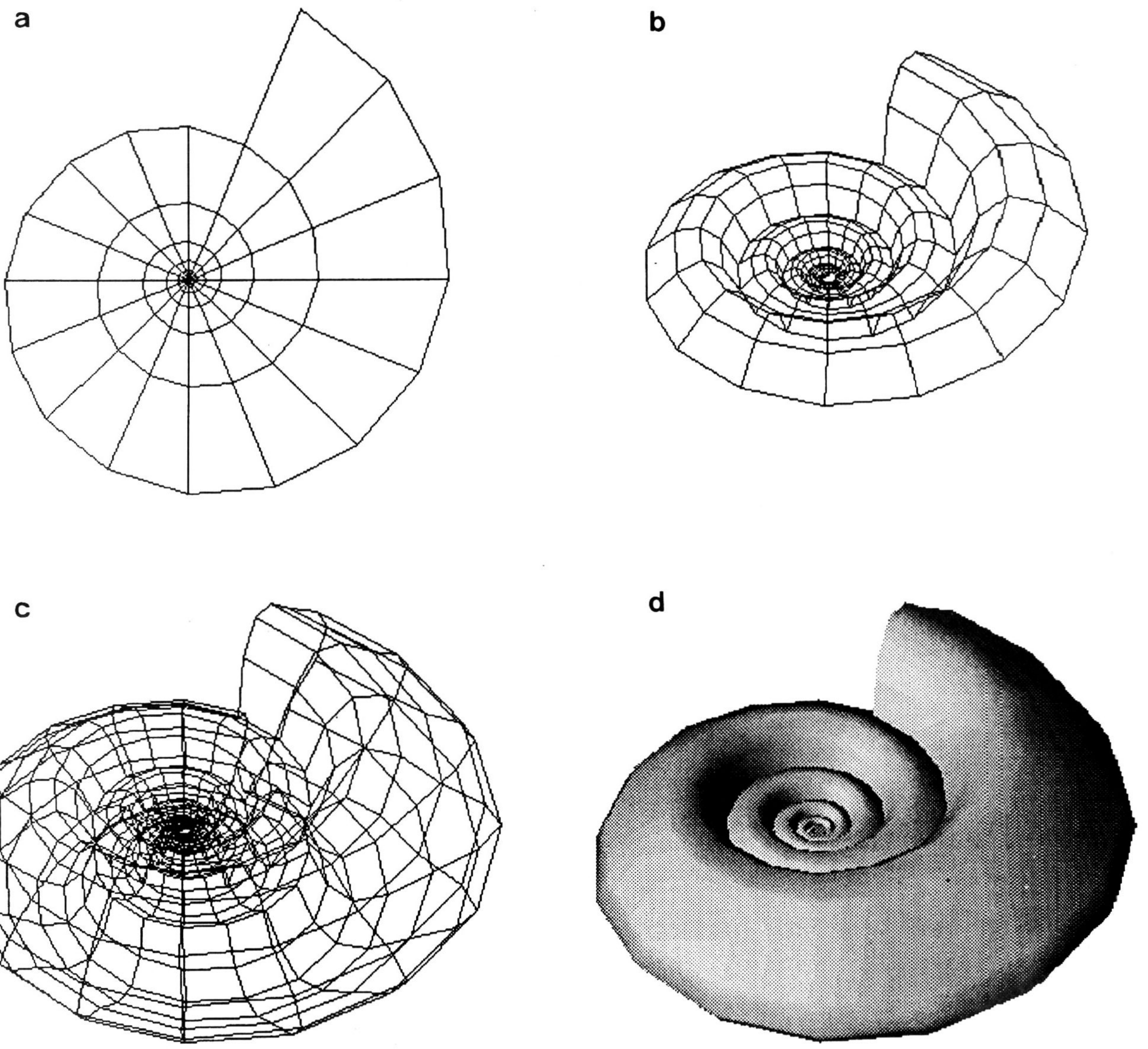

Fig. 10. Construction of a shell.

4. CONCLUSIONS

This paper presents a new method for defining three-dimensional shapes. It relies on constraint-based modeling of polygon meshes. The definition of an object in terms of constraints can be more straightforward and simpler than other types of definitions. In order to find vertices of a mesh defined using constraints, a system of nonlinear equations must be solved. In the general case, only numerical methods can be used for this purpose. Solving the system of equations can be made easier, if the mesh can be decomposed into rigid submeshes. Such decomposition is possible in many practical situations. Application of the constraint-based approach to the modeling of natural objects, for example flowers and shells, is an attractive topic open for further research.

ACKNOWLEDGMENT

This research was partially supported by grant number A0324 from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Borning, A. [1981]: The programming aspects of Thinglab, a constraint-oriented simulation laboratory. *ACM Trams. on Programming Languages* **3**, No. 4, pp. 353-387.

Conte, S. D. [1965]: *Elementary numerical analysis: An algorithmic approach.* McGraw-Hill, New York.

Foley, J. D., and van Dam, A. [1983]: *Fundamentals of interactive computer graphics.* Addison-Wesley, Reading.

Hain, K. [1967]: *Applied kinematics.* McGraw-Hill, New York.

Knuth, D. E. [1979]: *TEX and METAFONT.* Digital Press and American Mathematical Society, Bedford.

Lin, V. C., Gossard, D. C., and Light, R. A. [1981]: Variational geometry in computer-aided design. *Computer Graphics* **13**, No. 3, pp. 171-177.

Nelson, G. [1985]: Juno, a constraint-based graphics system. *Computer Graphics* **19**, No. 3, pp. 235-243.

Stevens, P. S. [1974]: *Patterns in nature.* Little, Brown and Co., Boston.

Sutherland, I. E. [1963]: *Sketchpad, a man-machine graphical communication system.* In *1963 Spring Joint Computer Conference*, reprinted in Freeman H. (Ed.): *Interactive Computer Graphics*, IEEE Computer Soc. 1980, pp. 1-19.

Van Wyk, C. J. [1982]: A high-level language for specifying pictures. *ACM Transactions on Graphics* **1**, Nr. 2, pp. 163-182.