

AN INTERACTIVE AND USER-ORIENTED THREE-DIMENSIONAL GRAPHICS EDITOR

Nadia Magnenat-Thalmann  
H.E.C., Dépt. des méthodes quantitatives  
Montréal, Canada

André Larouche  
Air Canada  
Montréal, Canada

Daniel Thalmann  
Dépt. D'I.R.O., Université de Montréal  
Montréal, Canada

ABSTRACT

HORIZON is a general-purpose three-dimensional graphics editor. It permits to create three-dimensional diagrams, to modify them by geometrical transformations and to assemble them to obtain a complete drawing. The drawing can be stored on a file; later, one may retrieve it for further modifications, or request a hard copy.

HORIZON is strongly user-oriented. It is easy to use even by persons who are not familiar with computer science. The user communicates with the editor by a set of simple commands, in a simple and efficient way. The editor provides various guiding messages and feedbacks to the user. Thus, a session of HORIZON is a friendly dialogue between user and editor.

HORIZON allows the user to define interactively any kind of projection. The visual parameters definition commands are compatible with those defined in the GSPC Core System. Diagrams may be created in three ways: by direct graphical input with the cursor, by placing a diagram from a catalog area into the diagram area or by combining existing diagrams. Combinations can be made by linking corresponding vertices of two diagrams or by forcing two diagrams to have a common edge. 3D graphical input is only available in a standard projection, which is a cavalier projection.

HORIZON is implemented in the MIRA-3D language, a graphical PASCAL extension based on abstract graphical types. HORIZON is mainly used in 3D computer animation.

KEYWORDS: Graphics editing, 3D graphical input, user interface.

RESUME

HORIZON est un éditeur graphique tridimensionnel d'utilité générale. Il permet de créer des diagrammes à trois dimensions, de les modifier par des transformations géométriques et de les assembler pour obtenir un dessin complet. Le dessin peut être ensuite conservé sur un fichier; on peut le retrouver plus tard pour le modifier ou en obtenir une copie sur papier.

HORIZON est spécialement orienté vers l'utilisateur. Il est facile de l'utiliser même par des personnes qui ne sont pas familiarisées avec l'informatique. L'utilisateur communique avec l'éditeur par un ensemble de commandes simples et de manière efficace. L'éditeur fournit des messages et des conseils à l'utilisateur. Ainsi une session de HORIZON est un dialogue agréable entre l'utilisateur et l'éditeur.

HORIZON permet à l'utilisateur de définir interactivement n'importe quel type de projection. Les commandes de définition de paramètres visuels sont compatibles avec le standard graphique de l'ACM. Les diagrammes peuvent être créés de trois façons: par une entrée graphique avec le curseur, en prenant un diagramme dans un menu ou en combinant des diagrammes déjà existant. Les combinaisons peuvent être effectuées en liant les sommets correspondant de deux diagrammes ou en forçant deux diagrammes à avoir une arête commune. L'entrée graphique tridimensionnelle n'est disponible qu'avec une projection standard qui est une projection cavalière.

HORIZON est implanté en MIRA-3D, une extension graphique de PASCAL, basée sur les types graphiques abstraits. HORIZON est principalement utilisé dans des projets d'animation tridimensionnelle.

MOTS-CLES: édition graphique, entrée graphique 3D, interface usager.

## INTRODUCTION

There are many applications requiring three-dimensional drawings. Architecture, advertising, animation, industrial design, all involve graphical representations of three-dimensional objects. Specific systems are available [1,2,3,4]. Unfortunately, general-purpose three-dimensional editors have not yet appeared as a standard feature of interactive computer systems.

Diagrams are often the most attractive part of publication, report or presentation. However, three-dimensional diagrams are very difficult, if not impossible, to produce with manual methods. It is time-consuming and costly to render them technically perfect. Moreover, once a drawing has been made, it is unthinkable to change the point of view or the direction of the projection.

But a drawing, like a text, is often subjected to various modifications. The repetitive process of hand-producing similar drafts, or modifying successive unsatisfactory sketches, can be tedious and costly. However, these operations - the creation and the successive modifications of a drawing - can be performed easily with a graphic editor.

A general-purpose graphical editor must possess two qualities. First, it should not require expensive hardware. Second, it should not demand an extensive knowledge of computer science. These are two concepts that guided us in the development of HORIZON, the three-dimensional editor described in this paper.

## SYSTEM OVERVIEW

HORIZON is a command-directed editor. There are forty commands, and almost all of them have one or more parameters. The concept of graphical menu is not used in HORIZON at the command level in order to economise the display area of the CRT. Instead, the user has to type the commands on the keyboard; the commands are echoed by the editor. The user is assisted by the responses, and error messages sent by the editor during the session.

The user can select interactively the visual parameters to define the projection used for the drawing. Because the choice of visual parameters demands some knowledge, HORIZON offers a predefined projection - the standard

projection - to accommodate the user who is not familiar with three-dimensional graphics. A standard mode corresponds to the standard projection; conversely, the general mode allows the user to define an original projection. Figure 1 shows a classical example of perspective effect.

The standard projection is parallel; the projectors form an angle of 45 degrees with the view plane - the XY plane. This projection has been chosen because it is the most convenient for graphical inputs. It can be done with very few computations - this is important when we consider the frequency of such an operation. This kind of input is similar to the polyedron input of PICAX [5].

Graphical inputs are only allowed in the standard mode. In the general mode, the user must utilize the variables to specify a point. There are ten variables, identified by a number from zero to nine. The variables are defined by a special command. Every time a pointing is requested on the screen, the user can avoid the direct pointing with the cursor by specifying the number of a variable that contains the coordinate of the desired point.

HORIZON allows the use of a graphical menu, the catalog, which contains basic figures which are frequently used as part of more complex drawings. An example of a catalog of regular polyhedra is presented in Figure 2. The characteristics of a catalog are described in the next section.

## THE USER INTERFACE

The user interface means the interface between humans and machines. This interface must offer a friendly and, possibly, a natural dialogue with the user.

At the beginning of a session, HORIZON divides the screen into four areas:

- i) The diagram area - or the drawing area - which occupies 9/10 of the screen, where the user creates diagrams.
- ii) The catalog area, which occupies 1/10 of the screen - at the left - contains a choice of standard or user-defined basic figures.

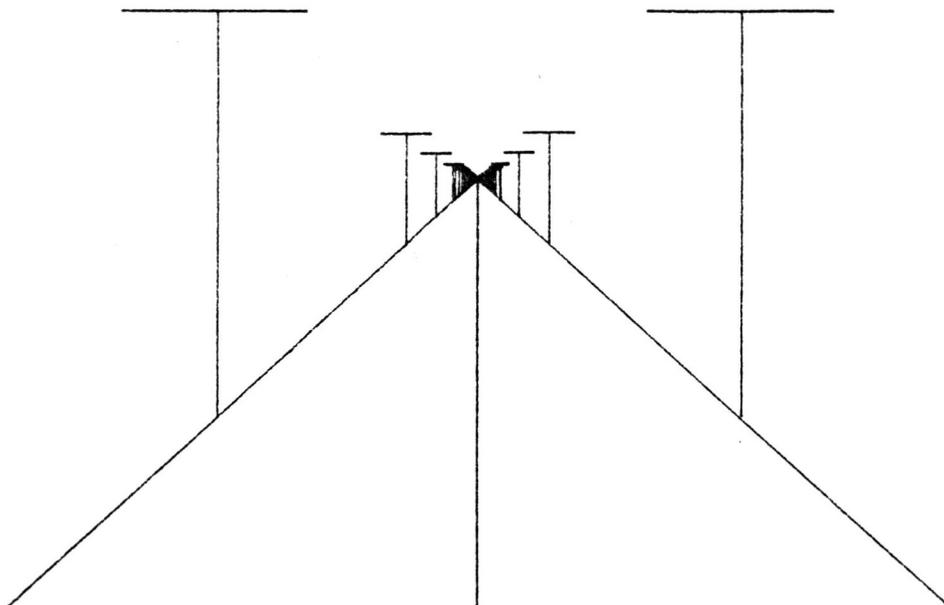


Figure 1 A classical example of perspective effect

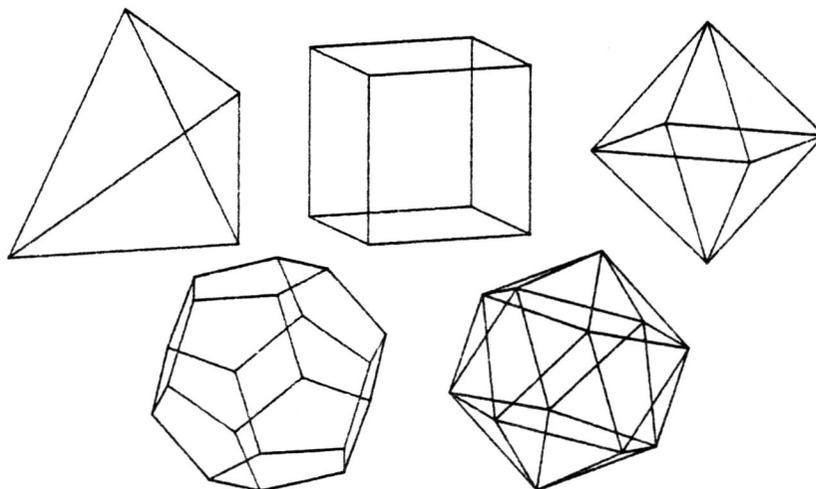


Figure 2 A catalog of regular polyhedra

- iii) The message area, which is alphanumeric and occupies a line at the bottom of the screen, where the editor sends responses and error messages.
- iv) The command area, which occupies a line above the message area, where the user types the commands.

The user is prompted for commands by the message "COMMAND :", at the left of the command area. Then the user may write several commands separated by a semicolon ";". When the user types RETURN, the editor analyses the line of commands; for each command, there is a feedback. If a command is syntactically incorrect, the editor sends an error message to indicate the source of the problem. If a command needs graphical inputs, such as the pointing of a diagram or of a point, a system message requests it. In any case, the user receives a response from HORIZON; one is never left in uncertainty. This is fundamental, because the frustration of having an interactive program bump or hang before completing its task can be overwhelming.

Each graphical area - the diagram and the catalog area - has a current diagram. Almost all commands refer implicitly to one or both of the current diagrams.

There are seven classes of commands in HORIZON:

- 1) Relation to environment commands, which allow the saving and retrieval of diagrams and catalogs, or communication with the operating system. When a drawing is retrieved, the editor draws it with the same projection which was used when the sketch was saved.
- 2) Variable definition commands, which allow the user to define or to modify the value of the variables and of the current diagrams.
- 3) Visual parameters definition commands, which allow the user to define an original projection. When one defines a new projection, the new visual parameters are accumulated until the user types a special command (REDRAW); then the editor tests the consistency of the new visual parameters and, if they are correct, redraws all diagrams with the new projection. The visual parameters definition commands are compatible with those defined in the GSPC Core System [6,7].
- 4) Creation/deletion commands, which allow the user to create or delete diagrams. The four ways of creating a diagram are discussed in the next section.

- 5) Transformation commands, which allow the user to modify the current diagram or all diagrams by various geometrical transformations: rotation, translation, symmetry, scaling, union, and so on.
- 6) Catalog commands, which allow the user to manipulate the diagrams in the catalog.
- 7) Special commands, which allow the user to select various global options - linestyle, color, axis, grid ... - or to examine the values of variables.

### DIAGRAM CREATION

There are three ways of creating a diagram:

- 1) by direct graphical input with the cursor  
In this case, the user draws an original diagram. Graphics reading is performed by a cursor which can be moved along the three axes. Depth is simulated by using a cavalier projection.
- 2) by placing a diagram from the catalog into the diagram area  
This operation, which is very simple, is generally followed by transformation operations on the diagram.
- 3) by combining existing diagrams  
Combinations can be made in three ways:
  - . by concatenating two or more diagrams into a single one.
  - . by linking corresponding vertices of two diagrams by segments. Both diagrams must have the same number of vertices. Figure 3 shows an example of a building, which has been made by direct graphical input and linking. Figures 4 and 5 are other views with different visual parameters.
  - . by forcing two diagrams to have a common edge. The user points both diagrams and, for each one, the edge that must be joined to the other diagram. Once the figures are joined, the editor performs a rotation of the second diagram around the common edge. The rotation angle is given by the user. Figure 6 shows the construction of a cube by successive steps. At each step, a square is attached with an angle of 90° to an edge of the figure to be built. Within three steps a cube is obtained.

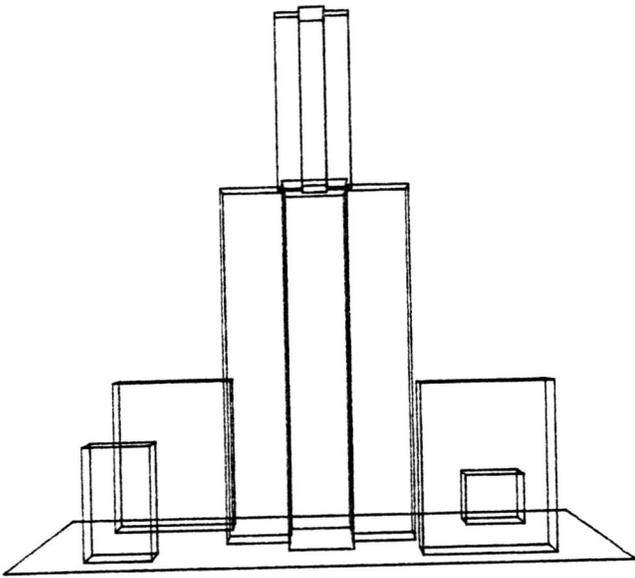


Figure 3 An example of building created by direct graphical input and linking

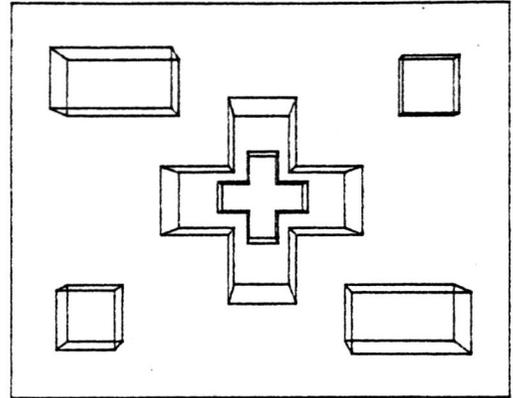


Figure 4 Another view of the building

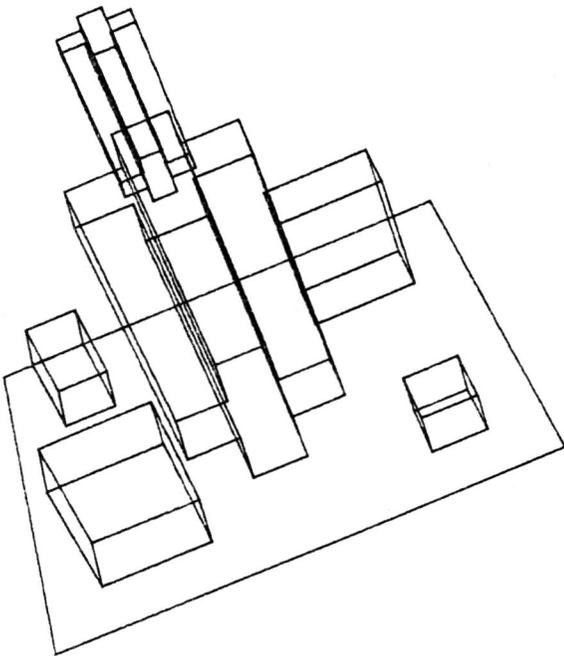


Figure 5 Another view of the building

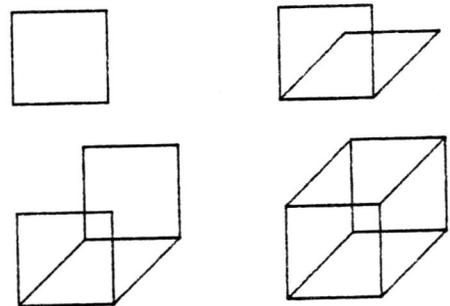


Figure 6 Construction of a cube by successive steps

4) by repetitive commands

Most commands may be preceded by a repetitive factor \* $\langle n \rangle$ ; the line of commands may also be prefixed by a repetitive factor X $\langle n \rangle$ . Those features allow the user to write little "programs" which are very useful for animation purpose. But the main purpose is the easy creation of drawings based on a very regular pattern. For example, in the sunrise of Figure 7, the sun has been made by repetitive rotations and the grid by repetitive translations. Figure 8 shows a drawing, which has been created by a symmetry. In Figure 9, a rectangle is placed far from the point of view and is approached by repetitive translations. Note that the point of view is not parallel to the tunnel which is created by this effect.

There are figures that cannot easily be created with HORIZON because the editor's means of direct graphical input are often limited to the cursor of the terminal. However, special diagrams can be constructed in a MIRA-3D program and included in a catalog file. The standard MIRA-3D library contains procedures to support this interface. The catalog file could later be handled with HORIZON exactly as if it has been created internally. An example of such a catalog is presented in Figure 10; it includes conical and cylindrical surfaces. Conversely, it is possible to create diagrams with HORIZON and to use them in a MIRA-3D program by reading a catalog or diagram file.

IMPLEMENTATION

HORIZON is an interactive program, presently implemented on a CDC Cyber 173 and a DEC VAX 11. The editor can be used with TEKTRONIX 4027, D.E.C. GIGI and HP 2648 A terminals. HORIZON is written in the MIRA-3D [8] language, a graphical PASCAL [9] extension based on abstract graphical data types [10]. Because of the structured approach to computer graphics [11], that is the base of MIRA-3D, HORIZON is a well-structured program which can easily be extended.

The whole system, including all auxiliary programs, consists of 6000 lines of MIRA-3D source code. The editor itself consists of 3000 lines of source code.

HORIZON is composed of five independent modules:

- 1) The monitor, which controls the execution of commands and handles manipulation errors.

- 2) The parser, which analyses the line of commands and detects syntactic errors. If a command is correct, the parser transmits an internal code to the monitor which executes it.
- 3) The communication module, which maintains the dialogue with the user. All interactive input/output functions are concentrated in this module.
- 4) The list manipulation module, which handles the main data structure. For each new diagram, HORIZON dynamically creates a descriptor record; these records are arranged into linear lists: one for each graphical area. This module permits to add or delete a record in a list, or to draw or erase all diagrams in a list, and so on.
- 5) The interface module, which permits communication with the operating system by executing control cards. Transfers between the screen and graphical files are also handled by this module.

The monitor and the parser communicate by means of global variables. The three other modules communicate by means of parameters. Each module can be extended or modified independently; for instance, an adaptation of HORIZON to another operation system would only imply the modification of the interface module.

CONCLUSION

The general-purpose graphical editor described in this paper possesses most of the basic features needed for drawing almost any kind of three-dimensional diagrams. However, a certain number of desirable commands and options are not implemented in HORIZON, because the size of its code must be kept within reasonable bounds.

The basic idea behind HORIZON is the concept of general-purpose graphical edition. If a change has to be made on a given figure, only the part to be changed has to be modified. It is not necessary to redraw it all over.

In the future, we would like to add commands to support high-quality text in three dimensions. Another improvement could be done by allowing graphical input with an arbitrary projection. Some desirable features, such as hidden surfaces or haloed line effect, will be introduced; such niceties, however, will decrease the efficiency of the editor (selective erasing, for instance, is no longer possible

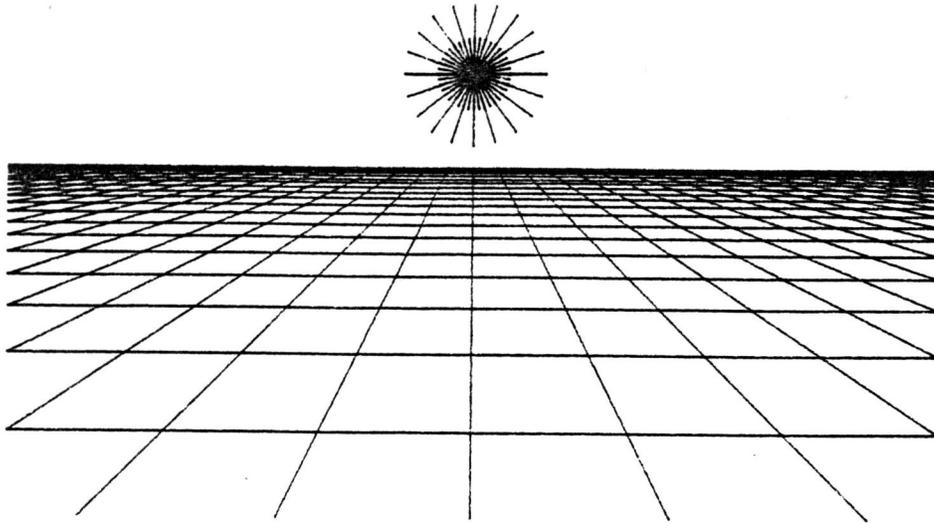


Figure 7 An example of the use of repetitive commands

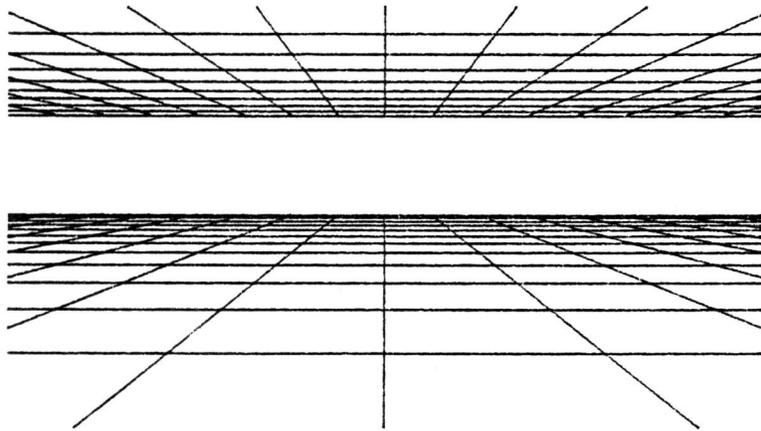


Figure 8 Application of a symmetry operation

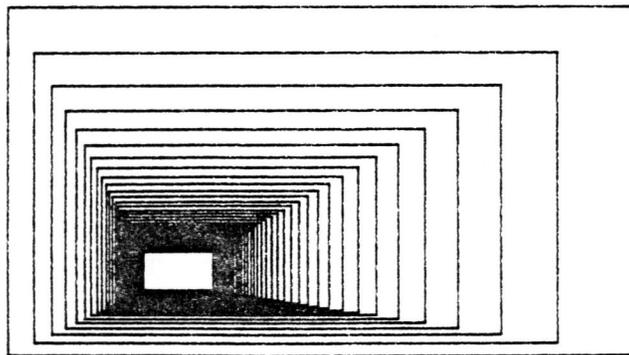


Figure 9 A tunnel effect

with hidden surfaces).

The integration of HORIZON into teleprocessing, text processing and other software provides a powerful tool for communicating ideas in graphical form.

HORIZON is also part of a complete 3D animation system including the CINEMIRA [12] computer animation language based on actor and camera data types and a Multiple Track Animator System [13].

ACKNOWLEDGEMENT

This work has been sponsored by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[ 1 ] Herot, C.F. et al., "A Prototype Spatial Data Management System", Proc. Siggraph'80, Computer Graphics, vol. 14, nr 3, 1980, pp. 63-70.

[ 2 ] Shectman, R.M. and Bering, D.E., "The Electronics Engineer's Design Station", IEEE Computer Graphics and Applications, vol. 1, nr 2, 1981, pp. 15-16.

[ 3 ] Weber, J. et al., "Applications of an Interactive Vector Graphics System to Chemistry", Proc. Eurographics'80, North-Holland, 1980, pp. 139-147.

[ 4 ] Rogers, G., "Dynamic 3D Modelling for Architectural Design", Computer Aided Design, vol. 12, no 1, 1980, pp. 13-20.

[ 5 ] Liardet, M., "PICAX - Polyhedron Input to the Computer Using an Axonometric Drawing", Proc. Intern. Conf. on Interactive Techniques in Computer Aided Design, IEEE Press, 1978.

[ 6 ] Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH, Computer Graphics, vol. 13, no. 3, 1979.

[ 7 ] Bergeron, R.D.; Bono, P.R. and Foley, J.D., "Graphics Programming using the Core System", Computing Surveys, vol. 10, no 4, 1978.

[ 8 ] Magnenat-Thalmann, N. and Thalmann, D., "MIRA-3D: A Three-dimensional Graphical Extension of PASCAL", Software-Practice and Experience, vol. 13, 1983.

[ 9 ] Jensen, K. and Wirth, N., PASCAL User Manual and Report, Springer-Verlag, 1974.

[10] Thalmann, D. and Magnenat-Thalmann, N., "Design and Implementation of Abstract Graphical Data Types", Proc. COMPSAC'79, Chicago, IEEE Press, 1979, pp. 519-524.

[11] Magnenat-Thalmann, N. and Thalmann, D., "A Graphical PASCAL Extension Based on Graphical Types", Software - Practice and Experience, vol. 11, no 1, 1981, pp.53-62.

[12] Magnenat-Thalmann, N. and Thalmann, D., "CINEMIRA: A Computer Animation Language based on Actor and Camera Data Types", Publ., Dép. Inform. et Rech. Opér., Univ. de Montréal, 1983.

[13] Fortin, D.; Lamy, J.F. and Thalmann, D., "A Multiple Track Animator System", Proc. SIGGRAPH/SIGACT Interdisciplinary Workshop on Motion: Representation and Perception, Toronto, 1983.

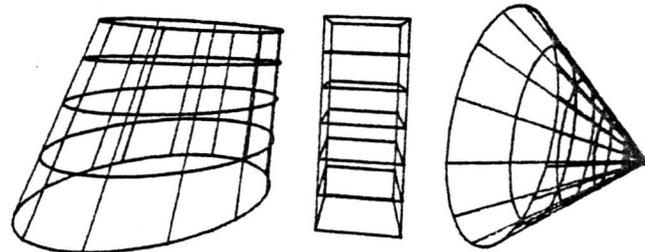


Figure 10 Conical and cylindrical surfaces