EG

EUROGRAPHICSEMINARS
Tutorials and Perspectives in Computer Graphics

M. M. de Ruiter (Ed.)

Advances in Computer Graphics III

Springer-Verlag

M. M. de Ruiter (Ed.)

Advances in Computer Graphics III

With 247 Figures

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo

EurographicSeminars

Edited by W. T. Hewitt, R. Gnatz, and D. A. Duce
for EUROGRAPHICS –
The European Association for Computer Graphics
P.O. Box 16, CH-1288 Aire-la-Ville, Switzerland

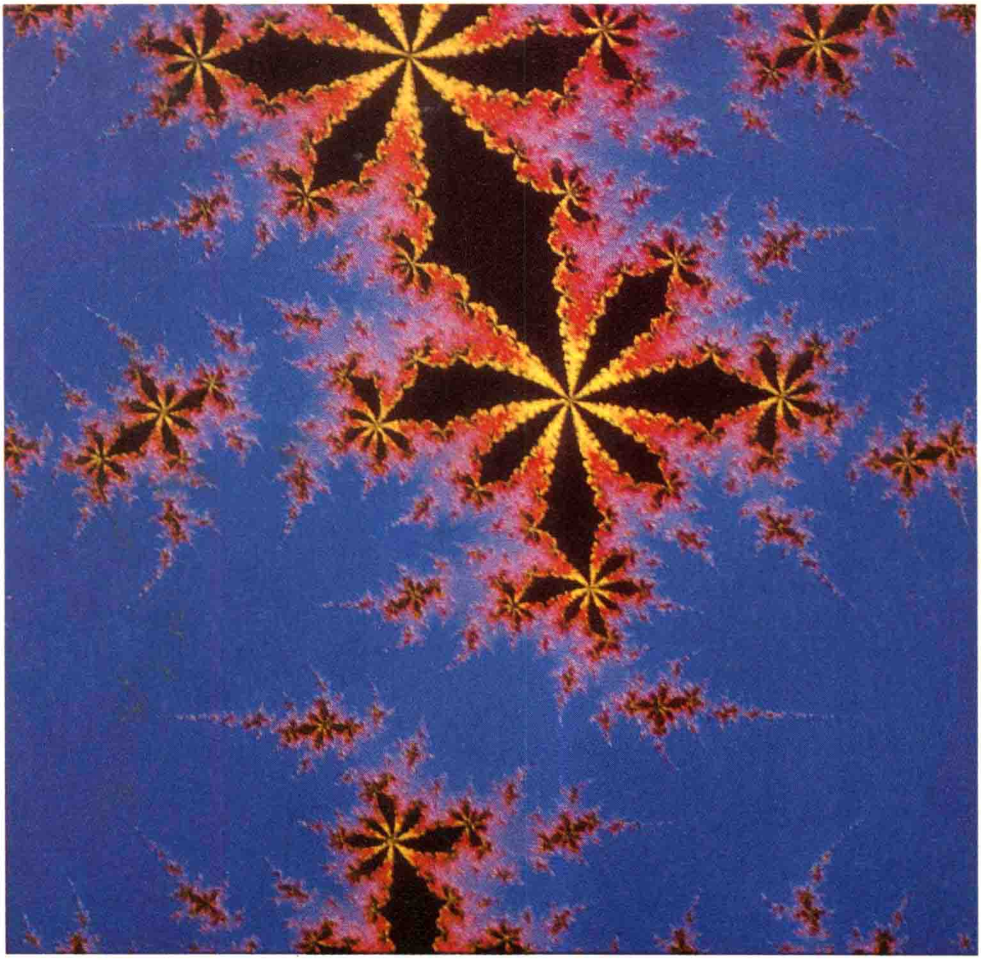
Volume Editor

Maurice M. de Ruiter
Stichting Mathematisch Centrum
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands

ISBN 3-540-18788-X Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-18788-X Springer-Verlag New York Berlin Heidelberg

Library of Congress Cataloging-in-Publication Data.
Advances in computer graphics III / edited by M. M. de Ruiter. p. cm.–
(EurographicSeminars)
ISBN 0-387-18788-X (U.S.)
1. Computer graphics. I. Ruiter, M.M. de, 1948-. II. Title: Advances in computer graphics 3. III. Title: Advances in computer graphics three. IV. Series: Eurographic seminars. T385.A364 1988 006.6–dc 19 88-4441 CIP

Printing: Druckhaus Beltz, Hemsbach/Bergstr.; binding: J. Schäffer GmbH & Co. KG., Grünstadt
2145/3140-543210



Basins of attraction of a complex mapping (see Part 5, page 198)

EurographicSeminars

Tutorials and Perspectives in Computer Graphics

Edited by W. T. Hewitt, R. Gnatz, and D. A. Duce

EurographicSeminars

Tutorials and Perspectives in Computer Graphics

Eurographics Tutorials '83. Edited by P. J. W. ten Hagen.
XI, 425 pages, 164 figs., 1984

User Interface Management Systems. Edited by G. E. Pfaff.
XII, 224 pages, 65 figs., 1985

Methodology of Window Management. Edited by F. R. A. Hopgood, D. A. Duce, E. V. C. Fielding, K. Robinson, A. S. Williams.
XV, 250 pages, 41 figs., 1985

Data Structures for Raster Graphics. Edited by L. R. A. Kessener, F. J. Peters, M. L. P. van Lierop. VII, 201 pages, 80 figs., 1986

Advances in Computer Graphics I. Edited by G. Enderle, M. Grave, F. Lillehagen. XII, 512 pages, 168 figs., 1986

Advances in Computer Graphics II. Edited by F. R. A. Hopgood, R. J. Hubbold, D. A. Duce. X, 186 pages, 96 figs., 1986

Advances in Computer Graphics Hardware I. Edited by W. Straßer.
X, 147 pages, 76 figs., 1987

GKS Theory and Practice. Edited by P. R. Bono, I. Herman.
X, 316 pages, 92 figs., 1987

Intelligent CAD Systems I. Edited by P. J. W. ten Hagen, T. Tomiyama. XIV, 360 pages, 119 figs., 1987

Advances in Computer Graphics III. Edited by M. M. de Ruiter.
IX, 323 pages, 247 figs., 1988

Preface

The material in this book was presented in the tutorial programme of the Eurographics '87 Conference, held in Amsterdam, The Netherlands, in August 1987. The contributors are leading experts in their respective fields. Nine major aspects of computer graphics are discussed.

Part 1, Solid Modelling and VLSI Implementations, describes proposals for extending the hardware support provided in display systems and workstations, to include facilities for handling solid models directly. These proposals are based on a series of important structuring ideas which have influenced the development of existing computer graphics systems, and which can therefore be taken as good initial guides to future developments, as more powerful technology becomes available. The proposals include a series of detailed design studies to illustrate what is currently possible and also to outline next stage developments and likely applications.

Part 2 presents an overview of the current advances in user interface management system technology and how these advances are encapsulated in a number of existing user interface management systems. After introducing fundamental concepts and the basic technology, a number of advanced topics are discussed and an evaluation of different commercially available user interface management systems is given.

The object-oriented programming paradigm offers new facilities for computer graphics programming. Environments such as the Smalltalk-80 system or Lisp-machine's flavour system provide powerful tools for prototying man-machine interfaces. **Part 3** explores how these tools can be used in classical computer approaches.

The extent to which computers can and will be used in art depends on the nature of computing, the nature of art, and human nature. To use computers in art means considering art in terms of the things computers can do. **Part 4** considers the role that a computer can play in art as a tool, a medium, a catalyst, and a 'smart apprentice'. Topics covered include paint systems (their possibilities and potential), modelling systems (what they are and what they can do), pictures by program (making pictures when you can't draw), pictures by rule (modelling creativity) and an art gallery (examples of current work).

Part 5, entitled *Fractals - Mathematics, Programming and Applications*, covers three topics. The first is the mathematics of fractals. Several classes of fractals, known as self-similar sets (for example Koch Islands), complex mappings (for example Mandelbrot and Julia sets) and strange attractors are discussed. The generation and visualization of fractal objects is the second subject. The interactive generation and representation of fractal objects is described. The third topic is the application of fractals to modelling objects from nature. In this part some examples are shown of how fractals may serve as mathematical models for biological objects.

Part 6 gives an overview of the theory and methods for modelling three-dimensional shapes and objects, with an emphasis on modelling solids. Both background material and advanced topics are covered. The material is organized into four sections: introduction, principal representations and algorithms, evaluation and visualization techniques, and model specification and shape operations.

Part 7 gives an overview of existing CAD-Interface specifications (IGES, VDAFS, SET, etc.) and the various aspects of data exchange between dissimilar CAD systems in practical applications in the automobile industry. It includes a detailed presentation of on-going national and international CAD-Interface specification work (STEP, CAD*I, etc.) in the area of wireframe, surface and solid-modelling geometry. In addition, there is an introduction to new techniques for processor development and test methodology.

The final part, **Part 8**, covers the major issues in the increasingly important area of Desktop Publishing. The introduction of high quality, low-cost, printers, particularly laser printers, has made it possible to produce professional-looking documents directly from a text processing system. The hardware and software used in Desktop Publishing is reviewed, and the state-of-the-art presented. The new programmes and hardware which tackle the problem of combining text with pictures are described and evaluated with respect to different types of graphics/text problems. Some of the new standards for document interchange are considered. The competing protocols DDL, Interpress and PostScript are looked at with respect to future applications. The Macintosh/PC worlds are also evaluated with respect to their Desktop Publishing software availability and applicability.

The book will be of interest to systems designers, applications programmers and researchers, who wish to gain a deeper knowledge of the state-of-the-art in the areas covered.

Amsterdam, January 1988 M.M. de Ruiter

List of Contributors

W.F. Bronsvoort
Faculty of Mathematics and Informatics, Delft University of Technology
Julianalaan 132, 2628 BL Delft, Netherlands

J. Ero
Philips ISA/CAD Centre
Postbus 80000
5600 MB Eindhoven, Netherlands

J.A. Kaandorp
Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, Netherlands

J. Lansdown
System Simulation Ltd
250M Bedford Chambers The Piazza
Covent Garden
London WC2E 8HA, England

H.A. Lauwerier
Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, Netherlands

R. van Liere
Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, Netherlands

K.M. Mittelstaedt
Kernforschungszentrum Karlsruhe
D-7500 Karlsruhe, FR Germany

F.H. Post
Faculty of Mathematics and Informatics, Delft University of Technology
Julianalaan 132, 2628 BL Delft, Netherlands

A.L. Thomas
Sussex University
Brigthon, Sussex BN1 9QT, England

D.E.E. Trippner
BMW AG
D-8000 München, FR Germany

H.A. Tucker
Institute of Datalogy, University of Copenhagen
Universiteitspark 1-3, DK-2100 København Ø, Denmark

P. Wisskirchen
GMD
Postfach 1240, Schloss Birlinghoven
D5205 St Augustin 1, FR Germany

Table of Contents

Part 1
VLSI for Solid Modelling

A. L. Thomas

In this tutorial, proposals are discussed for extending the hardware support provided in display systems and workstations to include facilities for handling solid models in a direct way. These proposals are based on a series of important structuring ideas, given below which have infuenced the development of existing computer graphic and object modelling systems, and which by extension seem to be good guides for planning future developments as more powerful technology becomes available.

The first idea is that the role of graphics in computer based systems has become primarily one of communication. In contrast, traditional forms of graphics, created by labour intensive, manual operations, had three almost equally important roles to support:

1. As a storage medium for information.
2. As a communication medium for the same information
3. As a model or data structure to support a wide variety of different forms of information analysis.

Computers have taken over many of the operations that derive new information from existing data, providing a greatly extended range of different forms of analysis that it is now practical and economic to perform. To carry out this work alternative forms of information modelling have had to be employed which have then also taken over the storage role from graphics. However what these new models have not been able to replace is the efficiency of the graphic forms when used in man to man communication, and more importantly now need to be used for effective man to machine communication.

The second idea is that the development of high speed display generating devices to support more efficient man-machine communication, has opened up a new dimension in the use of graphics. The appropriate display hardware will now permit displayed images to be generated fast enough to work with them interactively. It is difficult to define complex spatial relationships without some form of visual feedback to allow the user to monitor what he is doing. An interactive facility allows the generation of the image of a model to indirectly control the correct construction of the model itself. This allows internal data-structures to be maintained and manipulated automatically with less chances of errors.

The third idea is an extension of the second and consists of using high level constructs as building blocks for defining new models. If the purpose is to build models of solids then it is possible to combine together the appropriate collection of geometric elements such as point coordinates, edge lines and facet surfaces in a suitable structure. However it is easier to start with a set of simple solids and then to combine them together to give more complex solids. This approach allows the system to start with correctly structured object descriptions and by only permitting properly defined operations on these objects, it ensures that the system keeps its models correctly structured, however complex they finally become.

The fourth idea is that the ability to interact with high level objects at great speed makes it possible to have moving synthetic images controlled by very high level commands. Although the immediate idea which comes to mind is that of the pilot-training-simulator visual-display-system, the idea is more general than this, providing a very high level, wide band width communication channel for all kinds of man machine interaction. This is particularly relevant if such a facility can be provided at a reasonable cost.

The fifth idea is that the application of VLSI technology in an appropriate way can introduce a massive reduction in unit costs if the market is large enough. Circuit integration also produces units of a fraction of their previous sizes. The workstation or whatever the main man-machine interface-unit is ultimately called, must potentially have a large market, but to realise it fully, it needs to be cheap enough in any application area for each user to be able to afford one, and needs to be as small and portable as possible.

Introduction

The current method for creating graphic models of solid objects in a way which is simple, convenient and easy to interact with, was first developed by the French mathematician Gaspard Monge in the period 1768-1795, while he was working as an instructor at the military academy at Mezieres. Called descriptive geometry, its importance can be measured on one hand by the fact that it was treated as a military secret for quarter of a century, on the other by the fact that it still forms the basis of modern engineering drawing practice.

The ability to model solid objects is of great commercial, as well as military interest, today, which will explain in part the speed with which Computer Aided Design systems for solid objects have been developed since they emerged from university research at the end of the sixties and the beginning of the seventies. The potential applications of the many different modelling methods which have been opened up by computer technology have finally created a need after 200 years, to reassess traditional draughting methods. However even though changes are being made at a great rate, the consequences of this re-evaluation are on a scale which means that it is going to take some time for them to work themselves through all the design and production processes in engineering, to which they might be applicable.

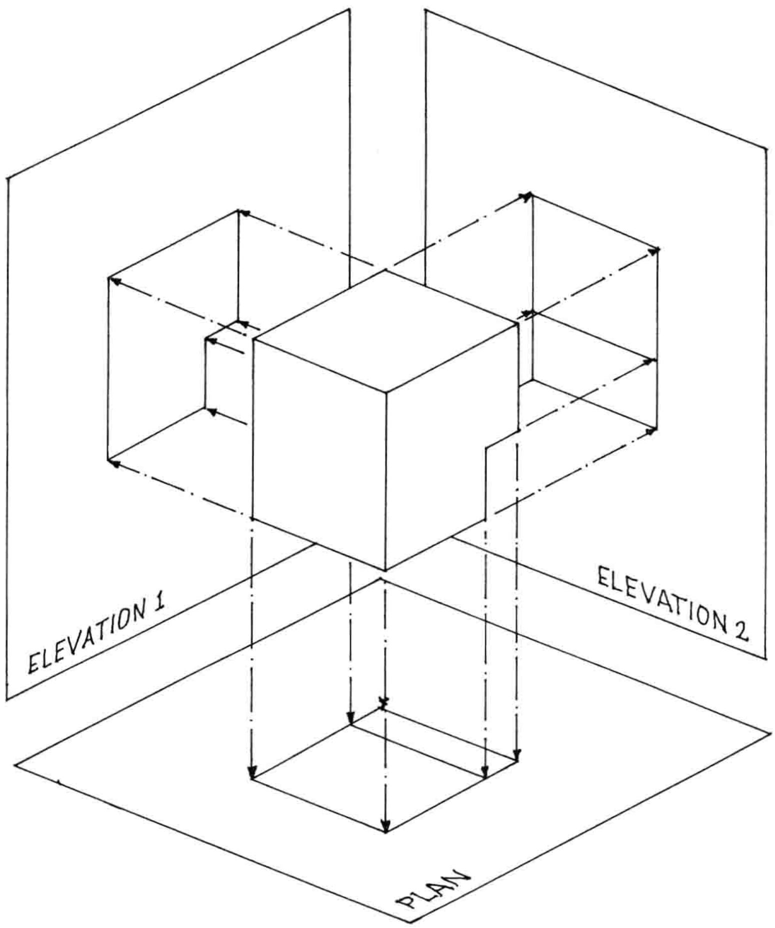


Figure 1. Engineering Drawings of Solid Objects

The Evolution of Interactive Display Hardware

The adoption of the cathode ray tube as a display device must be considered to be the starting point for interactive computer graphics. And must therefore be taken as an important step in the subsequent development of Computer Aided Design systems. All real-time display facilities appear to fan out from this point. In some ways this is not a surprising development since the CRT was used as a general purpose testing device for electronic circuits, and early on as a memory device in the form of Williams storage tubes. However the volume of computation required to support a CRT as a refresh display, where every displayed point had to be calculated by the host computer, was so high that it took the power of the Whirlwind computer developed in MIT to provide the first synthetic moving images, reported to be those of a bouncing ball.

The classical work, also MIT based, which provided the first effective interactive graphics system was the PhD research of Ivan Sutherland, in which he produced the SKETCHPAD system. This system introduced most of the important facilities needed by an interactive draughting system for use in computer aided design work. To reduce the computational load on the host computer, line interpolation hardware was developed. This permitted the display to be represented by a list of line segment, end-point coordinates rather than all the points needed in the display. Making the interpolation of the lines the task of the display system, to be carried out repeatedly for each refresh cycle of the CRT, made it possible to move a point in the display and have the lines linked to it move like rubber bands, following the point to whereever it was finally placed. This introduced a new form of interactive editing which it was clearly desirable to extend.

The next step was to allow whole objects to be moved around in a display as an editing operation. To do this it was necessary to identify an object's set of edges and then to transform all their end point coordinates to give the required consistent whole body motion. Simple two dimensional translations could be achieved using a form of base displacement addressing hardware which support absolute and relative coordinates. However because the system was based on point coordinates, it was possible to apply a range of geometric transformations to an object: scaling, rotation, translation, shear, and reflection transformations, by using homogeneous coordinates and applying the appropriate matrix operation to the list of object coordinates. This was a general approach which allowed three dimensional models to be handled within the same scheme, the coordinates being projected onto the CRT screen for display purposes. Again though these results were desirable, they placed a heavy computational load on the host computer:having to repeatedly recalculate the transformations, when continuous movement was required. This led to the development of specialised matrix processors to support real time display systems in simulators, but had to wait for technology to advance before it was economic to provide the same facilites in Computer Aided Design systems. Other line based primitive operations were developed that were suitable for hardware implementations such as line clipping and image windowing. However at this point difficulties emerged handling the displays of three dimensional objects which had to be solved before further steps in hardware development were worthwhile.

Although simple wireframe models were adequate for visualisation they had serious shortcomings when they were used to represent real solid objects. The problems encountered are well illustrated by the set of drawings given in Figure 2.

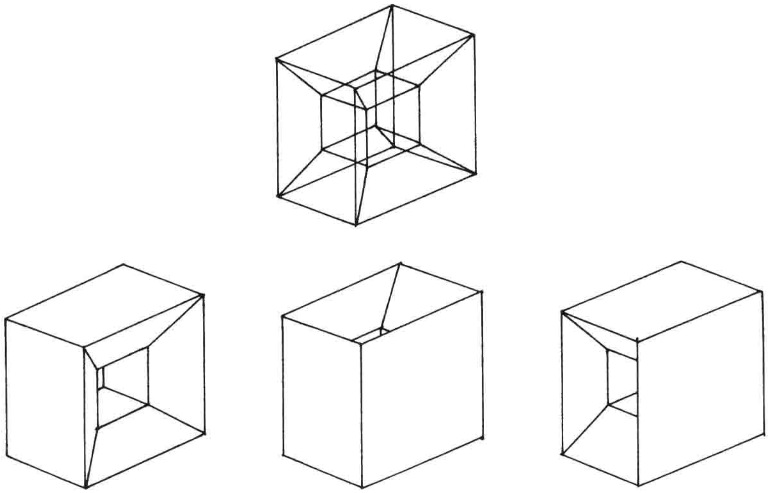


Figure 2. Wireframe Ambiguity and Hidden Line Removal.

Without further information the simple line model is ambiguous if it is meant to represent a solid object. Figure 2 also illustrates a further difficulty which is that without hidden line removal wireframe diagrams quickly become unreadable as two dimensional projections. However hidden line removal also requires a non ambiguous representation. Including hidden line removal in the display process effectively destroyed the systems response, necessary for interactive work, and therefore lost much of the advantage gained by providing fast point transformation and line interpolation hardware.

In order to provide sufficient information for solid modelling and display purposes, it was necessary to define the surfaces or boundary of objects. This could be done implicitly -- by structuring the list of coordinates in an appropriate way to imply the use of some standard, accepted interpolation scheme. Alternatively, it could be done explicitly by giving a functional definition for each piece of an object's surface. Both approaches incur difficulties of self consistency and accuracy of representation when an object is being manipulated or displayed. Objects with curved or scultured surfaces are the most difficult to manage, but even simplified polyhedral approximations to curved surfaced objects are not without their problems. It is interesting to note that one of the advantages claimed for this kind of methematical modelling was that it provided for the first time a complete and non ambiguous representation. Something which the draughtsman's model was unable to do except for plane faceted objects.

Better structures for modelling solid objects slowly evolved, but one important display problem was hard to solve. In interactive work it is desirable to be able to move objects around in a scene as part of the object building or editing process. While doing this it is relatively easy, unintentionally, to make objects overlap without being aware of the fact. The consequence of this "placement" problem was that simple hidden-line removal algorithms ceased to work. If two objects overlapped it was necessary to generate the edges where their surfaces intersected in order to get a correct display. Finding whether objects interfered with each other, added to the complexity of the display process even further. It was a search to find a solution to this problem which led to the adoption of the Boolean expression models discussed below, and which led to the development of alternative display algorithms based on raster based display devices such as the television monitor.

As soon as images could be generated fast enough to drive a television display then half-tone and full colour displays became a possibility. The task of generating realistic images fast enough to do this was started by the General Electric Corporation in work for NASA in the late sixties. The aim of this project was to produce electronic displays for pilot training simulators, which synthesised their images rather than televised them from scale models. Work also started in the University of Utah, about the same time, on a whole series of studies on raster graphics topics. For this work to be useful in an interactive environment it was necessary to find faster and more efficient ways of structuring the large amount of calculation needed to produce a raster image of even a relatively low resolution. Alternatively, it was necessary to design special display hardware that could reduce the display work carried out by the computer, in the way that the specialised interpolation facilities succeed in doing in the case of refreshed line displays.

The most important step taken in this direction so far, has been the introduction of the framestore. Memory as it became cheaper and packaged in large enough volumes, finally became fast enough and densely packed enough to store a complete TV image in a digital form. This information could then be accessed and converted into an analogue video signal at the rate required to drive a television. Work in the Computer Aided Design Centre in Cambridge produced the "Bugstore", a two port block of memory which allowed information to be written to it from the computer, while maintaining a constant flow of refresh data at the correct speed to a TV monitor. The hidden area removal algorithm of 'Newell, Newell and Sancha' was designed to use this kind of memory based hardware. Working with polygon facet models this approach employed the overwriting of memory to model the way in which nearer object facets obscured more distant object facets, by entering the property or colour values of more distant facets into the framestore first. This process required two supporting procedures. The first was depth ordering of polygons; the second was a form of polygon shading or polygon fill algorithm which was needed to determine which pixels or memory locations to overwrite.

Polygon facets can at times be impossible to sort into a simple order, and more importantly they can intersect each other as a result of the placement problem caused by interactive editing. A hardware solution to this problem was provided by the depth buffer. Any new polygon being painted into display memory was only entered at a particular pixel position if its depth value at that position was less than the depth value of the polygon already stored at the point. This approach was an ideal solution to the problem of preparing static displays using a refresh raster system, because no restriction was placed on the way that new depth or property values were generated. It permitted the results from a large variety of different object modelling systems to be merged for display, whatever the speed and the order in which the new data was prepared.

Both these approaches were beautifully simple but they had one drawback if speed was the ultimate objective. Every polygon had to be painted into memory, so for complex scenes every memory location could expect to be accessed several times per frame. Memory access time constitutes one of the limiting constraints on the resolution of raster displays. Since greater resolution can confidently be seen to be an unsatisfied demand until wall sized images are easy to produce it seemed unlikely that either of these two approaches represented the ultimate solution, particularly where real time editing and moving synthetic images were the goal. For such an objective greater speed in generating images had to be found. This appeared possible in two interrelated ways. Polygon data being entered into memory either had to be processed in parallel in some way, or polygon boundary lines had to be preprocessed to remove hidden zones, so that data for only one zone was entered at each pixel location during a frame time. Once the resolution of raster display systems was high enough then it was a natural step to include line interpolation facilities within the same system to enter line