EUROGRAPHICSEMINARS

Tutorials and Perspectives in Computer Graphics

Eurographics Tutorials '83

Edited by Paul J.W. ten Hagen

Springer-Verlag
Berlin Heidelberg New York Tokyo

Eurographics Tutorials '83

Edited by Paul J. W. ten Hagen

With 164 Figures

Springer-Verlag
Berlin Heidelberg New York Tokyo 1984

EurographicSeminars
Edited by G. Enderle and D. Duce
for EUROGRAPHICS –
The European Association for Computer Graphics
P. O. Box 16
CH-1288 Aire-la-Ville

Editor:
Drs. Paul J. W. ten Hagen
Stichting Mathematisch Centrum
Kruislaan 413
NL-1098 SJ Amsterdam

ISBN 3-540-13644-4 Springer-Verlag Berlin Heidelberg New York Tokyo
ISBN 0-387-13644-4 Springer-Verlag New York Heidelberg Berlin Tokyo

Library of Congress Cataloging in Publication Data.
Main entry under title: Eurographics Tutorials '83. (EurographicSeminars) Bibliography: p. 1. Computer graphics–Congresses. I. Hagen, P. J. W. ten. II. Series. T385.E975 1984 001.64'43 84-13905 ISBN 0-387-13644-4 (U.S.)

Printed in Germany

Printing: Beltz Offsetdruck, Hemsbach/Bergstr. Bookbinding: J. Schäffer OHG, Grünstadt
2145/3140-543210

Editors Introduction

This book is the first issue of a EUROGRAPHICS publication series in the field of computer graphics, an important field of research and a versatile tool for various application areas. The availability of powerful hardware at an affordable price and the evolution of high standard software have led to a rapidly increasing expansion of computer graphics and the penetration of computer graphics techniques and systems into a wide range of application areas.

This book series will cover state-of-the-art surveys as well as scientific contributions on specific areas of research and development.

The first book in the series contains the Tutorial Notes of the EUROGRAPHICS '83 conference, held in Zagreb, Yugoslavia, in September 1983. It covers four major aspects of computer graphics today:

- The first part contains a detailed **introduction into computer graphics,** its concepts, its methods, its tools, and its devices. It gives an easy access for the newcomer to the field and it offers an overview of the state of the art in computer graphics.
- The second part is devoted to **interactive techniques.** This is currently one of the most important fields of research in computer graphics. Important aspects of this research and its current state are reported. From the developments described here, in the near future powerful generally applicable user interface management systems are likely to evolve.
- The third part gives broad information on the most important software development in computer graphics in the past years: The first computer graphics standard, the **Graphical Kernel System,** GKS. Concepts, functions, and interfaces of GKS are described, a case study reports on implementation experiences.
- The fourth part covers important aspects of a major application area of computer graphics – namely the field of **three-dimensional models.** Contributions in this book describe both the fundamental concepts of surface design and of solid modelling.

The description of implemented solutions offers a bridge from the theoretical fundamentals to the reality of applications.

We are sure this book will serve as a thorough, detailed, yet easily comprehensible introduction to four important computer graphics areas. It will offer easy access to the fields of computer graphics fundamentals, interactive techniques, computer graphics standards, and three-dimensional modelling techniques.

Paul ten Hagen
EUROGRAPHICS '83
Tutorial Chairman

David Duce
Günter Enderle
EurographicSeminars
Series Editors

Table of Contents

I. Introduction to Computer Graphics (Part I)
R. J. Hubbold

1. Acknowledgement *1*
2. General Introduction *1*
3. Digitising *2*
   3.1 Manual Digitisers *2*
   3.2 Semi-Automatic Digitisers *3*
   3.3 Scanners and TV Cameras *3*
4. Graphical Displays *3*
   4.1 Direct View Storage Tube Display (DVST) *3*
   4.2 Refresh Displays *5*
   4.3 Raster Displays *8*
   4.4 Display System Architectures and More Advanced Features *14*
   4.5 Bit-Mapped Raster Displays *16*
5. Input Tools *18*
6. Hard Copy Devices *20*
   6.1 Flat-Bed Plotter *20*
   6.2 Drum Plotter *20*
   6.3 Electrostatic Plotter *21*
   6.4 Matrix Printer *21*
   6.5 Ink-Jet Plotter *21*
   6.6 Camera *22*
   6.7 Camera-Systems *22*
   6.8 Computer Output on Microfilm (COM) *23*
   6.9 Recommended Reading *23*

II. Introduction to Computer Graphics (Part II)
R. J. Hubbold

1. Projections and Viewing *25*
   1.1 Planar Geometric Transformations *25*
   1.2 Specifying Projections in a Graphics System *28*
   1.3 Clipping in 3-D *29*
   1.4 Representing Solid Objects *29*

1.5 Half-Toning Techniques 35
2. Colour 36
Recommended Reading and References 40

III. Introduction to Computer Graphics (Part III)
R. D. Bergeron

1. Graphics Concepts 43
1.1 Graphics Software Overview 43
1.2 Device Independent Graphics Standards 45
1.3 Graphical Output 46
1.4 Graphics-Based Input 51
1.5 Human Factors in Interactive Programming 53
1.6 Interaction Techniques 54
2. Modelling Concepts 65
2.1 Object Definition 65
2.2 Two-Dimensional Transformations 66
2.3 Three-Dimensional Transformations 69
References 72

IV. Interactive Techniques
P. J. W. ten Hagen

1. Introduction 73
2. User Interface 73
3. Interaction and Graphics 74
4. Dialogues 75
4.1 The Separation of Algorithm and Dialogue 76
4.2 Dialogue Cells 76
4.3 The Basic Cycle of a Dialogue Cell 78
4.4 Input Parsing 79
4.5 The Input Pool 80
5. The Interface to the Graphics System 81
5.1 A Symmetric I/O Layer on Top of GKS 82
Appendix A: Complete Syntax for Dialogue Cells 83
Appendix B: Examples for Dialogue Cells 93

V. Specification Tools and Implementation Techniques
H. G. Borufka, H. Hanusa, H. R. Weber

1. Introduction 97
1.1 Overview 97
1.2 Notions 98

2. Models and Design Strategy 100
2.1 Models 101
2.2 Design Strategy 105
3. Guidelines and Observations 123
3.1 Theoretical Aspects 123
3.2 Practical Aspects 144
3.3 Problems Arising 172
4. Tools and Techniques 177
4.1 System Model 177
4.2 Describing a Dialogue System 186
4.3 Dialogue Cells 197
5. Conclusion 227
6. Literature 228

VI. The Graphical Kernel System
J. Schönhut

1. Introduction 233
2. GKS Overview 233
3. Concepts and Programming of GKS 234
3.1 Graphical Output 234
3.2 Workstation Concepts 239
3.3 Coordinate Systems and Transformations 240
3.4 Segment Concept 242
3.5 Graphical Input 244
3.6 GKS Metafile Interface 246
3.7 GKS Level Structure 247
3.8 State of GKS 248
3.9 Error Handling 248
3.10 Special Interfaces 249
4. GKS Interfaces 249
4.1 Language Interfaces 249
5. Acknowledgement 250
6. References 250
Annex 1: GKS FORTRAN SUBROUTINE names 251
Annex 2: Correspondence of GKS data types and FORTRAN data types 255
Annex 3: GKS FORTRAN enumeration type mnemonics . 257

VII. Case Study of GKS Development
C. D. Osland

1. Introduction 264
1.1 Our Requirements 264
1.2 Objectives of GKS 267

1.3 Design Phases 268
1.4 Overall Structure of our GKS System 269
2. Output and Attributes 272
2.1 The Workstation Interface 272
2.2 Handling of Attributes 274
2.3 Handling of Transformations 274
2.4 Organization of 'Pipelines' 275
2.5 Design of Output Utilities 276
2.6 Problems with Error Handling 278
2.7 General Utilities 278
3. Input System 280
3.1 Problems with Level 'c' Input 280
3.2 Organization of Input System 280
3.3 Input Data Records 282
3.4 Two Problems with Input 282
4. Segments 283
4.1 Different Possible Design Structures 283
4.2 Communication with Central Segment Store 284
4.3 Design of Workstation-Independent Segment Storage 284
5. Metafiles 285
5.1 Developments in Metafile Standards 285
5.2 Requirements of GKS Metafile Interface 285
5.3 Some Problems with Metafiles 286
5.4 Future Developments in Metafile Support 286
6. Conclusions 287
6.1 Survey of Problems 287
6.2 Survey of Targets Achieved 288
6.3 Future Developments 288
Reference 289

VIII. Surface Design Foundations

W. Böhm, G. Farin

1. Local Coordinates 292
2. Polynomial Curves 294
3. B-Spline Curves 303
4. Rational Curves 311
5. Tensor Product Polynomial Surfaces 314
6. B-Spline Surfaces 321
7. Triangular Polynomial Patches 324
8. Scattered Data Interpolation 331
9. Transfinite Methods 334
References 340

IX. Geometric Modelling – Fundamentals
M. A. Sabin

1. The Objectives of Geometric Modelling *343*
2. Early Models *344*
3. Geometrically Complete Models *354*
4. Conversions between Models *363*
Appendix 1: 2D Construction *369*
Appendix 2: 3D Constructions *373*
Appendix 3: Profile Constructions *377*
Appendix 4: Sculptured Surface Sectioning Algorithms . . . *387*
Bibliography *389*

X. Solid Modeling: Theory and Applications
M. Mäntylä

1. Introduction *391*
2. The Role of Solid Modeling in CAD *393*
   2.1 Design Cycle *393*
   2.2 Models for CAD *394*
   2.3 System Architecture Aspects *395*
3. Solid Modeling Techniques *396*
   3.1 Taxonomy of Modeling Schemes *396*
   3.2 Wire Frames *398*
   3.3 Boundary Representations *399*
   3.4 Constructive Solid Geometry *405*
   3.5 Other Representations *408*
4. Solid Modeling Applications *409*
   4.1 Visualization *409*
   4.2 Integral Properties *412*
   4.3 Kinematic Analysis *416*
   4.4 NC Verification *420*
5. Concluding Remarks *423*
References *423*

I. Introduction to Computer Graphics (Part I)

R. J. Hubbold

1.0 ACKNOWLEDGEMENT

I should like to thank my colleagues Tony Arnold and Terry Hewitt for their help in producing these notes. The notes on graphics hardware are based partially on those written by Tony for Eurographics 82.

2.0 GENERAL INTRODUCTION

The main parts of a system for interactive computer graphics are:

- A computer. Used to store information from which pictures are constructed, to execute calculations, to generate the information needed to drive the graphical output device (e.g. display), and to monitor input tools controlled by the operator. This might be a large time-shared host. However, the popularity of this is diminishing because of poor response times - the trend is towards powerful personal computers.

- An interactive display. Used to show pictures.

- Input tools. Used to control the information which is displayed and how it is calculated. Examples include keyboard, joystick, graphic tablet, function buttons. (These will be described later.)

- Hard-copy devices. Plotters, printers etc., used to obtain a permanent record of pictures and other data.

Graphics equipment falls into two broad categories:

- Calligraphic. Pictures are drawn with lines. Suitable for many types of graphics and until recently the commonest type of equipment.

- Raster (or raster-scan). Pictures are made up from rows of dots which are drawn row by row in a similar manner to a domestic T.V.

There are three main aspects of computer graphics:

- Entering data into the computer, known as digitising. This can be performed manually, semi-automatically, or with a fully automatic scanner or T.V. camera.
- Viewing data graphically and updating it with some kind of display device and input tools.
- Obtaining hard-copy on paper with some kind of plotter or printer, or on film or video tape.

We will examine what equipment is available in each of these areas.

3.0 DIGITISING

There are three main ways in which information can be recorded from existing diagrams and other sources:

- Using a manual digitiser.
- With a semi-automatic line-following digitiser.
- With a scanner or T.V. camera.

3.1 MANUAL DIGITISERS

The manual digitiser consists of a flat drawing surface, rather like a draughtsman's drawing board, with many wires embedded in X and Y directions. Each wire carries a unique signal which is detected by a puck which is capacitively coupled to the tablet. These signals are decoded to produce the X and Y coordinates of the puck's position.

Usually there are several buttons on the puck which the operator may use to signal different types of data being recorded. Information can be picked off diagrams, charts, drawings, maps, X-ray films etc. by positioning the puck's cross-hairs over a point and pressing a button. Some systems also permit stream-mode recording, where points are sampled at either fixed time or distance intervals.

These digitisers produce coordinates of points and line segments.

3.2 SEMI-AUTOMATIC DIGITISERS

Manual digitising is tedious and error prone. One solution is to use a scanner which will follow lines. One such is the Laser-Scan Fastrack digitiser. This scans back and forth on either side of a line, recording the line's centre coordinates as it traces from one end of the line to the other. The system is interactive, so that ambiguities about which direction to take at nodes between different line segments can be resolved by the operator.

As with manual digitisers, the output is a set of coordinates of points and line segments.

3.3 SCANNERS AND T.V CAMERAS

These record pictures in the form of a raster image, with intensity information recorded for each point (pixel) in the picture. The various features within the picture must be derived using image analysis techniques. Such methods are increasingly popular with the growth of raster display systems, but need sophisticated software if they are to be used to input line diagrams.

4.0 GRAPHICAL DISPLAYS

Graphical displays vary enormously in cost, complexity and capabilities. At one end of the spectrum is the simple display terminal, and at the other are systems with considerable local intelligence, typified by the Evans & Sutherland PS300, which contains a Motorola 68000 together with highly specialised graphics processors.

We will examine three different types of display technology:

- o Direct view storage tube.
- o Refresh display.
- o Raster-scan display.

then look at various display processor architectures.

4.1 DIRECT VIEW STORAGE TUBE DISPLAY (DVST)

Marketed by Tektronix since early 1970s, this display has been very popular because of its relatively low cost and ease of programming and interfacing. It has become a de-facto standard and many newer displays employing other technologies (e.g. raster-scan) are "Tektronix compatible".

It is usually configured as a terminal operating via an RS232 interface. Drawing instructions received from a host computer are decoded by the display logic and used to trace the picture on a special type of CRT shown in Figure 1. The storage tube is a calligraphic display.

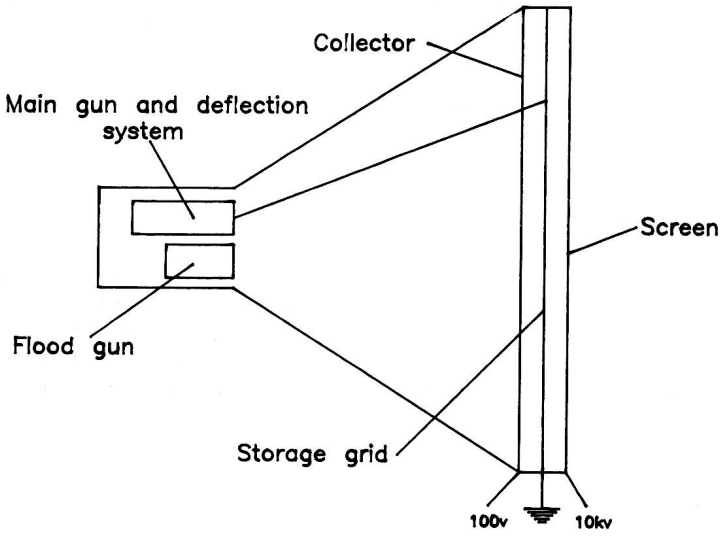

FIGURE 1 : DIRECT VIEW STORAGE TUBE (TEKTRONIX)

The picture is drawn on a storage grid by the main electron gun. This leaves a charge on the storage grid. Low-velocity electrons are emitted by the flood gun. The collector smooths the flow of electrons from the flood gun. In areas of the storage grid charged by the main gun, flood gun electrons can pass through and are attracted to the positively charged screen. These electrons strike the phosphor coating of the screen and produce a visible image.

The DVST is very popular mainly because:

- It's easy to program - uses ASCII codes.
- It's easy to interface - typically RS232.
- Can be used as an ordinary VDU.
- Picture is flicker free.
- It's relatively cheap (cheaper alternatives now available, however).

Problems are:

- Cannot remove charge selectively from grid - no selective erase. Screen must be cleared by applying high voltage to storage grid for up to half a second resulting in bright green flash. Limits its potential for interactive graphics.
- Charge leaks from storage grid eventually rendering picture invisible.
- Overall charge builds up on storage grid due to electron supply from flood gun resulting in overall background glow.

With an RS232 interface picture drawing times can be quite long. Because there is no selective erasure the whole picture must be re-drawn when deletions are made. Later models have local intelligence and only changes to the picture need to be re-transmitted from the host. This cuts re-draw times to about half a second.

4.2 REFRESH DISPLAYS

The refresh display is also a calligraphic device. Figure 2 shows a simple refresh display system. The picture is drawn on a CRT screen by controlling the deflection voltages so that the lines, text and other picture parts are traced by the CRT beam. The phosphor is excited by the electron beam and glows for a short period. To maintain a steady image, the picture must be re-drawn before the phosphor glow fades completely; usually 30 or 50 times a second.

The picture is refreshed from a stored description called the display file. If the display file contains more much data than can be processed in 1/50th of a second (i.e. a very complex picture is defined), the result is a flickering picture caused by fading of the image between frames.

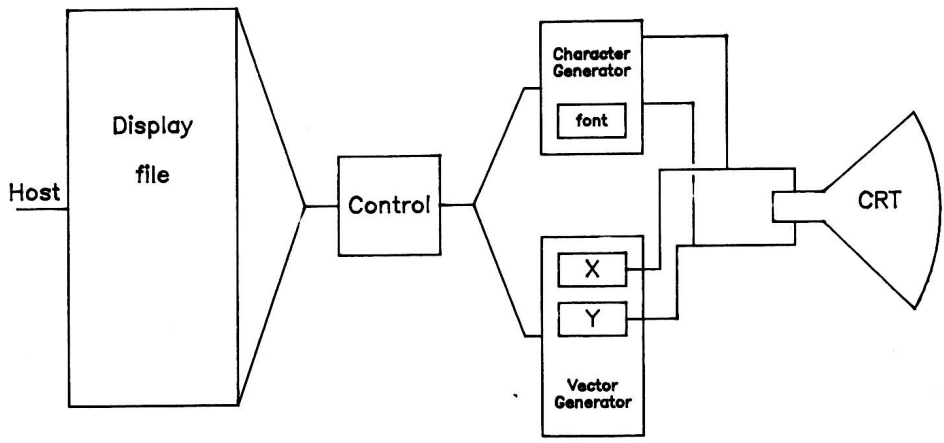

FIGURE 2: SIMPLE REFRESH DISPLAY

The display file may be modified dynamically by the host CPU, allowing selective erasure and animation. Display file may be in in host CPU store (cheap but requires high-speed DMA interface and puts considerable load on host I/O bus), or in display's own refresh buffer.

The display file contains two main classes of instruction:

- o Drawing instructions.
- o Control instructions.

Drawing instructions.

Figure 3 shows hypothetical instruction format. Some drawing instructions are:

- o MOVE - draw invisible line.
- o LINE - draw visible line.
- o TEXT - display character string.
- o ARC - draw part of a circle.

Lines and text are usually drawn by special vector and character generation hardware. More expensive systems can also draw arcs, curves etc. using special generators.

By modifying code within the display file, dynamically changing pictures can be drawn. Each time the code is altered the new picture appears instantaneously, because the display processor is continually refreshing

the picture using the current display file contents. This makes the refresh display a powerful interactive device.

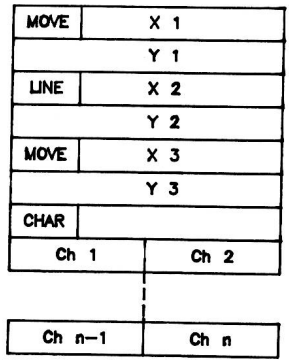

FIGURE 3: SIMPLE REFRESH DISPLAY INSTRUCTION FORMATS

Control instructions.

These control drawing parameters (e.g. line style, character size, intensity) and flow through display file. Typical instructions might be:

1. SET_LINE_STYLE - select solid or dashed line style.
2. SET_CHAR_SIZE - select character size.
3. SET_INTENSITY - select intensity of displayed information.
4. JUMP - specify address from which next display file instruction is to be fetched.
5. JUMP_TO_SUBROUTINE - as JUMP but remembers current address in display file.
6. RETURN - JUMP to address saved by JUMP_TO_SUBROUTINE + 1.

The JUMP instruction provides a way to add structure to the display file. Instead of a simple sequential list of instructions, a linked list of blocks of instructions can be constructed, making it easy to add new pictures and delete old ones by linking new blocks to the list and un-linking those no longer required.

The subroutine instructions permit more complex structuring. If a picture contains many copies of a sub-picture, JUMP_TO_SUBROUTINE allows its description to be stored once but invoked many times thus saving display file space. Coordinates specified inside subroutine need to be relative to current point.