Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

421

Tamiya Onodera Satoru Kawai

A Formal Model of Visualization in Computer Graphics Systems



Springer-Verlag Berlin Heidelberg New York London Paris Tokyo Hong Kong

Editorial Board

D. Barstow W. Brauer P. Brinch Hansen D. Gries D. Luckham C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

Authors

Tamiya Onodera IBM Research, Tokyo Research Laboratory 5-19, Sanban-cho, Chiyoda-ku Tokyo 102, Japan

Satoru Kawai Department of Graphics and Computer Science The College of Arts and Sciences, The University of Tokyo 3-8-1, Komaba, Meguro-ku Tokyo 153, Japan



CR Subject Classification (1987): 1.3.4

ISBN 3-540-52395-2 Springer-Verlag Berlin Heidelberg New York ISBN 0-387-52395-2 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this publication or parts thereof is only permitted under the provisions of the German Copyright Law of September 9, 1965, in its version of June 24, 1985, and a copyright fee must always be paid. Violations fall under the prosecution act of the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1990 Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr. 2145/3140-543210 – Printed on acid-free paper

Preface

BIBLIOTHEQUE DU CERIST

The field of computer graphics has expanded rapidly during the last decade. It provides a comprehensive way for computers and human beings to communicate, and is indispensable in modern computer systems. In order to facilitate the building of such systems, the kernel systems of computer graphics are designed and implemented to provide programmers with a set of fundamental graphics functions for producing pictures.

Visualization refers to the total process of transforming graphical objects from system-specified representations into their final representations. Unfortunately, visualization in conventional kernel systems is only poorly described: the specification documents are incredibly thick and most of the essential parts are written in natural language, relying on the incomplete framework of a viewing pipeline. The model of a viewing pipeline is insufficient, since it models only the viewing process, ignoring the important process of attribute rendering of graphical objects, and does not define the central concept of graphical output primitives. Moreover, the diversity of standards and proposed standards by the ISO for kernel systems embarrasses the computer graphics community, since we do not have a model on which two standards can be compared. All of these problems motivate the construction of a general, concise, and formal model of visualization.

The main purpose of this book is to present a formal model of visualization called a visualizing net. We give the model in a unified and axiomatic manner based on the mathematical concept of correspondence, which is a general form of mapping. Surprisingly, various attribute rendering processes, including pattern filling, are described there in a uniform way, and graphical output primitives are defined as mappings from finite generable objects. A visualizing net is a concise model that not only allows an unambiguous and comprehensive description of systems but can also be the center of the documentation. In addition, a visualizing net is far more general than a viewing pipeline. This implies that all visualizations in conventional kernel systems can be described by means of a visualizing net: we do so in Section 3.3.

This book is primarily intended for students or researchers interested in the formal aspects of computer graphics, which have been paid little attention by most researchers. One reason is that the research done so far in this field is not closely linked to actual systems.

We try in this book to link the model to reality. That is, we present the design and implementation of a kernel system called a *graphics nucleus*, which provides the full functionality of a visualizing net. The system is based on one general and expressive drawing technique, called *geometry-driven* drawing, which is the consequence of our abstraction of attribute rendition. For instance, it can draw roads or railways as simply as dotted lines.

Kernel system designers can also benefit from our work, since it contains a discussion of what a kernel system should be and expresses a view on how to avoid building complex, hopeless systems. Chapter 2 will be helpful for those who are interested in correspondence, since it contains indepth discussions on this subject, which has hitherto received far less attention than mapping or functions. Familiarity with computer graphics kernel systems and with first-order logic is assumed.

Acknowledgements

We are thankful to Messrs. T. Kamada, S. Matsuoke, H. Morishima, and C. Lee of the Kawai Laboratory of the University of Tokyo for their helpful discussions and valuable criticisms. We would also like to express my gratitude to Mr. H. Yokouchi of IBM Research, Tokyo Research Laboratory for his mathematical suggestions. Finally, We are grateful to Mr. McDonald of IBM Research, Tokyo Research Laboratory for his careful proof-reading.

Tokyo, January 1990

Tamiya Onodera Satoru Kawai

Contents



1 Introduction

1.1	The Outline	1			
1.2	Overviews of Computer Graphics Systems	3			
	1.2.1 Output Primitives and Primitive Attributes	3			
	1.2.2 Viewing Pipeline	6			
	1.2.3 Picture Storage	8			
1.3	Related Work	9			
	1.3.1 Specifying GKS	9			
	1.3.2 Mallgren's Specification	10			
	1.3.3 Detailed Viewing Pipeline	11			
	1.3.4 Comments	12			
N 4-					
wa	thematical Framework				
2.1	Correspondences	15			
2.2	Operations on Correspondences	16			
2.3	Equalities	17			
2.4	Extensions of Correspondences	21			
For					
FOI					
3.1	Formalization of Basic Concepts in Computer Graphics	25			
	3.1.1 Pictures and Picture Transformations	25			
	3.1.2 Picture Sum	26			
	3.1.3 Geometries and Geometry Transformations	26			
	3.1.4 Geometric Transformations	27			

2

3

	3.1.5	Restrictive Transformations	27	
	3.1.6	Replications	27	
	3.1.7	Pictorializing Transformations	28	
	3.1.8	Finite Ordered Geometries	28	
	3.1.9	Equalities	29	
3.2	Visual	lizing Nets	31	
3.3	Specif	ying Visualizations in Conventional Systems	33	
	3.3.1	Polyline in GKS	34	
	3.3.2	Polymarker in GKS	35	
	3.3.3	Line in Postscript	36	
	3.3.4	Fill Area in GKS	36	
	3.3.5	Comments	39	
Gra	phical	Nucleus		
4.1	Geom	etric and Pictorial Primitives	41	
	4.1.1	Formalization of Graphical Primitives	41	
	4.1.2	Sound Property of Primitives	43	
4.2	Graph	úcal Nucleus	44	
4.3	Imple	mentability of a Graphical Nucleus	44	
4.4	Optin	ization Considerations	47	
Imr	Implementation of a Craphical Nucleur			
5 1	Geom	atric and Pictorial Primitives in Nucleus	49	
5.2	Repre	sentation of Visualizing Nets	50	
5.2	Geom	atries and Geometry Transformations	51	
0.0	5.3.1	Restrictive Geometries	51	
	532	Geometries	51	
	0.0.4		50	
	533		1.1.2	

4

5

	5.4	Pictures and Picture Transformations	53	
		5.4.1 Pictures and Picture Naming	53	
		5.4.2 Picture Transformations	55	
	5.5	Pictorializing Transformations	56	
		5.5.1 Pictorializing a Point Geometry	56	
		5.5.2 Pictorializing a Lattice Geometry	57	
		5.5.3 Pictorializing a Line Geometry	58	
	5.6	Backend Display Function — peval	60	
6	Ext	ending a Graphical Nucleus		
	6.1	Simulating a Conventional Graphics Interface	63	
	6.2	Picture Storage	65	
	6.3	Generalized Texts	66	
7	Cor	telusion	69	
A	ppen	dices	75	
Appendix A Proofs of Propositions and Theorems			77	
\mathbf{A}_{j}	Appendix B pNucleus Reference Manual			
	ppen	dix B pNucleus Reference Manual		
	ppen B.1	dix B pNucleus Reference Manual Introduction	85	
	ppen B.1 B.2	dix B pNucleus Reference Manual Introduction	85 85	
	ppen B.1 B.2 B.3	dix B pNucleus Reference Manual Introduction	85 85 85	
	ppen B.1 B.2 B.3	dix B pNucleus Reference Manual Introduction	85 85 85 86	
	ppen B.1 B.2 B.3	dix B pNucleus Reference Manual Introduction Starting the pNucleus System Functions B.3.1 Control Functions B.3.2 Graphical Functions	85 85 85 86 87	
	ppen B.1 B.2 B.3	dix B pNucleus Reference Manual Introduction Starting the pNucleus System Functions B.3.1 Control Functions B.3.2 Graphical Functions B.3.3 Utility Functions	85 85 85 86 87 89	
	B.1 B.2 B.3 B.4	dix B pNucleus Reference Manual Introduction Starting the pNucleus System Functions B.3.1 Control Functions B.3.2 Graphical Functions B.3.3 Utility Functions Formal Syntax of the pNucleus System	85 85 86 87 89 90	
	B.1 B.2 B.3 B.4	dix B pNucleus Reference Manual Introduction Starting the pNucleus System Functions B.3.1 Control Functions B.3.2 Graphical Functions B.3.3 Utility Functions Formal Syntax of the pNucleus System B.4.1 Basic Data Types	85 85 86 87 89 90	

	B.4.3 Pictures	91	
	B.4.4 Invoking Transformations	91	
B .5	Examples	92	
Appen	dix C Implementation Issues in pNucleus		
C.1	Sectioning a Line Geometry	95	
C.2	Transformations for Mapping a Rectangle onto a Quadrangle	96	
	C.2.1 xy-Transformation	96	
	C.2.2 Two-Dimensional Projective Transformation	97	
Biblio	Bibliography		

List of Figures

1.1	Typical output primitives	5
1.2	GKS viewing pipeline	7
1.3	PHIGS viewing pipeline	7
2.1	Divergent and convergent correspondences	16
2.2	Correspondence for illustrating Proposition 3	19
2.3	Correspondences for illustrating the equality in Theorem 1	20
2.4	Correspondences for illustrating the equality in Theorem 2	21
2.5	Diagram representing both sides in the equality of Theorem 4	23
3.1	Picture transformation derived from a divergent correspondence	26
3.2	Replication defined by three survergent correspondences	28
3.3	Intrusion of a geometric transformation into a restrictive transformation when the application order is reversed	30
3.4	Pattern generation when Dom $r_1 eq$ Dom p	32
3.5	Visualizing net	33
3.6	Visualizing net of GKS polyline	34
3.7	Effects of two different restrictive transformations of polyline	35
3.8	Visualizing net of GKS polymarker	36
3.9	Effects of two different restrictive transformations of polymarker	37
3.10	Visualizing net of Postscript line	37
3.1 1	Visualizing net of GKS fill area	38
4.1	Examples of geometric primitives	42
4.2	Soundness of geometric and pictorial primitives	43
5.1	Region specified by the expression in Example 5.1	52

5.2	Picture to which the name *A is assigned	. 54
5.3	Pictorialization of a Point Geometry	. 57
5.4	Pictorialization of a Lattice Geometry	, 58
5.5	Development of Sections	. 59
5.6	Pictorialization of a Line Geometry	. 60
6.1	Conventional picture storing by pNucleus	. 66
7.1	Pictorializaiton of a line geometry	. 71
7.1	Pictorialization of a line geometry (continued)	. 72
7.2	Pictorialization of a polygon geometry	. 73
C.1	Determining section height vectors	. 96
C.2	Pictorialization of a line geometry by two-dimensional projective transformation	. 98

List of Tables

1.1	GKS primitives with parameters supplied in the invocation and with basic attributes	4
	controlling their appearances	-1
2.1	Results of applying correspondence operations	18
B. 1	Data type notation	86