Applications of Spatial Data Structures

Computer Graphics, Image Processing, and GIS

Hanan Samet
University of Maryland

The quadtree and octree are hierarchical data structures used to represent spatial data. This book shows how these spatial data structures can be applied in computer graphics, image processing, geographic information systems (GIS), and other domains. It illustrates the representation of region data, in particular, both in two dimensions with quadtrees and in three dimensions with octrees.

The advantage of hierarchical data structures is their ability to focus on interesting subsets of the data, resulting in more efficient representation and faster execution. Thus, they are particularly convenient for performing set operations. Even where other data structures might perform as efficiently, hierarchical data structures are often preferred for their conceptual clarity and ease of implementation.

The author begins with a general introduction to spatial data structures. He continues with explanations of how to convert between quadtrees and other representations, as well as applications such as geometric property computation, image transformations, ray tracing, and image compression. The approach is algorithmic. Each chapter after the introduction contains at least one detailed algorithm, written in pseudo code, to illustrate an application. Each critical step in these algorithms is described with a liberal use of examples. Each chapter also includes a large number of exercises, with solutions provided for many.

The book focuses on implementation techniques and the uses of spatial data structures. Readers desiring a more complete introduction to these representations, including fundamental concepts, basic techniques, and a broad range of possible uses, should refer to another book by the same author, *The Design and Analysis of Spatial Data Structures.* The introductory material in *Applications of Spatial Data Structures* makes the book self-contained. Each book contains an extensive bibliography, and the books are cross-referenced.

Hanan Samet is an internationally recognized authority on spatial data structures. Readers will find practical value in his work for a variety of applications involving all kinds of spatial data.

90000
9 780201 503005
ISBN 0-201-50300-X

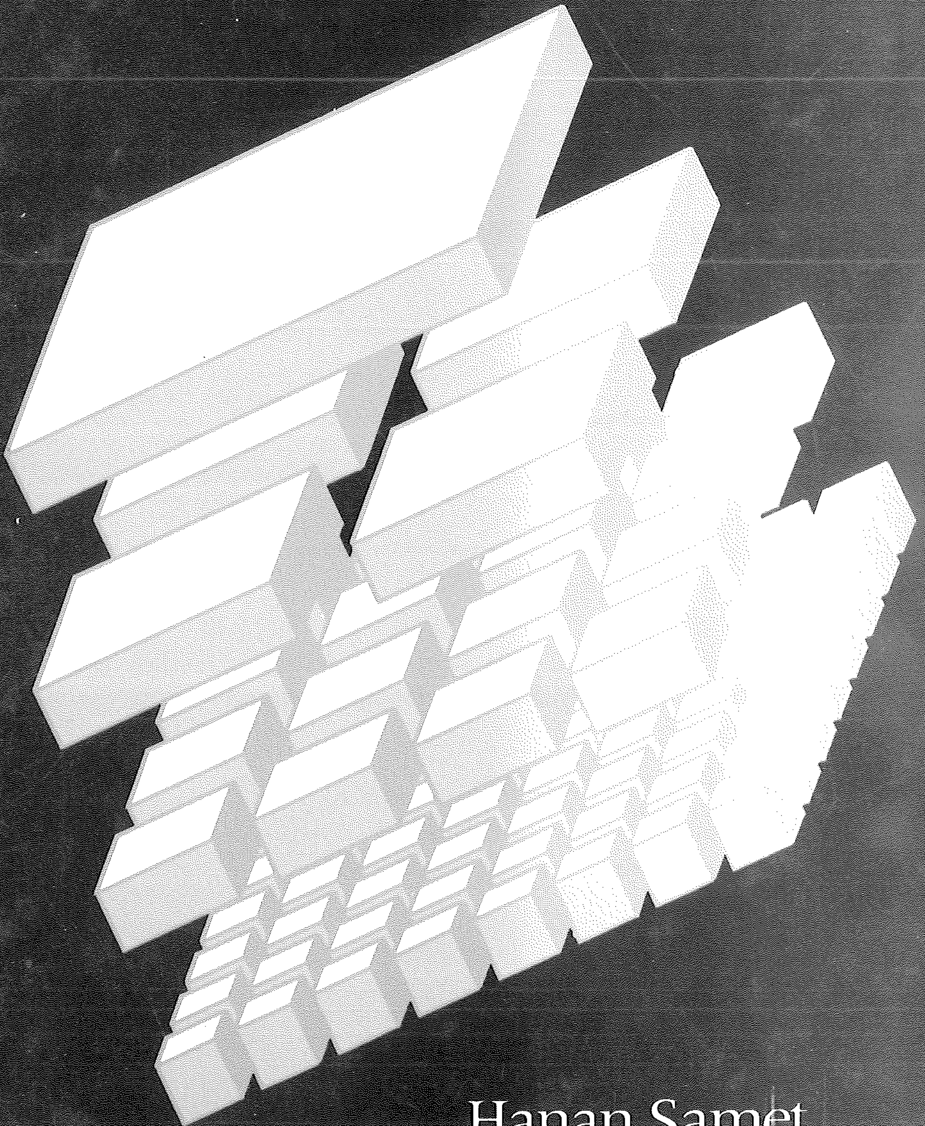
Graphics
Addison-Wesley Publishing Company

Applications of Spatial Data Structures
Computer Graphics, Image Processing, and GIS
Samet

50300

Applications of Spatial Data Structures

Computer Graphics, Image Processing, and GIS



Hanan Samet

Applications of Spatial Data Structures

Applications of Spatial Data Structures

Computer Graphics, Image Processing, and GIS

Hanan Samet
UNIVERSITY OF MARYLAND

ADDISON - WESLEY PUBLISHING COMPANY
Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam
Bonn • Sydney • Singapore • Tokyo • Madrid • San Juan

This book is in the Addison-Wesley Series in Computer Science
Michael A. Harrison: Consulting Editor

Library of Congress Cataloging-in-Publication Data

Samet, Hanan.
Applications of spatial data structures: computer graphics, image processing, and GIS / by Hanan Samet.
p. cm.
Bibliography: p.
Includes index.
ISBN 0–201–50300–X
1. Data structures (Computer science) 2. Computer graphics.
I. Title.
QA76.9.D35S25 1989 89–30365
006—dc19 CIP

Credits:
Thor Bestul created the cover art.
Figures 1.1, 2.3, 3.1, 4.1, 4.3, and 5.16 are from H. Samet and R. E. Webber, On encoding boundaries with quadtrees, *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 3(May 1984), 365–369. © 1984 IEEE. Reprinted by permission of IEEE.
Figures 1.2, 1.3, 1.17, 6.2, 6.4, 6.5, and 6.12 are from H. Samet and R. E. Webber, Hierarchical data structures and algorithms for computer graphics, Part I. Fundamentals, *IEEE Computer Graphics and Applications 8*, 3(May 1988), 48–68. © 1988 IEEE. Reprinted by permission of IEEE.
Figures 1.4 through 1.6, 1.9 through 1.14, 1.18, 3.3, 4.16, 4.28, 5.2, 5.17, 5.18, 6.1, 6.3, and 6.10 are from H. Samet, The quadtree and related hierarchical data structures, *ACM Computing Surveys 16*, 2(June 1984), 187–260. Reprinted by permission of ACM.
Figures 1.15 and 4.24 are from H. Samet and R. E. Webber, Storing a collection of polygons using quadtrees, *ACM Transactions on Graphics 4*, 3(July 1985), 182–222. Reprinted by permission of ACM.
Figures 2.1, 4.22, 4.23, and 8.1 are from C. R. Dyer, A. Rosenfeld, and H. Samet, Region representation: boundary codes from quadtrees, *Communications of the ACM 23*, 3(March 1980), 171–179. Reprinted by permission of ACM.
Figures 2.2 and 8.2 through 8.19 are from H. Samet, Data structures for quadtree approximation and compression, *Communications of the ACM 28*, 9(September 1985), 973–993. Reprinted by permission of ACM.
Figures 2.4 and 4.13 are from C. A. Shaffer and H. Samet, Optimal quadtree construction algorithms, *Computer Vision, Graphics, and Image Processing 37*, 3(March 1987), 402–419. Reprinted by permission of Academic Press.
Continued on p. 507

To my parents, Julius and Lotte

PREFACE

The quadtree and octree are hierarchical data structures used to represent spatial data. They are based on the principle of recursive decomposition (similar to *divide and conquer* methods [Aho74]). This book focuses on the use of quadtree and octree representations of region data (in two and three dimensions, respectively) in applications in computer graphics, image processing, and geographic information systems (GIS), as well as computer vision, robotics, pattern recognition, solid modeling, and other areas. For a comprehensive treatment of related hierarchical representations of spatial data including points, lines, rectangles, regions, and volumes, see [Same90a].

To many people, the terms *quadtree* and *octree* have taken on a generic meaning synonymous with the term *hierarchical data structure*. Hierarchical data structures are useful because of their ability to focus on the interesting subsets of the data. This focusing results in an efficient representation and in improved execution times. Thus they are particularly convenient for performing set operations. Many of the operations described can often be performed as efficiently, or more so, with other data structures. Nevertheless, hierarchical data structures are attractive because of their conceptual clarity and ease of implementation. In addition, the use of some of them provides a spatial index. This is very useful in applications involving spatial databases.

This book is organized as follows. Chapter 1 introduces hierarchical data structures such as the quadtree and octree. It reviews their key properties, traces their history, and gives an overview of their use in representing point and line data, as well as three-dimensional regions. All of these topics are covered in much greater detail in [Same90a].

Most hierarchical data structures are trees. As such they are most often implemented using pointers. Chapter 2 discusses alternative implementations that do not make use of pointers and compares their storage requirements with an implementation that does use pointers. Chapter 3 contains a detailed presentation of how to perform neighbor finding. This is an important technique in the efficient implementation of a number of algorithms that use quadtree and octree-like representations. It is used heavily in their construction (Chapter 4), computing geometric properties (Chapter 5), ray tracing (Chapter 7), and skeletons (Chapter 9). Chapters 2 and 3 may be skipped by readers who are interested only in applications.

The remaining chapters discuss the applications in greater detail. Chapter 4 shows the ease with which conversions can be made between the quadtree representation of two-dimensional regions and more conventional representations such as arrays, rasters, and boundary codes. The extension to three-dimensional regions is straightforward and is discussed only in the context of building an octree from a set of views of an object or a scene of three-dimensional objects. For details on converting between other representations of three-dimensional regions and octrees (e.g., boundary model [BRep], constructive solid geometry [CSG]), see Chapter 5 of [Same90a].

Chapter 5 examines the computation of geometric properties such as connected component labeling. Such operations arise in computer graphics (where it is known as polygon coloring or filling) and as a basic step in the processing of an image.

Chapter 6 discusses the implementation of set-theoretic operations, linear transformations, and other algorithms useful in computer graphics and image processing. Chapter 7 contains a detailed presentation of the use of hierarchical data structures in the display of graphical information. In particular, much attention is given to the problem of ray tracing and some to beam tracing. The radiosity method is covered with considerably less detail. Chapters 6 and 7 may be skipped by readers uninterested in computer graphics.

Chapter 8 treats the problem of image approximation and compression by use of hierarchical representations. This discussion is in the context of two-dimensional binary images. Chapter 9 examines the application of quadtree-like decomposition to the computation of skeletons. In particular, the concept of distance is formulated in the context of a quadtree, and its application is explored. Chapters 8 and 9 are somewhat specialized and may be omitted in the interest of time.

This book is designed to be used as a reference as well as the basis of a course on the implementation of a graphics, image processing, or geographic information system (GIS) based on quadtrees and octrees. It can also be used as a supplement in a general course on these topics.

There are a number of topics for which justice requires considerably more detailed treatment. Due to space limitations, I have omitted a detailed discussion of them and instead refer interested readers to the appropriate literature. The notion of a pyramid is presented only at a cursory level in Chapter 1 so that it can be contrasted with the quadtree. In particular, the pyramid is a multiresolution representation, whereas the quadtree is a variable resolution representation. Readers are referred to Tanimoto and Klinger [Tani80] and the collection of papers edited by Rosenfeld [Rose83a] for a more comprehensive exposition on pyramids. The use of quadtrees in finite element analysis is mentioned in Chapter 1; for more details, see Kela, Perucchio, and Voelcker [Kela86] and the references cited there. Similarly I discuss image compression and coding only in the context of hierarchical data structures. This is done in Chapter 8.

For more details on early results involving these and related topics, consult the surveys by Nagy and Wagle [Nagy79], Peuquet [Peuq84], Requicha [Requ80], Srihari [Srih81], Samet and Rosenfeld [Same80d], Samet [Same84b], and Samet and Webber [Same88c, Same88d]. A number of excellent texts contain material related to the topics that I cover. Rosenfeld and Kak [Rose82a] should be consulted for an encyclopedic treatment of image processing. Mäntylä [Mänt87] has written a comprehensive introduction to solid modeling. Burrough [Burr86] provides a survey of geographic information systems. For a comprehensive view of the literature, see Rosenfeld's annual collection of references in the journal *Computer Vision, Graphics, and Image Processing* (e.g., [Rose88]).

Nevertheless, given the broad and rapidly expanding nature of the field, I am bound to have omitted significant concepts and references. In addition, at times, I devote a disproportionate amount of attention to some concepts at the expense of others. This is principally for expository purposes; I feel that it is better to understand some structures well rather than to give readers a quick runthrough of buzzwords. For these indiscretions, I beg your pardon and hope you nevertheless bear with me.

My approach is an algorithmic one. Whenever possible, I have tried to motivate critical steps in the algorithms by a liberal use of examples. I feel that it is of paramount importance for readers to see the ease with which the representations can be implemented and used. In each chapter, except for the introduction (Chapter 1), I give at least one detailed algorithm using pseudo-code so that readers can see how the ideas can be applied. The pseudo-code is a variant of the ALGOL [Naur60] programming language that has a data structuring facility incorporating pointers and record structures. Recursion is used heavily. This language has similarities to C [Kern78], PASCAL [Jens74], SAIL [Reis76], and ALGOL W [Baue68]. Its basic features are described in the Appendix. However, the actual code is not crucial to understanding the techniques and it may be skipped on a first reading. The index indicates the page numbers where the code for each algorithm is found.

In many cases I also give an analysis of the space and time requirements of different data structures and algorithms. The analysis is usually of an asymptotic nature and is in terms of *big O* and Ω notation [Knut76]. The *big O* notation denotes an upper bound. For example, if an algorithm takes $O(\log_2 N)$ time, then its worst-case behavior is never any worse than $\log_2 N$. The Ω notation denotes a lower bound. As an example of its use, consider the problem of sorting N numbers. When I say that sorting is $\Omega(N \cdot \log_2 N)$, I mean that given any algorithm for sorting, there is some set of N input values for which the algorithm will require at least this much time.

At times I also describe implementations of some of the data structures for the purpose of comparison. In such cases counts such as the number of fields in a record are often given. These numbers are meant only to amplify the discussion. They are not to be taken literally, as improvements are always possible once a specific application is analyzed more carefully.

Each chapter contains a substantial number of exercises. Many of the exercises develop further the material in the text as a means of testing the reader's understanding, as well as suggesting future directions. When the exercise or its solution is not my own, I have preceded it with the name of its originator. The exercises have not been graded by difficulty. They rarely require any mathematical skills beyond the undergraduate level for their solution. However, while some of the exercises are quite straightforward, others require some ingenuity. Solutions, or references to papers that contain the solution, are provided for a substantial number of the exercises that do not require programming. Readers are cautioned to try to solve the exercises before turning to the solutions. It is my belief that much can be learned this way (for the student and, even more so, for the author). The motivation for undertaking this task was my wonderful experience on my first encounter with the rich work on data structures by Knuth [Knut73a, Knut73b].

An extensive bibliography is provided. It contains entries for both this book and the companion text [Same90a]. Not all of the references that appear in the bibliography are cited in the two texts. They are retained for the purpose of giving readers the ability to access the entire body of literature relevant to the topics discussed in them. Each reference is annotated with a key word(s) and a list of the numbers of the sections in which it is cited in either of the texts (including exercises and solutions). In addition, a name and credit index is provided that indicates the page numbers in this book on which each author's work is cited or a credit is made.

ACKNOWLEDGMENTS

Over the years I have received help from many people, and I am extremely grateful to them. In particular, Robert E. Webber, Markku Tamminen, and Michael B. Dillencourt have generously given me much of their time and have gone over critical parts of the book. I have drawn heavily on their knowledge of some of the topics covered here. I have also been extremely fortunate to work with Azriel Rosenfeld over the past ten years. His dedication and scholarship have been a true inspiration to me. I deeply cherish our association.

I was introduced to the field of spatial data structures by Gary D. Knott who asked "how to delete in point quadtrees." Azriel Rosenfeld and Charles R. Dyer provided much interaction in the initial phase of my research. Those discussions led to the discovery of the neighbor-finding principle. It is during that time that many of the basic conversion algorithms between quadtrees and other image representations were developed as well. I learned much about image processing and computer vision from them. Robert E. Webber taught me computer graphics, Markku Tamminen taught me solid modeling and representations for multiattribute data, and Michael B. Dillencourt taught me about computational geometry.

During the time that this book was written, my research was supported, in part, by the National Science Foundation, the Defense Mapping Agency, the Harry Diamond Laboratory, and the Bureau of the Census. In particular, I thank Richard Antony, Y. T. Chien, Su-shing Chen, Hank Cook, Phil Emmerman, Joe Rastatter, Alan Saalfeld, and Larry Tokarcik. I am appreciative of their support.

Many people helped me in the process of preparing the book for publication. Acknowledgments are due to Rene McDonald for coordinating the day-to-day matters of getting the book out and copyediting; to Scott Carson, Emery Jou, and Jim Purtilo for TROFF assistance beyond the call of duty; to Marisa Antoy and Sergio Antoy for designing and implementing the algorithm formatter used to typeset the algorithms; to Barbara Burnett, Michael B. Dillencourt, and Sandy German for help with the index; to Jay Weber for setting up the TROFF macrofiles so that I can keep track of symbolic names and thus be able to move text around without worrying about the numbering of exercises, sections, and chapters; to Liz Allen for early TROFF help; to Nono Kusuma, Mark Stanley, and Joan Wright Hamilton for drawing the figures; to Richard Muntz and Gerald Estrin for providing temporary office space and computer access at UCLA; to Sandy German, Gwen Nelson, and Janet Salzman for help in initial typing of the manuscript; to S. S. Iyengar, Duane Marble, George Nagy, and Terry Smith who reviewed the book; and to Peter Gordon, John Remington, and Keith Wollman at Addison-Wesley Publishing Company for their encouragement and confidence in this project.

Aside from the individuals already named, I have also benefited from discussions with many people over the past years. They have commented on various parts of the book and include Chuan-Heng Ang, Walid Aref, James Arvo, Thor Bestul, Sharat Chandran, Chiun-Hong Chien, Jiang-Hsing Chu, Leila De Floriani, Roger Eastman, Herbert Edelsbrunner, Christos Faloutsos, George (Gyuri) Fekete, Kikuo Fujimura, John Gannon, John Goldak, Erik Hoel, Liuqing Huang, Frederik W. Jansen, Ajay Kela, David Kirk, Per Åke Larson, Dani Lischinski, Don Meagher, David Mount, Randal C. Nelson, Glenn Pearson, Ron Sacks-Davis, Timos Sellis, Clifford A. Shaffer, Deepak Sherlekar, Li Tong, Brian Von Herzen, Peter Widmayer, and David Wise. I deeply appreciate their help.

CONTENTS