Visual Computing: Geometry, Graphics, and Vision
LIMITED WARRANTY AND DISCLAIMER OF LIABILITY

CHARLES RIVER MEDIA, INC. ("CRM") AND/OR ANYONE WHO HAS BEEN INVOLVED IN THE WRITING, CREATION, OR PRODUCTION OF THE ACCOMPANYING CODE IN THE TEXTUAL MATERIAL IN THE BOOK, CANNOT AND DO NOT WARRANT THE PERFORMANCE OR RESULTS THAT MAY BE OBTAINED BY USING THE CONTENTS OF THE BOOK. THE AUTHOR AND PUBLISHER HAVE USED THEIR BEST EFFORTS TO ENSURE THE ACCURACY AND FUNCTIONALITY OF THE TEXTUAL MATERIAL AND PROGRAMS DESCRIBED HEREIN. WE HOWEVER, MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, REGARDING THE PERFORMANCE OF THESE PROGRAMS OR CONTENTS. THE BOOK IS SOLD "AS IS" WITHOUT WARRANTY (EXCEPT FOR DEFECTIVE MATERIALS USED IN MANUFACTURING THE BOOK OR DUE TO FAULTY WORKMANSHIP).

THE AUTHOR, THE PUBLISHER, AND ANYONE INVOLVED IN THE PRODUCTION AND MANUFACTURING OF THIS WORK SHALL NOT BE LIABLE FOR DAMAGES OF ANY KIND ARISING OUT OF THE USE OF (OR THE INABILITY TO USE) THE PROGRAMS, SOURCE CODE, OR TEXTUAL MATERIAL CONTAINED IN THIS PUBLICATION. THIS INCLUDES, BUT IS NOT LIMITED TO, LOSS OF REVENUE OR PROFIT, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THE PRODUCT.

THE SOLE REMEDY IN THE EVENT OF A CLAIM OF ANY KIND IS EXPRESSLY LIMITED TO REPLACEMENT OF THE BOOK, AND ONLY AT THE DISCRETION OF CRM.

THE USE OF "IMPLIED WARRANTY" AND CERTAIN "EXCLUSIONS" VARIES FROM STATE TO STATE, AND MAY NOT APPLY TO THE PURCHASER OF THIS PRODUCT.
Visual Computing: Geometry, Graphics, and Vision

Frank Nielsen
To my family,
To Audrey 玲奈 and Julien 怜旺

Ariel by Audrey (3 years old)
Visual Computing: Geometry, Graphics, and Vision
# Contents

## Foreword

## Acknowledgments

## Chapter 1 Introduction

1.2. Target Audience ............................................. 2
1.3. Organization of the Book ..................................... 3
1.4. Future of Visual Computing .................................. 5
1.5. Companion Web Site .......................................... 6

## Chapter 2 Abstract Data Structures

2.1. Pointers, Arrays, Lists, and Graphs ........................ 8
2.1.1. Pointers .................................................. 8
2.1.2. Arrays ..................................................... 10
   Images as Bidimensional Arrays .................................. 12
   Multidimensional Arrays ....................................... 13
2.1.3. Singly and Doubly Linked Lists .......................... 15
2.1.4. Graphs .................................................... 17
   Adjacency Matrices ........................................... 18
   Adjacency Lists ............................................. 19
   Trees ......................................................... 19
   Images as Graphs ............................................ 20
2.2. Stacks and Queues ............................................ 21
2.2.1. Application: Area Flood-filling ......................... 21
2.2.2. Recursive Algorithm .................................... 22
2.2.3. Flood-filling Using Stacks .............................. 23
2.2.4. Flood-filling Using Queues .............................. 26
2.2.5. Implementing Stacks and Queues ....................... 28
## Chapter 2

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3. Dictionaries</td>
<td>29</td>
</tr>
<tr>
<td>2.3.1. Unordered Dictionaries</td>
<td>29</td>
</tr>
<tr>
<td>2.3.2. Ordered Dictionaries</td>
<td>29</td>
</tr>
<tr>
<td>2.3.3. Application: Detecting Any Segment Pair Intersection</td>
<td>30</td>
</tr>
<tr>
<td>2.4. Priority Queues</td>
<td>33</td>
</tr>
<tr>
<td>2.4.1. Primitives</td>
<td>34</td>
</tr>
<tr>
<td>2.4.2. Application: Reporting All Segment Pair Intersections</td>
<td>34</td>
</tr>
<tr>
<td>2.5. Disjoint Sets</td>
<td>37</td>
</tr>
<tr>
<td>2.5.1. Application: Image Segmentation</td>
<td>37</td>
</tr>
<tr>
<td>2.5.2. Union-find Data Structure</td>
<td>39</td>
</tr>
<tr>
<td>2.6. Geometric Hashing</td>
<td>44</td>
</tr>
<tr>
<td>2.6.1. Hashing by Chaining</td>
<td>44</td>
</tr>
<tr>
<td>2.6.2. Application: Object Recognition by Hashing Point Sets</td>
<td>45</td>
</tr>
<tr>
<td>2.7. C++ Standard Template Library and Traits Classes</td>
<td>48</td>
</tr>
<tr>
<td>2.7.1. Class and Function Templates</td>
<td>48</td>
</tr>
<tr>
<td>2.7.2. C++ Standard Template Library</td>
<td>49</td>
</tr>
<tr>
<td>2.7.3. Traits Classes</td>
<td>53</td>
</tr>
<tr>
<td>Handling Constants</td>
<td>53</td>
</tr>
<tr>
<td>Unifying Types</td>
<td>54</td>
</tr>
<tr>
<td>Geometric Traits Classes</td>
<td>55</td>
</tr>
</tbody>
</table>

## Chapter 3

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. Transformation Principles</td>
<td>63</td>
</tr>
<tr>
<td>3.1.1. Coordinate Systems versus Modeling Transformations</td>
<td>64</td>
</tr>
<tr>
<td>Local and Global Coordinate Systems</td>
<td>65</td>
</tr>
<tr>
<td>Modeling Transformations</td>
<td>67</td>
</tr>
<tr>
<td>3.1.2. Transformations as Matrices</td>
<td>69</td>
</tr>
<tr>
<td>3.1.3. Homogeneous and Inhomogeneous Coordinates</td>
<td>74</td>
</tr>
<tr>
<td>3.2. Geometry Pipeline</td>
<td>84</td>
</tr>
<tr>
<td>3.2.1. 2D Transformations</td>
<td>85</td>
</tr>
<tr>
<td>3.2.2. 3D Transformations</td>
<td>92</td>
</tr>
<tr>
<td>3.2.3. Rotations as Quaternions</td>
<td>99</td>
</tr>
<tr>
<td>3.2.4. Normal Transformations</td>
<td>103</td>
</tr>
<tr>
<td>3.3. Graphics Pipeline</td>
<td>106</td>
</tr>
<tr>
<td>3.3.1. Transformation Hierarchies</td>
<td>107</td>
</tr>
<tr>
<td>3.3.2. Scene Graphs</td>
<td>110</td>
</tr>
<tr>
<td>3.3.3. Viewing Transformations</td>
<td>112</td>
</tr>
<tr>
<td>3.3.4. Projections</td>
<td>113</td>
</tr>
<tr>
<td>Orthographic Projections</td>
<td>113</td>
</tr>
</tbody>
</table>
Contents

Perspective Projections .................................................. 115
3.3.5. 3D Clippings and Projection Transformations .................. 119
Orthographic Canonical View Volume ................................. 120
Perspective Truncated Pyramid ......................................... 121
3.3.6. View Plane Clipping ............................................... 122
3.4. Vision Pipelines .......................................................... 127
3.4.1. Homographies ......................................................... 127
3.4.2. Camera Calibrations .................................................. 138
3.4.3. Pinhole Camera: Extrinsic and Intrinsic Parameters .......... 138
3.4.4. Epipolar Geometry .................................................... 145
3.5. Advanced Coordinate Pipelines ...................................... 153
3.5.1. Log-Polar Transform ................................................ 153
3.5.2. Cylindrical and Spherical Coordinates ................................ 155
3.5.3. Polarity ................................................................. 160
3.5.4. Conics and Quadrics ............................................... 162
3.5.5. Plücker Coordinates ................................................. 166
3.5.6. Eigenvectors and Matrix Diagonalization ....................... 171
3.5.7. Singular Value Decomposition (SVD) ............................ 172
3.5.8. Taxonomy of Projections .......................................... 178
3.5.9. Cameras with Lenses ............................................... 179
3.5.10. Tangent Spaces ..................................................... 183
3.5.11. Environment Maps .................................................. 185
3.6. Summary and Perspectives ............................................ 191
3.7. Bibliographical Notes .................................................. 195

CHAPTER 4 IMAGES

4.1. Application: Warping and Morphing Images ....................... 200
4.1.1. Image Warping ....................................................... 201
4.1.2. Image Morphing ..................................................... 204
4.1.3. Compositing with Alpha Matting ................................ 209
4.1.4. View Morphing ...................................................... 213
4.2. Interpolations ............................................................. 216
4.2.1. Resampling and Antialiasing ..................................... 217
Convolutions .................................................................. 219
Discrete Convolution Kernels .......................................... 221
Fourier Analysis ............................................................. 224
Antialiasing ................................................................. 231
Supersampling ............................................................... 231
Reconstruction Filters ..................................................... 232
Phase Correlation ........................................................... 234
4.2.2. Barycentric Coordinate Interpolations ............ 235
4.2.3. Elliptical Weighted Average Interpolations ........ 239
4.2.4. Superresolution ................................ 241
4.3. Colors .......................................... 242
  4.3.1. Radiometry and Photometry .................. 242
  4.3.2. Color Spaces .................................. 247
      Color Matching and Chromaticity Diagram ........ 248
      XYZ Color Space .................................. 250
      RGB Color Space .................................. 251
      HLS and HSV Color Spaces ....................... 252
      CMY(K) Color Space .............................. 254
      L*a*b* Color Space ............................... 254
      Pseudocoloring and Other Miscellaneous Color Techniques 255
  4.3.3. Physical Color Phenomena .................... 259
      Fluorescence and Phosphorescence ............... 259
      Iridescence ..................................... 260
      Fresnel Effect ................................... 260
  4.3.4. Perceptual Color Phenomena .................. 261
      Simultaneous Contrast ............................ 262
      Mach Bands ...................................... 263
      Color-appearance Models ....................... 263
4.4. Halftoning and Dithering .......................... 264
  4.4.1. Digital Halftoning ........................... 265
  4.4.2. Dithering .................................... 266
4.5. High-dynamic Range Imaging ....................... 271
  4.5.1. Light Probes ................................. 274
  4.5.2. Tone Mapping .................................. 274
4.6. Image Pyramids .................................. 274
  4.6.1. Mipmappings .................................. 275
  4.6.2. Gaussian and Laplacian Pyramids ............... 278
  4.6.3. Applications of Scaled Representations ....... 280
4.7. Bibliographical Notes ............................ 282

Chapter 5: Meshes ........................................ 285

5.1. Prelude to Meshes ................................. 285
  5.1.1. Polygons ..................................... 285
  5.1.2. Polyhedra ................................... 288
5.2. Basic Mesh Descriptions ........................... 290
  5.2.1. Mesh Topology ................................. 299
  5.2.2. Modeling Meshes .............................. 304
Contents

5.2.3. Triangulations and Tessellations ................................. 305
5.2.4. Procedural Meshes ............................................ 311
5.3. Data Structures for Meshes ........................................ 314
  5.3.1. Static Meshes ............................................. 315
         Indexed Face List ........................................... 315
         Bandwidth Optimization .................................... 318
  5.3.2. Winged Edges ............................................... 320
  5.3.3. Half-edges .................................................. 321
  5.3.4. Quad-edges .................................................. 323
5.4. Operations on Meshes ............................................... 326
  5.4.1. Fairing Meshes ............................................. 326
  5.4.2. Smothing Meshes by Subdivisions .......................... 328
  5.4.3. Remeshing ................................................. 334
  5.4.4. Progressive Meshes ....................................... 344
  5.4.5. Parameterization ........................................... 346
5.5. Geometry Images ................................................... 352
5.6. Alternative to Meshes .............................................. 355
5.7. Bibliographical Notes ............................................. 358

Chapter 6 Animation .................................................... 363

  6.1. Kinetic Data Structures ........................................ 363
  6.2. Motion Capture .................................................. 368
       6.2.1. Tracking ............................................... 368
       6.2.2. Matchmoving ............................................ 374
  6.3. Computer Graphics Animation ................................... 375
       6.3.1. Keyframe Animation ................................... 376
       6.3.2. Forward and Inverse Kinematics ......................... 377
       6.3.3. Active versus Passive Dynamics ........................ 380
       6.3.4. Particle Systems ....................................... 381
       6.3.5. Animating Rotations by Quaternions ..................... 384
  6.4. Bibliographical Notes ........................................... 387

Chapter 7 Randomization .................................................. 389

  7.1. Randomized Analysis of QuickSort ................................ 389
  7.2. Random Sample Consensus ....................................... 400
  7.3. Monte Carlo Samplings ......................................... 406
  7.4. Randomizing Incremental Algorithms ........................... 410
  7.5. Randomized Incremental Optimization .......................... 414
  7.6. Skip Lists ..................................................... 419
  7.7. Bibliographical Notes ......................................... 422
# Chapter 8: Higher Dimensions for “3D”

## 8.1. Nearest Neighbors
- Application: 2D Texture Synthesis
- kD-Trees

## 8.2. Clustering
- Application: Color Quantization
- Clustering by kMeans

## 8.3. Mathematical Techniques
- Linearization
- Application: Closest Pair of Points
- Lifting for Designing Geometric Predicates

## 8.3.2. Approximating Distances in Large Dimensions

## 8.4. Bibliographical Notes

# Chapter 9: Robustness

## 9.1. Identifying Code Weaknesses and Defining Robustness

## 9.2. IEEE 754 Binary Floating-point Standard

## 9.3. Filtering Predicates

## 9.4. Predicate Degrees

## 9.5. Overview of Libraries

## 9.6. Bibliographical Notes

---

## Acronyms

## Notational Conventions

## Colophon

## Bibliography

## Index
Foreword

I am really excited to be able to write a foreword to Frank Nielsen’s new book *Visual Computing: Geometry, Graphics, and Vision*. The fusion of computer graphics, computer vision, computational geometry, and discrete algorithms that this book presents is truly unique and, in afterthought, so obvious. Geometry, graphics, and vision all deal in some form with the shape of objects, their motions, as well as the transport of light and its interaction with objects—yet historically they have been covered by separate courses in curricula, grown around a distinct set of conferences, and cultivated separate communities. This book clearly shows how much they have in common and the kinds of synergies that occur when a common core of material is presented in a way that both serves and is enriched by all three disciplines.

Take coordinates and coordinate transforms as a simple example. Everyone needs the common math: homogeneous coordinates, matrix representations for transformations, quaternion representations for 3D rotations, parametrizations for other flats such as lines in 3-space, and so on. In a graphics course, transform hierarchies in modeling, or clipping and projection transforms for viewing may get more attention. In computer vision, epipolar geometry and the relations among multiple projections may get special treatment. In computational geometry, Plücker coordinates for lines in 3D may be studied to prepare for problems in stabbing and visibility. Yet every one of these topics can be very useful in each of the three disciplines: image-based rendering needs the math of multiview geometry, indexing lightfields requires the geometry of lines in space, and clipping is an essential geometric computation problem. The *Visual Computing* book manages to cover all three points of view in a coherent yet concise way, to the benefit of all sides.

I have taught an algorithms course for over a decade now and topics like dictionaries and priority queues are its bread-and-butter. It was truly refreshing to see these same topics introduced early in this book, but with novel effective examples all motivated by visual computing. The same happens later on with randomization, a topic that
has been seriously studied across all these communities, but with different emphases. This text truly establishes bridges where they will make the most impact: early on in a student’s education. I can see this book being used for a separate integrated course of its own, or as a supplement to existing courses and other texts covering algorithms, computational geometry, computer graphics, or computer vision—thus providing a fruitful common ground between what are currently separate offerings.

The book can also benefit graduate students and researchers across all parts of computer science that deal with modeling or interacting with the physical world. The material is methodically organized, the exposition is rigorous yet well-motivated with plenty of instructive examples. Major techniques and algorithms are given in actual C++ code, and these programs and additional materials are available on a companion Web site. Additional references to the literature are given at the end of each chapter.

Leonidas J. Guibas
Professor of Geometric Computing
Computer Science Department
Stanford University

May 2005
Acknowledgments

I would like to thank my friends, colleagues, and peers who took time to review portions of the manuscript: Paul Agron (State University of New York at Stony Brook, USA), Pierre Alliez (INRIA, France), Hervé Brönnimann (Polytechnic University, USA), Patrick Degener (University of Bonn, Germany), Olivier Devillers (INRIA, France), Richard Nock (DSI-GRIMAAG UAG, France), Stéphane Nullans (Realviz, France), Shigeru Owada (University of Tokyo/Sony Computer Science Laboratories Inc., Japan), Sylvain Pion (INRIA, France), Mariette Yvinec (INRIA, France) and Imad Zoghlami (Siemens Corporate Research, USA).

I am particularly indebted to my colleagues Dr. Shigeru Owada who got the very early draft, and Pr. Richard Nock who was my first alpha release reader.

I would like to heartfully thank the following people and organizations for their kind image contributions: Bart Adams (KU Leuven, Belgium), Sue Addleman (Cyberware, USA), Pravin Bhat (Washington University, USA), Pierre Alliez (INRIA, France), Yuji Ayatsuka (Sony Computer Science Laboratories Inc., Japan), Patrick Coleman (University of Toronto, Canada), Patrick Degener (University of Bonn, Germany), Greg Downing (Consultant, USA), Heather Gentner (Stanford University, USA), David Xianfeng Gu (State University of New York at Stony Brook, USA), Stéphane Guy (INRIA, France), Sayed Ibrahim Hashimi (University of Florida, USA), Stephen Ingram (Emory University, USA), SpherOnVR (Thomas Köllig, Germany), Bruno Lévy (INRIA, France), Shengyou Lin (Zhejiang University, China), Stefan Maierhofer (Vienna University of Technology, Austria), Martin Marinov (RWTH Aachen, Germany), Pascal Müller (ETH Zurich, Switzerland), Stéphane Nullans (Realviz, France), Konrad Polthier (Zuse Institute Berlin, Germany), Christian Rochefort (InSpeck, Canada), Steven M. Seitz (University of Washington, USA), Junichi Rekimoto (Sony Computer Science Laboratories Inc., Japan), Greg Turk (Georgia Institute of Technology, USA), Emma Wixey (Vicon Motion Systems, USA) and Jingyi Yu (MIT, USA).

Unfortunately, sometimes for copyright reasons, I could not use some images I received. Nevertheless, I would like to acknowledge Sumanta Pattanaik (University of Central Florida, USA), Paul Rademacher (Dreamworks Animation, USA), and Szymon Rusinkiewicz (Princeton University, USA) for kindly providing me with such
material. I also thank Professor Marc Levoy (CS, Stanford University) for letting me use the 3D models (mostly the bunny model) of the Stanford 3D scanning repository models in the illustrations.

I am deeply honored that Professor Leonidas J. Guibas has kindly accepted to write the foreword.

This book became possible thanks to the warm encouragements and supports from Sony Computer Science Laboratories, Inc. I express my gratitude to Dr. Mario Tokoro and Dr. Hiroaki Kitano, as well as all other members of Sony Computer Science Laboratories Inc. (as known as Sony CSL for short). Last but not least, during this book project I particularly enjoyed lunch breaks with my colleague Dr. Yosuke Tamura who showed much interest in my writing progress!

Further, I would like to extend my thanks to Jenifer Niles who was my principal contact at Charles River Media. Jenifer with her team (Bryan Davidson, Lance Morganelli, Meg Dunkerley, and the copyeditor) managed to get the book ready much before I anticipated!

I apologize for any (involuntary) name omission and remaining errors. As Blaise Pascal said,¹ I have written you a thick book because I did not have time to write a short one.

Finally, my deepest thanks and love go to my family for their everlasting support.

Tokyo, Japan

F. N.

May 2005.

¹”I have written you a long letter because I did not have time to write a short one.”
## Notational Conventions

The following is a brief review of the mathematical notations used throughout the book.

### Scalar Numbers & Ranges:

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a, \alpha, \lambda, u_x$</td>
<td>Scalar values (e.g., 3, e, π)</td>
</tr>
<tr>
<td>$\theta, \phi$</td>
<td>Angular values (e.g., $\frac{\pi}{2}, 120$ deg)</td>
</tr>
<tr>
<td>$v_x, v_y, v_z$ or $v_1, v_2, v_3$</td>
<td>Coordinates of 3D vector $\mathbf{v}$</td>
</tr>
<tr>
<td>$m_{1,1}, \ldots, m_{i,j}, \ldots, m_{n,m}$</td>
<td>Coefficients of matrix $\mathbf{M} = [m_{i,j}]_{n,m}$</td>
</tr>
<tr>
<td>$(x_1, \ldots, x_k)$</td>
<td>A $k$-tuple (for $k = 2$, a pair; for $k = 3$, a triple)</td>
</tr>
<tr>
<td>$[a, b]$</td>
<td>Closed real interval ${x \mid a \leq x \leq b}$</td>
</tr>
<tr>
<td>$(a, b)$</td>
<td>Open real interval ${x \mid a &lt; x &lt; b}$ (e.g., $(-\infty, \infty)$)</td>
</tr>
<tr>
<td>$[n]$</td>
<td>Integer set ${0, 1, \ldots, n}$</td>
</tr>
<tr>
<td>$[n]^*$</td>
<td>Integer set $[n] \setminus {0} = {1, \ldots, n}$</td>
</tr>
<tr>
<td>$([a, b], [a, b])$</td>
<td>Integer set closed at one end and open at the other: $([a, b]) = {a, \ldots, b - 1}$</td>
</tr>
</tbody>
</table>

### Vectors:

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{OP}$</td>
<td>Coordinate-free vector defined by points $O$ and $P$</td>
</tr>
<tr>
<td>$\mathbf{v}$</td>
<td>Vector notation in some given coordinate system</td>
</tr>
<tr>
<td>$\mathbf{v}^T$</td>
<td>Transpose vector</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>$d_2(\mathbf{p}, \mathbf{q})$</td>
<td>Distance between $\mathbf{p}$ and $\mathbf{q}$, length $</td>
</tr>
<tr>
<td>$\mathbf{e}_1, \ldots, \mathbf{e}_d$</td>
<td>Unit vector basis of a $d$-dimensional vector space (or eigenvectors)</td>
</tr>
</tbody>
</table>

### 2D Vectors:

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{p} = [p_x \ p_y]^T$</td>
<td>2D Inhomogeneous vector</td>
</tr>
<tr>
<td>$\mathbf{p} = [p_x \ p_y \ p_w]^T$</td>
<td>2D Homogeneous vector, often obtained by appending to $\mathbf{p}$ the $w$-coordinate set to 1 ($\mathbf{p} = [\mathbf{p} \ 1]^T$), except for ideal points ($w = 0$)</td>
</tr>
</tbody>
</table>

### 3D Vectors:

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{s} = [s_x \ s_y \ s_z]^T$</td>
<td>3D Inhomogeneous vector (&quot;s&quot; for space)</td>
</tr>
<tr>
<td>$\mathbf{s}$</td>
<td>3D Homogeneous vector, often obtained by</td>
</tr>
</tbody>
</table>
appending to \( s \) the \( w \)-coordinate set to 1 (\( s = [s \ 1]^T \)),
except for ideal points (\( w = 0 \))

\[ v^\perp \quad \text{Arbitrary perpendicular vector to } v \text{ of same magnitude} \]

\[ v_1 \cdot v_2 \quad \text{Dot or inner product.} \]
(equivalent to vector multiplication \( v_1^T v_2 = v_2^T v_1 \);
\( v_1 \cdot v_2 = ||v_1|| ||v_2|| \cos \theta \), where \( \theta \) is the angle made by
vectors \( v_1 \) and \( v_2 \))

\[ v_1 \times v_2 \quad \text{Cross product: } ||v_1 \times v_2|| = ||v_1|| ||v_2|| \sin \theta \]

Arrays & Matrices:

| \( S, I \) | 1D or 2D array of elements |
| \( S[k] \): the \( k \)th element of \( S \) |
| \( I[x, y] \): the pixel of \((x, y)\) coordinates |
| \( (or I[y, x] \text{ depending on the indexing context) } \) |
| \( M \) | Matrix or image notation |
| \( M[y, x] \): Matrix coefficient at \( y \)th row and \( x \)th column |
| \( M[y, x] = m_{y,x} \) |
| \( M^T \) | Transpose matrix |
| \( \mathbf{I}, \mathbf{I}_d \) | Identity \((d \times d)\) square matrix |
| \( \det M = |M| \) | Determinant of matrix \( M \) (zero if singular) |
| \( [u]_x = M \) | Skew antisymmetric matrix \((M^T = -M)\) defined as follows: |
\[
[u]_x = \begin{bmatrix}
0 & -u_z & u_y \\
 u_z & 0 & -u_x \\
-u_y & u_x & 0 \\
\end{bmatrix}
\]
and \( u \times v = [u]_x \times v \)

| \( \text{rank}(M) \) | Rank of matrix \( M \) |
| \( \text{trace}(M) \) | Trace of matrix \( M \) (sum of its diagonal elements: \( \sum_i m_{ii} \)) |
| \( \text{adj}(M) \) | Adjoint of matrix \( M \) (elements are signed cofactors) |
| \( \mathbf{R}, \mathbf{P}, \mathbf{T} \) | Inhomogeneous matrices |
| \( \mathbf{R}, \mathbf{P}, \mathbf{T} \) | Homogeneous matrices |

Quaternions:

| \( \mathbf{q} \) | Quaternion \( \mathbf{q} = [s \ v]^T \) (unit quaternion \( \mathbf{q} = \cos \theta + u \sin \theta \)) |
| \( \bar{\mathbf{q}} \) | Quaternion conjugate \( \bar{\mathbf{q}} = [s - v]^T \) |
| \( \hat{\mathbf{q}} \hat{\mathbf{q}}_2 \) | Quaternion multiplication: |
\[
\hat{\mathbf{q}}_1 \hat{\mathbf{q}}_2 = [s_1 s_2 - u_1 u_2 \ u_1 \times u_2 + s_1 v_2 + s_2 v_1]^T \\
\]
| \( ||\mathbf{q}|| \) | Quaternion norm \( \sqrt{\frac{q_0}{q_0}} = \sqrt{s^2 + ||u||^2} \) |
| \( \mathbf{q}^{-1} \) | Quaternion inverse \( \frac{\mathbf{q}}{||\mathbf{q}||} \) |
| \( \mathbf{q}^\lambda \) | Quaternion power: \( \mathbf{q}^\lambda = (\exp(\theta u))^\lambda = \cos \lambda \theta + (\sin \lambda \theta) u \) |
| \( \log \mathbf{q} \) | Quaternion logarithm: \( \log \mathbf{q} = \log \exp(\theta u) = \theta u \) |
Notational Conventions

Space:
- \( \mathbb{R}^2, \mathbb{R}^3, \mathbb{R}^d \): 2D, 3D, and dD Euclidean spaces
- \( \mathbb{P}^2, \mathbb{P}^3, \mathbb{P}^d \): 2D, 3D, and dD projective spaces
- \( \text{SO}^3 \): Space of 3D rotations
- \( \mathbb{C}^2 \): Space of complex numbers \( \mathbb{C}^2 = \{ a + ib \mid (a, b) \in \mathbb{R}^2 \} \)
- \( \mathbb{Q} \): Set of quaternions \( \mathbb{Q} = \{ a + bi + cj + dk \mid (a, b, c, d) \in \mathbb{R}^4 \} \)
- \( \mathbb{A} \): Affine space

Functions & Partial Derivative Operators:
- \( n! \): Factorial \( n! = 1 \times 2 \times ... \times n \)
- \( \binom{n}{k} \): Binomial coefficient \( \frac{n!}{k!(n-k)!} \)
- \(| \cdot |\): Absolute value
- \( \lfloor \cdot \rfloor \): Floor of \( a \) (e.g., \( \lfloor 2.312 \rfloor = 2 \))
- \( \lceil \cdot \rceil \): Ceiling of \( a \) (e.g., \( \lceil 2.312 \rceil = 3 \))
- \( \text{sign}(a) \): Sign of \( a \) (e.g., \( \text{sign}(-3) = -1, \text{sign}(5) = 1, \text{sign}(0) = 0 \))

Dirac function \( \delta(x, y) = \begin{cases} 1 & \text{if and only if } x = 0 \text{ and } y = 0, \\ 0 & \text{otherwise}. \end{cases} \) (Krönecker symbol)

Function \( f(x_1, x_2, ..., x_k) \): Function \( f \) of \( k \) variables \( x_1, ..., x_k \)

Gradient \( \nabla f \)

\[ \nabla f = \left[ \frac{\partial f}{\partial x_1} \ldots \frac{\partial f}{\partial x_d} \right]^T \]

Laplacian \( \Delta f \)

\[ \Delta f = \frac{\partial^2 f}{\partial x_1^2} + \cdots + \frac{\partial^2 f}{\partial x_d^2} \]

Jacobian \( \mathbf{J} f \)

\[ f : \mathbb{R}^d \rightarrow \mathbb{R}^n, \mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_d} \end{bmatrix} \]

Hessian \( \mathbf{H} f \)

\[ f : \mathbb{R}^d \rightarrow \mathbb{R}^d, \mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d x_1} & \cdots & \frac{\partial^2 f}{\partial x_d x_1} \end{bmatrix} \]

\( (\mathbf{H} f = \mathbf{J} \nabla f) \)
Geometric entities:

- $P$  
  Point $P$
- $S$  
  Line segment $S$
- $O$  
  Object $O$
- $H$  
  Plane $H$
- $H : n^T v = 0$  
  Plane $H$ defined by homogeneous equation:
  
  $n_1 v_1 + n_2 v_2 + n_3 v_3 + n_4 v_4 = 0$
- $\text{Ball}(B,r)$  
  Ball of circumcenter $B$ and radius $r$
- $\triangle PQR$  
  Triangle with vertices $P$, $Q$, and $R$
- $\mathcal{S} = \{S_1, ..., S_n\}$  
  Set of objects
- $\text{aff}(\mathcal{S}) / \text{conv}(\mathcal{S})$  
  Affine/convex space defined by set $\mathcal{S}$

Probabilities, Statistics, and Inequalites:

- $\Pr[\text{Event}]$  
  Probability of event Event
- $\hat{S}$  
  Random variable
- $\mathbb{E}(\hat{S})$  
  Expectation of random variable $\hat{S}$
- $\text{Var}(\hat{S})$  
  Variance of random variable $\hat{S}$
- $\sigma(\hat{S})$  
  Standard deviation of random variable $\hat{S}$
  
  $(\sigma(\hat{S}) = \sqrt{\text{Var}(\hat{S})})$
- $\hat{X}_E = I(E)$  
  Indicator random variable of event $E$
- $\Pr[\hat{S} \geq c] \leq \frac{\mathbb{E}(\hat{S})}{c}$  
  Markov inequality
  
  (non-negative discrete random variable)

Data Structures:

- $Q$  
  Data structure $Q$ (e.g., a queue)
- $\text{S.procedure()}$  
  Call function procedure without argument on $S$
- $\text{S.procedure(}\mathcal{P}\text{)}$  
  Call function procedure with argument $\mathcal{P}$ on $S$
- $P \leftarrow \emptyset$  
  Initialize an empty data structure $P$

Asymptotic Complexity:

- $O(f)$  
  Upper bound
- $\Omega(f)$  
  Lower bound
- $\Theta(f)$  
  Optimal bound
- $o(f)$  
  Nonasymptotically tight upper bound
- $\tilde{O}(f)$  
  Expected running time of randomized algorithms
- $\bar{O}()$  
  Average running time of deterministic algorithms
Notational Conventions

Radiometric Quantities and Units:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radiance</td>
<td>$\frac{W}{sr.m^2}$</td>
<td>Watt per steradian per meter square</td>
</tr>
<tr>
<td>Luminance</td>
<td>$\frac{cd}{m^2}$</td>
<td>Candela per meter square</td>
</tr>
<tr>
<td>Irradiance</td>
<td>$\frac{W}{m}$</td>
<td>Watt per meter</td>
</tr>
<tr>
<td>Illuminance</td>
<td>$lx$</td>
<td>Lux</td>
</tr>
</tbody>
</table>

In this book, we either present algorithms in C++-style language or at a higher-level pseudocode abstraction style. The argmin and argmax notations are often used in those algorithms. In practice, in a C++-style, we often compute both the min and argmin, or max and argmax. Let us consider a $d$-dimensional vector $\mathbf{v}$. We mathematically write the maximum of its coordinate as $m = \max_{i=1}^{d} v_i$. The maximum coordinate is obtained from the $k$th principal axis, with $k = \arg \max_{i=1}^{d} v_i$. Thus, we have the following max/argmax relationship: $v_{\text{argmax}} = \max$.

Here is the C++ code for performing those max and argmax operations:

```cpp
class vector { public:
    int d; double *x;
    // create a random vector
    vector(int dim) { d=dim; x=new double[d];
        for(int i=0;i<d;i++)
            {x[i]=rand()/(double)RAND_MAX;}
    }
    // delete a vector ~vector() {delete [] x;}
    // maximum coordinate value
    double max() { double value=x[0];
        for(int i=1;i<d;i++)
            if (x[i]>value) value=x[i];
        return value; }
    // coordinate axis that has
    // maximum value
    int argmax() { int axis=0;
        for(int i=1;i<d;i++)
            if (x[i]>x[axis]) axis=i;
        return axis; }
};
```
Colophon

This book has been typeset using \LaTeX{} system on Microsoft\textsuperscript{®} Windows\textsuperscript{®} operating system. A special \LaTeX{} class has been written to fit the typesetting rules of Charles River Media. Most of the figures have been prepared with the Ipe extensible drawing editor environment of Professor Otfried Cheong (available online at http://ipe.compgeom.org/). The book project has been managed using the excellent TeXnicCenter tool (http://www.toolscenter.org/) that provides a visual integrated environment for editing and compiling a full \LaTeX{} book project made of many \LaTeX{} files. I used XnView image browsing software (http://www.xnview.com/) to capture screenshots and extract frames from movies. XnView was also instrumental for the various image conversions I needed (bmp, png, eps, etc.). The Internet was a much appreciable resource for finding related materials and references. I can hardly imagine how I could have typeset this book without this Web information. I used the ACM (http://portal.acm.org/) and IEEE (http://www.computer.org/publications/dlib/) digital libraries for retrieving original papers, and sometimes their \BibTeX{} labels.
The cover page image was created by compositing images of a photogrammetric set of a 3D reconstruction of the Vatican library (no religious meaning):

Those images are courtesy of © Stéphane Nullans (France). Used with permission.

There is quite a lot of famous data in visual computing that is constantly used as a test bed for evaluating and comparing algorithms. We succinctly present the hall of fame on the next page.
The 512 × 512 24-bit lena picture and narrative story on how it became a de facto test image in the image processing and computer vision communities can be read online at http://www.lenna.org
Warning: the full picture contains nudity.

The Utah 3D teapot has traditionally been used in computer graphics. There is even a function named glutWireTeapot in the glut API to draw it. The Utah teapot history, including the descriptions of the original one that differs from the glut version, is summarized online, at http://sjbaker.org/teapot/.

The 3D bunny model of the computer graphics laboratory of Stanford’s university is often used for testing mesh algorithms. The 69,451-triangle bunny was acquired from a real-world model in 1993. The bunny history can be enjoyed online at either http://graphics.stanford.edu/data/3Dscanrep/, or http://www.gvu.gatech.edu/people/faculty/greg.turk/bunny/bunny.html.

The Cornell box is a traditional data set used for computing radiosity or simulating global illumination algorithms in computer graphics. The Cornell box history, data set, and experiments (including comparisons with the measured real-world box) are further described online at http://www.graphics.cornell.edu/online/box/. PovRay (http://povray.org) provides a script file cornell.pov to render your own radiosity box image.
Main Conferences and Publications in Visual Computing:
(List is nonexhaustive)

Main conferences:
- Annual Conf. Comp. Graphics and Interactive Techniques, ACM SIGGRAPH
- Annual Conference on Human Factors in Computing Systems, ACM CHI
- Biannual Symposium on Interactive 3D Graphics and Games, ACM I3D
- Annual Conference on Vision and Pattern Recognition, IEEE CVPR
- Annual International Conference on Computer Vision, IEEE ICCV
- Biannual European Conference on Computer Vision, IEEE ECCV
- Biannual Conference on Pattern Recognition, IAPR ICPR
- Annual Computer Graphics International, CGS CGI
- Annual Eurographics, European Association for Computer Graphics
- Annual International Conference on 3D Web Technology, ACM WEB3D
- Annual Conference on Point-Based Graphics, IEEE/Eurographics
- Annual Symposium on Geometry Processing, Eurographics SGP
- Annual Canadian Conference on Graphics Interface, GI
- Annual Symposium on Computational Geometry, ACM SoCG
- Annual Symposium on Solid and Physical Modeling, ACM SPM
- Annual Symposium on Information Visualization, IEEE IV
- Annual Canadian Conference on Computational Geometry, CCCG
- Annual Conference on Smart Graphics, SG
- Annual Game Developers Conference, GDC
- Annual Pacific Conference on Computer Graphics and Applications, PG
- Annual Conference on Voronoi Diagrams in Science and Engineering, VD

Main journal publications:
- Transactions on Computer Graphics, ACM TOG
- Transactions on Pattern Analysis and Machine Intelligence, IEEE TPAMI
- Transactions on Visualization and Computer Graphics, IEEE TVCG
- Discrete & Computational Geometry, Springer-Verlag DCG
- The Visual Computer, Springer-Verlag VCJ
- Computational Geometry: Theory and Its Applications, Elsevier CGTA
- Transactions on Applied Perception, ACM TAP
- Graphical Models and Image Processing, Elsevier GMIP
- Computer Graphics and Applications, IEEE CGA
Colophon

Major tradeshows:
- Annual International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH
- Annual Electronic Entertainment Expo, E3
- Annual Game Development Conference, GDC
- Annual National Association of Broadcasters, NAB
Bibliography


URL http://graphics.stanford.edu/papers/mpr/

URL http://www.realtimerendering.com/

URL http://www.inria.fr/rrrt/rr-4966.html

URL http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/


Bibliography


URL http://www-sop.inria.fr/prisme/publis/bdsty-arsol-92.ps.gz

URL http://www.inria.fr/rrrt/rr-5434.html

URL http://www.inria.fr/rrrt/rr-3270.html

URL http://www.cs.unc.edu/~snoeyink/papers/papers.html


URL http://www.inria.fr/rrrt/rr-1140.html

URL http://www-sop.inria.fr/geometrica/team/Mariette.Yvinec/livre.html

URL http://www.intel.com/technology/techresearch/people/bios/bouguet_j.htm

URL http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

URL http://www-star.stanford.edu/starlab_web_20030912/people/bracewell.html

URL http://isg.cs.tcd.ie/spheretree/


[65] Burt PJ and Adelson EH (1983) A Multiresolution Spline with Application to Image Mosaics. ACM Trans Graph (TOG) 2(4), 217–236. ISSN 0730-0301. DOI:
Bibliography

10.1145/245.247
URL http://web.mit.edu/persci/people/adelson/publications.html

URL http://math.ucsd.edu/~sbuss/MathCG/index.html

URL http://math.ucsd.edu/~sbuss/ResearchWeb/spheremean/


URL http://graphics.cs.cmu.edu/~pleno/papers/ps_projectpage.htm


URL http://www.cs.princeton.edu/~chazelle/

URL http://www.cs.princeton.edu/~chazelle/


URL http://www.cs.princeton.edu/~chazelle/


Bibliography

URL http://accad.osu.edu/~waynec/history/ACCAD-overview/overview3.html

URL http://db.uwaterloo.ca/~imunro/

URL http://www.cs.uu.nl/geobook/

DOI:10.1145/344779.344855
URL http://www.debevec.org/Research/LS/

DOI:10.1145/258734.258884
URL http://www.debevec.org/Research/HDR/

URL http://www.debevec.org/HDRI2004/

URL http://www.stat.harvard.edu/

URL http://www.pixar.com/

URL http://www.inria.fr/rrrt/rr-3451.html

URL http://www.inria.fr/rrrt/rr-4351.html

DOI: 10.1145/378583.378643
URL http://www.inria.fr/rrrt/rr-4120.html


Bibliography

  URL http://www.cis.rit.edu/people/faculty/fairchild/CAM.html

  URL http://www-sop.inria.fr/robotvis/personnel/faugeras/faugeras-eng.html

  URL http://www.lirmm.fr/~fiorio/

  URL http://www.ti.inf.ethz.ch/ew/courses/ApproxGeom05/paper/fischer_gaertner_03.pdf

  URL http://www.ai.sri.com/people/fischler/

  URL http://seattleweb.intel-research.net/people/fishkin/index.html

  URL http://heim.ifi.uio.no/~michaelf/papers/papers.html

  URL http://heim.ifi.uio.no/~michaelf/papers/surfparam.pdf

  URL http://www.cc.gatech.edu/fac/ Jim.Foley/foley.html

[120] Fontijne D and Dorst L (2003) **Modeling 3D Euclidean Geometry**. IEEE Comput Graph Appl **23**(2), 68–78. ISSN 0272-1716. DOI:10.1109/MCG.2003.1185582
  URL http://staff.science.uva.nl/~leo/clifford/CGA3.pdf
URL http://bailey.cs.uop.edu/fordtopp/datastruct.html


URL http://cm.bell-labs.com/who/sjf/

URL http://www.dcis.rutgers.edu/cs/people/

URL http://www.fftw.org/~athena/papers.html

URL http://www.cs.unc.edu/~fuchs/publications/PreDetermVis(PrelimRep)79.pdf

URL http://www.cs.unc.edu/~fuchs/publications/

URL http://www.dgp.toronto.edu/~funge/ai4games/

URL http://www.cs.colorado.edu/~hal/


URL http://graphics.cs.uiuc.edu/~garland/research/quadrics.html
Bibliography

URL http://graphics.cs.uiuc.edu/~garland/research/quadrics.html


URL http://www.math.ucla.edu/~gilboa/pub/PAMI_cmplx04_GSZ.pdf

DOI:10.1145/103162.103163
URL http://docs.sun.com/source/806-3568/ncg_goldberg.html


URL http://sccm.stanford.edu/faculty/nf-golub.html

URL http://cpp.datastructures.net/


DOI:10.1145/237170.237200
URL http://www.research.microsoft.com/~cohen/lumia.ps.gz

DOI:10.1145/882262.882276
URL http://www.cs.technion.ac.il/~sheffi/papers/SigCDV.pdf

URL http://www.debevec.org/ReflectionMapping/
URL http://mgrewal.ecs.fullerton.edu/

URL http://www.cs.cornell.edu/Info/People/gries/gries.html

URL http://graphics.idav.ucdavis.edu/publications/print?pub_id=448


URL http://www1.cs.columbia.edu/~mdog/pubs.htm

URL http://www.intel.com/technology/techresearch/people/bios/grzeszczuk_r.htm


URL http://geometry.stanford.edu/member/guibas/

URL http://www.dcc.unicamp.br/~stolfi/
URL http://www.eecs.umich.edu/~guskov/hm-coarse.pdf


URL http://www.multires.caltech.edu/~iwguskov/tunnels.pdf

URL http://valis.cs.uuc.edu/~sariel/papers/03/lloyd_kmeans/kmeans.pdf

URL http://www.math.technion.ac.il/~rl/docs/uniform.pdf


[160] Heckbert PS (1986) **Survey of Texture Mapping.** *IEEE Comput Graph Appl (CGA)* 6(11), 56–67. ISSN 0272-1716


URL http://mrl.nyu.edu/projects/image-analogies/

URL http://research.microsoft.com/~thoare/

URL http://research.microsoft.com/~thoare/
URL http://research.microsoft.com/~thoare/

URL http://www1.bell-labs.com/user/qbi/

URL http://www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/siggraph99.pdf

URL http://www.intel.com/research/mrl/research/opencv/


URL http://www.cs.unc.edu/~isenburg/oocc/

URL http://graphics.cs.cmu.edu/projects/bdtree/

URL http://graphics.ucsd.edu/~henrik/papers/book/

URL http://dcl.ee.washington.edu/papers/README.html


URL http://research.microsoft.com/users/kajiya/
URL http://www.cs.unc.edu/~welch/kalman/


URL http://www.cs.umd.edu/~mount/Projects/KMeans/

URL http://cs.nyu.edu/chenli/papers/core.pdf

URL http://www-math.cs.uni-saarland.de/~kettner/pub/nonrobust_esa_04_a.html

URL http://www-cs-faculty.stanford.edu/~knuth/taocp.html

URL http://www-cs-faculty.stanford.edu/~knuth/taocp.html


URL http://graphics.stanford.edu/papers/camera/

URL http://www.cse.iitd.ernet.in

URL http://www.cc.gatech.edu/cpl/projects/graphcuttextures/


Bibliography

URL http://www.cad.zju.edu.cn/home/lsy/

URL http://www.nada.kth.se/~tony/

URL http://www.gvu.gatech.edu/people/peter.lindstrom/papers/siggraph2000/

URL http://www.gvu.gatech.edu/people/peter.lindstrom/papers/i3d2003/


URL http://www.crd.ge.com/~lorensen/

URL http://research.microsoft.com/~hoppe/geomclipmap.pdf

URL research.microsoft.com/~hoppe/sgim.pdf


Bibliography


URL http://www.mines.edu/~dmehta/

URL http://www.rug.nl/rc/hpcv/people/arnold/index


URL http://web.media.mit.edu/~minsky/

URL http://www.acm.org/jgt/papers/MollerTrumbore97/

URL http://theory.stanford.edu/~rajean/

URL http://www.cs.umd.edu/~mount/Papers/ALMS.ps

URL http://www.cse.iitb.ac.in/~ketan/

URL http://www.engin.brown.edu/~mundy/

URL http://math.mit.edu/people/faculty/munkres.html

URL http://math.mit.edu/people/faculty/munkres.html

URL http://db.uwaterloo.ca/~imunro/

URL http://athos.rutgers.edu/~muthu/stream-1-1.ps

URL http://www.cantrip.org/traits.html


URL http://www.csl.sony.co.jp/person/nielsen/

URL http://www.csl.sony.co.jp/person/nielsen/

URL http://www.csl.sony.co.jp/person/nielsen/

URL http://www.csl.sony.co.jp/person/nielsen/
URL http://www.csl.sony.co.jp/person/nielsen


URL http://www.siam.org/meetings/sdm04/proceedings/

URL http://www.univ-ag.fr/~rnock/

URL http://www-ece.rice.edu/ece/faculty/Orchard.html

URL http://maven.smith.edu/~orourke/books/compgeom.html

URL http://www.faqs.org/faqs/graphics/algorithms-faq/

URL http://www.math.ucla.edu/~sjo/

URL http://www.iro.umontreal.ca/~ostrom/ImportanceSampling/


URL http://www-ui.is.s.u-tokyo.ac.jp/~o/VolumetricIllustration/

the IEEE Service Center, Piscataway, NJ, USA.
URL http://grouper.ieee.org/groups/754/

[260] Paterson MS and Yao FF (1990) Efficient Binary Space Partitions for Hidden-
surface Removal and Solid Modeling. Discrete and Computational Geometry
(DCG) 5, 485–503
URL http://www.dcs.warwick.ac.uk/people/academic/Mike.Paterson/

Graphics and Applications (CGA) 24(4), 22–23
URL http://www.merl.com/people/pfister/

012553180X
URL http://pbrt.org/

URL http://www.csl.sony.co.jp/person/nielsen/PT/hypermask/hypermask.html


IEEE Computer Vision and Pattern Recognition (CVPR)

URL http://keithp.com/~keith/porterduff/

and Visualization. Springer. ISBN 3-540-42058-4
URL http://www.geometrie.tuwien.ac.at/pottmann/

for CG, HDTV, and D-Cinema. Course Notes #2, SIGGRAPH
URL http://www.poynton.com/

Trans Graph (TOG) 22(3), 340–349. ISSN 0730-0301. DOI:10.1145/882262.882274

[270] Preparata FP and Hong SJ (1977) Convex Hulls of Finite Sets of Points in Two
DOI:10.1145/359423.359430
URL http://www.cs.brown.edu/people/franco/
URL http://www.cs.brown.edu/people/franco/

URL http://www.nr.com/

URL http://www.cs.umd.edu/~pugh/

URL http://www.cs.umd.edu/~pugh/

URL http://www.cs.unc.edu/~rademach/mcop98.html


URL http://www.cs.ucf.edu/~reinhard/

URL http://web. engr. oregonstate. edu/~saurabh/research/info-vis.pdf


URL http://www.csl.sony.co.jp/person/rekimoto/navi.html


Bibliography


URL shoemake@graphics.cis.upenn.edu


URL http://www.boost.org/

URL http://silvela.org/jaime/BFSpaper.pdf

URL http://www.cns.nyu.edu/~eero/steerpyr/

URL http://www.graphicsinterface.org/

URL http://www.cs.sunysb.edu/~skiena/

URL http://alvyray.com/

URL http://alvyray.com/


URL http://www.dcc.unicamp.br/~stolfi/
Bibliography


URL http://www.cs.dartmouth.edu/~scot/

URL http://www.simplex.t.u-tokyo.ac.jp/~sugihara/

URL http://www.cse.cuhk.edu.hk/~leojia/all_final_papers/mattinig.siggraph04.pdf


URL http://research.microsoft.com/~szeliski/publications.htm

URL http://www.cs.princeton.edu/~ret/

URL http://www.cs.princeton.edu/~ret/

URL http://www.cs.princeton.edu/~ret/

URL http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf

URL http://www.dgp.toronto.edu/people/tu/papers/sig94.ps

URL http://www.cs.umd.edu/~varshney/

URL http://graphics.stanford.edu/papers/veach/

URL http://www.csd.uwo.ca/faculty/olga/Papers/cvpr03-a.pdf

URL http://www.worldserver.com/turk/computergraphics/FixedPointTrigonometry.pdf


URL http://www.shef.ac.uk/dcs/research/graphics

URL http://www.shef.ac.uk/dcs/research/graphics

URL http://graphics.stanford.edu/papers/texture-synthesis-sketch-sig03/

URL http://graphics.stanford.edu/papers/texture-synthesis-sig00/

URL http://graphics.stanford.edu/papers/texture-synthesis-sig01/

for CAD Applications pp. 3–36
URL http://www.cs.rpi.edu/

URL http://www.cs.fiu.edu/~weiss/

URL http://www.inf.ethz.ch/personal/emo/

URL http://www.math.tau.ac.il/~michas/

URL http://encyclopedia.thefreedictionary.com/Lancelliams

URL http://www-cs.engr.ccny.cuny.edu/~wolberg/diw.html

URL http://www.math.tau.ac.il/~wolfson/


URL http://www.cv.cs.ritsumei.ac.jp/~xu/index.html

URL http://www.cs.princeton.edu/~yao/


URL http://web.mit.edu/kevinz/www/


URL http://www.i2r.a-star.edu.sg/

URL http://research.microsoft.com/~zhang/calib/

URL http://research.microsoft.com/~zhao/wsnbook.html
