

# VISIBILITY ALGORITHMS IN THE PLANE

SUBIR KUMAR GHOSH

*School of Computer Science, Tata Institute of Fundamental Research,  
Mumbai 400005, India*



**CAMBRIDGE**  
UNIVERSITY PRESS

# Contents

<b>Preface</b>	<i>page</i> xi
<b>1 Background</b>	1
1.1 Notion of Visibility	1
1.2 Polygon	2
1.3 Asymptotic Complexity	5
1.4 Triangulation	6
1.5 The Art Gallery Problem	8
1.6 Special Types of Visibility	11
<b>2 Point Visibility</b>	13
2.1 Problems and Results	13
2.2 Computing Visibility of a Point in Simple Polygons	16
2.2.1 Non-Winding Polygon: $O(n)$ Algorithm	16
2.2.2 Removing Winding: $O(n)$ Algorithm	23
2.3 Computing Visibility of a Point in Polygons with Holes	31
2.4 Recognizing Simple Polygons Visible from a Point	38
2.5 Notes and Comments	43
<b>3 Weak Visibility and Shortest Paths</b>	46
3.1 Problems and Results	46
3.2 Characterizing Weak Visibility	51
3.3 Computing Weak Visibility in Simple Polygons	58
3.3.1 Scanning the Boundary: $O(n \log n)$ Algorithm	58
3.3.2 Using Shortest Path Trees: $O(n)$ Algorithm	65

3.4	Computing Weak Visibility in Polygons with Holes	66
3.5	Recognizing Weakly Internal Visible Polygons	68
3.5.1	Using Visibility Graph: $O(E)$ Algorithm	68
3.5.2	Scanning the Boundary: $O(n)$ Algorithm	73
3.6	Computing Shortest Path Trees	82
3.6.1	In Simple Polygons: $O(n)$ Algorithm	82
3.6.2	In Weak Visibility Polygons: $O(n)$ Algorithm	87
3.7	Recognizing Weakly External Visible Polygons	95
3.8	Notes and Comments	102
<b>4</b>	<b><i>LR</i>-Visibility and Shortest Paths</b>	<b>105</b>
4.1	Problems and Results	105
4.2	Characterizing <i>LR</i> -Visibility	108
4.3	Computing <i>LR</i> -Visibility Polygons	110
4.4	Recognizing <i>LR</i> -Visibility Polygons	113
4.5	Walking in an <i>LR</i> -Visibility Polygon	115
4.6	Computing Shortest Path Trees using <i>LR</i> -Visibility	124
4.7	Notes and Comments	135
<b>5</b>	<b>Visibility Graphs</b>	<b>136</b>
5.1	Problems and Results	136
5.2	Computing Visibility Graphs of Simple Polygons	138
5.3	Computing Visibility Graphs of Polygons with Holes	143
5.3.1	Worst-Case: $O(n^2)$ Algorithm	143
5.3.2	Output-Sensitive: $O(n \log n + E)$ Algorithm	146
5.4	Computing Tangent Visibility Graphs	161
5.4.1	Convex Holes: $O(n + h^2 \log h)$ Algorithm	161
5.4.2	Non-Convex Holes: $O(n + h^2 \log h)$ Algorithm	165
5.5	Notes and Comments	169
<b>6</b>	<b>Visibility Graph Theory</b>	<b>171</b>
6.1	Problems and Results	171
6.2	Recognizing Visibility Graphs of Simple Polygons	174
6.2.1	Necessary Conditions	174

6.2.2	Testing Necessary Conditions: $O(n^2)$ Algorithm	180
6.3	Characterizing Visibility Graphs of Simple Polygons	183
6.4	Recognizing Special Classes of Visibility Graphs	187
6.4.1	Spiral Polygons: $O(n)$ Algorithm	187
6.4.2	Tower Polygons: $O(n)$ Algorithm	195
6.5	Characterizing a Sub-Class of Segment Visibility Graphs	201
6.6	A Few Properties of Vertex-Edge Visibility Graphs	205
6.7	Computing Maximum Clique in a Visibility Graph	208
6.8	Computing Maximum Hidden Vertex Set in a Visibility Graph	214
6.9	Notes and Comments	216
<b>7</b>	<b>Visibility and Link Paths</b>	<b>218</b>
7.1	Problems and Results	218
7.2	Computing Minimum Link Paths in Simple Polygons	221
7.2.1	Using Weak Visibility: $O(n)$ Algorithm	221
7.2.2	Using Complete Visibility: $O(n)$ Algorithm	224
7.3	Computing Minimum Link Paths in Polygons with Holes	231
7.4	Computing Link Center and Radius of Simple Polygons	238
7.5	Computing Minimum Nested Polygons	242
7.5.1	Between Convex Polygons: $O(n \log k)$ Algorithm	242
7.5.2	Between Non-Convex Polygons: $O(n)$ Algorithm	248
7.6	Notes and Comments	253
<b>8</b>	<b>Visibility and Path Queries</b>	<b>255</b>
8.1	Problems and Results	255
8.2	Ray-Shooting Queries in Simple Polygons	259
8.3	Visibility Polygon Queries for Points in Polygons	267
8.3.1	Without Holes: $O(\log n + k)$ Query Algorithm	267
8.3.2	With Holes: $O(n)$ Query Algorithm	272
8.4	Path Queries Between Points in Simple Polygons	278
8.4.1	Shortest Paths: $O(\log n + k)$ Query Algorithm	278
8.4.2	Link Paths: $O(\log n + k)$ Query Algorithm	289
8.5	Notes and Comments	292

**Bibliography**  
**Index**

*Contents*

295  
311