

# Contents

*Colour plates appear between pages 506 and 507*

<i>Preface</i>	xvi
<i>Acknowledgements</i>	xxi
<b>1 Mathematical fundamentals of computer graphics</b>	<b>1</b>
<b>1.1 Manipulating three-dimensional structures</b>	<b>1</b>
1.1.1 Three-dimensional geometry in computer graphics – affine transformations	2
1.1.2 Transformations for changing coordinate systems	8
<b>1.2 Structure-deforming transformations</b>	<b>9</b>
<b>1.3 Vectors and computer graphics</b>	<b>11</b>
1.3.1 Addition of vectors	12
1.3.2 Length of vectors	12
1.3.3 Normal vectors and cross products	12
1.3.4 Normal vectors and dot products	14
1.3.5 Vectors associated with the normal vector reflection	15
<b>1.4 Rays and computer graphics</b>	<b>17</b>
1.4.1 Ray geometry – intersections	17
1.4.2 Intersections – ray–sphere	18
1.4.3 Intersections – ray–convex polygon	19
1.4.4 Intersections – ray–box	21
1.4.5 Intersections – ray–quadric	23
1.4.6 Ray tracing geometry – reflection and refraction	23
<b>1.5 Interpolating properties in the image plane</b>	<b>25</b>
<b>2 Representation and modelling of three-dimensional objects (1)</b>	<b>27</b>
Introduction	27
<b>2.1 Polygonal representation of three-dimensional objects</b>	<b>33</b>
2.1.1 Creating polygonal objects	37

2.1.2	Manual modelling of polygonal objects	38
2.1.3	Automatic generation of polygonal objects	38
2.1.4	Mathematical generation of polygonal objects	39
2.1.5	Procedural polygon mesh objects – fractal objects	44
2.2	<b>Constructive solid geometry (CSG) representation of objects</b>	46
2.3	<b>Space subdivision techniques for object representation</b>	51
2.3.1	Octrees and polygons	53
2.3.2	BSP trees	55
2.3.3	Creating voxel objects	56
2.4	<b>Representing objects with implicit functions</b>	56
2.5	<b>Scene management and object representation</b>	58
2.5.1	Polygon mesh optimization	59
2.6	<b>Summary</b>	64
<b>3</b>	<b>Representation and modelling of three-dimensional objects (2)</b>	66
	Introduction	66
3.1	<b>Bézier curves</b>	69
3.1.1	Joining Bézier curve segments	75
3.1.2	Summary of Bézier curve properties	77
3.2	<b>B-spline representation</b>	78
3.2.1	B-spline curves	78
3.2.2	Uniform B-splines	80
3.2.3	Non-uniform B-splines	84
3.2.4	Summary of B-spline curve properties	90
3.3	<b>Rational curves</b>	90
3.3.1	Rational Bézier curves	91
3.3.2	NURBS	93
3.4	<b>From curves to surfaces</b>	94
3.4.1	Continuity and Bézier patches	98
3.4.2	A Bézier patch object – the Utah teapot	100
3.5	<b>B-spline surface patches</b>	101
3.6	<b>Modelling or creating patch surfaces</b>	106
3.6.1	Cross-sectional or linear axis design example	107
3.6.2	Control polyhedron design – basic technique	110
3.6.3	Creating patch objects by surface fitting	115
3.7	<b>From patches to objects</b>	121
<b>4</b>	<b>Representation and rendering</b>	123
	Introduction	123
4.1	<b>Rendering polygon meshes – a brief overview</b>	124

4.2	<b>Rendering parametric surfaces</b>	125
4.2.1	Rendering directly from the patch descriptions	125
4.2.2	Patch to polygon conversion	128
4.2.3	Object space subdivision	128
4.2.4	Image space subdivision	135
4.3	<b>Rendering a CSG description</b>	138
4.4	<b>Rendering a voxel description</b>	140
4.5	<b>Rendering implicit functions</b>	141
<b>5</b>	<b>The graphics pipeline (1): geometric operations</b>	142
	Introduction	142
5.1	<b>Coordinate spaces in the graphics pipeline</b>	143
5.1.1	Local or modelling coordinate systems	143
5.1.2	World coordinate systems	143
5.1.3	Camera or eye or view coordinate system	143
5.2	<b>Operations carried out in view space</b>	147
5.2.1	Culling or back-face elimination	147
5.2.2	The view volume	147
5.2.3	Three-dimensional screen space	149
5.2.4	View volume and depth	152
5.3	<b>Advanced viewing systems (PHIGS and GKS)</b>	156
5.3.1	Overview of the PHIGS viewing system	157
5.3.2	The view orientation parameters	159
5.3.3	The view mapping parameters	159
5.3.4	The view plane in more detail	162
5.3.5	Implementing a PHIGS-type viewing system	164
<b>6</b>	<b>The graphics pipeline (2): rendering or algorithmic processes</b>	167
	Introduction	167
6.1	<b>Clipping polygons against the view volume</b>	168
6.2	<b>Shading pixels</b>	171
6.2.1	Local reflection models	173
6.2.2	Local reflection models – practical points	177
6.2.3	Local reflection models – light source considerations	179
6.3	<b>Interpolative shading techniques</b>	179
6.3.1	Interpolative shading techniques – Gouraud shading	180
6.3.2	Interpolative shading techniques – Phong shading	181
6.3.3	Renderer shading options	182
6.3.4	Comparison of Gouraud and Phong shading	183
6.4	<b>Rasterization</b>	183
6.4.1	Rasterizing edges	183
6.4.2	Rasterizing polygons	185



6.5	Order of rendering	187	8.5	Light maps	240
6.6	Hidden surface removal	189	8.6	Environment or reflection mapping	243
6.6.1	The Z-buffer algorithm	189	8.6.1	Cubic mapping	245
6.6.2	Z-buffer and CSG representation	190	8.6.2	Sphere mapping	247
6.6.3	Z-buffer and compositing	191	8.6.3	Environment mapping: comparative points	248
6.6.4	Z-buffer and rendering	192	8.6.4	Surface properties and environment mapping	249
6.6.5	Scan line Z-buffer	193	8.7	Three-dimensional texture domain techniques	251
6.6.6	Spanning hidden surface removal	193	8.7.1	Three-dimensional noise	251
6.6.7	A spanning scan line algorithm	194	8.7.2	Simulating turbulence	252
6.6.8	Z-buffer and complex scenes	196	8.7.3	Three-dimensional texture and animation	254
6.6.9	Z-buffer summary	198	8.7.4	Three-dimensional light maps	256
6.6.10	BSP trees and hidden surface removal	199	8.8	Anti-aliasing and texture mapping	256
6.7	Multi-pass rendering and accumulation buffers	202	8.9	Interactive techniques in texture mapping	260
7	Simulating light-object interaction: local reflection models	205	9	Geometric shadows	263
	Introduction	205		Introduction	263
7.1	Reflection from a perfect surface	206	9.1	Properties of shadows used in computer graphics	265
7.2	Reflection from an imperfect surface	207	9.2	Simple shadows on a ground plane	265
7.3	The bi-directional reflectance distribution function	208	9.3	Shadow algorithms	267
7.4	Diffuse and specular components	211	9.3.1	Shadow algorithms: projecting polygons/scan line	267
7.5	Perfect diffuse – empirically spread specular reflection	212	9.3.2	Shadow algorithms: shadow volumes	268
7.6	Physically based specular reflection	213	9.3.3	Shadow algorithms: derivation of shadow polygons from light source transformations	271
7.6.1	Modelling the micro-geometry of the surface	214	9.3.4	Shadow algorithms: shadow Z-buffer	271
7.6.2	Shadowing and masking effects	214			
7.6.3	Viewing geometry	216	10	Global illumination	275
7.6.4	The Fresnel term	216		Introduction	275
7.7	Pre-computing BRDFs	219	10.1	Global illumination models	276
7.8	Physically based diffuse component	221	10.1.1	The rendering equation	277
8	Mapping techniques	223	10.1.2	Radiance, irradiance and the radiance equation	278
	Introduction	223	10.1.3	Path notation	281
8.1	Two-dimensional texture maps to polygon mesh objects	228	10.2	The evolution of global illumination algorithms	283
8.1.1	Inverse mapping by bilinear interpolation	229	10.3	Established algorithms – ray tracing and radiosity	284
8.1.2	Inverse mapping by using an intermediate surface	230	10.3.1	Whitted ray tracing	284
8.2	Two-dimensional texture domain to bi-cubic parametric patch objects	234	10.3.2	Radiosity	286
8.3	Billboards	235	10.4	Monte Carlo techniques in global illumination	288
8.4	Bump mapping	236	10.5	Path tracing	292
8.4.1	A multi-pass technique for bump mapping	238	10.6	Distributed ray tracing	294
8.4.2	A pre-calculation technique for bump mapping	239	10.7	Two-pass ray tracing	297
			10.8	View dependence/independence and multi-pass methods	300

10.9	Caching illumination	301	<b>13</b>	<b>Volume rendering</b>	370	
10.10	Light volumes	303		Introduction	370	
10.11	Particle tracing and density estimation	304		<b>13.1</b>	<b>Volume rendering and the visualization of volume data</b>	373
<b>1</b>	<b>The radiosity method</b>	306		<b>13.2</b>	<b>'Semi-transparent gel' option</b>	377
	Introduction	306		13.2.1	Voxel classification	378
11.1	Radiosity theory	308		13.2.2	Transforming into the viewing direction	379
11.2	Form factor determination	310		13.2.3	Compositing pixels along a ray	379
11.3	The Gauss–Seidel method	314		<b>13.3</b>	<b>Semi-transparent gel plus surfaces</b>	380
11.4	Seeing a partial solution – progressive refinement	315		13.3.1	Explicit extraction of isosurfaces	382
11.5	Problems with the radiosity method	318		<b>13.4</b>	<b>Structural considerations in volume rendering algorithms</b>	384
11.6	Artefacts in radiosity images	319		13.4.1	Ray casting (untransformed data)	385
11.6.1	Hemicube artefacts	319		13.4.2	Ray casting (transformed data)	387
11.6.2	Reconstruction artefacts	321		13.4.3	Voxel projection method	388
11.6.3	Meshing artefacts	323		<b>13.5</b>	<b>Perspective projection in volume rendering</b>	390
11.7	Meshing strategies	325		<b>13.6</b>	<b>Three-dimensional texture and volume rendering</b>	391
11.7.1	Adaptive or <i>a posteriori</i> meshing	325	<b>14</b>	<b>Anti-aliasing theory and practice</b>	392	
11.7.2	<i>A priori</i> meshing	332		Introduction	392	
<b>2</b>	<b>Ray tracing strategies</b>	342		<b>14.1</b>	<b>Aliases and sampling</b>	393
	Introduction – Whitted ray tracing	342		<b>14.2</b>	<b>Jagged edges</b>	397
12.1	The basic algorithm	343		<b>14.3</b>	<b>Sampling in computer graphics compared with sampling reality</b>	398
12.1.1	Tracing rays – initial considerations	343		<b>14.4</b>	<b>Sampling and reconstruction</b>	400
12.1.2	Lighting model components	344		<b>14.5</b>	<b>A simple comparison</b>	401
12.1.3	Shadows	345		<b>14.6</b>	<b>Pre-filtering methods</b>	402
12.1.4	Hidden surface removal	346		<b>14.7</b>	<b>Supersampling or post-filtering</b>	404
12.2	Using recursion to implement ray tracing	347		<b>14.8</b>	<b>Non-uniform sampling – some theoretical concepts</b>	406
12.3	The adventures of seven rays – a ray tracing study	350		<b>14.9</b>	<b>The Fourier transform of images</b>	411
12.4	Ray tracing polygon objects – interpolation of a normal at an intersection point in a polygon	352	<b>15</b>	<b>Colour and computer graphics</b>	418	
12.5	Efficiency measures in ray tracing	354		Introduction	418	
12.5.1	Adaptive depth control	354		<b>15.1</b>	<b>Colour sets in computer imagery</b>	419
12.5.2	First hit speed up	355		<b>15.2</b>	<b>Colour and three-dimensional space</b>	420
12.5.3	Bounding objects with simple shapes	355		15.2.1	RGB space	423
12.5.4	Secondary data structures	357		15.2.2	The HSV single hexcone model	424
12.5.5	Ray space subdivision	363		15.2.3	YIQ space	427
12.6	The use of ray coherence	364		<b>15.3</b>	<b>Colour, information and perceptual spaces</b>	427
12.7	A historical digression – the optics of the rainbow	367		15.3.1	CIE XYZ space	429
				15.3.2	CIE xyY space	433
				<b>15.4</b>	<b>Rendering and colour spaces</b>	435



<b>15.5 Monitor considerations</b>	436		
15.5.1 RGB <sub>monitor</sub> space and other monitor considerations	436		
15.5.2 Monitor considerations – different monitors and the same colour	437		
15.5.3 Monitor considerations – colour gamut mapping	439		
15.5.4 Monitor considerations – gamma correction	440		
<b>16 Image-based rendering and photo-modelling</b>	443		
Introduction	443		
<b>16.1 Reuse of previously rendered imagery – two-dimensional techniques</b>	444		
16.1.1 Planar impostors or sprites	445		
16.1.2 Calculating the validity of planar impostors	445		
<b>16.2 Varying rendering resources</b>	447		
16.2.1 Priority rendering	447		
16.2.2 Image layering	448		
<b>16.3 Using depth information</b>	452		
16.3.1 Three-dimensional warping	452		
16.3.2 Layered depth images (LDIs)	456		
<b>16.4 View interpolation</b>	458		
16.4.1 View morphing	460		
<b>16.5 Four-dimensional techniques – the Lumigraph or light field rendering approach</b>	463		
<b>16.6 Photo-modelling and IBR</b>	465		
16.6.1 Image-based rendering using photographic panoramas	469		
16.6.2 Compositing panoramas	469		
16.6.3 Photo-modelling for image-based rendering	470		
<b>17 Computer animation</b>	473		
Introduction	473		
<b>17.1 A categorization and description of computer animation techniques</b>	476		
<b>17.2 Rigid body animation</b>	477		
17.2.1 Interpolation or keyframing	477		
17.2.2 Explicit scripting	479		
17.2.3 Interpolation of rotation	483		
17.2.4 Using quaternions to represent rotation	484		
17.2.5 Interpolating quaternions	488		
17.2.6 The camera as an animated object	492		
<b>17.3 Linked structures and hierarchical motion</b>	493		
17.3.1 Solving the inverse kinematics problem	500		
		<b>17.4 Dynamics in computer animation</b>	504
		17.4.1 Basic theory for a rigid body – particles	505
		17.4.2 The nature of forces	506
		17.4.3 Rigid bodies – extended masses	507
		17.4.4 Using dynamics in computer animation	510
		17.4.5 Simulating the dynamics of a lumped mass	511
		17.4.6 Space-time constraints	515
		<b>17.5 Collision detection</b>	517
		17.5.1 Broad phase/narrow phase algorithms	518
		17.5.2 Broad phase collision detection with OBBs	519
		17.5.3 Narrow phase: pairs of convex polyhedra – exact collision detection	522
		17.5.4 Single phase algorithms – object hierarchies	524
		<b>17.6 Collision response</b>	526
		<b>17.7 Particle animation</b>	529
		<b>17.8 Behavioural animation</b>	531
		<b>17.9 Summary</b>	534
		<b>18 Comparative image study</b>	536
		Introduction	536
		<b>18.1 Local reflection models</b>	537
		<b>18.2 Texture and shadow mapping</b>	538
		<b>18.3 Whitted ray tracing</b>	539
		<b>18.4 Radiosity</b>	541
		<b>18.5 RADIANCE</b>	543
		<b>18.6 Summary</b>	543
		<b>References</b>	544
		<b>Index</b>	553