

Contents

Contents	vii
Preface	xv
Acknowledgments	xx
Glossary of terms, products, and acronyms.	xxi
Software and hardware availability.	xxiv
1 Machine vision for industrial applications.	1
1.1 Natural and artificial vision.	1
1.2 Artificial vision	3
1.3 Machine Vision is not Computer Vision	6
1.3.1 Applications.	9
1.3.2 Does it matter what we call the subject?	10
1.4 Four case studies.	12
1.4.1 Doomed to failure.	13
1.4.2 A successful design	14
1.4.3 Moldy nuts.	14
1.4.4 A system that grew and grew	15
1.5 Machine Vision is engineering, not science.	17
1.6 Structure, design, and use of machine vision systems	18
1.6.1 Using an IVS for problem analysis	19
1.6.2 Structure of an IVS.	22
1.6.3 Other uses for an IVS.	22
1.7 Other design tools	23
1.7.1 Lighting Advisor	24
1.7.2 Prototyping kits for image acquisition	26
1.8 Outline of this book.	27
2 Basic machine vision techniques	31
2.1 Representations of images.	31
2.2 Elementary image-processing functions	33
2.2.1 Monadic pixel-by-pixel operators.	33
2.2.2 Dyadic pixel-by-pixel operators	35
2.2.3 Local operators	35
2.2.4 Linear local operators.	36
2.2.5 Nonlinear local operators.	38
2.2.6 N-tuple operators	40
2.2.7 Edge effects	41
2.2.8 Intensity histogram [hpi, hgi, hge, hgc]	41
2.3 Binary images	42
2.3.1 Measurements on binary images.	47
2.3.2 Shape descriptors	48

2.4	Binary mathematical morphology	49
2.4.1	Opening and closing operations	51
2.4.2	Structuring element decomposition	52
2.5	Grey-scale morphology	53
2.6	Global image transforms	55
2.6.1	Hough transform	56
2.6.2	Two-dimensional discrete Fourier transform	57
2.7	Texture analysis	58
2.7.1	Statistical approaches	59
2.7.2	Co-occurrence matrix approach	61
2.7.3	Structural approaches	62
2.7.4	Morphological texture analysis	63
2.8	Further remarks	63
3	Algorithms, approximations, and heuristics	65
3.1	Introduction	65
3.1.1	One algorithm, many implementations	67
3.1.2	Do not judge an algorithm by its accuracy alone	69
3.1.3	Reformulating algorithms to make implementation easier	71
3.1.4	Use heuristics in lieu of algorithms	71
3.1.5	Heuristics may actually be algorithmic	72
3.1.6	Primary and secondary goals	72
3.1.7	Emotional objections to heuristics	74
3.1.8	Heuristics used to guide algorithms	74
3.1.9	Heuristics may cope with a wider range of situations	75
3.2	Changing image representation	75
3.2.1	Converting image format	75
3.2.2	Processing a binary image as a grey-scale image	80
3.2.3	Using images as lookup tables	81
3.3	Redefining algorithms	82
3.3.1	Convolution operations	82
3.3.2	More on decomposition and iterated operations	86
3.3.3	Separated Kernel Image Processing using finite-State Machines	87
3.3.4	Binary image coding	88
3.3.5	Blob connectivity analysis	89
3.3.6	Convex hull	92
3.3.7	Edge smoothing	97
3.3.8	Techniques based on histograms	104
3.4	Approximate and heuristic methods	108
3.4.1	Measuring distance	108
3.4.2	Fitting circles and polygons to a binary object	116
3.4.3	Determining object/feature orientation	123
3.5	Additional remarks	129
3.5.1	Solving the right problem	130
3.5.2	Democracy: no small subset of operators dominates	130

3.5.3	Lessons of this chapter	131
4	Systems engineering	135
4.1	Interactive and target vision systems	135
4.2	Interactive vision systems, general principles	136
4.2.1	Speed of operation	136
4.2.2	Communication between an IVS and its user	137
4.2.3	Image and text displays.	139
4.2.4	Command-line interfaces.	141
4.2.5	How many images do we need to store in RAM?.	141
4.2.6	How many images do we need to display?	143
4.3	Prolog image processing (PIP)	144
4.3.1	Basic command structure.	145
4.3.2	Dialog box	155
4.3.3	Pull-down menus.	157
4.3.4	Extending the pull-down menus	158
4.3.5	Command keys	158
4.3.6	Journal window.	159
4.3.7	Natural language input via speech.	161
4.3.8	Speech output	165
4.3.9	Cursor	165
4.3.10	On-line documentation	166
4.3.11	Generating test images	170
4.3.12	PIP is not just an image-processing system	171
4.4	Advanced aspects of PIP	171
4.4.1	Programmable color filter	171
4.4.2	Mathematical morphology.	175
4.4.3	Multiple-image-processing paradigms	179
4.4.4	Image stack and backtracking	180
4.4.5	Programming generic algorithms	181
4.4.6	Batch processing of images	187
4.4.7	Simulating a line-scan camera.	188
4.4.8	Range maps	188
4.4.9	Processing image sequences.	189
4.4.10	Interfacing PIP to the World Wide Web	194
4.4.11	Controlling external devices.	194
4.5	Windows image processing (WIP).	196
4.6	Web-based image processing (CIP)	198
4.7	Target (factory floor) vision systems	201
4.8	Concluding remarks.	202
5	Algorithms and architectures for fast execution.	203
5.1	Classification of operations.	203
5.2	Implementation of monadic pixel-by-pixel operations	204
5.3	Implementation of dyadic pixel-by-pixel operations	205
5.4	Implementation of monadic neighborhood operations.	207

5.5	Implementation of monadic global operations.....	209
5.6	SKIPSM – a powerful implementation paradigm.....	211
5.6.1	SKIPSM fundamentals	212
5.6.2	SKIPSM example.....	1 213
5.6.3	SKIPSM example.....	2 216
5.6.4	Additional comments on SKIPSM implementations	218
5.7	Image-processing architectures.....	218
5.8	Systems with random access to image memory	219
5.8.1	Systems based on personal computers or workstations	219
5.8.2	Systems based on bit-slice processors	221
5.8.3	Systems based on multiple random-access processors	221
5.9	Systems with sequential image memory access only.....	224
5.9.1	Classification of operations needed for inspection	224
5.9.2	Pipeline Systems	225
5.9.3	Single-board system capable of many operations in one pass	226
5.9.4	Hardware implementations of SKIPSM	228
5.10	Systems for continuous web-based processing	228
6	Adding intelligence	231
6.1	Preliminary remarks	231
6.1.1	Basic assumptions.....	231
6.2	Implementing image-processing operators.....	234
6.3	Very Simple Prolog+ (VSP)	238
6.3.1	Defining the # Operator	240
6.3.2	Device Control	242
6.3.3	VSP Control Software.....	243
6.4	PIP	251
6.4.1	System considerations	251
6.4.2	Why not implement Prolog+ commands directly?	252
6.4.3	Infrastructure	253
6.4.4	Storing, displaying, and manipulating images.....	254
6.4.5	Prolog–C interface	254
6.4.6	PIP infrastructure	257
6.4.7	Defining infrastructure predicates.....	259
6.4.8	Implementing PIP's basic commands.....	262
6.5	WIP	265
6.5.1	WIP design philosophy.....	265
6.5.2	Host application	266
6.5.3	Command line function.....	268
6.5.4	Parser	269
6.5.5	Calling function	270
6.5.6	Interface function	270
6.5.7	Image function	270
6.5.8	Returning results and errors	271
6.6	Concluding remarks.....	271

7	Vision systems on the Internet.	273
7.1	Stand-alone and networked systems	273
7.2	Java	274
7.2.1	Platform independence	274
7.2.2	Applets, applications, and servlets.	275
7.2.3	Security	276
7.2.4	Speed	277
7.2.5	Interactive vision systems in Java	277
7.3	Remotely-Operated Prototyping Environment (ROPE)	278
7.3.1	What ROPE does.	279
7.3.2	Assistant.	280
7.3.3	Vision engineers should avoid travel.	282
7.3.4	Current version of ROPE.	282
7.3.5	Digitizing images.	284
7.3.6	Java Interface to the Flexible Inspection Cell (JIFIC)	286
7.3.7	Other features of ROPE	287
7.4	CIP.	288
7.4.1	Structure and operation of CIP	292
7.5	Remarks.	293
8	Visual programming for machine vision	295
8.1	Design outline.	295
8.1.1	Graphical user interface (GUI)	296
8.1.2	Object-oriented programming	296
8.1.3	Java image handling	297
8.1.4	Image-processing strategy	298
8.2	Data types	300
8.2.1	Image	300
8.2.2	Integer	301
8.2.3	Double	301
8.2.4	Boolean	301
8.2.5	String	302
8.2.6	Integer array	302
8.3	Nonlinear feedback blocks	302
8.3.1	Feedback	303
8.3.2	FOR loop	304
8.3.3	IF ELSE.	305
8.3.4	Nesting.	305
8.4	Visual programming environment.	305
8.4.1	Interpretation of graphics files.	306
8.4.2	Plug-in-and-play architecture	307
8.5	Image viewer and tools	309
8.5.1	Horizontal and vertical scans.	309
8.5.2	Histograms	309
8.5.3	Pseudocolor tables	309

8.5.4	3-D profile viewer	310
8.6	Sample problems	311
8.6.1	Low-level programming	311
8.6.2	High-level programming	312
8.6.3	Convolution	312
8.6.4	Fourier transform	313
8.6.5	Isolate the largest item in the field of view	314
8.6.6	Character detection using the N-tuple operator	315
8.7	Summary	315
9	Application case studies	317
9.1	Preliminary remarks	317
9.2	Taking a broad view	318
9.2.1	Automobile connecting rod (conrod)	318
9.2.2	Coffee beans	319
9.2.3	Table place-setting	319
9.2.4	Hydraulics manifold	320
9.2.5	Hydraulics cylinder for automobile brake system	320
9.2.6	Electrical connection block	321
9.2.7	Electric light bulb	321
9.2.8	Analysis of industrial X-rays	322
9.2.9	Highly variable objects (food and natural products)	323
9.3	Cracks in ferrous components	324
9.4	Aerosol spray cone	325
9.5	Glass vial	326
9.6	Coin	327
9.7	Metal grid	328
9.8	Toroidal metal component	329
9.9	Mains power plug (X-ray)	331
9.10	Conclusions	332
10	Final remarks	335
10.1	Interactive prototyping systems	336
10.2	Target vision systems	337
10.3	Design tools	339
10.4	Networked systems	340
10.5	Systems integration	341
10.6	Algorithms and heuristics	342
10.7	Concluding comments	343
Appendix A	Programmable color filter	345
	Representation of color	345
	Color triangle	346
	Mapping RGB to HSI	348
	Programmable color filter (PCF)	348
	Software	351

Appendix B	A brief introduction to Prolog	353
	Prolog is different	353
	Declarative programming	353
	Facts	354
	Simple queries	354
	Rules	355
	Queries involving rules	356
	Backtracking and instantiation	356
	Recursion	357
	Lists	357
	Other features	358
	Further reading	359
Appendix C	PIP commands and their implementation	361
	References	369
	Further reading	374
	Index	377