A MULTIPLE-SYSTEMS APPROACH IN THE SYMBOLIC MODELLING OF HUMAN VISION

A Thesis

submitted to the Faculty of Engineering of the

University of Glasgow

for the Degree of

Doctor of Philosophy

by

Thomas McIndoe

September 1992

This Thesis is dedicated to my mother,

CAROLINE

and to the memory of

THOMAS McINDOE, Senior

ABSTRACT

For most of the thirty years or so of machine vision research, activity has been concentrated mainly in the domain of metric-based approaches: there has been negligible attention to the psychological factors in human vision. With the recent resurgence of interest in neural systems, that is now changing. This thesis discusses relevant aspects of basic visual neuroanatomy, and psychological phenomena, in an attempt to relate the concepts to a model of human vision and the prospective goals of future machine vision systems. It is suggested that, while biological vision is complex, the underlying mechanisms of human vision are more tractable than is often believed. We also argue here that the controversial subject of direct vision plays a crucial role in natural vision, and we attempt to relate this to the model.

The recognition of massive parallelism in natural vision has led to proposals for emulating aspects of neural networks in technology. The systems model developed in this work demonstrates software-simulated cellular automata (CAs) in the role of mainly low-level image processing. It is shown that CAs are able to efficiently provide both conventional and neurally-inspired vision functions.

The thesis also discusses the use of Prolog as the means of realising higher level image understanding. The symbolic processing developed is basic, but is nevertheless sufficient for the purposes of the present demonstrations. Extensions to the concepts can be easily achieved.

The modular systems approach adopted blends together several ideas and processes, and results in a more robust model of human vision that is able to translate a noisy real image into an accessible symbolic form for expert-domain interpretation.

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my Supervisor, Professor John Barker, for his expert discussions and guidance during the course of this work. This was particularly important given the diverse nature of the Project's goals. In this research, I have been able to realise an ambition of more than twenty years—to begin an understanding of the difficult and abstract nature of biological vision. I am also grateful to my Supervisor for introducing me to the methods of cellular automata (CAs) as alternatives to artificial neural networks in computer simulations. His concept of CA-simulated soap films coupled the Project to Gestalt theory: thus forging an important link between psychology and technology.

I must also record my appreciation of the considerable freedom and flexibility that has been accorded me in my research studentship, in recognition of the personal problems of visual disability. For this I have to again thank my Supervisor, and the then Head of Department, the late Professor John Lamb, CBE DSc FRSE FEng.

I take this opportunity to recall my debt to Dr. Richard Fryer, my previous Supervisor at Strathclyde University's Department of Computer Science. Without his help and encouragement during my earlier studies for MPhil. I would not have had the confidence to undertake my present research.

Primary funding and support in the form of Research Studentship no. 89314186 awarded by the UK Science and Engineering Research Council (SERC) is gratefully acknowledged. Thanks are due to Professor Peter Laybourn for arranging this in 1989, prior to the commencement of my studentship. I also acknowledge limited financial assistance from the Royal National Institute for the Blind (RNIB), the Royal National Institute for the Deaf (RNID), and the Department of Electronics and Electrical Engineering at Glasgow University.

I am grateful to Professor John Frisby of the AI Vision Research Unit at the University of Sheffield for his valuable advice and commentary on relevant aspects of the Marr-Gibson paradox. I also appreciate his suggestions for selected readings in the complex subject of visual psychology.

Finally, I wish to thank academic and secretarial staff in Glasgow University's Department of Electronics and Electrical Engineering for help of various kinds during the course of my studies, especially the following: Mrs J Reid, Mrs K Phillips, Miss K Edwards, Miss A MacKinnon, Dr. D Muir, and Dr. J MacLeod.

T. McIndoe, CGIA BSc MPhil DipTech FIAP CEng MIEE MBCS
September, 1992

LIST OF THESIS FIGURES AND TABLES

Figure 1.1a	Complexity in natural images	7
Figure 1.1b	The kind of data generated in machine vision	7
Figure 1.2	Illustrating the Optic Frame paradox	20
Figure 2.1	The simple caricature of a vase	27
Figure 2.2	Marr's concept of the 2.5D sketch	31
Figure 3.1	Simple models of biological neurons	42
Figure 4.1	The three common CA neighbourhoods	47
Figure 4.2	A 2D cellular lattice model	51
Figure 4.3	Demonstrating SOAP film mechanisms	64
Figure 5.1a	The Kanizsa Square illusion	70
Figure 5.1b	A variety of chair styles	70
Figure 5.2	Airborne radar example	74
Figure 5.3	The "Dalmation-in-the-Park"	75
Figure 6.1	The perception of a "circle"	80
Figure 6.2	The Gestalt Laws of Organisation	81
Figure 6.3	A conventional image segmentation	81
Figure 7.1	Cotter's diagram of the Optic Tract	97
Figure 7.2	Arbib's schematic of the Optic Tract	98
Figure 7.3	A technological interpretation of vision	100
Figure 7.4	Treisman's preattentive vision concept	102
Figure 8.1	Multi-level evidence array [Jr]	105
Figure 9.1	The direct image array [Ar]	124
Figure 9.2	Arrangement of CA image processor	124
Figure 9.3	Local-global interactions in GM model	126
Figure 9.4a	Diagram of the ECM mechanism	130
Figure 9.4b	Schematic of CA rule feedback	130
Figure 9.5	General concept of ECM-HRS-IRM	132
Figure 9.6	Operation of the ECM mechanism	132
Figure 9.7	Concentric contexts in Prolog	141

Figure 10.1 General concept of PROVIS system 148
Figure 10.2a PROVIS screen – original image 151
Figure 10.2b PROVIS screen – process image 151
Figure 10.3 Test results – 4-dots image 155
Figure 10.4 Test results – global features 158
Figure 10.5 Test results – edge detectors 160
Figure 10.6 Test results – COMPLETE rule 164
Figure 10.7 Operation of the COMPLETE rule 165
Figure 10.8 Test results – Kanizsa's Square 168
Figure 10.9 Demonstrating the COC illusion 169
Figure 10.10 Demonstrating CA segmentation 172
Figure 10.11 Demonstrating ECM-HRS mediation 174
Figure 10.12 The energy changes in TEXTURE rule 176
Figure 10.13 Test results – convex hull 178
Figure 10.14a A PROVIS test image 181
Figure 10.14b Image highlights display 181
Figure 10.15 Test results – Treisman-like preattentive images 182
Figure 10.16 Demonstrating the basic ECM-HRS system 189
Figure 10.17a The PROVIS main menu 191
Figure 10.17b The VPC processor menu 191
Figure 10.18a The floppy diskette test image 192
Figure 10.18b Connectivity matrix for the floppy disk 192
Figure 10.19a PROVIS output report for disk image 193
Figure 10.19b The PROVIS database upgrade information 193
Figure 10.20a A natural street scene test image 196
Figure 10.20b The ECM-generated edge map 196
Figure 10.20c The symbolic regional map 197
Figure 10.20d The regional image masks 197
Figure 10.21a Connectivity matrix for street scene 198
Figure 10.21b Image node data for street scene 198
Figure 10.22a An additional object in street scene 199
Figure 10.22b New object creates its own symbolic mask 199
Figure 10.22c Object masks can be positioned anywhere 200
Figure 10.22d New image created by mask manipulation 200

Table 2.1 Marr and Gibson Attributes 34
Table 7.1 Dualism in Natural Vision 92
Table 8.1 Typical VPC Calls from Prolog 110
Table 10.1 Parameters Returned by GET_REG 187

THESIS CONTENTS

Abstract i

Acknowledgements ii

List of Figures and Tables iii

Thesis Contents v

1 : INTRODUCTION 1

1.1 Goals of the Current Project . . . 3

1.2 Natural and Machine Vision . . . 4

1.3 Complexity in Natural Imagery . . . 5

1.4 Peripheral Issues in Machine Vision . . . 8

1.4.1 Textual Reading Devices . . . 8

1.4.2 Retinal Acuity and Retinal Inhomogeneity . . . 8

1.4.3 Colour Perception . . . 9

1.4.4 Stereopsis and 3D Vision . . . 10

1.4.5 Optic Flow . . . 11

1.4.6 Neural Networks . . . 11

1.4.7 Supercomputers and Parallelism . . . 12

1.4.8 Computer Graphics . . . 12

1.4.9 Autonomy in Localised Neural Control . . . 13

1.5 Current Approaches in Machine Perception . . . 14

1.5.1 First-Generation Machine Vision . . . 15

1.5.2 Second-Generation Machine Vision . . . 16

1.6 Marr's and Gibson's Theories of Vision . . . 17

1.7 The "Optical Frame" Paradox . . . 19

1.8 Towards a Vision Model . . . 21

1.9 Chapter Summary . . . 21

2 : NATURAL AND ARTIFICIAL VISION 23

2.1 Levels of Vision . . . 24

2.1.1 Low-level Vision . . . 24

2.1.2 High-level Vision . . . 24

2.2 Marr's Computational Theory of Vision . . . 25

2.3 The Ecological Theory of Vision . . . 32

2.4 Marr and Gibson Compared . . . 34

2.5 Other Theories of Natural Vision . . . 34

2.6 Data-Driven or Goal-Driven Vision? . . . 35

2.7 Chapter Summary . . . 35

3 : NEURAL NETWORKS 36

3.1 The Neural Network Revival . . . 37

3.2 Why Neural Networks? . . . 38

3.3 Parallel Distributed Processing . . . 38

3.4 The Basis of Artificial Neural Systems . . . 39

3.4.1 McCulloch-Pitts Neuron Model . . . 40

3.5 Neural Networks and Memory . . . 43

3.6 Learning in Neural Networks . . . 43

3.7 Chapter Summary . . . 44

4 : CELLULAR AUTOMATA 45

4.1 General Description of CAs . . . 46

4.2 CAs and Artificial Neural Nets . . . 50

4.3 CAs and Image Processing . . . 52

4.4 Characteristics of "Pure" CAs . . . 53

4.4.1 Parallelism . . . 53

4.4.2 Locality . . . 53

4.4.3 Homogeneity . . . 54

4.5 Cellular Automata Algorithms . . . 54

4.6 CA Rules and Lookup Tables . . . 55

4.7 CA Neighbourhoods and Kernels . . . 56

4.8 Morphological Operators . . . 57

4.9 Examples of Simple CA Rules . . . 57
4.9.1 Totalistic Rules . . . 58
4.9.2 Semitotalistic Rules . . . 59
4.10 Defining CA Rules—Lookup Tables . . . 60
4.11 CA Rules for Machine Vision . . . 61
4.12 The "Soap Film" Rule . . . 62
4.13 CA Hardware Possibilities . . . 66
4.14 Chapter Summary . . . 66

5 : PSYCHOLOGICAL FACTORS IN VISION 67

5.1 Low-level Vision . . . 67
5.2 High-level Vision . . . 69
5.3 The Relationship Between LLV and HLV . . . 71
5.3.1 The "Kelvin" Exemplar . . . 71
5.3.2 Airborne Radar Analogy . . . 72
5.3.3 The "Dalmation-in-the-Park" . . . 75
5.4 Visual Psychology and Prolog . . . 76
5.5 Chapter Summary . . . 77

6 : ASPECTS OF PERCEPTUAL GROUPING 78

6.1 The Problem of Perceptual Grouping . . . 78
6.2 Perceptual Organisation . . . 79
6.3 The Gestalt Psychology . . . 82
6.4 A Gestalt Link to Technology . . . 83
6.5 Recent Research in Perceptual Organisation . . . 84
6.6 Perceptual Organisation and Machine Vision . . . 85
6.7 Chapter Summary . . . 86

7 : RECENT THEORIES OF BIOLOGICAL VISION 87

7.1 Natural Neural Networks . . . 87
7.1.1 The Brain . . . 89
7.1.2 The Superior Colliculus . . . 90
7.1.3 The Lateral Geniculate Nuclei . . . 91

7.2 Technological vs. Biological Vision 91
7.3 The Nature of Biological Vision 92
7.4 Gibsonian-Style Vision 93
7.4.1 The Optical Tracts 94
7.4.2 Massive Retinal Parallelism 95
7.4.3 The SC and the LGN-Cortex 95
7.5 Modelling the Optic Tract 99
7.6 Preattentive Vision 101
7.7 Chapter Summary 103

8 : COMPUTING REQUIREMENTS 104

8.1 Low-Level Image Processing Software 104
8.1.1 VPC Low-Level Routines 105
8.1.2 CA Lookup Table Functions (Rules) 107
8.1.3 Turbo Prolog Predicate Calls 109
8.2 High-level Vision Functions 111
8.3 Hardware Facilities 113
8.4 Chapter Summary 113

9 : CONSTRUCTING A VISION MODEL 114

9.1 General Considerations 114
9.2 Scope of an Adequate Vision Model 115
9.3 Pre-Project Evaluation Studies 117
9.4 Requirements of a Vision Model 118
9.5 The Grossberg-Mingolla Model 120
9.6 CA-based Vision Processing 121
9.7 Local and Global Processes 125
9.7.1 Global Propagation in the G-M Model 125
9.7.2 Global Propagation in the CA Model 127
9.8 Justifying the Current Approach 131
9.9 The ECM–HRS Process 133
9.9.1 The Prefiltering Stage (1) 134
9.9.2 The Preattentive Vision Stage (2) 134

9.9.3 The Oriented Edge Masks (3) 134
9.9.4 Oriented Edge-Cell Competition (4) 135
9.9.5 Weight-Cell Thresholding (5) 136
9.9.6 The Combined ECM + HRS Stage (6) 138
9.10 The Prolog-Based IRM 140
9.10.1 Concentric Contexts 141
9.10.2 Prolog Rule-Logging 142
9.10.3 Prolog and Fuzzy Logic 142
9.10.4 Image-Tree Representation 143
9.11 Chapter Summary 144

10 : DEMONSTRATOR RESULTS AND DISCUSSION 146

10.1 The Software Modules 146
10.2 Defining Test Images 149
10.3 The User Interface 150
10.4 Basic CA Processes 152
10.4.1 CA Lookup Tables and Rules 152
10.4.2 CA LUT Rule—SOAP 154
10.4.3 CA LUT Rule—EDGE 159
10.4.4 CA LUT Rule—COMPLETE 161
10.4.5 Analysis of the COMPLETE Rule 162
10.5 Synthetic Illusory Images 166
10.6 Demonstrating Direct Vision 170
10.6.1 Textured Images 170
10.6.2 Direct Image Mediation 173
10.6.3 Energy in CA Computations 175
10.6.4 The Convex Hull 177
10.6.5 Demonstrating Preattentive Vision 179
10.7 Demonstrating Image Understanding 183
10.7.1 Brief Review of ECM-HRS 184
10.7.2 The IRM Functions 186
10.8 A Demonstration of the ECM-HRS 188
10.9 A Synthetic Image Demonstration 190

10.10 A Natural Scene Example 194
10.11 Discussion of Results 201

11 : CONCLUSIONS AND FUTURE WORK 205

11.1 Interpretation of the Results 206
11.2 Final Conclusions 207
11.3 Future Work 210

APPENDICES

A : VPC PRIMITIVES AND PROLOG CALLS 214

A.1 VPC Primitives 214

A.2 Prolog Functions 217

B : BRIEF NOTES ON PROLOG 219

B.1 Prolog Language Development 219

B.2 The Historical Roots of Prolog 220

B.3 Turbo Prolog 222

C : BASIC IMAGE TRANSFORMATIONS 224

C.1 Conventional Image Processes 224

C.2 Convolution 228

D : PARALLEL COMPUTERS AND CAs 229

D.1 A Historical Perspective 229

D.2 Computer Architecture—Flynn's Taxonomy 232

D.3 A Brief History of CAs 233

D.4 CAs: Some Applications and Possibilities 236

E : CELLULAR AUTOMATA AND ANNs 237

E.1 CAs vs. ANNs 237

F : THE CONCEPT OF MEMORY MAPPING 240

F.1 Memory Mapping 240

F.2 Sensors Within Memory Address Space 241

G : INTERFACING TURBO PROLOG 242

G.1 General Concepts 242

G.2 Turbo Prolog Compatibility Problems 244

G.3 Linking Prolog with C OBJ Modules 245

G.4 System Initialisation Call 246

G.5 Underbar Generation in C Code 246
G.6 Calling Convention for VPC Functions 247

H : NEURAL SIGNAL COMPRESSION 248

H.1 Introduction 248
H.2 The Signal Form 249
H.3 The Processing Elements 250
H.4 Discussion 251

I : MATHEMATICAL NOTES 252

I.1 Neural Cell Processes 252
I.2 A Simple Digital Filter 259

J : GRAPHICAL REPRESENTATIONS 261

J.1 Simple Image-Tree Graphs 261
J.2 Attneave's Image Highlights 266

REFERENCES 270

GLOSSARY 279

CHAPTER 1

INTRODUCTION

"The objects of perception and the space in which they seem to lie are not abstracted by a rigid metric but a far looser one than any philosopher ever proposed or any psychologist dreamed." — Jerome Lettvin, 1981.

Perception is not just about vision. Perception is that most elusive property of the human psyche which can enable us to perceive a "red" London Transport bus even when that bus is seen reflecting brilliant white sunshine; perception allows one to recognise and sustain the image of a familiar human face even when the features are distorted by frowning, or viewed in deep shadow; perception is invoked to generate and update visual memories of events past and present; places, aromas, concepts, and even synthesize entirely new mental images. These feats are all evidently directly linked with the essential faculty we call "vision," but vision by itself cannot be responsible for all of this. There must be interaction with stored dynamical memory, and a built-in ability to reason about the nature of our retinal images. The London Transport bus exists as much as a mental "concept" as a real-world, or "vehicle-class image" that has to be visually analysed and recognised. Artists can synthesise images that have existed only in their own minds, and architects can visualise buildings that are still only in their imaginations.

Somehow, the visual image representations of the physical real world have to be blended-in with our extremely complex psychological and mental processes. However, as Lettvin observed, this is achieved in Nature, not by a set of complex and rigid metrics, but by strangely "loose" processes. The writer considers that it is the discovery of cogent paradigms of the human mind-brain that will eventually result in substantial advances in both artificial vision, and in artificial intelligence (AI).

Commonsense and experience suggest that vision is really a dualist phenomenon. Intuitively, we realise that there is something "direct" about our normal capability to instantly "see" and understand a fast moving image, or assess a complex visual scene. We also intuitively know that our ability to reason about the images we see must somehow be connected with our knowledge, learning, and past experience of the real world. Thus we would appear have a dual kind of visual mechanism. It is strange, therefore, that scarcely any research has been reported in connection with the dualist phenomena of intelligent vision. To the writer's knowledge, there have been very few attempts, if any, to directly link the practicalities of image processing technology with the philosophical ideas of cognition and visual psychology.

The challenge within the science of vision is to discover how our physical retinal images, created from complex patterns of direct and reflected light in the external environment, translate into meaningful internal psychological concepts and models. When we can understand the nature of these obscure processes, we will be in a position to apply advanced technology to the design of practical artificial vision machines for human benefit.

The work described in this thesis develops a "systems" approach to the problems of machine vision. Visual-psychological concepts, and novel image processing mechanisms, act together to produce high-level, semi-symbolic, descriptions of real-world scenes. The resulting image symbolisms are then more easily interpreted within an AI programming environment—such as Prolog, or an expert system.

As will be described, the Gestalt psychologists in the early part of this century provided the basis of important links between psychology and technology in the form of the Gestalt Laws of Organisation. These links are seen by the writer as very important in a much more complete understanding of vision in humans. However, the contributions of the Gestalt School of Psychologists most certainly did not embrace all of the relevant aspects of human perception. There remain to this day daunting problems in the generation of internal perceptual models (for example), and the totally subjective aspects of vision. There is also a need to be able to quantify and refine these abstract Gestaltist concepts, before they can be considered for practical use in computer algorithms, or as solutions in pragmatic engineering applications.

1.1 Goals of the Current Project

The principal goal of the present work is to begin an understanding of certain neglected aspects of biological vision—in particular, to review the phenomenon known as "direct vision." As will be seen later in this thesis, the writer considers that such an understanding may hold the key to the comprehension of the entire range of phenomena which are of interest in artificial intelligence (AI). However, we wish only at this stage to investigate how direct images (referred to later as "Gibsonian images") can *mediate* the processes of image transformation.

Included in the above goal is a requirement to develop a computational theory of key aspects of intelligent vision—by which we mean here tangible models, or analogues, of human vision. We wish thereby to suggest how these models might relate to the Gestalt ideas, and hence to visual psychology. Despite the superiority of the many forms of metric-based artificial vision, which permit the determination of highly accurate measurements of image parameters and attributes, the human vision model remains the goal for many AI researchers. The unifying principle in this work is the "computational" approach; that is, it should be possible to understand the processes of vision independently of whether they are implemented in computer hardware, or in biological wetware.

Another objective of the present project is to investigate how massive-parallelism, realised as simulated fine-grained cellular automata (CAs), can provide both the low-level and the higher level visual processes. Parallelism has been shown to be most successful in low-level vision, and in many areas of image processing where pixel manipulative processes are well-posed and robust. But higher level vision includes complex image abstraction, image segmentation and restructuring, and of course the many intangible, psychological, aspects of perception. Natural vision is therefore an ill-posed problem.

It will be shown here that there are links to Gestalt visual psychology which can be exploited at the lower levels of vision and image processing—particularly using our developing CA models of basic image processing.

A further aspect of the project is a demonstration of how the many symbolisms created by such image processing can be understood in an AI model of image comprehension which directly invokes the phenomenon of direct vision. The Prolog computer programming language will be used for this purpose, as it is likely to be a very long time before biological neural circuitry and functionality can be understood in a symbolic sense.

In short, what is sought is a robust and plausible paradigm which defines the main steps in the processes mentioned. Clearly, any one of the above subgoals can be pursued to a high degree of specialism: hence an important factor in the project must be the application of suitable goal constraints which can permit progress without incurring the usual penalty of philosophical or computational bottlenecks.

For a long time, artificial vision research has remained at the lower levels of achievement, with only an occasional foray into intermediate and higher levels. Research on the highest planes has always tended to be confined to the specialist interests of the psychologist, and the cognitive scientist. rather than the physicist, the technologist, or the vision systems engineer. There exists a wealth of knowledge which has been gleaned over the more than three decades of psychological, physiological, and neuroanatomical research which can be put to use in technology. This makes the study of convincing vision models a truly interdisciplinary and challenging activity.

1.2 Natural and Machine Vision

The arguments in favour of developing artificial vision have had a full exposure over the years. This section briefly reviews them.

Natural vision is the most important faculty possessed by humans, and by most living species. The goal of vision is that of understanding, in collaboration with other senses and stored information, the complex patterns of visual stimuli to which an animal is ordinarily exposed. For most animals, the main requirements of vision are the location of food and the avoidance of predators. These are the basic requirements for survival in a potentially hostile environment. For humans, vision plays a much more complex role, especially in our civilised societies. Here, high visual acuity is important, together with the ability to function in complex environments. The success of human vision, and the demands of modern manufacturing and production, have inspired many attempts over the past four decades to emulate natural vision. These efforts have often been disappointing. They suggest that the problems of vision are much more philosophical and deeply rooted than the early researchers had anticipated.

Descriptions like "machine perception," "machine vision," "artificial perception," and "computer vision" relate to these attempts, which have traditionally tended to be closely linked with robotics and industrial manufacturing. However, many other types of technological vision have existed, particularly within the fields of surveillance, and medical imaging. For present purposes, the generic term "machine vision" can conveniently be used in most cases to define implementations coming within the scope of artificial vision. Normally, the closely related subjects of **image processing**, and **pattern recognition**, are regarded as quite separate activities, although a considerable overlap exists between these two disciplines and machine vision.

In general, natural vision is characterised by speed, tolerance, reliability, robustness, and of course "intelligence," whereas machine vision has usually only included a small subset of these desirable attributes.

1.3 Complexity in Natural Imagery

A reason for the failure of much of artificial vision is the sheer complexity of natural imagery, coupled with a lack of knowledge about the real world. Figure 1.1 shows a typical scenic image, in which grey levels representing featural aspects of the external world are typically represented as a series of integers. Within such images, the "information" is usually assumed to be the image itself.

Natural colour images are, of course, even more complex, and real-time processing of moving sequences of such images is beyond the capability of most present-day computers. [1]

Another problem is the ever present effect of **noise** in its many forms, as found in images and processing systems. Noise can come from a large number of sources and may have nontrivial implications. For example, extraneous pixels appearing within a computer tomographic (CT) image scan might be due to "noise" – or a detected tumour.

[1] Note, however, that there is usually no problem when processing complex images "off-line" as there need be no imposed restrictions on the available processing time.

A third phenomenon is the effect of luminance (brightness) and its apparent perceptual stability in biological vision. The red London Transport bus cited earlier is perceived as "red" even when seen in partial shadow. So dealing with the subjective and perceptual effects of luminance (and of illuminance) variation is a very complex problem for artificial vision. This is often referred to in the literature as "discounting the illuminant."

Confronted with this kind of input, most artificial vision systems are hard-pressed to compute, and hence classify, imaged objects within an acceptable processing time. Typical machine vision systems may take many seconds—possibly even minutes—to perform a classification. Clearly, this will be unacceptable in most industrial assembly lines. Very much longer image processing times can usually be tolerated in, for example, medical imaging, and the other kinds of specialist visual analysis that can be carried out off-line. The achieved success of machine perception has traditionally tended to be judged on the speed of image processing, and by the results of analysis when related to a specific application.

Despite the massive data content and textural complexity of natural images, there is usually insufficient supportive information within the images themselves to permit reliable object classification and/or scene understanding. This might seem strange, but even human observers can find everyday object recognition difficult in situations deprived of context. This is an important factor. The point to be emphasised here is that, even in a vision-based system, pictorial information by itself is not enough. It is the intelligent interpretation of visual data, in association with visual memory and an acquired knowledge-base of real-world experience and information, that is needed. In addition, the phenomenon of direct vision (to be discussed) is regarded by the writer as possibly the most important single aspect of vision, as well as being a factor in AI in general.

Figure 1.1a Complexity in natural imagery. A typical image of the kind usually found in machine vision applications. This is a grey-level picture of a small English village, re-digitized to give a low-resolution image of 64x64 pixels.

11	8	6	5	5	3	2	2	3	3	4	5	4	6	6	5
12	10	12	9	6	5	5	3	3	2	3	5	6	5	5	6
6	7	7	7	6	6	6	5	3	2	3	5	6	6	5	4
7	15	15	15	5	7	9	7	7	3	3	3	5	5	5	5
11	15	15	15	6	8	6	5	7	5	3	4	2	5	6	5
12	15	15	15	6	7	7	6	8	7	7	4	3	3	5	5
13	15	15	12	7	6	8	8	8	8	7	6	5	3	5	4
15	15	15	13	6	7	7	7	8	9	7	8	8	5	3	4
15	15	15	11	8	6	7	6	5	5	5	6	6	3	1	2
15	15	15	14	9	7	7	7	5	5	9	11	8	8	2	2
15	15	15	14	12	9	12	11	9	10	11	13	10	13	6	6
15	15	15	8	10	12	14	12	12	10	12	10	10	15	15	15
15	15	15	7	10	14	14	12	11	9	12	12	9	15	15	14
15	15	13	6	4	5	8	10	12	8	13	12	10	15	13	3
15	15	7	2	2	2	5	10	14	5	9	10	11	15	10	4

Figure 1.1b Illustrating the kind of data generated in machine vision. This is a small portion of the above image, the grey levels being represented as a set of integer values; in this case on a scale range of 0–15. Already, at this early stage of processing, the subjective nature of direct vision appears to have been lost.

1.4 Peripheral Issues in Machine Vision

Before proceeding to describe aspects of the present project, it is necessary to clarify certain issues which can potentially come within the scope of artificial vision. Some of these factors do not have an immediate relevance in the present approach to vision research. Their exclusion from the work of this thesis seems justified on the argument of pursuing the central theory, whose principles must be exposed.

1.4.1 Textual Reading Devices

The subject of text reading, although ostensibly a goal of machine vision, is usually regarded as a separate specialism. The reason for this is that text reading and text processing can range from the very general and much constrained—as when reading numbers on cheques—to the extremely specialised and context-sensitive—as when studying the detailed information contained in a textbook. Hence, despite a similar kind of dichotomy in some forms of industrial machine vision, it has usually been the custom to consider text reading as being an entirely separate research objective. This is the attitude adopted here.

However, this stance may not be viable in future machine perception research, if the goal of replicating human vision is to be tackled in earnest. This is because human vision is *able* to deal with many different kinds of visual information. For an an excellent overview of a "neural" hierarchical artificial vision model of text reading, consult the paper by Fukushima (1988).

1.4.2 Retinal Acuity and Retinal Inhomogeneity

The human retina, unlike conventional image sensor devices, is very nonlinear in its acuity response. The *fovea*, lying on the optic axis in the centre of the retina, is the area of the greatest concentration of photoreceptor neurons; and so in humans and most animals the eye is moved in such a way as to bring objects of interest within the foveal region. Almost all commercially available image sensors are totally linear in their resolution over the active surface; indeed, the cost of these devices is normally related to the manufacturing guarantees as to the fidelity of the very many photosites in such an array package. The consequence of this is that we have to regard the *total* active area of photosensors as the equivalent of foveal imaging.

This means that peripheral information in an image sensor does not have the same significance as peripheral responses within a biological retina. For example, unlike the eye, a video-camera will not be required to oscillate about a fixation point in order to maintain photoreceptor activity and stimulation. Another factor is the existence of retinal veins and arteries, and the blind spot, which would seem to obscure areas of the retina. This does not in fact happen—because of neural network compensation. The only relevance of retinal nonlinearity and inhomogeneity in this work might be in the postulated role of the midbrain (the superior colliculus) in the organisation of visual stimulus tracking and eye fixation.

There are other differences resulting from retinal inhomogeneity, the implications of which cannot be discussed here. Recent attempts to produce much more realistic artificial retinas, having concentric acuity and many of the features of biological retinas, are described in recent papers such as Mead (1988), and Meuller et al. (1987).

1.4.3 Colour Perception

Another issue of difference between machine vision and natural vision is the subject of colour imaging. Human, and most animal retinas are sensitive to a wide spectrum of radiant energy frequencies (colours) and this can be crucial to both direct vision and image understanding. Traditionally, most machine perception systems have ignored colour, and preferred instead the use of a grey scale for the representation of subjective colours. The grey scale is normally within the integer range 0–255, with subranges of 0–63, 0–31, or 0–15 being common. In the present work, for reasons of economy and speed, we restrict our use of colour images to the small subrange 0–15.

However, we are reminded that colour by itself can often be the most significant parameter or attribute in the recognition of an image, or in the understanding of a scene in context. For example, industrial parts on an assembly line could be colour-coded as an aid to robotic manipulation and assembly using vision. Colour is also of importance in psychological understanding and categorisation of everyday objects. As mentioned, the characteristic attribute of the ubiquitous London Transport bus is probably that it always seems to be perceived as a "red" bus—even while turning a corner, and thereby reflecting obliquely brilliant ("white") sunlight!

This crucial perception is in accordance with modern theories of colour vision: ideas that concur with several higher order concepts developed in this thesis. Colour vision is now to be understood as the ability of natural vision to identify those qualities of an object that remain constant, regardless of ambient lighting and viewpoint. It is, consequently, not merely a matter of the spectral wavelength. Indeed, as will be discussed later in this thesis, colour would seem to be one of the many elusive perceptual "invariants" identified by the vision pioneer and cognitive researcher, James Gibson.

1.4.4 Stereopsis and 3D Vision

Machine vision research has in recent years become obsessed with the idea that stereoscopic vision and 3D imaging can provide answers to many of the daunting problems in artificial vision. This partly stems from Marr's use of three-dimensional (3D) image pairs in his work on natural vision. Marr (1982), and Barrow and Tenenbaum (1978) are among the many who believed that the exposition of 3D surface elements is a goal of low-level and natural vision processing. They argued that the visual interpretation task is far easier when surface descriptions are used. Indeed, this notion formed the basis of Marr's formulation and conceptualisation of the so-called "2.5D Sketch," to be described in Chapter 2, and mentioned elsewhere in this thesis.

Another possible reason for the overemphasis on 3D is computational convenience and elegance. Stereopsis models facilitate mathematical analysis, and hence impressive mathematical results can be achieved for stereo matching, image fusion, and so on. Stereoscopic algorithms lend themselves to implementation in a wide variety of novel ways; such as energy minimisation, optic flow analysis, cellular automata, as well as the more conventional serial computer methods. In addition, there is a considerable interest in active sensors involving special kinds of striped lighting, surface polarimetry, and so forth. Examples of these methods include the recent work by Fryer and Miller (1991), Siebert et al. (1991).

The writer considers that the additional information derived from 3D and stereopsis does not contribute significantly to the understanding of the nature of psychological vision. One has only to remember that (except possibly in unusual circumstances) monocular vision does not appear to seriously hinder humans who have never experienced normal binocular vision.

It is acknowledged, however, that 3D information may be valuable in, say, robotic vision, where a need is for robots to be able to locate and grasp mechanical components. Nevertheless, the addition resources necessary to tackle 3D images would only deflect from the present goal of developing psychology-inspired machine vision. Consequently, this thesis does not discuss stereo or 3D vision in any detail.

1.4.5 Optic Flow

Strictly speaking, this is not a peripheral issue in machine vision, but is a relevant aspect of the present study. Optic flow can be extremely useful and powerful in many real vision tasks, such as image segmentation, and the separation of figures from background. It is an aspect of Gibson's ecological theory of direct vision, to be discussed later in this thesis. The problem is that we have come to expect to deal with only grey levels and single static images in machine vision. This is part of the legacy of the image processing and the pattern recognition origins of machine vision.

Unfortunately, optic flow analysis cannot be included within the goals of the present project, mainly because of the restrictions on computer resources and project time.

1.4.6 Neural Networks

The neural network revival which began in the early 1980s provides a renewed impetus for seeking novel solutions to the vision problems. However, despite the mass of research material and the supporting publications of recent years, there still does not seem to be a clearly defined route to the understanding of natural vision in terms of these complex neural processes. In some respects, this excess of research may even hinder advances in machine vision, by obscuring the more obvious and direct routes. However, some recent neural network discoveries relating to the vertebrate retina provide evidence for new approaches in the modelling of biological vision.

We now know that natural neural networks are able to "do" complex "psychological things" and that they can be represented as, and to some extent replicated by, simple circuit descriptions. Nevertheless, one cannot make the leap from neural network psychology to computing technology directly. For the foreseeable future, researchers will be compelled to continue using a conceptual crutch, in the form of the high-level AI computer programming languages—such as LISP or Prolog.

Eventually, solid links between psychological processes and biological neural network architecture may emerge. However, for the purposes of this thesis, neural networks shall be regarded as robust and efficient target models of parallel image processing, which can potentially guide our theoretical ideas and our technological implementations.

1.4.7 Supercomputers and Parallelism

It is often believed that supercomputers (by which we mean here very fast serial computers with special cached memory, etc.) are the answer to most computing problems. Similarly, parallelism (of several kinds) has become more prominent in recent years. These mechanisms, although certainly extremely important concepts within computer science, cannot by themselves solve the very deeply entrenched problems of AI, and of artificial vision.

In an honest interview prefacing a collection of his published papers in the book by Yager et al. (1987), Lofti Zadeh (champion of the fuzzy logic approach) makes the following observation:

> "... there are many problems in AI that will not be helped to an appreciable extent by the availability of supercomputers, whether they will be von Neumann-type supercomputers or some other kind. The reason why this is so is because the limitation is not so much computing power, but our lack of understanding of some of the processes required to perform even simple cognitive tasks."

The view of this writer is that success in artificial vision—in relation to human vision—will depend much more on the development of a robust and plausible vision paradigm, rather than computing power *per se*. Nevertheless, adequate computing power—in the form of massive parallelism in both hardware and software—is regarded as crucial in the practical realisation of advanced "computer" vision.

1.4.8 Computer Graphics

There is much common ground between computer vision, image processing, and computer graphics—particularly in the mathematics of projective geometry. Computer graphics can suggest mechanisms which may be useful in machine vision. For example, in the modified Grossberg-Mingolla model described in the present project a concept similar to "flood filling" is invoked to implement regional colourisation. These, and other related aspects are discussed more fully in later chapters.

1.4.9 Autonomy in Localised Neural Control

The final point to be considered in this section is *autonomy*. In natural vision the control over peripheral aspects of vision, such as the diameter of the pupil or iris, the focussing of the lens, and the *foveation* mentioned above, are delegated to purely local neural subprocesses. The brain only provides target goals and global signals. This delegation of control applies in most other aspects of biological subsystems, such as muscular coordination, digestion, blood pressure and blood sugar level maintenance, and so on. Thus, many of these purely local (and subconscious) visual processes are nevertheless extremely complex, and would need an inordinate amount of computing resources and control to model them.

In much of artificial vision research, one can safely ignore this level of complexity, since it normally has no direct bearing on the conceptual goals of the research. For example, a video camera with a manual lens control is a sufficient representation of the eye for most experimental purposes. In the present work, the neglect of such details does not materially detract from the fundamental principles which we hope to elucidate.

1.5 Current Approaches in Machine Perception

In the introduction to the collected conference papers within the volume "Issues on Machine Vision" Pieroni, (1989), wrote

> "A machine vision system should be able to analyze images and produce descriptions of what it sees."

This "analyze-and-then-match" philosophy has been the mainstay of artificial perception, particularly industrial machine vision, for the past three decades or so. The following is a list of just some of the approaches—and the many mathematical models—that researchers have developed over the years:

- vector analysis
- Fourier analysis
- projective geometry
- special relativity
- analytic functions
- neural networks
- cellular logic
- image morphology

These can interpret and explain but some of the visual effects. The various methods are usually not unified, and there is no solid general principle. Indeed, as many different kinds of mathematics seem to be applied to visual analysis as there are problems in machine vision.

Early industrial artificial vision developed in a piecemeal fashion, with highly specific solutions being applied to particular problems. There seemed to be no need in those earlier days for general-purpose robot vision systems. For example, a linear array of photodetectors can be used, in conjunction with dedicated electronics, to measure and verify the principal features of an industrial component. These might typically include measurements of area, perimeter, centre-of-area, and so on. Indeed, to the present day, all that is necessary in a great many industrial vision applications is the verification that holes and patterns have been formed in the mechanical parts. The "tailor-made" solutions were—and still are—entirely viable in many kinds of manufacturing and production processes. They will remain successful provided that they can continue to operate within tightly-constrained factory environments. These specialist industrial environments usually require uniform and stable general illumination schemes, with silhouette back lighting. This has not always been cost-effective or easy to achieve within the normal hazardous factory production and assembly plant.

Increasingly complex production and assembly processes, and the rising cost of human labour, dictated that more sophisticated vision systems should be developed. It was at this point that the real difficulties and the conceptual problems began to emerge. It was soon realised that the problems of artificial vision are indeed daunting. Considering for the present *industrial* applications of machine vision, the following sections briefly describe typical developments. It is worth noting at this point that, to the best of the writer's knowledge, no truly advanced system using "psychological" concepts has so far been demonstrated.

1.5.1 First-Generation Machine Vision

The main principle here is that of extracting salient features from a usually static image, with a view to matching these against prestored lists of features for a limited range of known objects. By "known" is meant that the system has previously been exposed to images of typical objects or artefacts, and that feature lists have been derived and manually associated with the corresponding objects.

The following sequence of operations is typical:

1. Capture and temporarily store a grey-scaled image.

2. Apply a suitable threshold, converting the image to binary form.

3. Extract appropriate features from the binary image. These will typically be geometrical properties of the image, including edges, regions and areas, lines, blobs, clusters, and possibly pixel colour values.

4. Compare the derived image features against prestored feature lists describing the set of known objects.

5. Compute a best match, and hence select (classify) the object.

This is, of course, a simplistic procedure. "Object recognition," *per se* may not be appropriate in all classes of industrial processes; for example, in seam welding by robots. Many manufacturing tasks will continue to need tailor-made solutions, not involving the recognition of objects in the sense implied here. Further details of conventional machine vision approaches can be found in quite recent publications by Chin and Dyer (1980), Freeman (1988), Pieroni (1989).

The relative simplicity of these approaches has been achieved at a cost in generality of the application. There is no simple way by which a First-Generation system can "generalise" to classify objects whose parameters were not explicitly predefined; or presented earlier to the system as target sets of prestored image-feature lists.

1.5.2 Second-Generation Machine Vision

The severe restrictions imposed by first-generation machine vision, and the consequent lack of generality, have forced artificial vision researchers to come up with more advanced proposals. The model for these remains biological vision, but that does not mean that future advanced systems may not develop in entirely unforeseen ways.

It is generally agreed that the failure of first-generation systems is due to a lack of "intelligence" —that is, detailed knowledge about the real world. For example, we are interested in knowing what kinds of objects could project a given pattern on an artificial retina or image sensor. This is often called *image understanding*. Additionally—and more importantly—the *context* in which images occur must be considered, if vision is to be regarded as intelligent.

In keeping with Marr's paradigm for vision (to be discussed later), there has been much activity over the past decade or so to implement three-dimensional (3D) vision systems, in the belief that these can more realistically emulate biological vision. The crucial question, however, is this: Does the acquisition of 3D object knowledge, with its need for two distinct processing channels and hence additional complexity, add significantly to a system's real-world knowledge?

In this thesis we submit that it does not, and offer as evidence the simple fact that image recognition from a photograph, or a television screen (two examples in which all direct 3D information has been lost) is scarcely affected in humans. One cannot easily envisage situations in which a human with normal vision can recognise common objects with both eyes, but fails to do so with one eye covered up! Strange and puzzling medical vision cases do exist, but real examples of the kind of 3D vision situation just mentioned must be extremely rare indeed. Humans can recognise a familiar face with one or both eyes: 3D does not seem to be at all relevant in this situation. In addition, most of us have for many years viewed 2D, grey-tone, 405-line, TV screens and were able to visually understand what was seen—at least, within the context of the TV programme, or scene. This explains why 3D image processing is regarded as only an interesting peripheral issue in the context of the present work.

The 1970s also saw the introduction of the concept of *intrinsic images* (Barlow and Tenenbaum, 1978) in which iconic images could be processed to yield various descriptive (intrinsic) representations of the scene. These include depth information, surface characteristics, reflectance, orientation, and so forth. This is the theme which was also adopted by Marr, in what later became known as Marr's paradigm for vision. This approach continued into the 1980s.

Thus, second-generation machine vision includes proposals and methods for dealing with real-world scenes and images, rather than the often artificial and idealised views obtained in the research laboratory, or on highly constrained and specially illuminated production lines. In practice, it has proved to be extremely difficult to implement many of the ideas. This thesis argues that a reason is lack of understanding of the interactive and complex system aspects of natural (especially human) perception, and the omission of proposals for dealing with the psychological (or "direct" perception) issues that are involved.

It should be noted that there are also differing opinions as to what constitutes a "generation" in machine vision chronology, reflecting the detail of the approaches. For a review, see Batchelor (1991). The writer's view of considering only two generations (basically involving 2D and 3D imaging) concurs with McCafferty (1990).

1.6 Marr's and Gibson's Theories of Vision

The next chapter of this thesis examines the main aspects of two very important, but mutually exclusive theories of intelligent vision. The significance of these two researchers, in particular, stems from their demonstrations of the dual nature of animal vision. That is, although these two theories appear to be totally incompatible, they are in many ways complementary. This compatibility is seen as one of the principal assertions of the present thesis.

David Marr (1945–1980) is probably the best known of the vision researchers. A Cambridge (England) graduate, he left Britain in the late 1960s to study at Cambridge, Massachusetts. His stated reason for doing so was to access the powerful computing resources which were only available in the United States at that time. Marr is linked with the so-called computational approach to understanding natural vision. His work, being of a mathematical nature, has always been relevant to vision as studied by the scientific and engineering communities.

Marr was able to describe the nature and goals of biological vision in terms of his computational theory of vision, which consists of three levels of description, namely: (1) computational, (2) algorithmic, and (3) hardware-based. These points are covered later in this thesis.

James Gibson's ideas appear to be the exact opposite of Marr's. That is, whereas Marr and his followers attempt to demonstrate the complex computational mechanisms in vision, Gibson's adherents reject all such notions of computation. Gibson's is a "direct" theory—meaning that visual information is sensed or "picked up" directly from the visual environment. There is no need, in Gibson's view, for time consuming processing of data. In fact, according to Gibson, there is clearly insufficient time in most fast-moving situations for natural vision to be able to afford the luxury of "computation." This view stems from the very nature of Gibson's early work and career, which included an involvement in the training of military pilots during World War II. Gibson's theory is often referred to as the "ecological" theory. For an overview of Gibson's work, see Frisby (1990b).

Thus, Marr and Gibson apparently represent two entirely opposing approaches to the daunting problems of intelligent natural vision. Marr claims that visual signals require to be processed and analysed, in order that the information can be distilled, and made "explicit." Gibson says, among many other things, that animal and human practical and survival activities, such as catching fast-moving prey, or the playing of tennis, are too rapid for even massively-parallel neural computers (biological brains) to cope with.

In this thesis, we hope to demonstrate that both Marr's and Gibson's theories are tenable. They both support two very different aspects of natural vision, which can both be understood by reviewing some recent neuroanatomical and physiological discoveries. This has significant implications for those seeking to implement natural vision systems in technology. It also provides a contribution to the ongoing debate on natural vision within the relevant disciplines.

A good review of both the Marr and Gibson theories is presented in the respected text by Bruce and Green (1985). Further recent discussions can be found in Frisby (1990a, 1990b). The case against Gibson's ideas is expounded in a number of sources—for example, Ullman (1980).

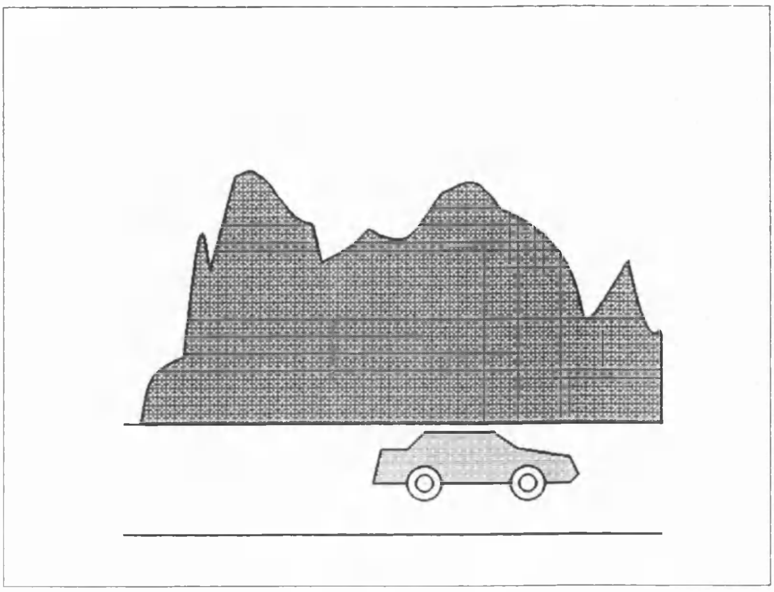
1.7 The "Optical Frame" Paradox

Yet another phenomenon can give an insight into the kind of problems that have to be faced by researchers working on advanced psychological models of vision.

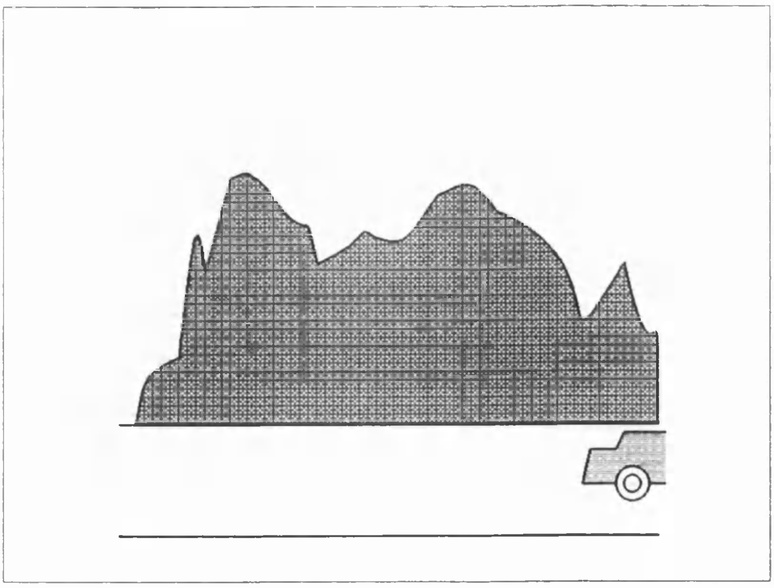
Figure 1.2 shows two simple images, which can be regarded as "frames" or snapshots of a dynamic scene. The images are almost identical except that in figure 1.2b the car has moved to the right, partly out of the frame. The question arises: Are these two images to be regarded as representing the same scene? A human might say "yes" because all significant elements are in both images. An industrial vision system's decision will be based on some sensitivity or distance metric derived from the images themselves.

This problem is caused by the need for an "optical frame"—not to be confused with the AI frame (see e.g. Rich, 1983)—which defines the active area of the image sensor. When viewing this "scene" in a real-world context this phenomenon does not arise for humans because, despite the finite size of our retinas, we do not appear to perceive images in a field-restricted way.

The image-frame problem alludes to the ideas of James Gibson, and his concept of the "optic array" which will be discussed later. Thus at almost every theoretical junction there are perceptual and conceptual differences between biological and machine vision.



(a)



(b)

Figure 1.2 Illustrating the "optical frame" paradox. The above two images are almost identical, except that the "car" in the lower picture has partly gone out of the scene. This kind of thing is typical in machine vision, but humans do not seem to have mental "frames." The question of how many objects—or how much information—the lower image has lost represents an interesting puzzle.

1.8 Towards a Vision Model

In respect of the arguments above, the writer considers that the undernoted goals must be included within any convincing model of human vision:

1. A means of dealing with the abstract psychological processes of human vision, particularly the phenomenon of "direct" vision. Psychology has been a much-neglected subject in machine vision, but is the key to advanced developments in the future.

2. A model of fine-grained, massive parallelism for rapid image processing. This will be provided in the project by simulated cellular automata (CAs), rather than the currently fashionable artificial neural network (ANN) developments.

3. A model of human knowledge and intelligence. This will be provided by a Prolog-based demonstrator, similar in operation to a simple expert system. However, the domain knowledge will be restricted to that required for the understanding of images.

4. A feedback mechanism. It is important that an intelligent vision model is able to interact—via feedback—with the original detailed (Gibsonian) input image.

In order to realise these goals, it is necessary to review some background material in the relevant subject matter. This is done, albeit somewhat superficially, in the chapters that follow. The vision model proper will be developed in Chapter 9.

1.9 Chapter Summary

It is seen that, despite a considerable research effort over three decades—including substantial advances within artificial intelligence and computer science—progress towards the ultimate goal of creating robust and humanlike artificial vision systems has often been disappointing.

Machine vision is highly task-dependent. That is, the kind of vision system appropriate to an application depends on the nature of the application. There is at present no such thing as a "general purpose" vision model that may be applied to a wide range of visual tasks—as human vision can. Nevertheless, many of these artificial vision systems can actually surpass human visual capabilities in restricted application domains. Such systems typically measure and re-present image information that is not readily perceived by humans; an example is the enhancement and subsequent spectral analysis of medical images.

The development of specialised vision systems for both industry and medicine have been the most successful applications to date, and this trend is likely to continue into the future. However, research must now concentrate on the difficult areas involving, in particular, the psychology of human vision. Novel approaches and methods are called for, in order to leapfrog the many philosophical bottlenecks.

This thesis will be concerned mainly with a description of the following aspects of our proposed model of human vision:

- A technical interpretation of the phenomenon of "direct" vision.
- Massive parallelism—realised as fine-grained cellular automata models.
- AI-based vision processes—developed within a Prolog framework.

It is hoped that the proposd broad-front systems approach can expose the vision problems to further discussion, and encourage cooperative interdisciplinary research into the truly difficult areas of natural vision. It will be shown by this approach that many psychological vision phenomena are considerably less daunting than is generally believed.

CHAPTER 2

NATURAL AND ARTIFICIAL VISION

Scientist have been working for at least four decades in an effort to understand why natural vision is so successful. The hope is that a deeper understanding of the many facets of natural vision may lead to advanced technological models of vision. In addition, philosophical spin-offs are likely to be of immense value in the understanding of the other sensorineural functions, and in AI in general.

This chapter considers two seemingly opposing theories of natural vision, as proposed by two of the most prominent vision researchers, David Marr and James Gibson. Although, of course, these two are by no means the only theorists of merit, the works of Marr and Gibson are often cited for illustrative purposes, because they embody what some researchers consider to be the fundamental issues in vision. However, much research has been concluded since Marr's death in 1980, and it is only to be expected that major revisions of theory may be required.

The hypotheses developed by Marr and Gibson are usually regarded as diametrically opposite—antagonistic, even. However, the writer believes that both concepts are compatible with recent neuroanatomical discoveries. For example, Gibson's approach to vision exhibits the commonly observed characteristics of biological systems, such as speed and "direct" perception. As we shall see, these are aspects which can be shown by psychovisual experiments to be real phenomena. Marr has provided us with plausible mechanisms by which perception can be made **explicit**—in terms of the (usually subjective) qualities of retinal imagery. Marr proposed a **computational** approach to the understanding of visual processes. This is something which is tangible, and also seems well suited to the requirements of the physical, and the computing and information sciences.

Gibson's is the kind of fast and "direct" seeing which is needed by an animal, hunting prey or avoiding predators. It is associated with the ideas of reflexive interaction with the environment (hence the use of the term "ecology" in Gibson's writings). Marr's computational vision is said to make objects "explicit," and in this way provides a link to discussions concerning the role of the cortical areas of the brain. But, precisely what Marr meant by the term "explicit" is not always clear:

for one thing, we must not be left with yet another image that—in itself—needs to be further processed and analysed. This is a problem with many of the current stereopsis demonstrators and 3D vision models.

Thus, rather than being forced to choose between two apparently opposing—but probably equally valid—hypotheses, this thesis suggests that both approaches are not only compatible, but necessary. Both researchers represent different aspects of what we call natural, or intelligent, vision; so we recognise that it requires at least these two conceptually different theories to explain it. Later in this thesis, the work of Grossberg and his associates will be shown to modify Marr's paradigm—and thereby providing a "neural network" explanation of Marr-like visual processing.

The implications for the design of artificial vision (machine vision) systems is that any technique which seeks to replicate useful aspects of human vision—perhaps the ultimate goal—must take both kinds of theory into account.

2.1 Levels of Vision

Before continuing with the present discussions it may be worthwhile attempting to establish, if possible, the meanings of the descriptions "low-level" and "high-level" when applied to vision models.

2.1.1 Low-level Vision

Low-level vision (LLV) usually means the processing and manipulation of raw image data with the objective of making explicit basic image features, such as lines, edges, and regions. Low-level vision will normally be carried out "bottom-up" using purely local information, and usually without reference to stored intelligence. In computational terms, low-level vision deals with the manipulation of image picture elements (pixels), and so is often said to be "pixel-oriented."

2.1.2 High-level Vision

High-level vision (HLV) is considered to invoke complex mental and psychological processes, in order to "understand" retinal images. In some sense, high-level vision may not be purely vision at all, but an amalgamation of information derived from vision, the other senses, stored data, intelligence, and psychological factors, including the current context. In contrast to low-level vision, high-level processes are usually "top-down" and "goal-driven." The essential point to note is that much of what one may believe is derived directly from "vision" does not come from physical images at all.

2.2 Marr's Computational Theory of Vision

The logical foundations of a computational approach rest on the Turing proof of computability—see Boden (1988:259). At the practical level, computer logic simulations have provided a rigorously brutal means of translating the often vague ideas of psychology into technologically plausible demonstrators. The conceptual framework for understanding the complex and intangible subject of vision, in terms of a computer scientific approach, was advanced by David Marr (1976). He has had a greater impact on cognitive psychology and visual perception than any other individual vision researcher in modern times. Nevertheless, he is far from being a lone voice: the computational literature on visual perception is an explosively increasing one. This is particularly so since the mid-1980s, with the rapid "connectionist" (neural network) revival, and advances in traditional AI.

Marr's computational theory attempts to explain natural vision in terms of his three levels of abstraction, as mentioned in Chapter 1. The three levels comprise: (1) the visual image computations required, (2) the mathematical algorithms which can achieve these computations, and (3) the necessary hardware to support them.

Marr's work is significant because it is very closely linked with mathematical biology and the study of biological perceptual systems. However, Marr never produced a comprehensive theory of natural vision *per se*. Conventionally, Marr's ideas of visual computations are introduced through Laplacian edge-finding, and the three-dimensional (3D) stereo correspondence problems. However, as was mentioned in Chapter 1, the writer considers that the importance and relevance of 3D and stereopsis may have been overemphasised in the literature.

It is not the modelling of 3D vision systems, however interesting and challenging that work may be, which may solve the daunting perceptual problems. Rather, it is a solid understanding of the essential processes and psychology of vision, and the nature of the visual pathways, which seem to be important. Marr has provided a valuable insight into the visual computational processes, by demonstrating how natural networks of neural cells are able to "do" conventional image processing algorithms, such as zero cross-over edge detection. He provides plausible ideas and explanations, and mathematically sound demonstrators. This is seen by the writer as being the significant contribution by Marr to the science of natural vision.

Thus, Marr's work is characterised by computations, and the notion of data-driven, tokenised, representations of images at an early stage. His approach is commonly referred to as "Marr's paradigm" for vision, within which the derivation of the so-called "2.5D Sketch" is the important subgoal.

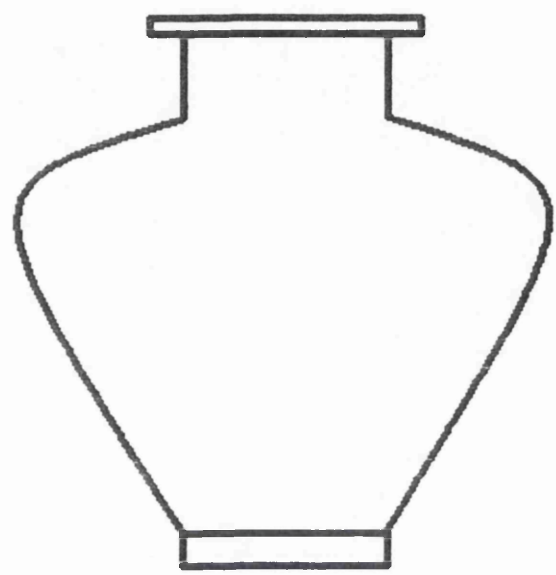


Figure 2.1 A simple caricature representation of a vase. Although there is no internal featural detail in this image, most people readily perceive the outline of a vase.

Leaving aside the 3D stereoscopic vision aspects, and the complexities originally proposed by Marr, the following summarises the sequential steps of Marr's paradigm in the derivation of low-level vision tokens:

1. Capture a grey-scaled image at an adequate level of pixel image resolution and scale.

2. Produce the Raw Primal Sketch. This is a map of local image features, especially line segments, edge segments, and small blobs. This map is derived from image intensity changes, and is normally represented in Cartesian coordinate space.

3. Use further edge detection, together with any relevant cues, including 3D information where available, to produce the Full Primal Sketch. This results from a combination and linking of Raw Primal Sketch features into a more explicit representation: a grouping of sets of edge and region points.

4. Invoke the Full Primal Sketch in association with the original grey-level image to produce what Marr calls the "2.5D Sketch." This uses surface contour information and various "shape-from" processes to further group image tokens.

5. Apply the 2.5D Sketch, in association with stored real-world knowledge (such as memory), to re-create a 3D Object Model.

This so-called 2.5D Sketch is the most complete representation which is achieved by a purely data-driven, low-level, bottom-up approach. It is the most advanced representation of an image which can be obtained under Marr's paradigm for demonstrating natural visual computations. The 2.5D Sketch uses the derived feature maps, in association with "shape-from" algorithms, to deduce the frontal contours of the original image. The final step, if it has to be implemented, is the full object identification and volumetric description—the 3D Object Model.

The terminology "2.5D Sketch" is used by Marr in preference to, say, the "3D Sketch" because it does not represent the projection, nor the full analysis of a complete 3D object. There can only be a single perspective in the retinal image at any one time, and so real-world information and knowledge are needed to reconstruct, or hypothesise, the unseen "rear" of the image. This is one reason why Marr's 2.5D Sketch is likened to a "bas-relief"—the hidden surface is flat. See figure 2.2.

The 3D Object Models stage (object recognition or classification), as far as the writer is aware, was never pursued by Marr in experiment, although Marr and Nishihara (1978) did propose the use of generalised cones and cylinders. Their purpose was to discover the possible 3D image models that could produce a given 2.5D Sketch (a so-called inverse problem). However, there is difficulty here because some objects—for example, a crumpled sheet of paper—cannot be represented in terms of any such generalised cylinders, generalised cones, a "blocks world," nor indeed in any simplistic object primitives description.

It should be noted at this stage that in his step (2) computation, and possibly even in his step (3) algorithm, Marr used stereoscopic (3D) information, in conjunction with Gaussian filtering (or blurring) to extract the relevant spatial image primitives (edges, blobs, etc.) of the Primal Sketch. However, as we have stated above, natural vision does not seem to need binocular vision in order to extract many kinds of information for Primal Sketch maps—assuming that such maps are indeed produced in natural vision. It may be more appropriate to use original colour or grey-scaled image during these earlier steps to compute intensity differences, and thereby find edges, and so forth.

But this anticipates the arguments to be developed later in this thesis. There the ideas of Gibson, and the writer's notion of the "Gibsonian" image—and its perceived relationship to quite recent neuroanatomical evidence—will be be discussed.

Further evidence for the existence of problems in Marr's theory is discussed in Haken (1990:235), in connection with the Marr-Hildreth method of edge detection. Briefly, according to Marr-Hildreth, sharp changes in image intensity are are found by blurring the image with a convolution kernel of Gaussian shape, and then filtering the resulting image with a Laplacian. The spatial distribution of the input image can be recovered by locating the zero-crossing points in the resulting image. The method goes back to Ernst Mach, but Marr's novelty is the use of filters at many spatial resolutions so that *coincidence* of many edge maps suggests the existence of objects. The point that has to be emphasised here is that "edges" themselves do not contain textural detail—information is thrown away, or pooled (as happens in the notion of the so-called "Grandmother Cells").

Methods of edge detection, similar to those just described, are used by the writer in his Edge Constraint Map (ECM) to be discussed later in the thesis. However, Marr-like edge detection provides only part of the evidence accumulated within the ECM.

Figure 2.2 illustrates the essential steps of Marr's paradigm, using as an example Marr's famous "teddy bear" image. It is seen how the 2.5D Sketch must appear as a kind of bas-relief, because the hidden information needed to re-create the 3D Object Model is not available. As mentioned before, this representation arises because the imaging system (even one using a 3D stereo pair) cannot, by itself, know what lies behind the planar image. It could, of course, make a reasoned and informed guess, but this suggests higher level knowledge which Marr had never provided. Note particularly that there is no mechanism within Marr's paradigm itself that specifies how images should be processed to yield information for the storage and subsequent recall of a computer-based—or a biological—visual memory.

The mechanisms of Laplacian neural cell edge detection are described in Marr and Hildreth (1980). Marr's paradigm, although not by any means a complete theory of vision, is widely regarded as being among the best and the most advanced approaches to date. However, to the writer's knowledge, no practical demonstrator using *all* of Marr's principles has yet been built.

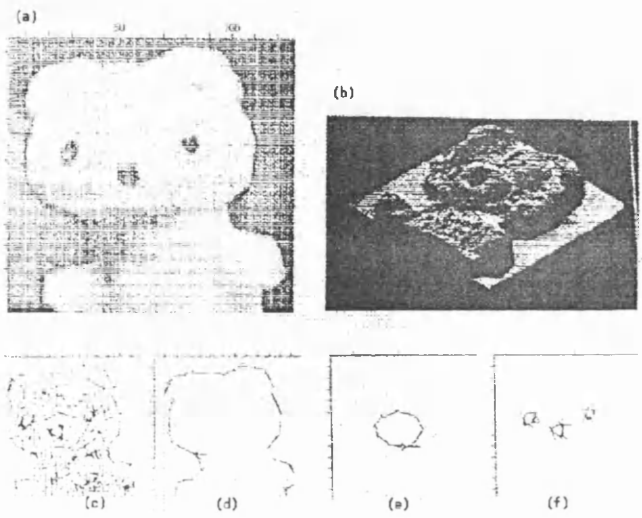


Figure 2.2 Details adapted from Bruce and Green (1985) and Marr (1976), showing some aspects of Marr's derivation of the 2.5D Sketch.

(a) The well-known "Teddy Bear" image.
(b) An intensity map derived from (a).
(c) Location of small edge segments in Raw Primal Sketch.
(d) A featural grouping set.
(e) A featural grouping set.
(f) A featural grouping set.

The grouping sets at (d), (e), and (f) emerge as a result of purely low-level (bottom-up) processes, and do not require any high-level knowledge about the image context. This is different from some other theories of vision where expectancy and object-hypothesis can play a role in featural determination and segregation.

The final 2.5D sketch emerges as a result of applying "shape-from" processes to yield 3D information. The final form of the 2.5D Sketch in fact resembles the intensity level image shown in (b) above, the difference being that range information—rather than intensity—is depicted. The flat rear of the image accounts for Marr's use of the unusual term "2.5D Sketch."

2.3 The Ecological Theory of Vision

This somewhat grandiose subheading is concerned with the work and controversial theory of James Gibson—a vision researcher who had been active for more than 35 years. See Gibson (1950, 1966, 1979). His last known publication appears to be Gibson (1979). Gibson's work is formally known as the "direct" theory of visual perception. In its mature form, it states that the flux of light (which Gibson calls the ambient optic array) reaching the visual sense organs of an animal is richly structured by the nature of the environment, and by the movements of objects within it. The "optic array" is, therefore, assumed to be made up of variants and invariants, created by the patterns of light flux due to both the ambient reflectance and transmittance. The optic array exists at any arbitrary point in 3D space, and the concept at this level is really only a restatement of the physical laws of light, such as are used in optics, and for the lighting design in buildings. However, it has important implications for psychological vision.

It is the invariants which are held to be sufficient, in themselves, to specify relevant aspects of the visual environment. For example, in simple species the movement of an object, and the corresponding changes in the light patterns of the optic array (as sampled by the retina) due to both static and dynamical features, will be sufficient to trigger a reflexive response from an animal. The near-instantaneous responses would typically be "attack" or "flee" — depending on the prestored neural memories (responses) of the particular animal species. This is often called "instinct."

Gibson's theory can be extremely complex, and involves such notions as "affordance", "resonance", etc. We shall not pursue this in any detail, but we note the importance of the term "direct perception," and Gibson's claim that all of the information needed to perceive the environment is contained within the optic array. We also need to be aware of the claims by supporters of the direct theory of vision that image processing, and hence visual memory storage, are not needed in natural vision. This is possibly the most controversial aspect of what is an extremely controversial theory. Other Gibson ideas, such as those of optic flow, have been rather well received. For example, Koenderink (1986), who has mathematical research expertise in optic flow, regards Gibson (a non-mathematician) as a "genius." So it is easily understood why the debate on Gibson's work continues even now.

Further details and good summaries of Gibsonian ideas are to be found in recent texts, such as Bruce and Green (1985). Frisby (1990ab), and Roth and Frisby (1986) give useful introductory material on both Marr and Gibson. These sources contain references to selected readings, which can be used to build detail on the controversial theory.

Gibson's theory of direct perception, as we have suggested, contrasts sharply with the indirect theories, especially those which propose a computational mechanism for the analysis and subsequent classification of visual data—as developed by Marr. There is a relatively small number of vision researchers who believe in Gibson (although not all are uncritical of the extreme aspects of the theory) and many more who reject him completely. For an example of the kind of arguments ranged against direct perception, see Ullman (1980). The struggle that many have felt in trying to do justice to Gibson is well brought out in a personal comment by Hinton, who wrote

> "... how could someone who says so many sensible things about perception maintain that perception is direct and does not involve computation? Either Gibson is being very silly, or there is a deep misunderstanding of what it means for perception to be direct."

Referenced in Frisby (1990b), and quoted originally in Ullman (1980:387).

This thesis adopts the view that Gibson's basic concepts are a manifestation of real and fundamental neurophysiological aspects of biological vision, which we shall consider in outline later. We also agree with Ullman (1980), in that there is a strong tendency towards convergence in theories of vision today: most differences are more a question of focus and emphasis, than genuine reasons for conflict. It is hoped to justify this view in later discussion on neuroanatomical discoveries.

2.4 Marr and Gibson Compared

The following table lists, very informally, pairs of characteristics or attributes of both Marr's and Gibson's vision theories. This is by no means a rigorous taxonomy. It is provided to highlight the opposing nature of these two important approaches to natural vision. Not all of these characteristics will necessarily be present in any single specific vision concept, or model implementation.

TABLE 2.1: MARR AND GIBSON ATTRIBUTES

Gibson Attribute	Marr Attribute
fast axons	slow axons
direct	indirect
iconic	symbolic
immediate	inferred
explicit	implicit
reflexive	inflexive
spontaneous	delayed
higher-level	lower-level
non-memory	memory-based
detailed images	impoverished images

2.5 Other Theories of Natural Vision

Besides Marr and Gibson, there have been many other theorists and researchers in this field. We shall not attempt here to cover these, but shall mention only one: this serves to illustrate the severity of the problem, and the variety of the proposed solutions. Minsky (1975) argued that our knowledge of an object is often a synthesis, built up from our viewing it from a large number of different perspectives. He gives as the prime example the recognition of the human face. We can easily recognise a familiar face (even from a "flat" two-dimensional photograph) when viewed the right way up, but most people find great difficulty when the same face is viewed upside-down. This very basic observation contradicts the theories of some vision researchers, since when a face is viewed upside-down all the points of it are maintained in the same spatially relative positions with respect to the internal coordinates in the the brain and head.

2.6 Data-Driven or Goal-Driven Vision?

Linked with the discussions above is the question of whether visual processes should be data-driven (low-level) or hypothesis-driven (high-level). In data-driven systems, low-level processes proceed to completion bottom up—that is, from raw sensory input data to a final image interpretation—without any input or control from high-level processes, such as psychology. In goal-driven systems (that is, the hypothesis-driven approach) a goal is created top-down by some active brain state, or *schema*. The vision system is supplied with certain relevant parameters to seek support about specific hypotheses concerning the scene: for example "now go and find the red pen on the desk." A schema (or "demon") that can potentially satisfy the goal gets instantiated, and the system proceeds until some predetermined stopping condition is reached.

We argue in this thesis that human vision does, and machine vision should, make full use of **both** bottom-up (iconic) and top-down (symbolic) mechanisms. This leads to the idea of using multiple-level processing within machine vision.

2.7 Chapter Summary

Natural vision is clearly extremely complex and psychological-based. There have been many theories which have attempted to explain visual function in humans and animals, but those of Marr and Gibson appear to offer the best conceptualisations so far. Marr's is a more readily understood computational approach, while Gibson's is a controversial and "direct" phenomena—the so-called "ecological" theory.

A serious hindrance in artificial vision research is the lack of the equivalent of a "schematic diagram," or fundamental knowledge of how biological vision actually works. In the present thesis, Marr's and Gibson's theories can provide us with a conceptual crutch—a useful starting point for the important task of merging both biological and technological concepts to produce a plausible model of vision.

CHAPTER 3

NEURAL NETWORKS

After a promising start with the publication of the now classical study by US researchers McCulloch and Pitts (1943), which led to Frank Rosenblatt's much-acclaimed Perceptron device in 1963, interest and research in neural networks and neural computing waned, and faded into near oblivion. This state of affairs lasted for almost two decades, with only some "clandestine" material being published (often in heavily disguised forms) in a variety of obscure journals and psychology textbooks. The present neural revival began in the early 1980s, and the study of neural systems is presently among the most rapidly expanding areas of research and technology, attracting academics from a wide variety of related disciplines.

In this chapter, an attempt will be made to review a few relevant aspects of neural networks. We shall concentrate only on those ideas and methods which appear to be of relevance to machine perception; in particular, those "neural" mechanisms and local processes which are active in natural vision processing. Later in this thesis, some of the interesting artificial neural network (ANN) mechanisms will be shown to have much in common with cellular automata (CAs).

As already mentioned, the study of neural networks, both natural and artificial, is an explosively increasing activity. One must therefore be restrictive in the selection of techniques, and in the topics for research. Only those aspects which can potentially contribute to the understanding of natural vision can be considered. The objective in all simulations must be to secure maximum processing capability with a minimum of complication.

Fortunately, much of the complexity in the literature on neural systems can be disregarded here, since only a very small subset of concepts needs to be distilled from the mass of research material generated over the past five years. For example, discussions and theories relating to membrane potentials, axonal pulse-coding and timing, inter-neuronal oscillation, and so on, will probably be of little significance in image processing. Consequently, it is not proposed to review that kind of material here. There is a rapidly growing number of monograph texts and research proceedings published since about 1987, covering introductory, intermediate, and advanced levels in neural systems which the interested researcher can consult. Recommended among the many are Arbib (1989), Beale and Jackson (1990), Aleksander and Morton (1990), Muller and Reinhardt (1990), and Domanay et al. (1991). In addition, at the time of writing, several new neural network titles are in the final stages of publication.

3.1 The Neural Network Revival

Neural networks have been receiving much research attention since the beginning of the 1980s. The modern beginnings of the subject can be traced to a 1943 landmark paper by McCulloch and Pitts. These two US scientists used the mathematical notation of Whitehead and Russell to establish a theoretical basis for the logical properties of a matrix of interconnected (idealised) model neurons. The concept is broadly similar to the programmable logic designs of today. This theory was received enthusiastically, and during the next two decades useful results were obtained for a number of mainly electronic neural network demonstrators.

The most well-known include Rosenblatt's 1963 *Perceptron* for simple visual pattern recognition. However, two other prominent US researchers, Marvin Minsky and Seymour Papert, demonstrated important computational defects in the Perceptron-class models, and effectively scuttled academic interest (and funding) in artificial neural network research worldwide. This happened (around 1969) because of Minsky's personal and academic standing, and his considerable influence on computer science in the USA. It was not until 1982 that another US researcher, John Hopfield, proposed a simple answer to the challenge posed by the Minsky and Papert book. Since then, the study of neural nets has once again become a fashionable subject.

It is generally accepted that the study of neural networks is a study of both natural and biological networks. That is, there is now a very strong interdisciplinary involvement, and simultaneous investigations of both natural and artificial processes. Therefore, the use of the broad term "neural network" must always be related to the current context. The science of neural networks benefits from an interchange of ideas between natural and artificial neural research, and between the interested disciples. Closely related subjects, such as cellular automata, can benefit by borrowing relevant concepts from neural network theories. Such methods include the theory of "spin-glasses" and the simulated annealing of metals—ideas that have been borrowed from physics. See Meuller and Reinhardt (1990), Domanay et al. (1991). But these techniques are not in any sense restricted to artificial neural networks: cellular automata and information theory can hope to benefit also from the convergence of ideas and methods.

3.2 Why Neural Networks?

Historically, the interest in neural networks has two origins: (1) a desire to understand the basic principles on which biological brains work, and (2) the need to build machines which can carry out complex tasks for which the conventional serial computer is not well suited. The first goal, although fascinating in itself, is of less importance to the present work than the second. Although conventional computers are generally excellent for modelling and numerical processing, they are quite inefficient when it comes to the elementary and everyday tasks which we humans take for granted.

The goal of machine vision must be to discover the essential neurobiological and associated psychological or mental processes of natural vision, in order that these factors can be captured in technology. But this does not mean that technology needs to emulate Nature's methods in a slavish way.

3.3 Parallel Distributed Processing

One of the techniques which Nature obviously employs is massive parallelism, and the delegation of neural function to many areas of both the central nervous system (CNS), and the peripheral nervous system (PNS). This has many important advantages, some of which will be discussed later. However, it is still not understood how the many neural mechanisms are integrated, coordinated, and restrained to form stable (i.e. non-oscillatory) and robust control systems. Positive and negative feedback loops obviously play their part, but there are many neural processes that are not yet understood—even in simple terms.

Common observation shows that the brains of the lower animals are capable of performing tasks that are far beyond the capacity of even the best and most powerful of today's electronic (serial) computers. Indeed, even the microscopic brains of tiny insects are more powerful than a battery of microprocessors. However, the computers which are usually compared with biological brains are the conventional serial processing types—often referred to as the standard von Neumann-type architectures. [1]

These computers, developed over the past four decades, have proved to be ineffective in AI applications for a number of reasons, one of which is the relatively slow speed of conventional solid-state logic elements—the most basic components of all computers. Developments in microelectronics are rapidly approaching a point where little or no increase by orders of magnitude in component density can be expected from traditional silicon technologies. This creates the impetus for basic research into novel mechanisms and architectures. A brief review of these problems, with some potential solutions, is presented in the paper by Barker (1990), and the references contained therein.

3.4 The Basis of Artificial Neural Systems

It would be futile to pretend that a proper analysis of these very complex systems could even be attempted here. The present goal is to describe in outline the basics of artificial neural systems (ANS), and the state of understanding of neural networks. Section 3.4 considers how this knowledge is being utilized in current artificial neural systems research. The starting point for the discussion is the neuron model of McCulloch and Pitts. It will be convenient hereafter to refer to this basic model as the McCulloch-Pitts Neuron (or MP Neuron).

An important issue in neural networks is the classification and the taxonomy of their neuron-like functions. For example, neural networks may be implemented as associative memories, I/O controllers, filters, oscillators, transducers, effectors, amplifiers, and so forth. Thus, although one may talk about "neural networks," the purpose and relevance of neural computational models have to be matched to their equivalent biological functions. An example is the electronic neural networks used as associative memories, where the sought characteristic is the network's ability

[1] After John von Neumann (1903-1957), scientist, mathematician, and pioneer in the design and development of digital computer architectures.

to complete or "fill-in" partially obscured images, etc. The speed and efficiency of such systems is often quite irrelevant, since it is only a specific property of the neural network that is desired. The example of associative memory is one of the classical applications of neural networks, pioneered by the Finnish researcher Teuvo Kohonen (1990), and developed by others.

On the other hand, the efficiency of a modelled neural network can be extremely important. For example, an associative memory model may be of little practical use if it can store only a few images or patterns. Understanding—and predicting—the pattern storage capacity of the various models of artificial neural networks is a research subject in its own right, and will not be discussed here. As will be seen later, cellular automata implementations of equivalent neural processing and transducing functions may in many cases provide a superior solution to artificial neural network implementations.

3.4.1 McCulloch-Pitts Neuron Model

The basis of our present understanding of the central nervous system (CNS) and the peripheral nervous system (PNS) is the hypothesis that neural function is determined by the passage and mediation of pulses through networks of cells called *neurons* (UK: neurones); although there are in fact many more *glial* cells in the CNS and PNS than neurons. For the present it is convenient regard both the CNS and the PNS as vast networks of idealised neurons. In the CNS (which includes the brain and the spinal cord) neurons are arranged in laminar structures, having a complex and dense matrix of interconnections. In the PNS, neurons are mostly superficial and of the sensor type.

There are many different kinds of neurons, as suggested by figure 3.1a. Figure 3.1b illustrates what might usefully be called a "typical" biological neuron. The neuron is a cell, and so contains a nucleus which is situated within the *soma* or body of the cell. The inputs to the neuron are the *dendrites* which form tree-like filaments or fibres connected to the soma. The *axon* is the single narrow output fibre which normally branches or splits into many thousands of finer filaments contacting with the dendrites, the soma, or sometimes even the axons of other neurons. Note that there is only one signal output from each neuron, which divides or "fans out" to distribute the neuronal response to neighbouring cells. The axon in CNS neurons is usually *myelinated* to enable the amplification of pulses which often have to travel relatively long distances. Myelin is really just another name given to the glial cells which completely surround CNS axons. The shorter axons of PNS neurons (e.g. those of the skin) do not normally have myelinated sheaths. See, for example, the numerous sources quoted in Churchland (1986), and Brown (1991).

The points of contact between the axonal fibres of a neuron and the dendrites of other neurons (in the same or different neural layers) are very important and relevant. These are called *synapses*. Each axonal fibre terminates in a tiny endbulb synapse, which is in contact with either the dendritic inputs or the soma membrane of adjacent cells. It is the influence of the summed and thresholded "weighting" factors of the many input synapses which is significant. This mechanism provides the complex processing capability, and the adaptability (or learning potential) of biological nervous systems. The neuron internal transform function can take several forms, from a simple stepped function to more complex sigmoidal shapes.

In the simplest theory, each neuron soma integrates and thresholds the input impulses from its many synaptic connections, and may then fire a series of impulses along its axon depending on the internal status of the nucleus, membrane potentials, and the neuron threshold function. Thus neuronal activity is based on pulse frequency principles, rather than on analogue potential levels: see Duggan (1991). Two forms of synapse are known to exist: *excitory* and *inhibitory*. The former, as the name suggests, excite internal neuronal activity, while the latter suppress it. It seems to be a complex electrochemical balance which decides whether a particular neuron will "fire" or not. Figure 3.1b shows the pseudo-logical model of the MP neuron. It must be emphasised, however, that the accepted analytical methods of conventional binary logic system gates, logic circuit diagrams, and Boolean logic truth tables are not considered appropriate by present researchers to the deeper understanding of neurobiological function. The MP neuron is not simply a model of an binary AND gate, despite appearances.

Not all neurons are typical. Some are extremely untypical, but the general principles of operation apply equally to all neurons, in much the same way that transistors can be categorised into high-frequency, low-frequency, power drivers, and other specialist types, although all are basically three-terminal, solid-state, amplifying devices.

(a)

(b)

Figure 3.1 The upper figure shows but four of the many and varied forms of biological neurons: see Brown (1991). The lower diagram shows a "typical" neuron, and its pseudo-logical equivalent diagram. The latter is a McCulloch Pitts (or MP) neuron.

3.5 Neural Networks and Memory

One of the most important functions of biological neural networks is the provision of **memory**. This is an emergent property, since neural networks appear to store information in both the synapses and in vast cellular interconnection matrices existing throughout nervous systems. This is very different from electronic computer memories, where data is stored and accessed in sequential locations. Neural memory is one of many distinct categorisations of neural function.

The book by Cohen et al. (1986), an Open University standard text, provides an up-to-date survey of current memory theory from the perspective of the cognitive scientist. Concepts of associativity are manifest in the associative memory forms of artificial neural network, as advocated by Kohonen (1984, 1988, 1990), and others.

On the other hand, memory, as a "computer" science or a physics discipline is now studied in great detail from the mathematical viewpoint. Kohonen (1987), for instance, cites more than 1900 references in his standard text on content-addressable memories.

For the purposes of our model, it is convenient to adopt the computer science concept of memory as a sequentially-addressed storage and retrieval mechanism. As will be seen, this concept of memory appears in the present work in two guises. The first is the storage and reproduction of knowledge-based material, as required by our Prolog model of intelligence. The second application is related to the concept of "virtual memory" as advocated by Kohonen (1990). This latter form is used to model *expectancy* in high-level vision, and will be discussed again later in this thesis.

3.6 Learning in Neural Networks

This is another huge and potentially boundless subtopic. Learning in neural networks is believed to occur when modifications are made to the effective coupling between neural cells, at synaptic junctions. However, as was mentioned before, a neural network is a matrix of interconnected cells, and so there are actually **two** mechanisms by which the effect of the network can be altered: (1) by adjustment of synaptic strength (weights), or (2) by changes in the axon/dendrite interconnection matrix (the axonal circuitry).

3.7 Chapter Summary

This chapter has provided only a superficial review of natural and artificial neural systems. It has of necessity concentrated only on a very few of the many neural network theories and mechanisms which can potentially contribute to better technological models and systems. The recent revival of the science of neural networks, and the expanding "neural engineering" philosophy, are among the fast developing topics in science. Entire volumes could easily be devoted to a single aspect of the subject—hence the selection of relevant biological neural network concepts is a vital part of neural systems emulation.

In this project, we are interested in neural networks as pointers to novel methods and algorithms for a computer simulation of appropriate image processing and vision mechanisms. In general, only the simplest and the most relevant mechanisms are sought.

The list of useful neural properties is large, and includes the various kinds of memory models, mass-action (shunting) and adaptive neural signal processing, and the modelling potential of the many specialised kinds of regional neural network. Specifically, we are interested in those forms of neural processing that can potentially be carried out by simpler discrete cellular automata (CA) mechanisms.

CHAPTER 4

CELLULAR AUTOMATA

The recent neural network revival has been accompanied by a renewed interest in the properties exhibited by arrays of relatively simple processors having local neighbour connections. Such cellular array systems are currently being used in the modelling of mathematically complex physical processes, such as lattice gas properties and the heat equation. They are also closely associated with the developing mathematical interest in chaos theory and fractals.

This chapter considers cellular automata (CAs) as especially relevant mechanisms in the modelling of machine vision. It is certainly not a requirement of the approach described in this thesis that CA methods need be used: indeed, most image processing applications are carried out in the standard serial computing domain. Nevertheless, there are compelling reasons why CAs are rapidly becoming important in image and vision processing, and in many similarly related fields of scientific application. Potential benefits of CAs include: massively-parallel computation, very high processor speeds, the simplicity of rules and programming, and the ability to scale problems. In addition, future developments in very high-density hardware and molecular computing are likely to be directly concerned with CA implementation.

Most of the above advantages apply also to artificial neural networks: so the arguments in favour of ANNs will be applicable to CAs. Indeed, the very close relationship between CAs and ANNs will no doubt prove to be profitable. One can even speculate that CAs, like ANNs, will likewise benefit from future discoveries in fundamental neuroanatomy and neurophysiology.

A special reason for proposing CAs in vision work is the recognition that the Gestalt Laws of Organisation (to be discussed in Chapter 5) are framed within the concept of "magnetic force fields" (and by implication the physical mechanisms of soap films, rubber membranes, and elastic bands). These are properties which the Gestalts held to be responsible for attracting (that is, enforcing) perceptual groupings in biological vision—especially in human perception. CAs appear particularly well suited to the modelling of most kinds of physical phenomena, such as dynamic fluid flow, turbulence, thermal, magnetic, electric and electromagnetic fields, and therefore can be shown to be relevant in the present work. More specifically, certain physical phenomena such as the annealing of metals and the theory of atomic spins (or spin-glasses) can be considered as useful analogues of Gestaltist vision theories.

An important aspect from the perspective of this thesis is the ability of CAs to emulate "conventional" artificial neural networks. It will be demonstrated that many local and global features of ANNs can be modelled by CAs. In particular, the neural net model of Grossberg and Mingolla (1987) can be simulated on CAs. As will be shown, this is a basic image processing paradigm that can support many of the features discovered in biological neural networks concerned with vision.

In essence, what we seek from cellular automata models in this work are their image-pattern modification capabilities: that is, cellular computing mechanisms and CA algorithms that can be applied directly to the processing and subsequent logical (morphological) modification of both natural and synthetic images.

4.1 General Description of CAs

The essentially local nature of CAs makes this model eminently suited to image processing and machine vision, where localised processing and massive parallelism are essential. Despite the localised nature of CA processing, important global properties can emerge.

Cellular automata (CAs) are fully discrete and dynamical systems, and thus can be described in terms of a set (often taken as infinite) of finite-state automata (FSAs) which "fill" a defined space or lattice in d-dimensional Euclidian space. The terms CELL, LATTICE-SITE and AUTOMATON are used interchangeably. Every cell or lattice-site has a small set of neighbours, usually those sharing a "face" or "edge." The number of neighbours is typically 4, 6, or 8, depending on the neighbourhood geometry. Sometimes more complex neighbourhoods are defined. A cell having eight neighbours is said to be 8-connected and square (Moore neighbourhood). A cell having only four neighbours is likewise 4-connected: the neighbourhood is cross-shaped (von Neumann neighbourhood). A cell having six neighbours is hexagonally-connected. These are illustrated in figure 4.1.

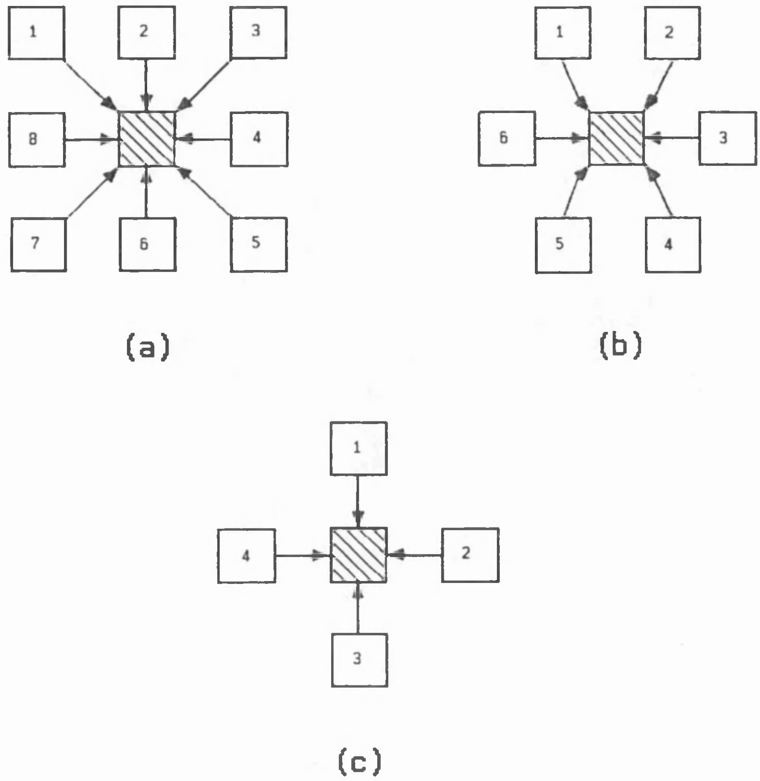


Figure 4.1 Showing the three most common CA cell neighbourhoods, or surround directions.

(a) The SQUARE neighbourhood—preferred in the present work.
(b) The HEXAGONAL neighbourhood.
(c) The CROSS neighbourhood.

The square cell neighbourhood is 8-connected, and is thus similar to the 9-element kernel (or template window) used in conventional image processing algorithms. The square configuration suffers from the so-called "connectivity paradox" with respect to pixels lying on a diagonal dividing line. This aspect of CAs and cellular-based systems is discussed more fully in a useful text by Duff and Fountain (1986).

The central element (shaded) is the current cell. In most CA work the inclusion of the central cell in a 9-sum constitutes a kernel; if only the eight nearest neighbours are involved the resulting 8-sum is often referred to as the neighbourhood sum.

Each automaton has a set of "states" $\{q_0, q_1, \ldots, q_m\}$ its "next state" being a prescribed function F of the state transition function of its present state and the current state of its neighbours. It is usual, though not essential, to have the same (universal) transition function applied to all cells in the array or lattice.

An assignment of states to all cells is a configuration c_t of the CA at time t. A global transition function F can be defined as

$$c_{t+1} = F(c_t) \tag{4.1}$$

in terms of function F applied simultaneously (that is, synchronously) to all cells or lattice-sites.

The initial cell configuration is c_0 and a "computation" by a cellular automaton then consists of a sequence of configurations $c_0, c_1, c_2, \ldots,$. It is usual to have a "halt" configuration c_h such that entering it stops the computation. It is clear that c_0 will "grow" to c_1, c_2, etc. on iteration of the transition function, thereby unfolding the dynamics of the system. The transition function is usually called a "rule."

The maximum rate at which active states "propagate" into the quiescent cellular surround is given by the size of the "steps" as defined in F. It has become customary to call this the "velocity of light," and is usually taken as one cell per iteration (or clock pulse). Conventional CAs often use the "bucket brigade" method of passing information over and out of cellular array.

Practically, each cell site can be realised as a discrete computing element—or processing element (PE). A cellular automaton is said to be "tightly-coupled" if it interacts only with its immediately adjacent nearest neighbours. It is this inspection of neighbouring data (or states) which allows two-dimensional CAs to be so useful in the analysis of 2D arrays or data structures, particularly those representing pictorial patterns and images.

As mentioned, operations are assumed to occur in discrete time, with each time step being a generation, iteration, or cycle. It is further assumed that all changes take place simultaneously; that is, the action of elements within the cellular array is *synchronous*. Finally, it is assumed that the same transformation rule applies at every individual site in the array, and that every transform is deterministic. These are the usual assumed conditions but important variations have been proposed.

In more precise mathematical terms, a cellular automaton is given by a tuple $\langle G, V, Q, F \rangle$ where:

G is the *cellular space* (usually a regular lattice)

$V = \{ i_1, \ldots, i_s \}$ is the neighbourhood

Q is the finite set of states

F is the local transition function, or "rule"

The function F associates a new state to each state configuration in the CA neighbourhood, that is

$$F : Q^s \longrightarrow Q; \quad (x_1, \ldots, x_s) \in Q^s \longrightarrow F(x_1, \ldots, x_s) \in Q \tag{4.2}$$

where $s = |V|$ is the cardinality of the set V.

As stated, the dynamics of the automaton is given by the synchronous application of the same local function to every site in the cellular space; that is, for any given site i

$$x_i(t+1) = F(x_{i-i_1}(t), \ldots, x_{i-i_s}(t)) \tag{4.3}$$

Although CAs are relatively simple in concept, potentially powerful processing capability and emergent properties, including "neural" computations, have been demonstrated.

Individual cellular sites can be realised in VLSI hardware, as arrays of simple microprocessors. They may be controlled by a common set of simple arithmetical and logic (Boolean) instructions, or be software simulations in general-purpose computers. There are prospects for the future development of very high-density molecular electronics which can support CA mechanisms. See, for instance, Barker (1990).

In the present project, CAs are software-simulated on conventional PC platforms in order to conveniently develop and evaluate suitable image processing and vision algorithms. The relevant details will be covered later.

4.2 CAs and Artificial Neural Nets

Artificial neural networks (ANNs), as discussed in Chapter 3, are based very loosely on their biological counterparts: the state-space can be very large indeed. CAs, on the other hand, usually have a much smaller state-space, and they often have no more than eight local neighbours. There are exceptions, but for most applications, and especially artificial vision, eight local neighbours has been found to be adequate. In this thesis, CAs are regarded as simplified models of ANNs.

Figure 4.2 illustrates a binary CA lattice. As in all cellular-based models, it is the massive interconnection and interaction of the many individual cells (or network neurons in the case of ANNs) that can produce meaningful computations, and the useful emergent properties.

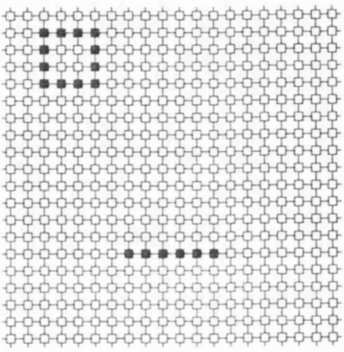


Figure 4.2 The figure above shows a two-dimensional, 4-connected, cellular lattice or automaton. This consists of a regular spatial array of identical cells located at positions X,Y in the lattice. Each cell has data input/output ports, and every site is characterised by a finite discrete-valued state function. In this project, three-state arrays are used to represent one plane of a 64x64 grey-scaled or colour image. Binary patterns are represented in an obvious way, but more complicated procedures are required for the representation of multi-state and multi-plane images.

At successive discrete time intervals each cell makes a transition to a new state—determined by a transition function or RULE (mapping) which depends on the previous state values of the connected neighbours and the cell's own status. The above diagram shows a simple initial configuration of the automaton. Iteration of the mapping will unfold the dynamics of the CA, which is thus able to provide the meaningful computational properties in the domain of machine vision algorithms.

Practically, each individual cell can be realised as a microprocessor, a hard-wired logic function, or an embedded systolic-array element. In the the present work the CA, and its associated transform rules, have to be software simulations.

4.3 CAs and Image Processing

It was mentioned previously that CAs are finding applications in many diverse fields, including computer graphics, the modelling of complex physical processes, and the simulation of biological cellular systems. What is sought in the present work are mechanisms of visual image pattern modification which can be further developed to model the Gestalt Laws of Organisation (including the need to hierarchically order image and related data). These basic laws, as mentioned, are concerned with image mechanisms of soap bubbles, soap films, rubber membranes, and elastic bands, and are discussed further in the thesis. Unpredictability and instability in CAs, as modelling concepts, may be considered a problem in vision and image processing. This is due to the essentially chaotic and stochastic nature of cellular interaction. However, as will be seen, it is possible to control such phenomena by the use of appropriate stabilising feedback mechanisms.

Practically, in CA methods a digitised computer image is regarded as a 2D array or lattice of cell sites, which can be considered as coloured dots (pixels) or rectangles. Each cell in the lattice is repeatedly "updated" by changing its old cell numerical value (or its colour) for some new value, in accordance with a transformation "rule" which is applied to all cells simultaneously. A mechanism of this kind is called a **cellular automaton (CA)** if these processes can be shown to be:

1. parallel
2. local
3. homogeneous

This is the strict and formal definition of a "true" cellular automaton. However, in future applications including machine vision, it may not be necessary—or practical—to comply rigidly with all three of these conditions simultaneously. It could be that locality, for example, need not be confined to a cell's nearest neighbours as this could be uneccessarily restrictive. This prospect arises when considering the so-called "bus-automata" (BAs) of Rothstein (1987). Nevertheless, in any pure CA implementation the above mentioned three requirements should be strictly adhered to. The principal features of pure CAs are discussed in the following sections.

A further point should be made here. CAs intended for image and vision work share much in common with traditional image processing algorithms—particularly those based on conventional 3x3 template or window operations. As will be seen, the association between kernels and 8-connected neighbourhoods is a useful one. CAs clearly have much in common with cellular logic image processing, as exemplified by the recent work of Duff and his Group in the UK, and their very successful CLIP (Cellular Logic Image Processor) range of machines.

4.4 Characteristics of "Pure" CAs

The three characteristics of the stricter definition of a cellular automata are discussed briefly below. Important work on the **statistical** properties of one-dimensional (1D) CAs was performed by Wolfram (1983, 1984), but is not considered here.

4.4.1 Parallelism

This requires that every individual lattice site or cell be updated independently. It may also require that the system be "clocked" such that each cell is updated simultaneously. This is *synchronous* operation. It is possible that non-clocked or *asynchronous* systems will be proposed, depending on the mechanics of the model or the technology in which the CA is implemented. In a standard serial computer implementation of CAs parallelism is not possible, and so must be simulated. A buffer memory is required to store intermediate cell states until the entire array, or frame, of the CA is updated.

Implementing truly parallel CAs is likely to be a major challenge in the future. Each cellular site is ideally an independent processor having its own local memory, I/O channels, and microcode. Alternative implementations using Boolean logic functions have been proposed in the literature, e.g. Barker (1990).

4.4.2 Locality

This means that when a lattice cell is updated the cell's new value is based *only* on the "old" value of the cell, and its nearest neighbours. The pattern of a cell's nearest neighbours needs to be fully defined. In addition, the effect of the cell's own current state usually has to be considered. In simple CA rules the own-cell state may be included only in a bit-count (9-sum), or perhaps a complicated transformation may be defined (an 8-sum, plus some external signal defining action). These points are covered in more detail below.

4.4.3 Homogeneity

This requires that every lattice cell be updated in accordance with the same (current) transformation rule. Usually the summed value of a cell and its eight nearest neighbours are combined in accordance with some logico-algebraic function, or else are used to index the entries of a lookup table defining the totalistic state transformations.

4.5 Cellular Automata Algorithms

These are based, for the most part, on obtaining a summation of the numerical values (a *totalistic* process) describing the states of the neighbouring cells, and interacting with the cell's current state to produce the new cell state. Totalistic transformation rules can be defined by a lookup table (LUT). The objective here is to produce meaningful and relevant iconic image transforms, defined in the form of cellular state transformations. This information can be anything of interest. In the present work, CA cell states are related to image properties—such as intensity—and may define a parameter (or feature) space. In some cases the CAs may carry out symbolic forms of computation on image features, but this is difficult at present. The logical computational aspects of CAs is discussed in some detail by Reynolds, in Duff and Fountain (1986).

Cellular Automata algorithms may alter images by adding, subtracting, or otherwise modifying image information by cellular processing. It is even possible that the lookup table (rule) could be ADAPTIVE, implying that the entire CA processor can exhibit adaptation. Alternatively, critical entries indexed within the lookup table can be subjected to appropriate external influences. Such advanced possibilities, although extremely interesting in their own right, are not pursued here due to limitations on project time.

Image processing and transforming algorithms have to be devised within the context of meaningful image transforming operations, and it has been found necessary to conduct experiments with the objective of establishing relevant algorithms. This work had been carried out earlier in the project by the writer, using a customised evaluation and display utility.[1] Evaluation programs are able to demonstrate the effect of algorithms on a range of suitable real and synthetic test images.

4.6 CA Rules and Lookup Tables

Lookup tables (LUTs) are a convenient and efficient technique in intensive computer programming. There are many computationally demanding algorithms and applications which have yet to benefit from the use of lookup table methods—especially the adaptive forms. Basically, a lookup table is an array of precomputed values, indexed by function. Viewed in this way, the lookup table is a transfer function, where the input is an index (or at least can be processed to yield an index) and the output is the selected (precomputed) table entry. The combination of a CA mechanism and a lookup table rule constitutes a CA processor. As the cellular automata system iterates, or cycles, the rule-plus-CA combination causes the dynamics of the system to unfold. These could be quite complex.

In CA applications, especially where colour graphics displays[2] of cell states are required, video screen refresh speed is of the utmost importance. A relatively high-resolution display could involve the updating of some 640x480 screen pixels, plus an equal number of screen display memory locations. In these systems the application of lookup table methods is essential.

The cellular automata lookup table function F is expressed symbolically as follows:

$$\phi_t \leftarrow F[\sigma, \phi_{t-1}] \tag{4.4}$$

The new state of a CA cell is a function of the old state (ϕ_{t-1}) and some combination of the cell's neighbours. Given a cell transformation rule, it is possible to compute a cell's new state as some function of the sum of its nearest neighbours (σ). This computation is needed at every lattice site, and during each iteration of

[1] The writer's CAMDIS.EXE and CAMVIS.EXE executable programs can be used to construct and demonstrate useful user-defined CA vision rules.

[2] Note that it is not necessary for cellular automata states to actually be displayed on a screen. In future developments CAs may be used to carry out internal or intermediate computations which do not require to be visible.

the CA. In order to reduce the time overheads in simulated CAs, a lookup table is used. Each table entry state-pair can be precomputed and inserted into the lookup table at the appropriate location.

It should be remembered (see footnote) that any CA operations can be "silent" or "invisible." That is, although most CA demonstrators are necessarily concerned with the visual display of the CA's states (usually displayed in a wide range of colours), it is not strictly necessary for iterations or cell state-changes (generations) to be displayed continuously. It may often be acceptable, in a given model, for only CA start and finish states to be displayed, the intermediate states remaining invisible. In most cases this will result in a much faster CA processing throughput.

4.7 CA Neighbourhoods and Kernels

As mentioned, a common CA neighbourhood is the 8-connected cell layout shown below. This is often referred to as a **square** neighbourhood. The other common neighbourhood cell layouts include the hexagonal and cross forms. Because of its relationship to the standard 8-bit data (byte) format, the 8-connected neighbourhood is to be preferred in CA vision work. Note that in byte format, the central (own) cell cannot normally be included within the CA summation as this requires all eight local neighbours.

NW	N	NE
W	C	E
SW	S	SE

The term "neighbourhood" is normally reserved for operations involving a cell's eight nearest neighbours, but excluding the cell's own state. Where the cell's own state is included in the summation it has become customary to refer to the neighbourhood as a "kernel." See Preston and Duff (1984:13). Later in this thesis it will be seen that CA neighbourhoods and kernels are related to conventional image processing template and morphological methods in an obvious way.

Also note that in a majority of CA rules it is not necessary to use the "compass" positions of the individual neighbouring cells (W, NW, N, NE, W, SE, S, SW) because only the **numerical sum** of neighbour contributions is required. This is called a *totalistic* rule. By way of illustration, the neighbourhood and the corresponding lookup table for Conway's famous Life Game is shown below.

4.8 Morphological Operators

CAs have much in common with morphological operators, as discussed in Pratt (1991), and elsewhere. These processes are concerned with the modification of images; involving typically the dilation, erosion, thinning, and skeletonising of binary pictures. Algorithms for these processes are developed in the present work, but are considered to be CA rule definitions, or else are implemented as VPC procedures (see later). Consequently, specific morphological image operations are not discussed any further here.

Note, however, that in the concept of our vision model, an original image is **never** modified—only copies are processed. This is in keeping with the Gibsonian notion of direct vision, to be discussed later in this thesis.

4.9 Examples of Simple CA Rules

This section gives examples of simple CA rules and their corresponding lookup tables, to give some feeling for the method. Many of the rules given in CA textbooks, such as Toffoli and Margolus (1987), are often empiricist: they are not the result of deliberate design. However, in image processing and vision work, and in physics modelling, a CA rule must be logically developed and stated explicitly. The rules in such applications must have a definite purpose, and must achieve meaningful transformations and computations based on physical laws. The visual effect of the example rules quoted here can be seen in Toffoli and Margolus (1987) and elsewhere. However, so far as the writer is aware, these authors have not used CA methods directly in image processing.

4.9.1 Totalistic Rules

So-called TOTALISTIC rules are simple CA rules in which a cell's new state is decided mainly on the basis of a count of the states of the cell's neighbours. An example of such a rule is the MAJORITY rule, in which a simple majority of "votes" ensures that the cell changes to the "1" state, or ON. The "0" state represents OFF. There are normally eight votes, plus the cell's own vote. The eight votes provide an "8-sum" and all nine votes constitute the "9-sum." In the MAJORITY rule (VOTE rule), if a cell gets 5 or more votes it turns ON. This rule can be represented by a simple lookup rule table, as follows:

Majority: Totalistic Code 1111100000b = 992d										
9-Sum	0	1	2	3	4	5	6	7	8	9
NewState	0	0	0	0	0	1	1	1	1	1

It is seen that the NewState result depends only on the totalistic sum of the bit counts of the cell, and its eight neighbours. To represent rules more succinctly the binary pattern of 0s and 1s is reversed, and then the bit pattern is converted into an equivalent decimal number, (d), the rule ID number. That is:

$$0000011111b \longrightarrow 1111100000b = 992d$$

So this particular rule is represented by the decimal 992.

However, there are many patterns of 0s and 1s which can produce an identical 9-sum. These are all 10-bit patterns, so that the total number of entries required in a full lookup table of 9-sum rules is:

$$2^{10} = 1024$$

Another example of a totalistic rule is the PARITY rule found by Edward Fredkin. PARITY forms the 9-sum of a cell's neighbours, and then makes the NewState = 0 if the 9-sum is even, and 1 if 9-sum is odd. The rule of FREDKIN (or PARITY—see Toffoli and Margolus, 1987) is expressed by the following rule lookup table:

Parity: Totalistic Code 1010101010b = 682d										
9-Sum	0	1	2	3	4	5	6	7	8	9
NewState	0	1	0	1	0	1	0	1	0	1

Assuming a neighbourhood definition as shown in the diagram below, the FREDKIN (or PARITY) rule can be expressed in terms of the logical XOR (exclusive-OR) function $\oplus$ as the following Fredkin Formula:

NW	N	NE
W	C	E
SW	S	SE

$$NewState = C \oplus NW \oplus N \oplus NE \oplus E \oplus W \oplus SW \oplus S \oplus SE$$

where C is the cell's own current state. The 8-sum is given by the simple arithmetical sum of the cell's eight neighbours, thus:

$$\text{8-sum} = \text{NW} + \text{N} + \text{NE} + \text{E} + \text{W} + \text{SW} + \text{S} + \text{SE}$$

4.9.2 Semitotalistic Rules

Totalistic rules depend only on a cell's 9-sum. The so-called "semitotalistic" rules depend on the following two aspects:

- the 8-sum of a cell's eight nearest neighbours
- the cell's present state

Semitotalistic rules are 2D rules, whereas totalistic rules are only 1D. This table dimensionality classification is derived from the format of the rule table entries.

Probably, the most famous example of a semitotalistic rule is the Game of Life. The LIFE rule is specified in words, as follows:

1. Form the 8-sum of each cell's eight neighbours.

2. If a cell is 0 and its 8-sum is 3, the cell's new state is 1.

3. If a cell is 1 and its 8-sum is 2 or 3, the new state is 1.

4. In all other cases the cell's new state is 0.

This can be expressed as a CA rule lookup table:

Semitotalistic LIFE Rule									
	8-sum								
cell state	0	1	2	3	4	5	6	7	8
0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0

It is seen that the rule table fulfils the LIFE game specification, as in the steps 1–4 above. This "game" does not directly provide any useful image processing algorithms *per se* but it does give an insight into chaos and fractal graphics—topics which are enjoying considerable scientific attention at present.

4.10 Defining CA Rules—Lookup Tables

Given a rule specification, a lookup table can be constructed which will enable efficient computation of cell states based on that rule. The mechanism actually used for the CA rule table will depend on the proposed implementation of the CA system. That is, the form depends on whether the CA model is a hardware or a software one. There are few hardware systems known to the writer: the CAM-6 of Toffoli and Margolus being perhaps the only genuine attempt to date. Therefore the majority of CA implementations in the foreseeable future will be software simulations running on conventional mainframe or personal computers. In this case, the CA rules are easily constructed in the chosen computer programming language (Pascal, C, BASIC, etc.).

The Cellular Logic Image Processor (CLIP) of Duff and his Group at Imperial College is essentially a hardware system—which can also be programmed in a similar manner to a microprocessor. That is, the CLIP transforms are expressed in the mnemonics of a custom CLIP assembly language, although the computer is a parallel hardware device. It may be noted that, although delivering an impressive performance, the CLIP series may not be a cellular automata machine *per se*.

In the present work, CA demonstration systems have been developed in particular dialects of the Pascal and C programming languages, and so the CA rule tables are likewise defined in these languages. However, there is considerable scope for implementing CA demonstrators on the Intel Transputer network, or in fast 80x86 assembler. The chosen CA implementation does not materially affect the principles of advanced machine vision as described in this thesis.

4.11 CA Rules for Machine Vision

A number of basic CA rules have been developed by the writer for use in vision applications. Most of these CA rules perform specific image processing tasks of a conventional kind, while others use combinations of basic rules to carry out more complex functions.

There are two basic classes of CA rule in image and vision work:

- single-pass rules
- multiple-pass, or iterated sequence rules

Single-pass rules are usually of the totalistic variety, while most multiple-pass rules are semitotalistic. For speed reasons, it makes sense to always code single-pass rules wherever possible.

Combination rules apply two or more different rules in a sequence, to the image data: the order of the rule application is usually critical. Combination rules are similar to multiple-pass rules, the important difference being that in multiple-pass a single rule is iterated until some final termination state is reached. Combination rules can also be multiple-pass types: indeed any system of rules can be devised and assembled to produce any desired CA transformation function.

The obvious method of designing special image processing rules is to construct a custom CA rule demonstrator in software, and then apply evaluation rules to a range of simple test images. When suitable rules have been found they can be noted for later incorporation into the CA vision model. This is the approach used in the present work. Commercial—or self-developed—CA rule demonstrators can then be used to design custom image processing and computer vision rules.

The undernoted CA-related processes have been investigated by the writer for application to our high-level vision model:

- Perceptual organisation (high-level).
- Image-tree manipulation (in association with Prolog).
- Image-tree creation and manipulation.
- Visual target parameter specification.
- Visual knowledge and context modification/interaction.

4.12 The "Soap Film" Rule

One of the assertions of this thesis is that important features of the Gestalt Laws of Organisation—first mentioned in Chapter 2, and discussed above—can be modelled in both CAs and ANNs. The concept is that of rubber bands or membranes that can stretch between "pegs" (image high-energy points) or, in the case of sheet membranes, take up a smoothed profile of the underlying shape. The concept is visualised in figure 4.3 below, where two wire rings are seen have a soap film stretched between them. The soap film linking both rings from the rims assumes the form of a minimum surface area profile—similar to a catenary in an elevation view. However, if the "bubble" is burst, the soap film bifurcates, and collapses to form two separate soap films—one film adhering to each of the wire rings.

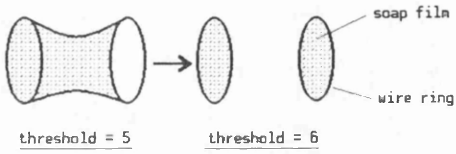
This assumption of minimal surface area is one of the analogies cited by the Gestalts in their Theory of Isomorphism. In the present study, only two-dimensional (2D) films will be assumed to exist, because we are interested only in 2D image forms. Even when the 2D images are those of 3D structures, a 2D image array will still represent the image. Thus, the 3D properties of a real-world model are not inferred (at least at this stage of the project) from 2D CA image processing. The soap film analogy is used here to infer only those visual features which would be imaged on a (planar) human retina. This is an important distinction to note.

It does not make any material difference whether rubber membranes, elastic bands, electric or magnetic fields, or soap bubbles are used as the Gestalt analogy—the basic principle is the same: namely, that of the assumption of minimum surface profile by the soap film. In this thesis we shall not present a mathematical analysis of soap films, because such detail does not confer any additional advantage to the model.[3] Details of soap bubble and soap film characteristics, and their formation, can be found in a very useful publication by Tieto Limited (Isenberg, 1978).

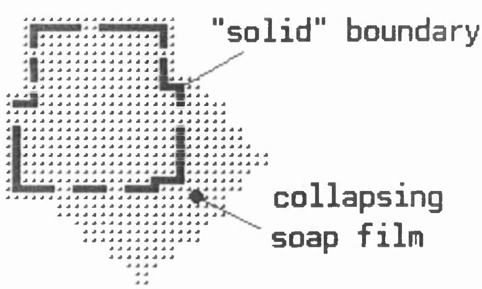
It turns out that it is relatively easy to model soap films using a CA semitotalistic rule. As the CA processor iterates this SOAP rule over an image, a soap film (search film or search field—all denoted by the abbreviation SF) commences from the boundary edge of the image and collapses or *propagates* inwards toward the centre of the image field. If there are no "solid" objects located anywhere within the image array, the SF disappears in the centre of the image. However if the SF encounters an object then—depending on the 8-sum threshold setting of the rule—a film will be formed in the neighbourhood of the solid object. If the 8-sum threshold is set at 6 or greater, the SF will simply form a surface film on the object. However, if the threshold setting is 5, the SF reaches the object as before, but establishes a "resistance" edge along the field.

There are both local and global effects. These will be discussed in a later chapter. Note that if using the SOAP rule it is necessary to create a border of 0s around the image, and restrict the start of the SF to the first row or column of image pixels. The soap film, in effect, tears away from the "black" edges of the image.

[3]This would certainly not be the case if one wanted to accurately model soap films as part of a physics demonstration model. In the present application the formulative characteristics can be obtained to a sufficient degree of accuracy using a simpler CA rule technique, as discussed in the text. We also assume here that planar or sheet soap films can exist.



(a)



(b)

Figure 4.3 In (a) a soap film (SF) is formed by surface tension across the gap between two wire rings, with the CA SOAP rule 8-sum threshold (S_T) set to 5. When the threshold is changed to 6, the film breaks, and collapses to form on the wire rings. In (b) a typical SF with S_T set to 5 is shown collapsing inwards towards the broken outline within a SOAP "search field"—also abbreviated here as SF.

The following semitotalistic SOAP rule table defines a soap film mechanism. This rule is iterated over the image array until a suitable termination or halt state is entered. Normally, the halt state is defined as some field "energy" metric. For example, a count of the 1s in the soap image can provide a convenient analogy of soap energy. When this count-rate becomes constant, the soap film is assumed to have reached a state of equilibrium, and so rule iteration stops.

The SOAP-0 rule has the 8-sum threshold set at 6, and collapses to meet the local surfaces of an imaged object:

Semitotalistic SOAP-0 Rule									
	8-sum								
cell state	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	1	1
2	2	2	2	2	2	2	2	2	2

The SOAP-1 rule forms an envelope or membrane over the salient features of a set of imaged objects, linking them together.

Semitotalistic SOAP-1 Rule									
	8-sum								
cell state	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1	1	1
2	2	2	2	2	2	2	2	2	2

Note that the above tables only show the 0–8 soap film states, the full lookup table requiring 73x3 entries to maintain the 2s in the representation of solid objects. Figure 4.3 shows typical soap films.

4.13 CA Hardware Possibilities

Many CA projects, as in the present work, make use of simulated CAs running on conventional serial computers. These are often developed using the standard high-level computer programming languages, such as Pascal or C. Assembler, too, has been used, but is normally rather difficult to code. In a few cases, CA-like algorithms (fractals) have been demonstrated on parallel hardware platforms, such as the Inmos Transputer system. This suggests that Transputer modules may continue to provide a suitable platform for future CA-based developments.

Increased processor clock speeds—up from 4.77 MHz to 66 MHz in the past decade—are helpful in CA simulations on conventional PC-based systems, and on the "vectored" mainframes. But the serious speed bottleneck that remains is the serial computing paradigm itself.

If a two-dimensional cellular automaton with, say, 256K of cells (or processors) was realised in advanced semiconductor technology it could compute 262,144 times faster than a serial processor accessing data bit-by-bit from memory. Yet, if engineered for volume production, such a device need not cost much more than a conventional RAM chip. This is the true potential of cellular automata. They map perfectly into the futuristic semiconductor manufacturing technologies, including even the prospect of molecular and massively-parallel computers. However, future advanced "computers" may not be recognised as such.

4.14 Chapter Summary

Cellular automata have been around for nearly four decades in the modern forms, and, like neural networks, have only recently been revived as a prospect for future computing. This revival has been triggered by a number of factors, including the need for new hardware concepts, ongoing neural network studies, massive parallelism, and a general scientific interest in chaos and fractal-style mathematics.

By means of rule definitions and lookup tables, CA transformations can yield virtually any computation which can be obtained by conventional computer algorithms. But CAs seem to be especially suited to the kinds of image processing and vision algorithms which are essential in the present work.

CHAPTER 5

PSYCHOLOGICAL FACTORS IN VISION

Chapter 2 introduced two different philosophies of natural vision, as advocated by James Gibson and David Marr. It was argued that these two approaches are usually regarded as mutually incompatible. This chapter examines in more detail aspects of high-level and low-level vision, with the objective of discovering a few of the attributes and mechanisms of each.

5.1 Low-level Vision

Low-level vision (LLV) is by far the easier of two broad categories of vision to deal with, since it is amenable to conventional computation. As such, it is more closely linked with the ideas of David Marr and the approachable techniques of conventional image processing. LLV is usefully described as "iconic"—because it is picture and image based. The use of parallel processing methods has often been advocated for LLV because of the essentially localised nature of most image processing algorithms—which are obviously well-suited to parallel and distributed processing. Another reason is the sheer mass of data that has to be processed—often now in real-time.

There is a large number of algorithms for dealing with LLV which have matured over the years. These have been designed principally for use with image processing on conventional serial computers. Freeman (1988) and Pieroni (1989) both contain a useful selection of papers dealing with machine vision topics. Recent overviews can be found in Batchelor (1991), and in the collective papers of BMVC-91 (1991), and ECCV-92.

The essential point here is that the vision researcher can now draw upon a wealth of tried, tested, and documented methods for almost every conceivable image processing task. One very important class of processes is edge or boundary detection for which there are almost as many algorithms as there are researchers in the field. Recent work by McCafferty (1990), and Grossberg and Mingolla (1985a, 1985b) has shown that "edges" can exist in images even where there are no detectable luminance differences. Such edges or boundaries are "illusory" and their detection is regarded as an indication of biologically-inspired artificial vision since only humans report seeing them.[1] A typical illusion of the Kanizsa type is shown in figure 5.1a.

Images derived from LLV processing can be converted to equivalent HLV *tokens* by a variety of techniques, including Hamming distance metrics, and the Hough transform. Hamming distance, and the processes based upon its many variations, are able to show changes in images due to spatiotemporal effects. Thus, the Hamming metric can be used to separate figures and objects from their backgrounds when the objects are moving—thereby revealing potentially interesting areas of images for detailed analysis by HLV. More complex methods may employ optic flow for this purpose: see Ballard and Brown (1982).

The Hough transform is also a useful mechanism for reducing the masses of iconic data to equivalent symbolic descriptions. The technique was originally developed as a detector of straight lines within images, but has subsequently been generalised for wider applicability and more general curve detection. A brief summary of the mechanism is given in Lewis (1990). Fuller discussions of the Hough transform, and related ideas of feature detection can be found in a number of sources, such as Ballard (1981, 1984), Ballard and Brown (1982).

Image preprocessing may include filtering of various kinds to reduce or remove noise. The use of colour is becoming more widespread as the cost of computer hardware continues to fall. Colour—rather than 3D or stereopsis—can provide extra clues for object identification.

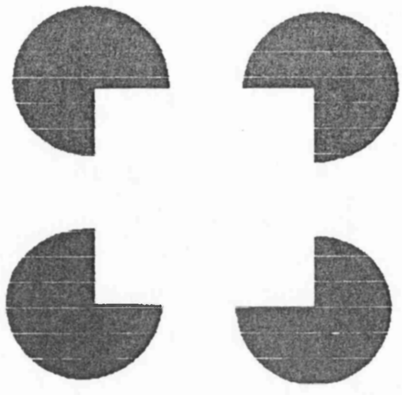
Therefore, the conclusion is that although low-level vision is rooted in the ideas and methods of conventional image processing, a careful application of a selection of well-established (low-level) techniques can result in emergent properties. That is, low-level vision can yield high-level tokens which, in turn, can be perceptually-organised, or grouped, as an important initial step towards object recognition.

[1] There is no direct evidence that the higher animals can see illusions in the same way that humans do. Equally, there is no evidence that they do not.

5.2 High-level Vision

High-level vision involves memory, knowledge, and experience. HLV is not usually considered to be "Gibsonian" vision, because Gibsonians have rejected all notion of image processing and visual memory traces. However, in this thesis, HLV is considered to be closer to Gibsonian philosophy than is generally acknowledged. HLV is usually regarded as perceptually "symbolic" because it is concerned with the many ideas and concepts derived from an interaction between LLV and our prestored memories, knowledge, and worldly experience.

For example, in figure 5.1b it is not the raw LLV "image" of any particular chair which is processed to tell us that the object is a chair. All of the objects are perceived as chairs—because of our worldly experience and knowledge of chairs. So, any low-level based process, or any group of procedures which attempt to analyse this picture on the basis of the image content alone (that is, without a specific knowledge of chairs) will probably fail to recognise all of the objects as chairs. Notice that 3D and stereopsis are not of much help in this kind of perceptual problem: indeed, this reinforces the comments made in Chapter 1 and elsewhere in this thesis regarding the relevance of 3D images and stereopsis to visual perception.



(a)

(b)

Figure 5.1 (a) shows a typical illusory figure of the Kanizsa type—in this case, an illusory square induced solely by the contrast edge features of the four PACMAN shapes.

Picture (b) shows some typical "chair" styles (classes), of which there is a great variety.

5.3 The Relationship Between LLV and HLV

What relationship might exist between what is usually distinguished as low-level and high-level vision? The following subsections give some idea of the subtleties of separating LLV and HLV. It is important to be aware of the fact that both forms of visual perception can be modelled by conventional image processing techniques. The close relationship between LLV and HLV suggests that HLV can often be derived directly from LLV without resorting to knowledge processing. That is, purely data-driven processes can reveal many aspects of higher level neural functions.

5.3.1 The "Kelvin" Exemplar

This analogy is used here to illustrate the concepts involved in the direct linking of LLV images and HLV in seemingly complex psychovisual processes. This model is cited by the writer to reinforce an important theme of this thesis—namely, that high detail in visual imagery is "blurred" by mental processes. This may be a natural data reduction process evolved by Nature to prevent perceptual or signal overloading in biological sensorineural processing: or it could be yet another of the enigmatic aspects of the human psyche.

The "Kelvin Way" model illustrates the idea that although the retinal images may alter quite drastically with every step as one walks (in this example) down the long tree-lined avenue adjacent to the University of Glasgow, the prevailing "mental image" of the scene will hardly alter at all—at least not until the far end of the avenue is reached when there is a marked change in the scene. That is, although one's visual senses may be exposed to masses of varying detail—the images of tree forms and leaves; the different sizes, shapes, and colours of motor vehicles parked at the pavement; the pedestrians, and so on—the prevailing character or "concept" of the avenue does not seem to change much as one walks.

This is clearly a high-level phenomenon, derived from stored memory and cortical interactions. The low-level scenic (iconic) images have become decoupled from the high-level "mental picture" of the avenue (such as might be painted by an artist from memory). The notion is that, if one can by some technical means organise and group raw image pixels in a structured and hierarchical manner, one can create a succession of levels in which the detail of the images becomes progressively reduced, more generalised, and fuzzy.

We can produce this easily, by using conventional image processing methods—in the form of standard grouping, segmentation algorithms. The novelty comes from realising that such segmentations can be *guided* by edge-driven processes that are themselves derived from both LLV and HLV—especially the latter.

These basic concepts of region merging are not new, of course. Image processing has for a long time combined pixels into regions, and then merged the resulting smaller regions into larger and potentially more meaningful groupings—that is, building image data trees from the leaves (pixels) to the trunk or root. An excellent recent example is presented in a paper by Beaulieu and Goldberg (1989).

The interesting observation from the perspective of this thesis is that high-level and "psychological" properties have apparently resulted from essentially low-level imagery—through the process of technical image grouping and pixel manipulation. This has not been stated explicitly before.

An added technical bonus is that information manipulation could be substantially reduced in artificial vision, as there seems no reason to overburden an "intelligent" vision processing system with masses of irrelevant detail and unnecessary processing. In other words, only "concepts" need be to stored at higher levels of machine perception.

This hierarchical grouping of images on levels suggests a relationship between technology and psychology—and a link between low-level and high-level vision. As was discussed in previous chapters, there is another link between technology and psychology: that of Gestaltist preferred groupings through "force field" perceptual organisation. This subject will be mentioned again in Chapter 6.

This simplified, data-driven, perspective of image transformations does not, however, satisfy the needs of psychology. There is no way of dealing, for example, with aesthetics; or the real phenomena of vision as a sensorineural experience. These issues are concerned with the notion of direct vision.

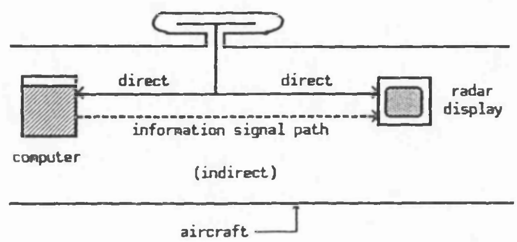
5.3.2 Airborne Radar Analogy

Let us now consider a technical analogy involving computerised radar systems as used in defence surveillance aircraft. (Similar arguments hold for ship radar, air traffic control, medical imaging, and so on). The purpose of this example is to again illustrate the advantages of the relationship between low-level imaging and vision processing, and higher level knowledge-based control strategy.

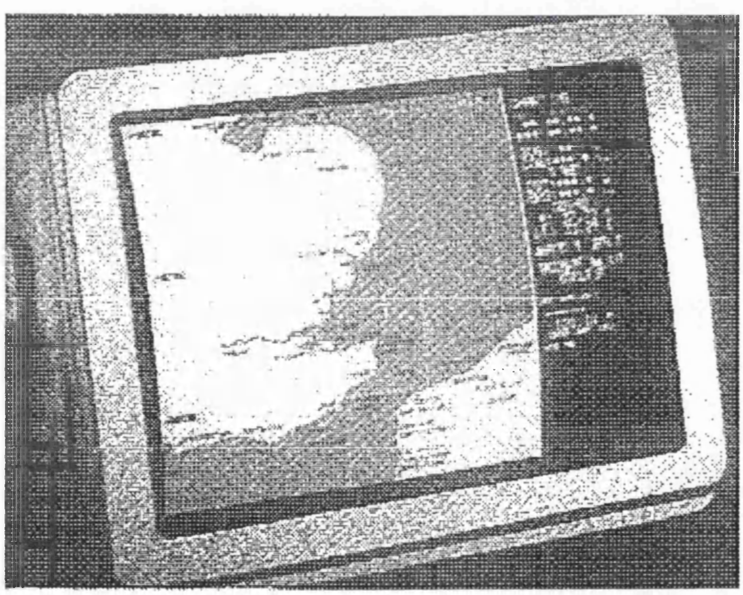
Figure 5.2 shows diagramatically a radar surveillance aircraft. The input radar image to the operator's console is a PPI (Plan Position Indicator) signal which can be directly displayed (we are not here interested in state-of-the-art radar electronics, or how an advanced radar display is deployed in practice). The relevant aspect is that a map-like image is produced directly on the radar display. This may be regarded as representing low-level "direct" vision. At the same time a "copy" of the radar signal is transmitted to the rear of the aircraft where an on-board computer is located. The computer contains a database of the characteristics and features of potential enemy aircraft and ship classes, as well as terrain information and other relevant data. The computer is able to use all available data and information to come up with a model of the current scenario, including details of any potential or actual threats of hostile action. This may be considered as the high-level informational response.

Computerised information is then combined with the low-level radar image to provide a comprehensive display for the aircrew. The result is a total representation, not only of the live radar image, but of the actual (present) and potential (immediate future) situation. The vision system in this case is a complex amalgamation of low-level radar (iconic) and high-level (symbolic and knowledge-based) visual information. The console operator is then able to track the hostile targets, and is updated by the computer on request by moving icons over the display area. This is an example of superposition, in which the iconic radar display is overlaid with high-level information which can highlight visually and strategically significant features. This is shown in the recent photograph of an actual display (Figure 5.2b).

The point being made in this analogy is that similar processes may be postulated for intelligent biological vision, in the sense that direct "Gibsonian" images are blended with the stored cortical knowledge and experience within the brain: a fusion of information.



(a)



(b)

Figure 5.2 Image information fusion.

(a) A hypothetical aircraft radar system with on-board computer. Both the radar display and the computer receive "direct" radar PPI signals.

(b) The technological advantage of blending information. A modern hybrid radar display can blend both "direct" and stored information. Raw radar images are signal-processed and combined with intelligence, terrain, and cartographical information to yield a comprehensive and informative visual display.

5.3.3 The "Dalmation-in-the-Park"

Figure 5.3 shows the well-known picture of the Dalmation-in-the-park camouflage scene. When viewing this image for the first time it is usually extremely difficult to see the Dalmation. However, once seen the opposite prevails, and it becomes more difficult not to perceive the animal's form. This demonstrates a strange linkage between HLV and LLV. Initially, LLV searches the image to try and discover edges, blobs, and other clues. In the total absence of any context it may take some considerable time for the Dalmation to be seen. However, once the scene has been recognised, the Dalmation image is committed to some higher perceptual level storage.

This phenomenon may contradict the theories of both Marr and Gibson. Marr's theory does not hold here because Marr considered that the 2.5D Sketch is a low-level grouping that occurs *before* high-level processes. The problem for Gibson's adherents is that they do not accept the idea of visual memory, so there is no simple way by which the "direct" theory can explain static camouflage effects. However, once a camouflaged animal or object begins to move, Gibsonian theory can explain the perception in terms of corresponding changes in the optic array disposition, and in the theory of optic flow. The HLV perception appears to have evolved from LLV—again suggesting that LLV images and HLV mental phenomena are linked.

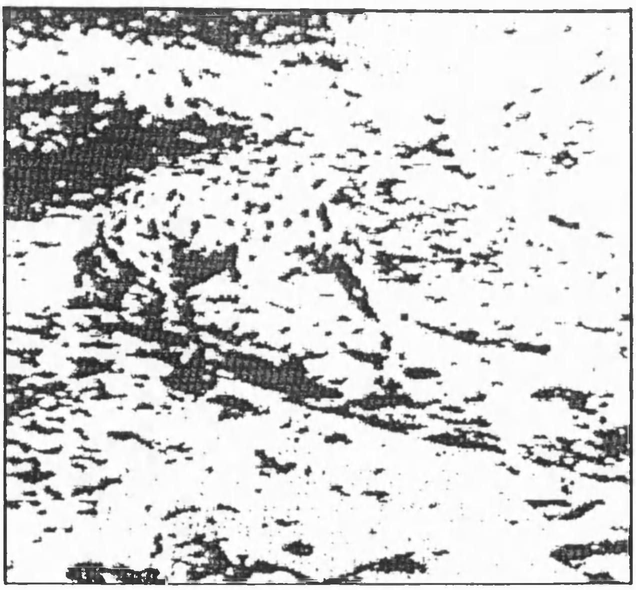


Figure 5.3 The well-known "dalmation-in-the-park" camouflage picture.

5.4 Visual Psychology and Prolog

Prolog is a main computing language of artificial intelligence (AI). In this project, Prolog is used to represent, in a relatively simple manner, some of the real-world contextual knowledge that is required to support advanced vision. For example, a spherical orange-coloured object might be fruit, a balloon, the planet Mars, a mechanical part, or any of perhaps hundreds of possibilities. The task of artificial vision—indeed any kind of vision—is made much easier if some kind of context can be used to constrain the range of possibilities. Thus one possible use of Prolog in artificial perception is to implement a useful database of knowledge and facts. This is a conventional Prolog application, as advocated and described in almost every Prolog text; for example, Bratko (1990), Stobo (1989).

Another—possibly more novel—use of Prolog in the present work is to examine visual data received from an CA image processing mechanism, and to then interact with stored knowledge in order to infer a set of parameters which can guide further image processing. That is, Prolog takes on the role of the human experimenter in such applications.

McCafferty (1990) demonstrates how perceptual organisation (to be discussed in the following Chapter 6) can be made to prefer stable groupings in mainly synthetic images. However, the impressive results obtained from McCafferty's "Grouping Engine" depend on the selection of appropriate parameters for his energy minimisation scheme. This is carried out by the human experimenter—using his own natural vision. This approach is perfectly reasonable in the context of experiments, but if the aim is to demonstrate machine perception then human vision should not be allowed to intervene: advanced machine vision should be autonomous.

In the present work, the aim is to use Prolog to establish numerical values for the cellular automata (CA) equivalent processing, and to further select appropriate image processing CA rules at every stage of the image processing algorithms—both high-level and low-level. In addition, a very simple form of expert system is constructed for the recognition and identification of images, and to thereby extract the parameters for CA rule selection.

5.5 Chapter Summary

This chapter has attempted to relate the goals of low-level vision (LLV) and high-level vision (HLV). It is seen that, even from the purely image processing perspective, there can be a useful interchange of data and information. For example, high-level percepts can come from purely LLV image processing.

The Prolog programming language can allow complex psychological properties to be expressed in an accessible manner. The mechanisms used by natural neural networks to encode mental concepts are not understood; and this will remain for the foreseeable future. Emergent properties are seen by the writer as being especially important.

CHAPTER 6

ASPECTS OF PERCEPTUAL GROUPING

Previous chapters have considered low-level vision (LLV) as a grouping phase. In LLV raw image pixels are grouped into meaningful regions and objects, mainly as a result of the detection of edges and boundaries within natural and synthetic images. The impetus for this approach stems from the realisation that edges and boundaries are likely to reveal gross structures within images, such as specific objects of interest. There are now many methods and novel techniques available for the grouping of image data—usually called *segmentation*.

These techniques frequently use only local image data—such as the nearest-neighbour similarity measures of pixel values, and so on. If we desire to develop a convincing technological model of human vision, we must consider grouping or segmentation methods which are likely to be found in biological vision. Human vision, of course, is natural vision—mediated by high-level processes resulting from complex psychological and memory factors. Clearly, the "groupings" in the case of humans can be very subtle indeed.

This chapter considers only limited aspects of image groupings within biological vision. The hope is that the techniques used by Nature may provide further inspiration for advanced machine vision.

6.1 The Problem of Perceptual Grouping

The interaction between LLV and psychology is termed *Perceptual Organisation*. That is, the organisation and structuring of images results from LLV processes, controlled to some degree by mental and psychological phenomena. Well-known examples are the many kinds of optical illusions, both natural and man-made. For instance, a common experience is the human brain's propensity to "see" human faces and other objects within the rich detail and patterning of floor carpets. At the same time, the *processes* involved in such perceptual groupings are considered to be firmly rooted in LLV.

In this chapter Perceptual Organisation is considered as a potentially important grouping mechanism in machine perception, but is still to be regarded as complementary to existing low-level techniques. It should, however, be noted that Marr (among others) considered that perceptual organisation takes place within the visual pathway *before* the primary cortical vision areas. That is, Marr considered perceptual organisation to be a purely low-level process of natural vision. Presumably, this would imply that both human and animal natural vision use a similar kind of perceptual structuring. This is a perfectly reasonable assumption. Nevertheless, many philosophical vision ideas are contradicted by the human experience of **optical illusions**.

In humans, optical illusions are likely to have very much deeper psychological origins: in other words, *preferred* perceptual groupings in humans are likely to be very much more complex than those of even the higher animals. Such factors have to be taken into account in any cogent model of advanced machine vision.

The problem of perceptual grouping, then, is that it is—at the same time—both an LLV grouping mechanism and a complex HLV phenomenon.

A recent review of Perceptual Organisation, containing references and pointers to further reading, is the text by McCafferty (1990). In the McCafferty approach, "energy" or "cost" minimisation is the key mechanism by which perceptual organisation is realised in practical demonstrations.

6.2 Perceptual Organisation

The term Perceptual Organisation refers to the human vision system's capacity (and propensity) for detecting groupings and structure in images. The tendency of humans to detect higher-order structure is a very important aspect of perception. Animals, too, must be capable of detecting shape, form, and structure, otherwise they would be unable to discriminate between prey and predator. However, if viewing an image such as figure 6.1, what would an animal perceive? A human being would most likely interpret figure 6.1 as a single object (a "circle" in this case) despite the fact that the object is itself composed of groupings of smaller elements. An animal, too, might perceive a circle—although such a form could be meaningless when "seen" by the animal in a situation devoid of natural context.

The pertinent question is this: How should artificial vision systems perceive an image such as that in figure 6.1? Within the context of this thesis we take the view that a machine vision perception should, wherever possible and appropriate, be consistent with human vision. That is, if it is meaningful in a given set of circumstances for a human to interpret the object in figure 6.1 as a circle, then machine vision should do likewise. This implies that the machine vision system must contain—or otherwise be able to access—a knowledge-base of relevant, contextual, high-level information and intelligence.

The subject of perceptual organisation subsumes a number of important topics, such as curve and edge detection, texture analysis, image and scene segmentation, preattentive vision, and figure-ground separation. These may all be regarded as aspects of perceptual organisation, but different grouping techniques will usually be applied in each distinct class of image processing task.

The study of perceptual organisation, or perceived groupings, began in Germany in the 1920s with the Gestalt School of Psychology. Following a discrediting of the Gestaltists, interest in perceptual organisation waned in the middle of the century. Recently, however, there has been an upsurge of interest in the subject, and perceptual organisation is once again an active field of research. Several aspects of perceptual organisation—relevant to the work described in this thesis—will be reviewed briefly in the following sections.

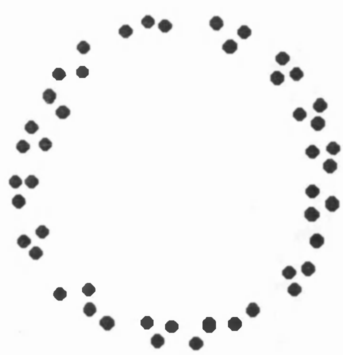


Figure 6.1 A "circle" is usually perceived by humans—even though there exists only a collection of dots on the page. This simple image raises a number of important philosophical issues in natural vision. See text.

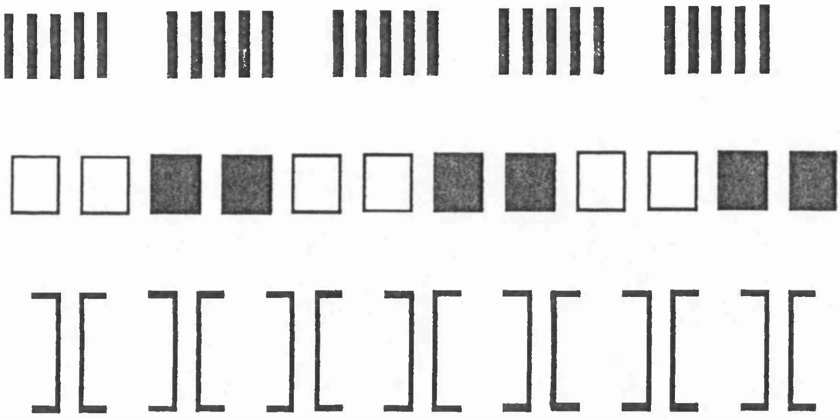


Figure 6.2 Illustrating some of the Gestalt Laws of Organisation in perceptual groupings. The black lines are grouped on the basis of **proximity**, the squares are linked by **similarity**, while in the bottom group, complete squares are perceived—rather than back-to-back brackets.

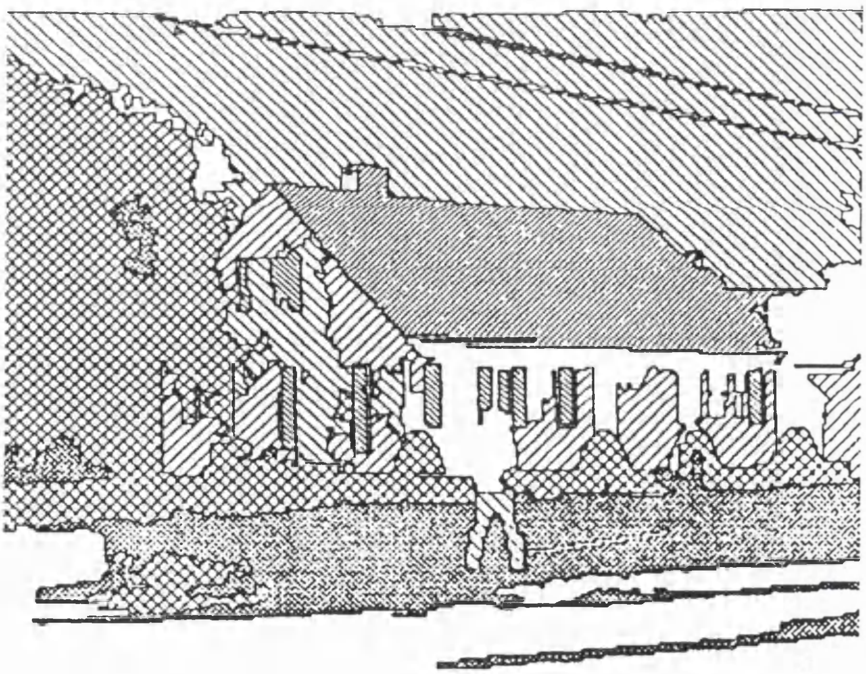


Figure 6.3 A typical conventional segmentation of an outdoor house scene which relies on purely low-level image data. From Arbib (1989).

6.3 The Gestalt Psychology

The psychology of human perception was dominated in the late 19th century by the Doctrine of Associationism—Bruce and Green (1985). This concept was based on the assumption that visual perception could be explained solely in terms of its component sensations; and that the association of such simple sensations resulted in substantially more complex perceptions. The emergence of the Gestalt School of Psychology in Germany in the 1920s and 1930s was a significant landmark in the history of modern visual psychology.

The Gestalts took a fresh approach. Their emphasis was to stress the importance of form and structure of sensations, and the relationships between them. The Gestalt tenet was that "the whole is greater than the sum of the parts." This, in turn, led to the Gestalt notion of "emergent properties" as discussed earlier. The example of two dots exhibiting the property of direction—which neither dot possesses on its own—was cited. There are, in fact, many commonly occurring examples of this kind, such as the perception of a "square" within an image containing only four dots at each of the four corners.

The Gestalt psychologists listed a number of factors which they held to be crucial in the perception of form and structure. These include:

- proximity
- similarity
- closure
- continuation
- figure-ground separation

Some of these are illustrated in figure 6.2. The grouping factors, collectively, became known as the Gestalt Laws of Organisation, and supposedly explained why grouping occurred. According to one of these laws—the Gestalt Law of *Pragnanz* (which is German for "simplicity" or "goodness-of-form")—only the simplest and most stable groupings are likely to be strongly perceived in biological or natural vision.

The process by which human (and animal?) vision supposedly implements the Gestalt Laws of Organisation, including the Law of Pragnanz, was detailed in the

Gestalt Doctrine of Isomorphism. This doctrine hypothesised that underlying every perceived visual sensation, there is a "brain event" which is structurally similar to the sensation. When combined with the concept of Pragnanz, the result was a theory of magnetic "force fields" and their associated mechanisms. A good analogy of this is the stretching of rubber bands across pins, or the surface tension effects observed in soap films and bubbles. It is the physical existence of such magnetic fields which, the Gestalts hypothesised, were responsible for the brain's tendency to always form the neatest, simplest, and the most stable perceptual groupings.

As will be discussed, Gestaltist theory lends itself to computer implementation—either as conventional algorithms or as simulations in cellular automata. As was mentioned before in Chapter 4, the latter are especially interesting since the long term developments in true parallelism and cellular-level computing are likely to be particularly relevant to image and vision processing.

As a description of possible neural processes, the Gestalt Doctrine of Isomorphism (in the physical sense) seemed highly unlikely, and so the theory fell into disrepute. However, the Gestalt Laws of Organisation have survived, together with the Gestalt notion of Pragnanz. Although this terminology was used descriptively (rather than quantitatively) by visual psychologists, we shall demonstrate that these Gestalt ideas can be implemented in current and emerging technologies. Indeed, the text by McCafferty (1990) demonstrates how such mechanisms of "energy minimisation"—being similar in principle to the surface tension of soap bubbles—can be used as the means of achieving stable perceptual groupings within images.

6.4 A Gestalt Link to Technology

If it is accepted that the Gestalt Laws of Organisation are indeed a valid concept in natural (especially human) vision, then we have at our disposal a useful paradigm for vision. What must be discovered is the means of relating the Gestalt Laws to measurable characteristics of technologically-derived images. McCafferty (1990:Ch4 and Ch5) discusses some of these topics in detail. However, he does not take much account of the possible high-level mechanisms in natural vision which may enforce groupings. That is, the McCafferty approach itself relies upon human intervention (human vision) to specify the grouping parameters for use in what he describes as a "Grouping Engine." In the present thesis, the view is taken that the superior processes of biological and human vision will act to enforce preferred perceptual groupings. These are the perceptions which are consistent with everyday human visual experience and intelligence.

This means that in the Gestalt Laws of Organisation, researchers have a valid set of rules and neural mechanisms which can potentially link the hitherto intangible—and often fragile—aspects of psychology to the much more robust and accessible mechanisms of technology.

Note, however, that it may not be necessary or desirable to model soap bubbles in mathematically precise detail, or in very-high resolutions. Relatively simple **approximations** of these phenomena have been found to be adequate for the cellular automata models developed in the present work.[1]

6.5 Recent Research in Perceptual Organisation

There has, until very recently, been comparatively little research into perceptual groupings in natural vision. Such work as has been published has been concerned mainly with measures of similarity, and has mostly ignored the other important grouping factors listed in Section 6.2 above. We shall not discuss this other work in detail. Hochberg and Brooks (1960), Olsen and Attneave (1970), Beck (1972), Julesz (1975), Rock (1975), are examples. Details of these and other researches can be found in Bruce and Green (1985), McCafferty (1990). All have demonstrated image segregation and perceptual groupings on the basis of lines, shading, dot density, continuity, and so on.

Interesting work, which has some relevance to ideas of this thesis, is reported in Treisman (1985). She used displays such as those of figure 7.4 to demonstrate the phenomenon of preattentive "popout" in human perception. Preattentive vision is held to be a visual response to a substantially changed scene which requires only 400-600 ms, and so occurs before the normal array of visual processes become fully operative. Because of the short times involved, such processing is unlikely to

[1] But of course there is nothing to prevent researchers using complex or mathematically accurate methods if they so desire.

be a result of Marr-like computations. Indeed, it is the contention of this thesis that preattentive vision is much more likely to be a manifestation of Gibsonian vision. In other words, Triesman's tests may have demonstrated aspects of direct vision—rather than validating the ideas of perceptual groupings *per se*.

6.6 Perceptual Organisation and Machine Vision

What connection, if any, exists between Gestalt ideas and the current approaches to image processing and machine vision? Most conventional image processing and machine vision systems have implicitly used some aspects of perceptual organisation. For example, pixel edge linking and region segmentation are grouping processes—but they are seldom considered as perceptual organisation in the sense implied here. Many other aspects of perceptual grouping, such as token continuity and closure, have been ignored completely by machine vision researchers.

The following is a list of established **conventional** image and vision processing techniques which can be considered to implement at least some limited aspects of perceptual organisation:

- edge detection
- edge linking
- region growing
- region splitting
- texture segregation
- curve detection
- curve representation
- shape completion
- histogramming

Most of these can use conventional methods, such as the Hough transform, or nearest-neighbour processes; or the recent cellular logic approach, as described in Preston and Duff (1984), Duff and Fountain (1986). Useful and recent reviews of well-established image processing techniques and algorithms are presented in the texts by Pratt (1991), and Lewis (1990). Mechanisms of image segmentation, including an interesting algorithm, are discussed in a paper by Beaulieu and Goldberg (1989). Figure 6.3, adapted from Arbib (1989), shows a typical conventional segmentation of a external house scene.

In this work, we are interested in discovering cellular automata type mechanisms which can carry out the above image operations. These basic processes should be guided by high-level vision concepts, so that the system is entirely autonomous. As discussed in the previous chapter, a difficulty with the otherwise excellent McCafferty (1990) energy minimisation approach is that it requires the intervention of human vision—and human intelligence—to prespecify the weighting factors to be applied to grouping energy mechanisms. This is undesirable.

A better approach is the Grossberg-Mingolla (G-M) neural network model of perceptual grouping. This is similar in concept to the writer's systems model of vision, and will be discussed again in Chapter 9.

6.7 Chapter Summary

This chapter has introduced the concept of Perceptual Organisation, including the principles and ideas from the Gestalt School of Psychology, in the recognition that these may be relevant in the identification of the difficult philosophical aspects of biological vision. Our hope is that the Gestaltists may have provided a link between the established techniques of image processing, and the totally abstract nature of biological and human vision. Although conventional image processing techniques implement perceptual grouping in some form, especially within image segmentation algorithms, the perceived need is to concentrate on ideas that can link with the Gestalt Doctrine of Isomorphism.

Such processes appear to be well-suited to modelling on conventional digital computers. Hence they can also be conveniently implemented as simulated CAs. By understanding the natural organisation of image elements—rather than the conventional image processing statistical and procedural groupings—we hope to be able to produce a better model of intelligent vision.

CHAPTER 7

RECENT THEORIES OF BIOLOGICAL VISION

"It is important for him who wants to discover not to confine himself to one chapter of science, but to keep in touch with various others." — Jacques Hadamard.

The previous chapters have discussed a number of important issues in machine vision, including the problems of implementing technological vision, diverse theories of Marr and Gibson, low-level and high-level mechanisms, the potential of cellular automata, and the Gestalt Laws of Organisation. It is seen that it is extremely difficult to grapple with the abstract nature of vision: the technological aspects of image processing are now well-understood, but the deeper psychological elements appear still to be beyond our comprehension.

This chapter attempts to highlight a few of the recent discoveries in neuroanatomical and biological vision, and to relate them to the goals of the project. It cannot do more than mention a few of the issues that are relevant in the present context: for instance, the recent discovery of dual massive parallelism within the optic tract.

The motivation for this interest is the writer's belief that the key to understanding vision lies in an appreciation of the direct sensory experience of vision—something that cannot yet be captured in our models of vision. Our notion of "direct" vision agrees—at least superficially—with Gibson's ideas; but we must not be drawn into a discussion of the many contentious issues. Some of the factors to be discussed in this chapter will play a role in the formulation, and in the justification of the vision model to be developed in Chapters 9 and 10.

7.1 Natural Neural Networks

Before we can begin to appreciate the significance of a biologically-inspired approach, we must consider some very basic neuroanatomy.

Distinct groupings of interacting biological neurons constitute neural networks, such as the Central Nervous System (CNS), and the Peripheral Nervous System

(PNS). However, it is not always apparent where one neural network ends and another begins. The expansive neural realm, which includes the CNS and PNS, is a complex and vast interconnected matrix. The brain is the largest single cluster, and has been further mapped into distinct regions which have become specialised.

The processing capacity of biological networks is determined by both the interconnection of the neurons and the synaptic weights operative at the dendrites. A "typical" brain neuron (if such exists) will have possibly 500-10,000 dendrites (a Purkinje Cell in the visual cortex has as many as 50,000) synapsing with an approximately equal number of axonal fibres from the other cells which form the neural networks. The brain is said to have in excess of 10^{10} neurons at birth, but other large cluster neural networks can be identified.

Examples include the optic nerve and the spinal cord. These are mainly two-way axonal fibre bundles, carrying electrochemical impulses, but much localised computation takes place, particularly within the grey matter of spinal column neural cells. The retina is an example of a **preprocessing** neural network, and the spinal cord also provides localised control over motor functions (limb movements) in response to commands from higher neural signals originating in the motor cortex region of the CNS (i.e. the brain).

The typical natural neural network has three layers: (1) an input or sensory layer, (2) the neural network proper, and (3) an output or response cell group. The input layer is associated with specialist neurons called **receptors**. Examples of receptor-type neurons include the *rods* and *cones* of the retina, the hot and cold receptors of the skin, and the stretch sensors within muscle fibres. Examples of response cells, or **effectors**, include muscle fibres and the numerous glands in the body. Neural networks themselves are usually laminar in construction (Brown, 1991), each separate layer interconnecting and synapsing with many other local, and distant, neural cell groups.

The shortest axons are the very localised skin receptors, while the longest fibres are to be found in the two-way axonal bundles of the optic nerve, spinal cord, and the "nerve" tracts running to the body periphery—such as fingers and toes. The latter may extend to more than one metre in length in humans. These axonal bundles represent two-way communication channels; because they carry both the input impulses from the receptors UP into the brain, and the response or command signals from the cortex DOWN to the effectors. A fourth type of neural network is known to exist. These are the oscillatory cell groups responsible for the biological clocks, needed to synchronize natural periodical and habitual activities. An obvious example is the heartbeat, controlled via systolic pulses generated in the oscillatory neural networks of the thalamic region.

In addition to the four types mentioned above (three layered classes, plus oscillators), neural nets can be classed as *feedforward* and *feedback* types. Feedforward neural networks, as the term suggests, act like pre-programmed (matched) filters, whereas feedback neural networks behave as adaptive signal processors. This is a much simplified description, of course, since the majority of the neural functions are still not very well understood.

Natural neural networks possess the highly desirable properties of massive parallelism, very-high density, robustness, and a graceful degradation in respect of damage and natural deterioration.

For further and detailed information on neural network structures the many specialist sources should be consulted. However, there are now a number of texts providing levels of neuroanatomical description that are better suited to the requirements of technologists, as distinct from the biological specialist. For example, Braitenberg and Schuz (1991), Eggermont (1991), Brown (1991).

7.1.1 The Brain

As was mentioned above, the brain is the largest and most distinct natural neural network, although itself composed of many separate regions, each responsible for different functions. In humans, this regional organisation is very marked, and has been understood and documented for some time. The largest volumes of the brain are the cerebral hemispheres, which occupy most of the interior of the skull. They are layered structures, the most complex being the outer layer, the cerebral *cortex* where neurons are densely packed to allow great interconnectivity. The cortex, and the cortical layers seem to be the centres of higher order functions and the focus of our natural intelligence.

The cerebral cortex has been the subject of investigation by many researchers over many years, but is only very slowly giving up its secrets. In this work, we are specifically interested in the visual cortex, which is the reception area for neural signals arriving from the various parts of the visual system. The analysis of the possible cortical functions is very difficult indeed, particularly because of the long-term evolutionary development (in the Darwinian sense) and the learning capacity due to synaptic processes. For example, the gross structure of the human brain is due to evolution and genetic predetermination, but learning in the individual also results in a self-organisation at a much finer level. Thus, a human brain is claimed to be a perfect representation of its owner, in both gross structure and in the individual personality and psychology.

The characteristic *grey matter* of the mammalian cortex is the mass of neuron cell bodies, while the *white matter* is due to the myelination of the axons of inter-connecting fibre bundles running between the neuron clusters, or layers. The white myelin sheathing of interneuronal axons is composed of "wrapped" glial cells, and is essential for the efficient transmission of neural signals. For a fuller description of the brain, and the cerebral cortex, see Arbib (1989), Braitenberg and Schuz (1991), Eggermont (1991), Brown (1991).

As mentioned, we are only interested in those specific regions of the brain which are concerned with biological vision. These will include the visual cortex, the retina (an outgrowth of the brain), and the two regions of the midbrain, known as the *Superior Colliculus* (SC) and the *Lateral Geniculate Nuclei* (LGN).

7.1.2 The Superior Colliculus

The Superior Colliculus (*optic tectum* in the lower species) is a subcortical structure in the midbrain, believed to be responsible for the generation of neural control signals in saccadic eye movement. A good model of the possible neural circuitry is presented in a paper by Fujita (1989). Fujita describes in some detail the processes by which a target, appearing on the retina, is mapped retinotopically to equivalent signal vectors producing oculomotor horizontal and vertical eye movements. A similar scheme, proposed by Pellionz (1986, 1987) is expounded in Churchland (1986).

The superior colliculus is phylogenetically older than the many other vision-related areas of the brain, and so is clearly involved in basic but very important low-level visual functions. It is proposed in this thesis that in its "imaging" role, the SC may be a source of direct vision, and of vision-related action and reaction.

It is possible that it is linked with the studies of the vision researcher James Gibson, whose theories were discussed briefly in Chapter 2.

7.1.3 The Lateral Geniculate Nuclei

The Lateral Geniculate Nuclei (LGN) are two clumps of neural tissue lying on the optic tract. Their function is not well understood, but the LGN appear to be closely involved in visual image processing, in collaboration with (and probably controlled by) the primary visual cortex. Our interest lies in modelling the LGN only as part of the dynamics of a vision system, rather than as a discrete module.

As with the SC, and its possible connection with Gibsonian vision, the LGN appears to be related to computational vision, and hence the image processing paradigm of David Marr. That is, the LGN appear to have evolved on the posterior optic tract for the purpose of carrying out specific image processing functions. However, unlike the feedforward class of neural networks, the LGN are greatly influenced by primary visual cortex feedback signals—see Churchland (1986).

7.2 Technological vs. Biological Vision

As was stated in Chapter 1, technically implemented vision systems, particularly those of the metric-based industrial kind, have achieved useful successes in recent years. This is because it is usually much easier to cast such vision problems in a technical mould. For example, the problems of implementing vision on production lines, or in medical imaging, are difficult—but they are at least *understandable*. The goals of such systems can normally be stated explicitly, and the means of achieving the objective can usually be formulated directly.

Thus, while it is easy to understand how edge and boundary detection can be accomplished using established computer algorithms, it is much less obvious why humans should perceive a SQUARE, say, when only four corner dots exist on a sheet of plain paper. The former processes are the result of low-level vision; and there seem to be few problems in the design of ever more sophisticated algorithms for image processing and segmentation. However, the route from pixels to predicates is not believed to be a direct and linear path.

The Gestalt Laws of Organisation, and their translation into sets of measurable quantities (e.g. McCafferty, 1990) is an important step in the direction of psychological understanding. However, this still does not address the problem of direct vision—how we humans (and animals) actually "experience" sensorineural visual stimulation. This is the great philosophical problem.

7.3 The Nature of Biological Vision

Natural vision seems to be dominated by many dualist characteristics; this was discussed in the previous chapter. Such a dualism creates difficulties of the kind experienced in physics, for example, where wave-particle duality is sometimes cited as a failure of physicists to understand natural philosophy. The following table lists some of the opposing attributes which are found in natural vision:

TABLE 7.1: DUALISM IN NATURAL VISION

cone-type receptors	rod-type receptors
photopic vision	scotopic vision
foveal vision	peripheral vision
high-level vision	low-level vision
concept-driven	data-driven
top-down	bottom-up
Marr's paradigm	Gibson's theory
computational vision	direct vision
LGN-C vision model	SC vision postulate
slow LGN-C axons	fast SC axons
BCS phenomena	FCS phenomena

The point here is that the evidence gathered over the years of vision research implies that there may be at least two major modes of visual perception. This suggests that Marr's and Gibson's ideas could both be regarded as manifestations of particular characteristics of biological vision. Both can be right.

Gibson's ideas of direct vision are psychological, vague, subjective, and often mysterious: hence Gibsonian vision is difficult to quantify. In this work, we shall use the notion of the "direct" Gibsonian image—or even a sequence of images if appropriate—to aid in the task of enforcing image segmentations and meaningful perceptual groupings within LLV. This process will be additional to any other high-level mechanisms which may be operative in the model. By such means, a crucial link between the Gibsonian image and high-level (i.e. neural-like) modelling processes can be established.

Note that our use of terminology referring to "Marr processing" and "Gibsonian vision" is intended only to exemplify certain general and philosophical ideas: these terms only provide a conceptual crutch, and so do not have absolute interpretations.

7.4 Gibsonian-Style Vision

As argued above, there seems to be at least two major duals of vision. This is the reason why the present thesis proposes a dualist approach to perception. Consider the following elementary fact: If an object is hurtling towards a human or an animal, there will normally (unless there is defective "eyesight") be an automatic, or reflexive action to avoid the missile. This happens almost instantaneously, and so is "preattentive" vision—in the sense of Treisman (1985). This was mentioned in Chapter 6, and will be considered again below. There is no time for image understanding in preattentive or reflexive vision: immediate action must be taken to avoid the imminent danger.

This simple fact illustrates the two principal features of natural vision: (1) the direct (or sensory) character, and (2) the analysis and subsequent image understanding, using cortical memory, neural image processing and archiving, and the extra-cortical associative processes. The latter include long-term memory (LTM) and other sensory interactions.

Thus, there is a commonsense understanding of vision as "seeing," and unconscious reasoning processes which are normally hidden from us. As stated previously, the former process we can relate, approximately, to Gibson's ideas of direct vision, while the latter can usefully be identified with Marr-like image (vision) processing.

Direct vision is purely sensory. This is similar to other kinds of perceptions, such as pain, pleasure, and so on. There is no need to "analyse" pain. We usually react in a manner so as to alleviate it. For example, we instinctively invoke reflex actions to remove a hand from a flame or scalding liquid. There is no requirement, nor is there any time for a "pain pattern analysis." Indeed, it has been argued that the retina is phylogenetically descended from the skin surface (Churchland, 1986). It is claimed that the evolutionary processes have produced eyes that are sensitive mainly to the visible portion of the electromagnetic wave spectrum: so our eyes can be considered to act as both direct light transducers, and as biological input devices to the complex natural computing system that we call vision.

We argue here that the commonsense ideas of eyesight and preattentive vision are more likely to have encouraged the Gibsonian theories of vision, than concepts of intelligence or complex neural memory. The following sections briefly consider the evidence for the existence of a direct and a separate Gibsonian kind of vision. This is necessarily only a very superficial treatment, and reference should be made to the original sources cited.

7.4.1 The Optical Tracts

In recent years, there has been reported new neuroanatomical evidence which indicates the existence of parallelism in the optic tract. See, for example, Churchland (1986:106), Cotter (1990). This physical path branching of the optic tract (optic nerve) is often confused with the more fundamental kind of mass-parallelism to be found in the retina, and in most parts of the nervous system. The localised parallelism of the retina is a strong indication of the totally interactive, and the simultaneous kind of neural processing that occurs in natural neural networks: the retina after all is itself a neural network. There is no argument or conflict in this concept. Indeed, this is exactly the kind of massive parallelism that appears well-suited to technological exploitation; for example, as CA-based vision implementations.

When it comes to parallel branching of the optic tract (mass axonal fibre bundles) there does seem to be difficulty. This stems from the realisation that neural signals transmitted by axonal bundles are the result of discrete parallel image filtering processes. In other words, the visual regions of the brain appear to be adapted for what the computer scientist calls *multi-tasking*. That is, they carry out more than one process simultaneously. Clearly, this is different from the kind of massive parallelism that occurs within a single task, or within a specific mapped region of the brain.

The terminus of the conduction pathways along which these signal pass is referred to as a **projection**. The optic nerve "projects" to the Superior Colliculus

(SC), or "projects" to the primary visual cortex via the lateral geniculate nucleus (LGN). Indeed, a major goal of current neurobiological and neuroanatomical research is to discover what these various visual sub-tasks are, and what they accomplish.

7.4.2 Massive Retinal Parallelism

The study of the retina, and of retinal function is a highly detailed subject in its own right and cannot be pursued in any detail here. What one wishes to do is mention a few significant aspects. The study of the retina, both as a neural network, and as an electronic circuit analogue, is detailed in Mead (1988). Our present interest is in the work of Egon Loebner, who has studied the biological retina in detail over a period of some thirty years (Loebner, 1987).

The Loebner reference depicts his "wiring diagram" representation of a typical or averaged vertebrate retina, culled from a very detailed study of over 100 research papers and 10 books.

It is noted that there exists a pre-ganglia branching of bipolar axons of the retinal inner plexiform layer (IPL)—as shown on the Loebner diagram. The relevant point here is that, because of this bifurcation, regional retinal signals are preprocessed by the *amacrine* cell groupings within the retina, but others are not. In terms of accepted neurophysiology, amacrine cells are the so-called "hidden units" of the retinal neural network. Thus, while a large percentage of retinal output neurons (ganglion cells) are nearly direct-connected to retinal receptor neurons (the RODS and CONES), others are mediated through the hardly-understood amacrine cell processes.

Our conclusion is that there are two kinds of axonal projections: those which are "direct" and those which are clearly indirect or "processed." These two types of axonal projection—direct and indirect—can usefully be linked with what this thesis identifies as direct Gibsonian vision. This then becomes the obvious complement of the established Marr-like image processing.

7.4.3 The SC and the LGN-Cortex

The Superior Colliculus (SC) is believed to be concerned with eye movements and foveation. This implies that the SC must be capable of "seeing" in its own right, since there could be no eye movement or foveation unless an image was actually and physically experienced. In other words, SC image processing must—by the very definition of its functionality—be iconic. Because of its direct image or "picture" function, it is appropriate here to regard SC vision as the potential source of Gibsonian vision. The hypothesis has never been verified by actual experiment, and so this view of SC function must remain purely speculative for the present.

Churchland (1986:228) gives an exposition the puzzling phenomena known as "blindsight" and "blindness denial." The characteristic of both conditions is that patients who have ostensibly lost all of cortical function are still able to "see" spatial patterns. They are unable to recognise the simple geometrical test shapes presented to them, but are able to sketch their basic forms. How could this occur in patients where the primary visual cortex is nonexistent?

Similarly, because the other major tract includes the LGN and the primary visual cortex, we regard the signals and processing here as supporting Marr-like computations. One can therefore physically trace two distinct pathways: the first, the Marr processing route, starting from the retina and projecting to the LGN-Cortex. This we have called LGN-C vision. The second pathway also originates from the retinal rods and cones (the receptor neurons), but projects directly to the SC. We shall for convenience refer to this route as exemplifying direct, or Gibsonian vision.

Figure 7.1, reproduced from Cotter (1990), shows diagramatically the main features of the human visual pathway. It may even be suggested by this diagram that the branches to the primary visual cortex at the rear of the brain are later developments within the definition of a Darwinian evolutionary process. This would suggest that biological intelligent vision—as opposed to primitive and direct SC perception—is a much more advanced and later function. Figure 7.2 shows a more schematic diagram of the optic tract, as understood by Arbib (1989).

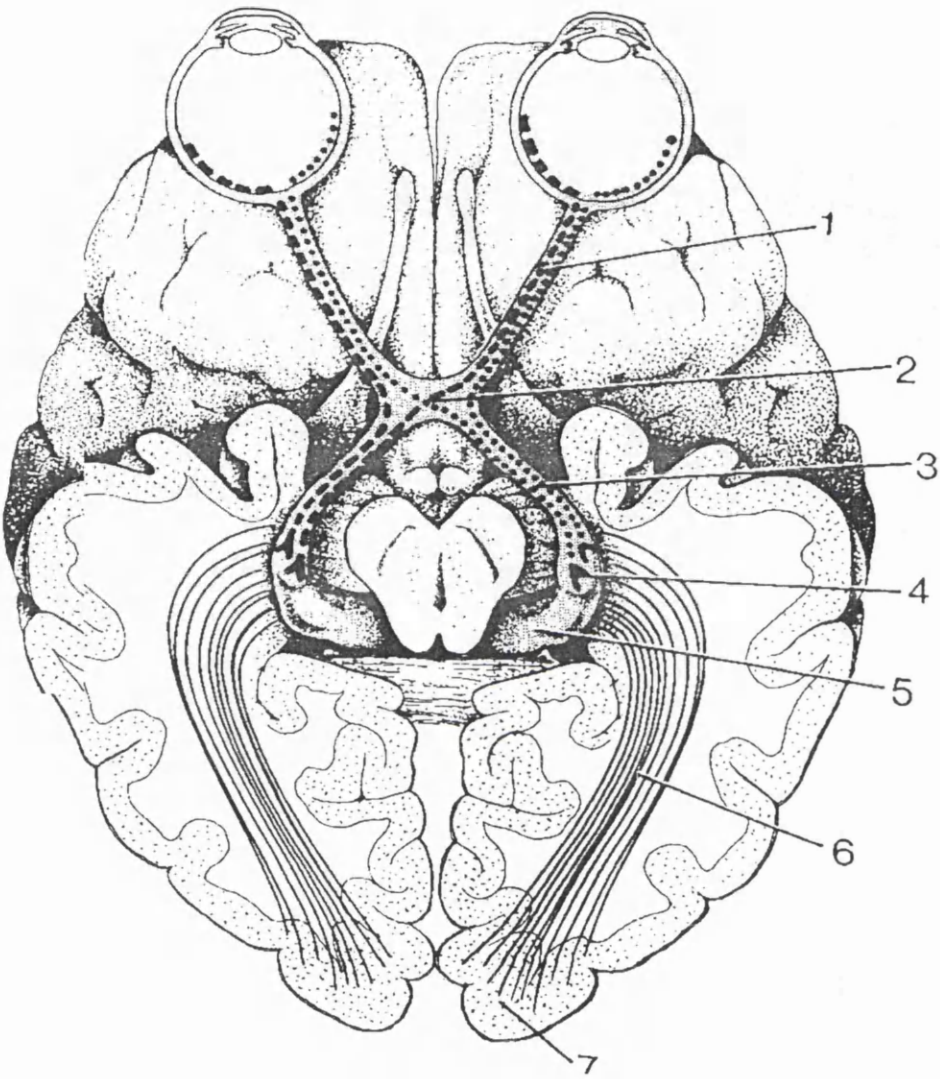


Figure 7.1 Diagram of the optic tracts—from Cotter (1990).
The annotated regions are:

(1) optic nerve
(2) optic chiasm
(3) optic tract
(4) lateral geniculate nuclei (LGN)
(5) superior colliculus (SC)
(6) retinal pathway
(7) visual cortex

It is seen even from this diagram that the evolutionary newer visual cortex is almost like an "add-on" to the older areas of the visual pathway, including the superior colliculus.

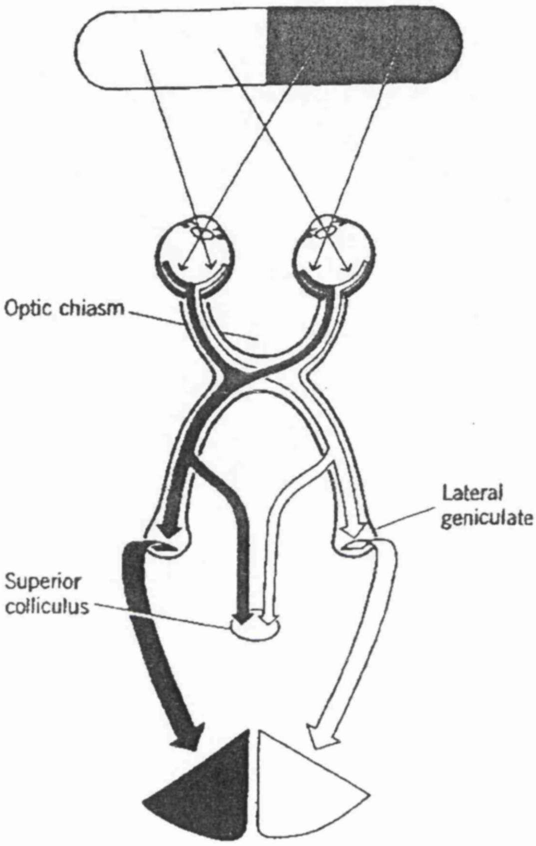


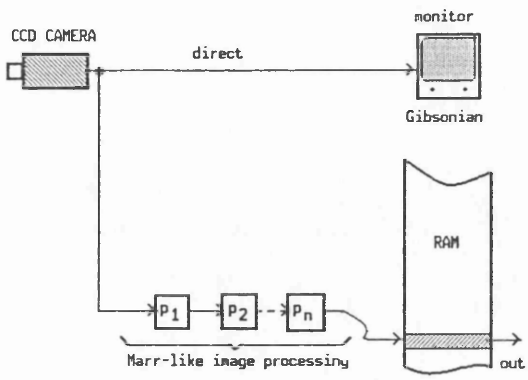
Figure 7.2 Another—more schematic—view of the optic tract. This illustration is reproduced from Arbib (1989). A large portion of the optic fibre axons (leading from the ganglia—see also the Loebner wiring diagram in reference) divides and projects to the superior colliculus (SC). The remaining axons project to the lateral geniculate nuclei (LGN), and thence to the receptor layer of the primary visual cortex.

The projection of mass axon fibres indicates that *iconic* (i.e. image-like) information is being sent simultaneously to two separate vision regions of the brain. Does this imply two distinct, parallel, vision mechanisms?

7.5 Modelling the Optic Tract

Figure 7.3 illustrates diagrammatically the writer's concept of the visual pathways, from a technological perspective. It is seen that the two main projections, or branches, lead to models of both SC and LGN-C vision. The SC receives direct images from the retina, as represented here by a CCD camera. The SC is modelled also as a video monitor. LGN-C processing is represented by a sequence of processes, $p_1, p_2, \ldots, p_n$, which ultimately access the contents of a distributed and associative memory—representing the cortex and any extra-cortical phenomena. It may be seen that the directly perceived image on the video monitor has no (direct) influence on cortical memory access or function.

The existence of mass-fibre (axonal) projections is the justification for our representing SC information as purely iconic in nature (with complex logarithmic mapping from retina to SC core). This is why the SC is represented as a picture "monitor" in figure 7.3. Following Marr-like image processing sequences $p_1, p_2, \ldots, p_n$ and a massive interaction with associative memory (cortex), the image data appears as a purely symbolic representation. By this much later stage data is represented as abstract neural signals—which can interact directly with cortical and extra-cortical signals to yield intelligent vision. Researchers do not as yet understand these processes; hence the need for conventional AI programming (e.g. Prolog) to define and encode the requirements of advanced neural function.



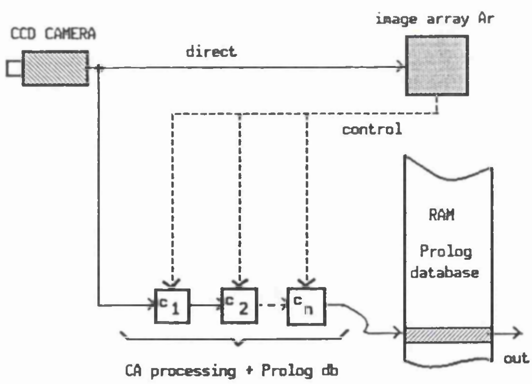


Figure 7.3 A technological interpretation of natural vision.

The upper diagram represents a technical interpretation of dualist vision. Marr-like image processing is represented by the processes $p_1, p_2, \ldots, p_n$ plus RAM accessed by associative or some other sparse-indexing mechanism. Gibsonian vision is represented by a video monitor.

In the lower figure the Marr image processing is replaced by CA planar processing, and an interactive Prolog knowledge-base. The monitor's function is replaced by video RAM, or other memory model.

7.6 Preattentive Vision

This phenomenon was investigated in some depth by Treisman (1985), and others. Their studies related to a human subject's ability to discern irregularities in patterns **preattentively**—by which is meant in less than about 400-600 ms following exposure to the visual stimulus. This is too short an interval for controlled eye movements to occur, and for the image recognition processes to respond; hence perception must be localised in some sense. Treisman performed a large number of psychological experiments in which subjects reported a segregation occurring in the display field despite the very short exposure times. She found that subjects could preattentively segregate test fields in which only a single property fully defined the segregation. An example of the many kinds of test field used in the Treisman experiments is reproduced here as figure 7.4. The phenomenon demonstrated by these Treisman tests is sometimes known as "popout."

Although McCafferty (1990) interprets these results as evidence of the existence of feature maps (or parameter maps) within the human visual system, this thesis takes a different view. It is proposed here that the phenomenon of preattentive vision is yet another manifestation of direct—that is, Gibsonian—vision. The speed of reaction in the Treisman tests is an indication that the deeper cortical and memory processes did not have time to respond.

The Treisman experiments seem to show that there is an immediacy to vision—as well as a quite deliberate and systematic analysis of the "meaning" of visual stimuli. The latter process is almost certainly associated with the slower axonal pathway of the LGN-C route. The former may be connected with the faster axonal projection to the SC. Researches on axonal information propagation are considered in some depth in Loebner (1987), and Churchland (1986). The Treisman results beg the question: If preattentive vision is not Gibson-like image processing, then what other kind of vision is it?

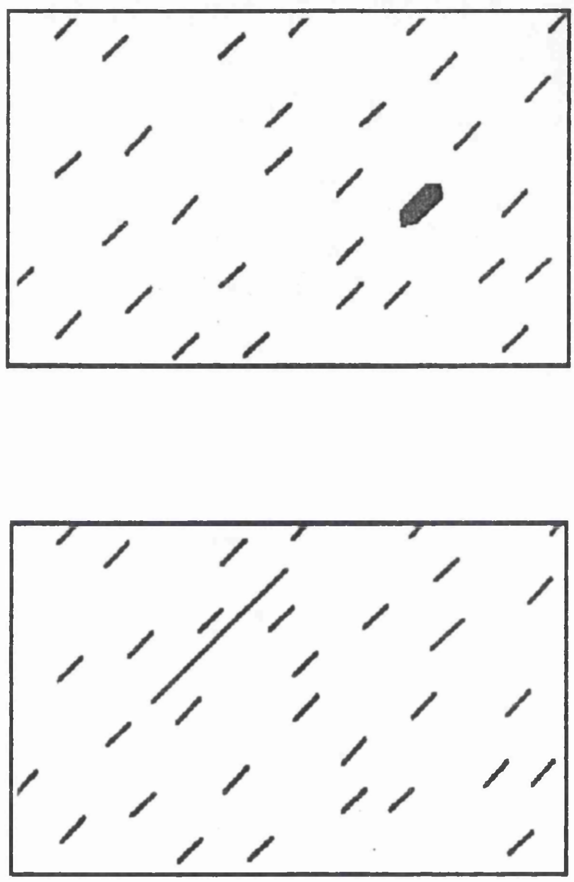


Figure 7.4 Preattentive vision as demonstrated by Anne Treisman's experiments.
The upper figure shows a thicker line which is preattentively detected. The long line in the lower illustration is preattentively noticed. A wide range of test figures has been evolved to study this phenomenon. The above illustrations were adapted from McCafferty (1990).

7.7 Chapter Summary

It appears that, despite a variety of implementational and technical problems, artificial vision is generally understandable and tractable. This is in sharp contrast to biological vision which appears dualist in nature—and intangible. This intangibility seems to stem from a direct and immediate kind of perception, which is different from the conventional approaches of Marr-like image processing. One has great difficulty in characterising the abstract phenomenological properties of direct vision—an experience that cannot as yet be understood, or captured in our models of technical vision. The dualist concept of Marr-Gibson is useful in the modelling of advanced vision, and seems to be justified on the basis of recent neuroanatomical findings.

The neuroanatomy of the optic tract and cortex appears to substantiate Marr-like theories of biological image processing, while the Treisman demonstrations of preattentive vision can appear to support Gibsonian concepts of direct vision.

CHAPTER 8

COMPUTING REQUIREMENTS

The choice of a suitable modelling environment is crucial to the success of any simulation, especially that of biological functions. The use of adequate computing facilities for project work is almost mandatory today, both as a means of realising the model itself and as a prerequisite to any envisaged hardware developments. The choice of appropriate software is very important, since the complexity of the model may require the facilities of more than just one programming language. This is the case in the present project, where C, Pascal, assembler, and Prolog are combined to produce a hybrid, multi-lingual computing environment. This is necessary because Prolog is still not the most suitable programming language for every purpose, although the scope and applicability of Prolog is increasing rapidly. Appendix B of this thesis provides some notes on the Prolog language.

This chapter briefly considers the software aspects of the project. Fully-detailed code listings will not be reproduced, but some code fragments deemed necessary to illustrate the concepts will be shown here and elsewhere as required. The following sections assume that the reader is familiar with the programming languages and concepts cited.

8.1 Low-Level Image Processing Software

A set of basic low-level image processing primitives (called "VPC") have been coded by the writer in the C programming language, having originally been developed and tested in Pascal, using the Turbo Pascal V6.01 dialect (Borland, 1988). A listing of the supported VPC code primitives is provided in Appendix A. These are essentially low-level image processing routines, and are of conventional design and purpose.

A similar range of CA functions is also available in the form of CA lookup table (LUT) definitions—Chapter 4. The combination of a CA algorithm and its defining lookup table constitutes a CA processor. These functions are also listed in Appendix A, as a set of Prolog calls to VPC. A number of specialised routines have been devised that use **iterated sequences** to reproduce the soap film models. As was discussed earlier in this thesis, the soap film concept is a necessary mechanism for the realisation of Gestaltist properties of natural vision in technology.

8.1.1 VPC Low-Level Routines

The VPC code implements a range of conventional image processing algorithms, based on a restricted 128x128 image format. This small image size is necessary because the system has been implemented on a standard IBM PS/2 Model 30-286 computer. The choice of PC development platforms was itself dictated by a number of factors, including the relative ease of interfacing Turbo C code with Turbo Prolog modules. The CA image planes and arrays have been modelled in Turbo C (as part of the VPC system), and obviously high-resolution images would run very slowly on a PC. All factors considered, a 128x128 resolution was regarded as adequate to demonstrate the conceptual principles of the project—the modern PC memory capacity can cope with this. However, the initial rule development work was done on an 64x64 image format, and so some illustrations are reproduced in this resolution.

The IBM PS/2 has been fitted with a VIDI-PC PLUS (16 grey-level) image digitizer and frame-grabber, plus an image capture program written in Turbo C V2.1. Captured images are further manipulated within an image array as shown in figure 8.1. This multi-level array also functions as a **parameter space** for the storage and accumulation of evidence for the edge-map process. This will be discussed in a later chapter.

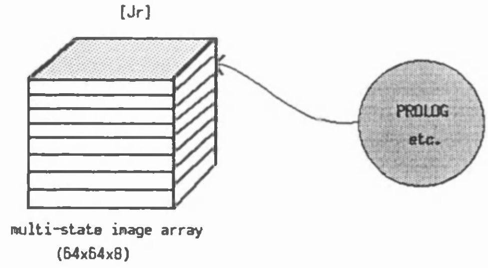


Figure 8.1 Multi-level image array [Jr] can be manipulated directly by low-level VPC procedures called from Prolog. It is also used to implement a Parameter Space.

By way of illustration, the following VPC code fragment shows the white pixel count function (cwp). This is typical of the VPC command set:

```
/***********************************************************/
 void cwp_0 (BYTE *Pr, WORD *CP)
/* Count the number of nonzero pixels in an image */
{
   BYTE
      i,
      j;

   WORD
      n;

      n = 0;
      ip_count = 0;

      for (j = 1;  j <= max-1;  j++)       /* max is preset to 128 */
      for (i = 1;  i <= max-1;  i++)
         {
         if (*(Pr + n) != 0)
         ip_count = ip_count + 1;
         n = n + 1;
         }
         *CP = ip_count;
}
/***********************************************************/
```

Notice how both the input and output parameters are specified by pointers. Thus the parameter addresses are passed to VPC by memory reference from the Prolog call.

8.1.2 CA Lookup Table Functions (Rules)

As mentioned, a similar set of image processing functions is available from the CA image processor, as defined by a CA lookup table, or rule. These procedures are binary functions, and grey-level image features have to be obtained by combining appropriate binary subimages. The CA rule sets, together with the CA mechanism, form the CA processor. These mechanisms are also coded in the C programming language.

The required image processes are obtained using CA lookup tables as explained in Chapter 4. Since we are only concerned here with binary subimages, extracted from a grey-scaled or colour original scene, the 2-state lookup table format needs only a 9x2 matrix to represent it. When "soap film" or search film (SF) processing is required, the full 3-state lookup table is used. This requires a 73x3 matrix array.

The following is an example of a 2-state totalistic lookup table rule which constrains the CA to perform image point noise removal:

Totalistic NOISE POINT Rule									
	8-sum								
cell state	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0

It is seen that if the 8-connected totalistic value is 0, then the central element is set to 0. If the totalistic value is 8, then the central element is set to 1 (8 x 1 = 8). In other words, the current pixel, if it is an isolated image element, is set to the background binary value. A similar filter process is available in conventional grey-scale or colour image processing: the current pixel is set to the surrounding value, thereby removing the point source. It is implicit in this operation that the presence of such isolated pixels represents genuine image "noise" elements that are to be removed.

The following C code fragment shows how a lookup table can be defined for a specific function. During an earlier initialisation call, the lookup table entries are initialised to the states 0, 1 and 2. This means that subsequent redefinitions normally only need to alter a few of the table entries to form a new CA processor rule. The example rule table shown below actually requires more extensive modification, which is carried out within a C code loop.

```
/**********************************************************/
 void otl_0 (void)
/* set up an OUTLINE lookup table for the CA processor */
{
   BYTE
      i,
      j;

      for (i = 0;  i < 72;  i++)
         {
         lkup1[i][0] = 0;
         lkup1[i][1] = 0;
         lkup1[i][2] = 2;
         }
         lkup1[72][0] = 0;
         lkup1[72][1] = 0;
         lkup1[72][2] = 0;
}
/**********************************************************/
```

8.1.3 Turbo Prolog Predicate Calls

Turbo Prolog (Borland, 1989) is a high-level artificial intelligence language which is used in this project to represent real-world facts and knowledge. The high-level aspects will be discussed in Chapter 9. The present requirement is that Prolog, if it is so desired, shall be able to activate low-level VPC or CA Rule processing directly. As a general principle, Prolog is not expected to handle dense pixel-based information processing, but may "command" that such image functions be invoked at a low-level via an appropriate VPC call.

A typical call is a Prolog predicate such as "cwp_0" — count the white pixels in a binary image. This predicate is normally included as part of an initial image structuring call to VPC, but could well be required separately at a later stage within a Prolog processing loop.

The "_0" suffixed to the call is a Borland convention to establish correct flow patterns in parameter passing. For example, the custom call "get_dta_0" has the flow pattern (i,i,o) — which means that the first two parameters are passed into the routine and the third is passed out as data—see below.

Parameters can be any standard Turbo Prolog types, including complex structures. Alternative flow pattern variants may be specified for the same Prolog predicate—for example (i,i,i). When more than one flow pattern is required, the predicate must be suffixed with different positive integers to identify each distinct flow pattern variant. For instance, this predicate call has two flow patterns:

_get_dta_0() — for the 1st flow pattern (i,i,o)
_get_dta_1() — for the 2nd flow pattern (i,o,o)

In the practical programming work required in this thesis only one flow pattern will be specified in most cases.

The following table illustrates some typical Prolog calls devised for this project, together with their associated flow patterns:

TABLE 8.1: TYPICAL VPC CALLS FROM PROLOG

VPC Function call	Flow Pattern	Language ID
_camd	–	language c
_get_reg(int)	– (o)	language c
_get_info(int,imdom)	– (i,o)	language c
_get_mat(int,int,int)	– (i,i,o)	language c
_get_area(int,int)	– (o,o)	language c
_get_imlist(llist)	– (o)	language c
_init_graphics(int)	– (i)	language c
_plt_pix(int,int,strg)	– (i,i,i)	language c
_mainloop(int)	– (i)	language c
_lda	–	language c

Refer to the Turbo Prolog Reference Guide (Page 77) and the Turbo C Programmer's Guide for further information (Borland, 1989 and 1988). Appendix A in this thesis lists the available VPC and Prolog calls, and Appendix G discusses the Borland interfacing protocol—as amended by the writer.

8.2 High-level Vision Functions

High-level vision functions are essentially Prolog procedures composed of a number of low-level predicates (calls) to either VPC or CA Rules, both of which are implemented in the C programming language.

The following Turbo Prolog code fragments illustrate the method of directly calling a VPC function from within a Prolog loop. This Prolog function collects data from a structure defined within VPC.

```
%-----------------------------------------------------------
%  datascan_1
% Scan VPC structured image data and extract region functor

      datascan_1 (I) :-
         I < 16, !,
         _get_info (I,X),          % call VPC function
         check_value (I,X), !,     % make tail-recursive
         I1 = I + 1,
         datascan_1 (I1).
         datascan_1 (_).

      check_value (No,X) :-
         X = f (Area,Perm,Xc,Yc,SF),
         Area > 0, !,
         I = No + 6,
         attribute (No),
         cursor (I,20),
         writef ("region %2d %8d %7d %7d %6d %8d",
                  No, Area, Perm, Xc, Yc, SF), nl,
         assertz (im_region (No, Area, Perm, Xc, Yc, SF)).
         check_value (_,_).
%-----------------------------------------------------------
```

The clause "datascan_1(I)" in effect copies the image regional structure from VPC into a Prolog functor: f(Area, Perm, Xc, Yc, SF), where Area is the regional area count, Perm is the perimeter, Xc and Yc are the numerical x-coordinate and the y-coordinate, respectively, of the image (region) centroid. SF is a simple regional shape factor.

The predicate "check_value()" ensures that only valid regions having a nonzero area count are analysed. This predicate also prints out the information stored within the Prolog functor, which is itself stored in the system database for later use.

In effect, Turbo C enables one to code new efficient image processing predicates that can be called from the Prolog main module.

In general, a series of Prolog calls can be made to VPC within a single procedural predicate. See Batchelor (1991) for an excellent discussion of these and related topics. For example, one could code:

```
can_see_sky :-
   _lda ("image_2"),     % load an image
   _cpy (Ar,Dr),         % copy image array
   _seg (Dr,11),         % segment the image
   _coa (X1,Y1),         % get the centre of area
   _thr (11,11),         % threshold the region
   _cwp (A1),            % count the area pixels
   A1 > 70,              % more than 70 units?
   Y1 > 40.              % above horizon?
```

8.3 Hardware Facilities

As mentioned, the project software has been developed on an IBM PS/2 Model 30-286 platform. The system comprises an 80286 processor running at 10MHz, 4 Mbytes of on-board RAM, 80287 maths coprocessor, and a 32 Mbyte hard disk. There are facilities to use 1.44 Mbyte and 1.2 Mbyte high-density floppy diskettes. The system monitor is an IBM 8513 VGA colour unit, and a mouse pointing device can be used. Output can be dumped to an HP Deskjet 500 printer, if required.

The VIDI-PC PLUS frame-grabber used in this project is an ISA plug-in adapter card, and can digitize video images to a maximum resolution of 512x512 pixels—in either 16 colours or grey-scales. As mentioned above, a 128x128 block-pixel image format was recoded in C for the project. A VIDI electronic filter is used to produce RGB colour frames for the display of colour images on the VGA monitor. Additional input in the form of a 400 dpi hand-scanner is available. All interfacing between VPC and Prolog is software-based, so there is no requirement for interfacing hardware or interconnecting cables.

8.4 Chapter Summary

This chapter has outlined the basis of using a multi-lingual approach to software development. This is required because no single language is sufficient, by itself, to provide the required performance. Tests conducted by the writer have shown that while Prolog is excellent for modelling intelligence and knowledge-based aspects, it is not suitable for direct image manipulation at the pixel level. Pascal and C are much better for this purpose, but are usually considered to be more difficult for the coding of high-level intelligence functions. The approach used here is therefore a compromise solution.

The hardware requirements for this project are conventional, although some modifications had to be made by the writer to the VIDI PC card's frame-grabber C code to customize the unit for the project.

CHAPTER 9

CONSTRUCTING THE VISION MODEL

This chapter attempts to construct a model of human visual perception—from low-level retinal input to higher level image comprehension—in satisfaction of the principles and goals discussed in previous chapters.

Basically, this approach combines a CA implementation of a novel edge detection approach with rudimentary concepts of cortical memory and analytic functions, implemented in the Prolog programming language. Given this relatively wide coverage, the proposed model can only be expected to provide the key features and some pointers to potential future developments.

It is hoped, particularly, that our model can draw attention to what the writer understands is meant by "direct" vision—and the severe philosophical and technological problems posed by this phenomenon. The system to be described accords with Marr-like image processing, but is also sympathetic to Gibson's theory of direct vision.

9.1 General Considerations

In addition to Marr and Gibson, theoretical work by Grossberg and associates at Boston University, in the USA, has provided a strong mathematical and neural network-based approach to biological vision. Grossberg's approach is clearly "Marr-like" — but he nevertheless severely criticises the inadequacy of, for example, the Laplacian edge detection mechanisms of Marr and Hildreth (1980). The 19th Century Hamiltonian and Laplacian "cost" methods are also criticised, since Grossberg advocates 20th Century dynamical and "synergetic" systems. See Grossberg (1987b:2). The Grossberg-Mingolla (G-M) model contains a weak element of what we would identify as Gibsonian concepts, and also focuses on important vision research completed since Marr's time.

The model developed in this thesis also coincides with the Grossberg concepts of boundary completion and featural "filling-in." However, the present work attempts to use cellular automata for this purpose, rather than the more complicated analogue models of artificial neural networks favoured by Grossberg. The present model additionally extends to ideas of higher level (usually extra-cortical) visual knowledge for a demonstration of simple scene interpretation: the Prolog programming language is used for this purpose. In addition, the CA approach can demonstrate perceptual effects that, to the writer's knowledge, cannot easily be produced by the G-M system. For example, the notion of four isolated corner dots forming a "square" (or other geometrical shape) cannot be generalised by the G-M model without additional data input.

An important by-product of the CA model is **data reduction**. Data compression is often a natural consequence of the translation from a pixel-based (iconic) form to an object-oriented symbol. That is, the pixel-level data is organised hierarchically into meaningful regions and objects, which in turn are recognised and classified by high-level knowledge. This, of course, happens in virtually every machine vision implementation where image grouping is based on **segmentation**. However, the data-reduction properties of object orientation (e.g. for the transmission or storage of images) are usually never emphasised, or even recognised as such. Massive data reduction—in its various guises—is a powerful feature of biological signal processing.

But data reduction poses problems of its own. For one thing, it is not possible to understand how we humans are able to "see" the fine details and characteristics of complex imagery if the image data is pooled—as happens in the classical "Grandmother Cell" postulate. This topic is discussed in the Rentschler and Caelli paper in Haken (1990:233), in which the essential thrust is directed towards the problem. This philosophical enigma is a main reason why the writer regards the basic concept of Gibsonian direct perception as being crucially important in the understanding of natural vision.

9.2 Scope of an Adequate Vision Model

The writer considers that adequate investigation into basic image processing and segmentation has been carried out in-depth in recent years, particularly during the past decade. The new problems therefore are concerned with developing a strong paradigm for vision, based on the known principles of human vision and neuroanatomy. The model to be discussed is developed upon the writer's belief that psychological factors are of crucial importance in visual understanding. The modern approach to artificial vision modelling is characterised by studies of the following topics:

- perceptual image grouping/organisation at low level
- visual and general psychological issues
- the principles of artificial intelligence
- the fundamentals of information theory
- dynamical (synergetic) models and cellular automata
- massive parallelism in image computations
- simulated biological neural signal processing

In addition to the above list of approaches and models, White (1992), discusses the new directions in AI, which include the undernoted three inter-related themes, or truths. These three principles are seen by a new generation of AI researchers as a means of finally breaking free of the current AI bottlenecks—allegedly caused by the von Neumann programming legacy. The new AI themes embrace the following:

1. The rise of contextualism—which holds that all processes in AI are deeply context sensitive, including logic itself.
2. The hypothesis that mathematical logic and set theory have philosophically serious limitations in their ability to give adequate expression to any real-world situation.
3. The idea that biological memory is fundamentally not quite the same thing as technological computer memory—that is, a system comprising sets of discrete and neatly stored facts.

This thesis is in full agreement with the three principles. The first point is incorporated within the writer's model of concentric context in Prolog (to be discussed). The second theme has been stated earlier in this thesis—namely, that advanced mathematics alone cannot solve the vision problem; and that fuzzy mechanisms need to be introduced. The third item was mentioned in Chapter 3 of this thesis, and in the writer's previous research—McIndoe (1988). As was suggested before, the concession that we make to Prolog is that, for the time being, it provides an adequate means of emulating aspects of complex neural network memory circuitry whose processes cannot as yet be understood in the **symbolical** sense.

The future practical implementation of advanced systems is likely to depend on massively-parallel computations, as in biological vision. It is proposed here that future hardware-based cellular automata, which can emulate neural networks, are likely to fulfil most of the above mentioned image processing requirements.

In conclusion, the model being proposed is both a broad-based and a systems-oriented one, in the conviction that in this approach it is necessary to take account of a wide variety of evidence, in order to construct a convincing vision paradigm. This means a reduced depth of coverage in many subtopics. However, in the modular approach being advocated, a framework will exist such that necessary details can be expanded in later research. New modules or algorithms can, in effect, be "plugged-in" as concepts develop.

9.3 Pre-Project Evaluation Studies

The writer has constructed several evaluation models in software. These models were intended to produce video and synthetic images for test purposes; and to evaluate potential mechanisms of CA-based image segmentation and the interaction between Prolog and image processes. The segmentation model shows, among other things, that simple colour or grey-scaled images can provide a fast and unambiguous segmentation.

In particular, and in accordance with the "Kelvin Way" principle described earlier, it was found that simple regionalisation of images was sufficient to perform adequate segmentation and recognition of a large class of real images. These image classes are more likely to be region-based, as opposed to being contour-defined or pixel edge-based. For example, the human ability to recognise politicians from simple caricatures would not normally be regarded as a region-based process. Note, however, that this simplification is an algorithmic one. The mechanisms that may enforce such simplified images may come from higher level processes.

The problem, then, becomes one of discovering how to create suitable restricted-range "cartoon-like" segmented colour images from complex natural colour or grey-scale video input. This is a semi-symbolic, or intermediate, representation, and forms the basis of the model to be developed here. [1]

9.4 Requirements of a Vision Model

In keeping with the general specification of Chapter 1, and the later requirements of this thesis, a convincing and robust model of human vision should be able to carry out at least the undernoted minimal tasks:

1. Build robust and noise-free representations which are invariant to image-plane absolute position, scale, and orientation.

2. Fill gaps in noisy or broken contours, and complete boundaries, including the classical illusory image models as discussed in the standard textbooks on visual psychology.

3. Dynamically extract the "high-energy" primitive image features at multiple continuous resolutions and scales (e.g. corners and endpoints, image preattentive features).

4. Guide segmentation by localising regions, and distributing the enclosed featural qualities in a meaningful way (i.e. make the images semi-symbolic, and fuse image data in accordance with the Gestalt Laws of Organisation). This is called "filling-in."

5. Find centroids, and determine basic image regional statistics of the resulting images for transmission to the higher cognitive or fully symbolic levels (e.g. a Prolog-based expert system).

A system to perform these tasks contains a number of distinct models and concepts of quite recent origin. These include diffusion-reaction mechanisms, parameter or feature space representations, feature maps, mass-parallel (shunting)

[1] It is interesting to note that Rich (1983) advocates such intermediate representations as a means of simplifying and (or) re-mapping problems within AI.

feedback, receptive fields, preattentive mechanisms, synergism, and so forth. Although often discussed under distinct headings, many of these ideas converge on a common set of **basic principles** as have been discovered by neuroanatomists, physiologists, and psychologists. In general, active dynamical, synergetic, and non-linear processes are now to be preferred to the conventional static, filter-based, linear, and image-processing oriented models of the past.

Neuroanatomists report that the visual pathway uses extensive local connectivity, simultaneous (asynchronous) processing, and a cortical "magnification factor" which can be modelled by logarithmic mapping. Physiologists report that imagery is rapidly processed by specialised gradient operators and motion detectors, and that these primitive biological feature detectors are widely distributed, and serve many purposes. Experimental psychologists report the assumptions inherent in natural visual processing, by finding visual illusions to which it is susceptible. However, there has not existed a linear path from the conventional image processing roots of machine vision to new models of biological vision. There is a fundamental conceptual gap that has to be bridged in order that the role of psychology can be understood, and subsequently realised in technology.

The present discussions and our systems-based model are offered as a starting point towards new concepts in machine vision. The unifying principle here is that complex images are reduced by the processes mentioned in the steps 1–5 above to yield **the simplest possible symbolic image**. This is perhaps the most important proposal of the present work. The concept of simplified (i.e. symbolic) images is supported by several of the prominent researchers in AI. For example, Kohonen (1988:120) writes

> **"... intelligent information processing seems in general to be the creation of simplified images of the observable world at various levels of abstraction, ... ".**

This observation is in keeping with the writer's "Kelvin Way" concept mentioned earlier in this thesis; and indeed with the entire philosophy of the present project.

9.5 The Grossberg-Mingolla Model

The following quotation is taken from Grossberg (1987b:20), in connection with what he calls the "filling-in dilemma". The writer considers that this problem is central to appreciating the concept of **direct** and **processed** vision, and the resulting Marr-Gibson paradox.

> "Any linear and feedforward approach to spatial vision is in fact confronted with the full force of the *filling-in dilemma*: If spatial vision operates by first attenuating all but the edges in a pattern, then how do we ever arrive at a percept of rigid bodies with ample interiors, which are after all the primary objects of perception? How can we have our edges and fill-in too? How does the filling-in process span retinal areas which far exceed the spatial bandwidth of the individual receptive fields that physically justify a Gaussian smoothing process?"

The solution to this problem delivered the Grossberg-Mingolla (G-M) model, which is remarkably similar to the concepts which form the basis of the approach developed in this thesis. The philosophy behind both of these new approaches is that a very small set of basic neural network functions (or their CA equivalents) can account for almost all of the observed phenomena of natural vision.

As discussed in Grossberg's submission to DARPA (1989), the use of consistent basic mechanisms leads to a uniform approach in neural modelling. A recurring problem is that many AI algorithms for machine vision have been much too specialised for application to real-world problems. Such algorithms are typically designed to deal with one type of information—for example, boundary, disparity, curvature, shading, or spatial frequency evidence. Moreover, such algorithms typically use different mathematical schemes and representations to analyse each distinct type of information. This means that their unification into a single general-purpose procedure is very difficult. For many such AI-based algorithms, other types of visual signals or image information are regarded as "noise" elements, rather than as cooperative sources of ambiguity-reducing data.

A severe criticism of Marr-Hildreth edge detectors—and of many other explanations of natural vision—is that they appear to "throw away" information. As every sighted person knows, humans and animals do not perceive a world consisting solely of edges or outlines. There is solidarity, and fine-grained, textural, colour in the images that we perceive. It is strange that, except for the G-M model mentioned, this aspect of natural vision has largely been ignored by most of the machine vision community.

9.6 CA-based Vision Processing

Figure 9.1 shows a portion of a 3-state image which which has to be processed using simulated CA mechanisms. (Appendix C gives the basic image array and convolution notation as favoured in the conventional image processing algorithms).

The concept of a CA "receptive field" is hereafter denoted by the 3x3 cell kernel or neighbourhood. Recall from Chapter 4 that a **kernel** involves a process or summation over nine cells, whereas a CA **neighbourhood** only includes the eight nearest neighbours. The CA receptive field (RF) is assumed to be centred on a cell (i, j) within an image array or field of $m \times n$ cells. The usual local neighbourhood summation includes all cells within the receptive field, except the centre or current cell $I(i, j)$.

From Chapter 4 it was seen that a CA transformation rule can be written as

$$\phi_t \leftarrow F[\sigma_t, \phi_{t-1}] \tag{9.1}$$

In terms of a predefined 8-cell local nearest-neighbour set $\{i : i \in 1, 2, \ldots, 8\}$ the RF sum at time t is

$$\sigma_t = \sum_{i=1}^{8} f(\phi_i) \tag{9.2}$$

In terms of an $m \times n$ image array indexed by (i, j) the local RF window $I(k, l)$ is scanned (in serial CA computer simulations) over the entire image, thus

$$\phi_t^{i,j} \leftarrow F\left[\left(\sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} I_{k,l}\right) - I_{i,j} \ , \ \phi_{t-1}^{i,j}\right] \tag{9.3}$$

where $F[\cdot]$ is a function defined by lookup table.

The term $I_{i,j}$ represents the current cell, and the RF central element, which can be either excluded from the summation (in the case of a CA-type neighbourhood) or included (in the case of a conventional or CA image-processing kernel). Thus the same RF neighbour summation mechanism can support both CA rules and conventional image processing. This could be an important and useful design factor in future hardware implementations of cellular-based circuitry.

The transformed state of a cell in the $m \times n$ image array is represented as ϕ_t, while the same cell's previous state was ϕ_{t-1}.

The result of scanning the CA 3x3 cell RF over the entire $m \times n$ cellular image array (denoted by $I_{\phi_{t-1}}$) is the **transformed** image I_{ϕ_t}. Thus we can write

$$I_{\phi_t} \;=\; \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} F[\cdot]\, \delta(i-k, j-l)\; I_{\phi_{t-1}} \tag{9.4}$$

where $F[\cdot]$ is the cellular automata transformation rule defined in eqn. (9.3) and δ is the Kronecker (delta) function which is needed to access individual image elements for the summation.

The above operation assumes that the CA is being simulated on a conventional serial computer, and represents a single pass, or one iteration, of the transformation rule. In a truly parallel computing environment, all cellular transformations may occur simultaneously on a synchronising clock pulse—or else neighbouring cells may function asynchronously and totally independently (i.e. in parallel).

The result of applying rule $F[\cdot]$ is that the previous image $I_{\phi_{t-1}}$, representing an array of cells at time $t-1$, is transformed to a new image I_{ϕ_t} at time t. If there are further iterations of the rule, then the current image I_{ϕ_t} is substituted in eqn. (9.4) for $I_{\phi_{t-1}}$, and so on.

As mentioned in Chapter 4, iteration of the transformation rule will continue until the cellular automaton enters a predefined stopping or halt state. The result is that an original CA image array held in a memory buffer, or represented in a video RAM array, is transformed (hopefully) to some new and meaningful array of image cell states. Thus, the power of CA methods is derived from the ability to define complex image transformations by CA rule lookup tables. In addition, the CA transformation rules can be **dynamically** altered while the CA is running. Such dynamical re-defining of a CA operation using the described mechanism of lookup rule tables (LUTs) is potentially very powerful—particularly when the entries within the CA lookup table may be subjected to *external* signals, or other influences.

Note that in eqn. (9.4) the image cell transformations are applied to all cellular array elements, except for a border equal to one cell in thickness at the image periphery. As was mentioned in Chapter 4, a border of 0s is usually required to allow CA rules to work correctly.

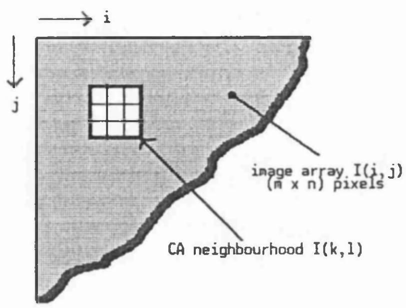


Figure 9.1 Illustrating the concept of a CA image array I(i,j) of dimensions $m \times n$ as scanned by a 3x3 local window, or receptive field (RF) denoted by I(k,l).

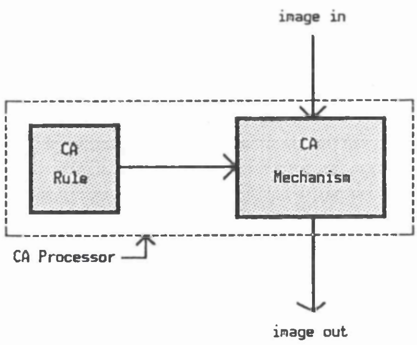


Figure 9.2 This diagram shows the general arrangement of a CA image processor that can transform an input image to some output image—as determined by a CA rule. It may often be more convenient to regard the two blocks contained within the dashed line as the complete CA image processor.

9.7 Local and Global Processes

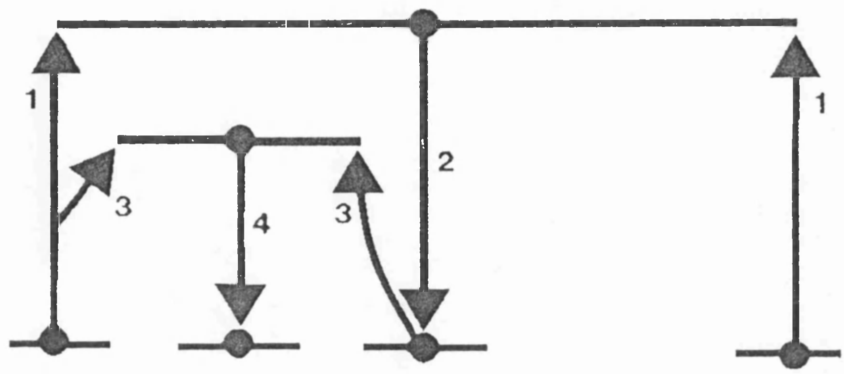
An important aspect of both the G-M approach and our proposed CA vision model is the interaction between local and global cellular processes. By definition, the receptive field (RF) of a specified CA cell is strictly *local*—involving eight (or nine) nearest-neighbour cells. Similarly, in the MP model neuron of Chapter 3, a neuron cell usually interacts with local neurons in both feedforward and feedback modes. The question thus arises: How are the *global* cellular properties derived from purely *local* cell interactions? The following subsections examine this in brief outline, while Appendix I derives the Gaussian ON-OFF receptive field model ("mexican hat") found in every mode of biological neural signal processing.

9.7.1 Global Propagation in the G-M Model

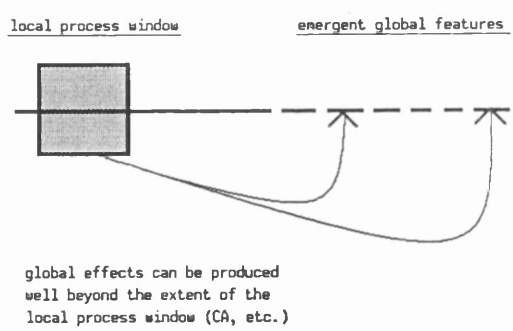
Figure 9.3 is a diagram of the basic BCS (Boundary Completion System) mechanism of the Grossberg-Mingolla (G-M) model, which can complete both local and global boundaries. The mathematics are described fully in Grossberg and Mingolla (1985a, 1985b), Grossberg (1987b), and DARPA (1989). The process is that of neuron excitation and inhibition. This is often compared to "voting" in a democratic system, but in neural networks the votes are redistributed by lateral axonal interconnectivity among neuron groups.

Referring to figure 9.3a, edge elements (1) which have a sufficiently strong activation at the lowest cellular level, can activate pairs of cells at intermediate or higher competitive levels. This results in what is called the "CC Loop" (the Competitive-Cooperative Loop). Depending on the accumulated activations from ON-CENTRE, OFF-SURROUND and OFF-CENTRE, ON-SURROUND neuron mass-shunting fields, there can be direct activation feedback to edge elements which lie *outwith* the receptive field of a given neuron (2). The local neighbourhood is the activity shown at locations (3) and (4). The G-M edge strength activations are determined initially as the output from orientated edge-detector masks.

By this process, the G-M model can generate **additional** global edge elements from purely local cellular interactions—including, especially, many classical illusory boundaries familiar from textbooks on psychological vision. In other words, the process can create edges which do not physically exist as image contrast differences.



(a)



(b)

Figure 9.3 Illustrating the mechanism of local-global interaction in the Grossberg-Mingolla (G-M) model. The diagram suggests how the BCS module can synthesize both real and illusory boundaries; that is, produce edges where no local physical image contrast exists.

(a) shows the basic concept of Grossberg's BCS mechanism.
(b) shows that local interactions can produce global effects.

9.7.2 Global Propagation in the CA Model

The G-M concept has demonstrated a fundamental property of local-global interaction, and Grossberg and his associates have been able to relate their vision model to neuroanatomical discoveries. In this section we show that **equivalent** properties can be achieved by using cellular automata processes—specifically, as a form of the iterated soap film rule. It is demonstrated that, although a CA receptive field includes only its eight nearest-neighbour cells, the iteration of a CA rule (any rule) is equivalent to the classical feedback concept.

Appendix I provides the mathematics of the simple RC digital filter, as discussed in Sullivan (1982). It is convenient to regard a cellular automaton as a form of **digital filter** in which the initial input is the raw image array states, and the resulting—iterated—output is the transformed, or processed, image states.

For convenience, and from Appendix I of the thesis, the simplest digital filter is shown to be approximated by the iteration (or feedback) formula

$$y(n) = b_0 x(n) + a_1 y(n-1) \tag{9.5}$$

where $y(n)$ is the current output, $y(n-1)$ is the previous output, and $x(n)$ is the input. The multiplying factors a_1 and b_0 are suitable constants.

This filter result is now used to show that our present ECM-HRS model (to be described below) is really a CA-type feedback mechanism, which is functionally equivalent to the G-M model. We use simple mathematics derived from the writer's software algorithm and source-code listings to describe the operation of the several steps of CA-based iteration.

Let the input to the CA processor block be represented by $Pr(n-1)$ and assume that some intermediate image, as may be transformed by an intermediate-state lookup table, L2, is $Cr(n)$. Then

$$Cr(n) \leftarrow L2[Pr(n-1)]$$

The intermediate-state array Br is defined as the 8-sum of the neighbour cells of intermediate image array Cr, so that

$$Br(n) \leftarrow \left[\sum_{8} Cr(n)\right]$$

The current output image array $Pr(n)$ can be related to its previous output image state $Pr(n-1)$ by using a delay function, D, which, in effect, represents storage of the previous state in the computer memory array, thus

$$Pr(n-1) \leftarrow D[Pr(n)]$$

Finally, a transformed output image $Pr(n)$ is obtained as a function of BOTH the 8-sum $Br(n)$ and $Pr(n-1)$ (as translated by the system's 3-state lookup table, L1—also denoted by $F[\cdot]$ in eqn. 9.4)

$$Pr(n) \leftarrow F[\cdot]\ Br(n)\ \delta(i-k, j-l) \tag{9.6}$$

where the symbols are as previously defined for eqns. 9.3 and 9.4 above.

Figure 9.4 shows the schematic diagram of the feedback concept of an iterated CA rule. Figure 9.4a depicts the transfer functions for the individual steps in the CA algorithm, while figure 9.4b diagrams the complete conventional feedback representation. The (Gibsonian) input (initial) image is denoted by $Ar(n)'$. This very simple analysis of the CA algorithm shows that, in effect, iterating a CA rule is equivalent to the mass-shunting feedback of other models. The significant advantage of the CA model appears to be the avoidance of explicit inter-cellular wiring for feedback. This could be important in IC fabrication. It may be further speculated that mechanisms and techniques used in the conventional—and well-established—digital filter theory may help to clarify certain design and implementational aspects of CAs. Note that, in figure 9.4, the input image to the summing element is $Ar(n)'$. This is the CA thresholded form of $Ar(n)$, and is maintained for only one (initial) cycle.

In essence, long-range propagation (that is, **global**) effects are obtained within CAs by iterating a rule (the equivalent of feedback). Short-range (or **local**) processing can be achieved with only one iteration (one pass) of an appropriate rule. An example of the former is the SOAP film rule, and a process of the latter kind is a CA-based image OUTLINE procedure, based on a CA rule. CA rules are defined by lookup tables (discussed in Chapter 4) and form part of the current VPC command set. These will be discussed later. For the present, it is important to appreciate the equivalence of our CA mechanism and that of Grossberg's neural network (G-M) mechanism—a system that can be justified on the basis of neuroanatomical evidence. This completes the first stage of our vision model development and description.

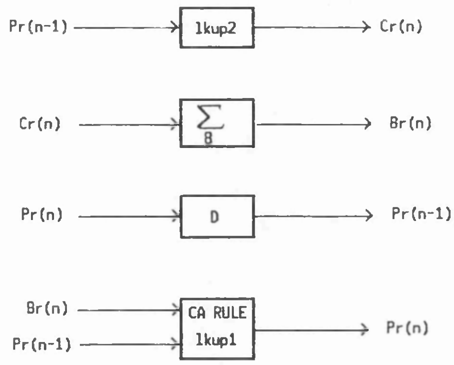


Figure 9.4a Illustrating the main algorithmic steps of the ECM model. These are implemented within the software source code for the project.

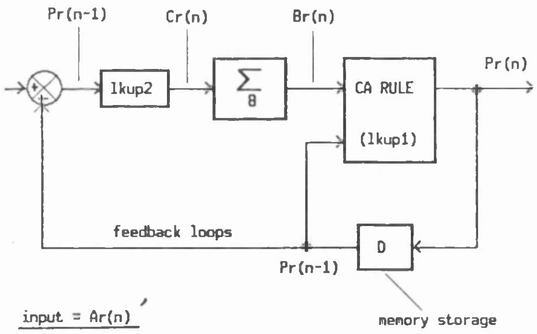


Figure 9.4b A schematic diagram illustrating the writer's interpretation of an iterated CA rule as a conventional feedback mechanism—similar in concept to a digital filter. Refer also to Appendix I.

9.8 Justifying the Current Approach

Given that Prolog cannot efficiently "understand" pixel-based (that is, fine-grained) detail or information, a vision model must produce basic features as discussed earlier. In particular, it requires a system that can deliver a noise-free and cartoon-like (symbolic) image for recognition by Prolog. Prolog should then be able to analyse such an image—which may have been constructed using both actual and illusory edge and region data.

For example, the Kanizsa Square illusion mentioned elsewhere in this thesis delivers **five** image components for analysis by a Prolog interpreter—not just the expected four PACMAN shapes which would be seen by **conventional** machine vision. The end result of this process is a representation of an image that includes **expected** psychological image features—if relevant.

The postulated direct vision phenomenon plays an important role in our proposed vision model—by directly mediating what is hereafter called the Homogeneous Regional Segmentation (HRS) process. As shown in the figures below, our proposed symbolic vision model combines the following processes:

1. An Edge Constraint Map (ECM) module.
2. An Homogeneous Regional Segmentation (HRS) module.
3. An Image Recognition Module (IRM).

Module (1) is the most important subsystem, and is really the core of the entire vision model. The ECM specifies objects of interest, but the definition of what constitutes a distinct "object" is always an open question in any advanced vision model. (If a window, say, is an object, is then a window pane a discrete object? Clearly, the answer will always depend on the current context.)

Module (2) includes a concession to the Gibsonian concept of direct vision, which is believed to be crucial. The HRS functions with the layered diffusion process to distribute and mediate featural qualities of the original image. This is the only mechanism known to the writer which explicitly considers a direct vision philosophy as expounded in this thesis.

Module (3) includes the Prolog database system for the interpretation of the symbolic meaning of images.

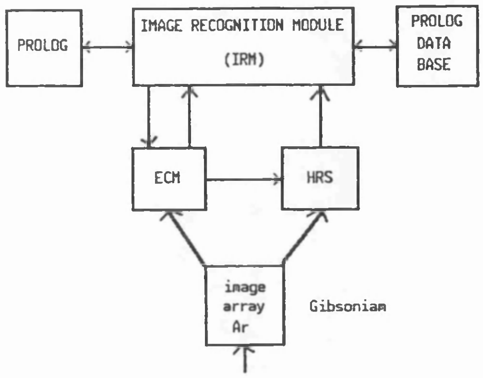


Figure 9.5 Illustrating the general concept of the proposed ECM-HRS-IRM model as developed by the writer for use in the present project.

ECM = Edge Constraint Map.
HRS = Homogeneous Regional Segmentation.
IRM = Image Recognition Module.

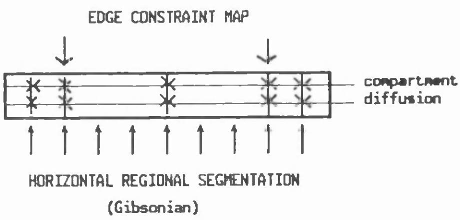


Figure 9.6 This diagram shows the important part of the present system. The above mechanism allows the Edge Constraint Map (ECM) to constrain lateral featural diffusion, as generated by Homogeneous Regional Segmentation (HRS). Unlike the GM system of Grossberg-Mingolla (1987), this model permits mediation and interaction by the original (Gibsonian) image.

Using the "soap film" mechanism, the CA-based model could build a perceived square from only the four corner pixels. Similar shapes can likewise be constructed from their energy maxima points. [2]

The definition of illusory contours in the present work, and in the studies of Grossberg and Mingolla (1985a, 1985b), Grossberg (1987b), and elsewhere in the literature, is the perception of image edges or boundaries where there are **no actual physical image contrast differences**. Since no physical edge contrast exists, the perceived boundaries must be self-generated—that is, they are "illusory." The generation and the interpretation of illusions is considered to be one of the definitive tests of the validity of any vision model which aspires to emulate biological vision.

9.9 The ECM–HRS Process

The following is the step-by-step derivation of the ECM-HRS vision model. An early stage in the ECM module involves applying orientation contrast masks as an obvious approach to finding potential image edges and boundaries. This stage is followed by a more robust evidence-based scheme for finding illusory boundaries. The CA mechanisms are based on the previous soap film iteration rule—which can be applied to either thresholded or multi-level (i.e. 3-state) CA images. All edge evidence is accumulated within a multilevel array [Jr] for subsequent analysis through a voting (i.e. competitive) system. This voting mechanism is neural-like, and is regarded as an important feature of our model.

The ECM produces the Edge Constraint Map—the equivalent of the BCS in the G-M model (q.v.). The next step invokes the HRS (Homogeneous Regional Segmentation) module, which effectively "fills-in" the webs defined by the ECM. The bounded areas are filled-in with either up to 16 colours or shades of grey (a more ambitious model may use more than one thousand subtle shades—but the general principle is the same). This stage results in a cartoon-like (semi-symbolic) rendering of the original scene, which can subsequently be applied to the Prolog-based IRM (Image Recognition Module) for full symbolic interpretation.

[2] An interesting and relevant concept is Attneave's Cat—described in Attneave (1954)—in which recognisable forms can be reconstructed from straight-line segments joining only points of maximum curvature. Is this similar to the well-known demonstration in which lights are attached to the joints of the human frame, thereby highlighting the human form when the figure is in motion? See appendices.

The IRM in this project contains a simple Prolog-based database which attempts to use derived image relationships to come up with a valid scenic description. The IRM initiates segmentation of the cartoon-like image by using a conventional histogramming approach, following which Prolog can produce a full symbolic representation and recognition.

9.9.1 The Prefiltering Stage (1)

Before presentation to the model, the original (Gibsonian) image is prefiltered to remove point noise. This ensures that a relatively clean image is available at the oriented masks. The method used is to high-pass filter the original image, but obviously requires care to avoid filtering-out relevant or significant image highlights.

9.9.2 The Preattentive Vision Stage (2)

Before the orientation masks are applied, the cleaned-up image is scanned in a search for possible preattentive image features, such as right-angled corners and other features which "stand out" from the general scene—see previous notes on Triesman's work in Chapter 6. This information is stored in a preattentive image array for possible later presentation to the IRM. This aspect of the work is an ongoing task and at the time of writing has not been finalised. Our hope is that a preattentive feature map acting within array [Jr] can help in advanced image identification.

9.9.3 The Oriented Edge Masks (3)

A set of oriented masks is used to find the contrast-defined and orientation-specific edge elements. This is accomplished by using oblong masks, divided into left and right fields L_{ijk} and R_{ijk} where (i,j) is the pixel position and k is the mask orientation. The particular values of i,j,k depend on the image resolution of the model. In this case 7x4 masks are used. The output integer value from the masks is given by

$$J_{ijk} = \frac{[U_{ijk} - \alpha V_{ijk}]^{+} + [V_{ijk} - \alpha U_{ijk}]^{+}}{1 + \beta(U_{ijk} + V_{ijk})} \tag{9.7}$$

where

$$U_{ijk} = \sum_{(p,q)\in L_{ijk}} S_{pq} \tag{9.8}$$

$$V_{ijk} = \sum_{(p,q)\in R_{ijk}} S_{pq} \tag{9.9}$$

and the notation $[p]^+ = \max(p, 0)$. In the above equations, α measures the contrast between the two halves of the mask field, and the expression containing β can be used to scale the output. In the present work we set $\beta = 0$.

9.9.4 Oriented Edge-Cell Competition (4)

In order to minimise spurious edge signals, a competition between edge orientations is established via the cellular mechanism of ON-CENTRE, OFF-SURROUND receptive fields. Appendix I discusses a mathematical derivation of such a field, as found in biological vision and many other types of neural circuitry—often realised as a difference of Gaussians (DOG) interaction. However, for the purposes of a simple demonstration such a high degree of modelling accuracy is not needed. At present, the simpler, and conventional, 3x3 receptive field shown immediately below is deemed to be a sufficiently valid representation for the current model's 128x128 pixel resolution.

ON-CENTRE, OFF-SURROUND FIELD

−	−	−
−	⊕	−
−	−	−

The numerical outputs (J_{ijk}) from the oriented edge masks of the previous step are used to activate a parameter space array of oriented weight-cells (w_{ijk}) via the above 3x3 surround field. The central cell w_{ijk} receives a positive activation potential, while the eight surrounding neighbours are inhibited. To achieve a dis-inhibitory activation, a bias potential V_b must be introduced into the model. The dynamics of the cell activations are described in analogue form by the following differential equation

$$\frac{d}{dt} w_{ijk} = -w_{ijk} + V_b + f(J_{ijk}) - w_{ijk} \sum_{(p,q)} f(J_{pqk})\, D_{pqij} \tag{9.10}$$

where D_{pqij} represents inhibitory interaction strength between neighbouring image-cell positions (p,q) and (i,j), and, as above, $f(J_{ijk})$ is the output from the orientation masks.

Suppose, for simplicity, that the following approximation is valid

$$f(J_{ijk}) = \gamma J_{ijk}$$

where γ is a positive constant determined by experiment. Also let w_{ijk} reach equilibrium instantly, so that suitable values can be obtained for the input to the weighted cell array. Setting $\frac{d}{dt} w_{ijk} = 0$ for the steady-state solution gives

$$w_{ijk} = \frac{V_b + \gamma J_{ijk}}{1 + \gamma \sum_{(p,q)} J_{pqk}\, D_{pqij}} \tag{9.11}$$

Thus, by an appropriate choice of parameters V_b and γ, as substitutions in the 3x3 field, suitable values for the activation weight-cell array w_{ijk} can be obtained.

9.9.5 Weight-Cell Thresholding (5)

The oriented weight-cells w_{ijk} are now thresholded to yield an output suited to CA manipulation by setting the auxiliary image array $Br[i,j]$ according to the function

$$Br[i,j] = \begin{cases} 2 & \text{if } w_{ijk} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.12}$$

The above threshold function is applied for each orientation (k) in turn, the array elements being OR-ed together to produce the required oriented-edge map array. The mathematics are readily translated into appropriate VPC functions.

Included within the competition steps is the accumulated edge-evidence within array [Jr]. This evidence can be assessed in a number of ways, including Bayesian inference, the Dempster-Shafer approach, and so on. However, the method actually used by the writer was a **vertical** summation of the "votes" through each layer of parameter space array [Jr]. For instance, the coincidence of a pixel in more than some set threshold (ten, say) of the sixteen levels of this array could be regarded as sufficient evidence of the presence of an edge element. But there are other equally suitable voting mechanisms that can be applied. The following very simple VPC code fragment illustrates the edge-evidence mechanism that creates the defined ECM edge array:

```
/**********************************************************/
 void sev_0 (void)
/* Count edge-evidence votes (EEV) in 3D array [Jr]:
   then threshold 2D edge-image array [ECM] accordingly   */
{
   BYTE  i,j,k;
   WORD  Vsum;                          /* evidence votes */

      for (i = 1;  i < max-1;  i++)     /* max preset 128 */
      for (j = 1;  j < max-1;  j++)
         {
         Vsum = 0;                         /* reset count */
         for (k = 0;  k < 16;     k++)     /* 16 levels   */
            {
            if (Jr[i][j][k]) != 0) Vsum = Vsum + 1;
            /* now threshold Vsum to get robust edge-map */
            if (Vsum >= ethres) ECM[i][j] = 1;
            else                ECM[i][j] = 0;
            }
         }
}
/**********************************************************/
```

The important point is that the existence of accumulator array [Jr] allows **separate** processes to pool their evidence for subsequent analysis and interpretation. This is quite unlike most conventional (e.g. filter-based) mechanisms whose outputs immediately become the input data to succeeding processing stages. The pooling of evidence also allows psychologically-derived image information—including illusory data, if relevant—to be incorporated into the ECM.

The result of this intercellular competition is that a sufficiently **robust** edge or boundary can emerge from a Gibsonian grey-level, or colour, input image. Unlike the Grossberg-Mingolla model, there is no requirement within the present model for line-end or termination effects—"sealing-off"—to prevent leakage or featural flow-out from the boundary. This need is easily satisfied from within the CA model, and the soap film rule. It will be recalled from Chapter 4 that the soap film (search film, or search field—SF) can directly provide enclosure of high-energy points. By the use of our custom CA rules, properties such as closure or completion of broken or occluded boundaries is also possible. However, one problem with this approach is that the relevant CA rules must somehow be **automatically** selected to suit a given application at an appropriate stage of the vision model's processing. This topic requires further research.

9.9.6 The Combined ECM + HRS Stage (6)

The final stage of the model operates to fill-in featural qualities, such as average grey-level or colour, within the webs created by the Edge Constraint Map (ECM) module just described. There are several approaches to this problem, all based on the recent neuroanatomical evidence of actual neural network circuits.

One method is for the edge colours immediately adjacent to the ECM to propagate into the constraint space, the intensity either remaining constant or decaying with penetration according to a mathematical law. The propagation can continue until either the value decays below a set threshold, or the propagation wave hits another edge or boundary. Thus the featural qualities are constrained within the image edge-web matrix defined by the ECM—a process like "painting" the interior.

An important by-product of the above edge-diffusion process is the desired "discounting of the illuminant" (i.e. the colour perceptual stability) advantage which has been observed in natural vision. This effect can happen because the propagation of colour (or other featural quality) occurs **after** mediation with the original (Gibsonian) image's high-resolution detail.

A bonus is the ability of this featural filling-in mechanism to demonstrate—and even explain—the familiar Craik-O'Brien-Cornsweet (COC) illusion of brightness perception. It is the ability of our model to explain these hitherto enigmatic effects that encourages belief in the model's fidelity and future capabilities.

9.10 The Prolog-Based IRM

This module attempts to recognise objects or scenes which have been rendered in semi-symbolic form by the ECM-HRS process. It may also use any preattentive information derived directly from the image, and can aid the earlier processes by producing expectancy. The IRM module is otherwise a conventional application of Prolog—the logic programming language favoured by many AI researchers today.

As mentioned before in this thesis, a main reason for using Prolog is that it is not possible at the present state of research to understand how biological neural network circuits are able carry out psychological processes and functions—and to thereafter explain how they arrived at their decisions. The specific Prolog code developed for the IRM by the writer contains several useful features which may be of general interest. These mechanisms include the following:

1. A database mechanism of **concentric contexts** which can concentrate the search for meaningful object descriptions and relationships. Contexts are activated by any event or clue which may be relevant in the current situation. This is a demon-like feature.

2. A database **logging** mechanism which keeps track of when a Prolog rule or procedure has succeeded. This information can subsequently participate in a self-documenting (and on-line) report on the system's progress in object or scene recognition. The method also allows a user to question the model's reasoning and decisions at a later time.

3. The use of **fuzzy logic** methods in the formulation, and in the subsequent characterisation, of image regions and objects. This includes techniques of feature tolerancing.

4. A database of **image-tree** relationships which can assist in the identification of images by the use of a hierarchical, image-component, descriptive approach.

These four mechanisms form the basis of the IRM, and are briefly discussed in the following sections.

9.10.1 Concentric Contexts

This method broadly resembles a "demon" in the AI literature, which is a system that awaits its opportunity to latch-on to an occurrence of a prespecified trigger condition. As an example, the presence and instantiation of "sky" (by some Prolog rule mechanism) can itself initiate a search for other contexts—for instance "bird." Once a demon has been activated, the concentric context can rapidly narrow the search field. Concentric contexts are database-defined, and are activated and de-activated as required within the Prolog IRM module.

By the use of the usual Prolog database methods, many contexts can be active simultaneously. The idea of concentric context is then likened to a system of successive approximations, where the Prolog system is able to home-in on the target as detail becomes available (more and more contexts become **set** or instantiated). Alternatively, it can be considered as a rough initial search, which is narrowed down as more evidence becomes available. Figure 9.7 below shows a very simple example. Because "sky" has been activated, both "bird" and "plane" are regarded as being concentric within "sky" since normally they would be expected to be associated with objects seen in the sky. Notice, however, both are equally probable at this stage. If "plane" is subsequently reinforced, then "bird" is deactivated by **retracting** it from the Prolog context database.

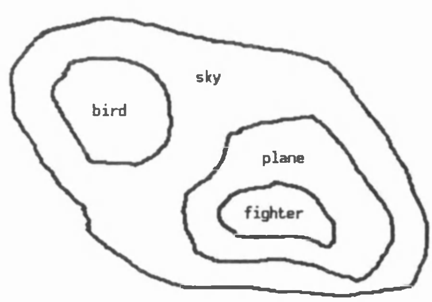


Figure 9.7 Illustrating the idea of concentric contexts in Prolog.

9.10.2 Prolog Rule-Logging

The ability to self-explain is central to the concept of intelligent knowledge-based systems (IKBS), and has been much discussed in the literature. In the present proposal, Prolog rules or clauses that succeed are logged. This information can then be used to support an on-line compilation, or an off-line systems progress report. The idea of rule-logging has been defined here as a Prolog database function, and gets around one of the criticisms often levelled at artificial neural networks—namely, that they cannot explain their decisions.

This idea is really just like an extension to Prolog's **trace** function, and there is nothing particularly problematic about it. The novelty stems from actually using and combining standard functions in an interesting and informative way.

9.10.3 Prolog and Fuzzy Logic

The use of fuzzy logic (Yager et al., 1987) is characterised by the use of vague and imprecise descriptive terms, such as

- small
- large
- quite small
- rather large

which are found in the classical fuzzy logic theory. This can be very useful in machine vision systems—as suggested in Batchelor (1991). The novelty in the present project is that we recognise that we are dealing with a fuzzy system—a point which is not often emphasised in conventional Prolog textbooks. This relationship between fuzzy logic and Prolog can be very close, as exemplified by the following simple Prolog code fragment:

```
very_big (X) :-
   area (X, Area),
   tolerate (Area, 20, 100),
   sh_f (X, SF),
   SF  >  20.
```

The above rule means that some image region (X) can be described as being **very big** if its (approximated) area is 100 scale units AND its shape factor is greater than 20 defined units. This further implies the use of **relative** scaling factors in the model, which requires that areas, for example, need to be related to the optical frame size (the image array dimension). Unfortunately, this ties the present model to conventional image processing concepts, but cannot be avoided at this stage of natural visual understanding. There must also be a mechanism for defining tolerances within Prolog, as above.

9.10.4 Image-Tree Representation

Appendix J shows how a simple hierarchical image-tree system can be developed from a set of primitive 2D graphical elements. Although essentially a graph-based concept, it is not strictly necessary for image graphs to actually be displayed on the computer terminal by the model. The essential image inter-relationships are coded into a VPC interconnection matrix, which itself can be realised as an image-tree using the Prolog functor procedure. It is the analysis of the tree interconnect matrix that results in the symbolic interpretation and representation of an image. This is really a **syntactic** process. Indeed, the methods of syntactic pattern recognition are applicable to the image-tree model developed here.

Note that the use of an image inter-relationship tree makes the model relatively immune from the "image frame" problem, discussed earlier in Chapter 2. That is, by the use of a symbolic representation, an image can be recognised independently of any position and orientation variations—within reasonable limits.

Noise immunity is a consequence of this symbolic representation which ignores pixel-based information. This further reinforces the ideas of data-reduction, image decoupling, and some other characteristics of biological vision, discussed earlier in the thesis.

Although the IRM may appear to be the most important module within the present system, this is not so. The writer considers that the ECM-HRS is our main contribution to a technological interpretation of natural vision. Our use of Prolog within the present research is otherwise conventional, and all of the model's CA rule definitions and features can be extended at any time as ideas are developed. Nothing is set in concrete.

9.11 Chapter Summary

This chapter has discussed the development of a CA computer model of human vision which attempts to produce a pure symbolic representation from natural images. It does this by constructing a semi-symbolic (cartoon-like) intermediate form, which is in keeping with current approaches to AI—the principle of simplified and (or) symbolic images is supported by the writings of a number of eminent AI researchers, including Zadeh, Arbib, Kohonen, and Minsky.

A distinction was made between local and global processes within CA models. Generally, local (i.e. nearest-neighbour) CA processes can be non-iterative, while global features can only be obtained as a result of iterating an appropriate rule, with storage of intermediate results. This was linked with the feedback model of a simple RC digital filter. It was also mentioned that iterated CA rules are equivalent, at least in principle, to models of artificial neural networks (ANNs) developed by Grossberg (1987) and his associates. This effectively links our CA model to cogent explanations of neuroanatomical vision—which appears to be based on only a relatively small set of common functions.

The relevant sections of this chapter derived the writer's ECM-HRS model in discrete stages, and then discussed in outline the needs of the Prolog-based image recognition module (IRM). The latter contains some of the writer's personal ideas for object identification based on Prolog developments of the (now-standard) IKBS approach. The important modules discussed above are:

1. The ECM – a robust means of detecting both real and illusory edges and boundaries within natural and synthetic images, including the well-known Craik-O'Brien-Cornsweet (COC) illusory edge profile.

2. The HRS – a system of filling-in the ECM boundaries to render images "cartoon-like" (i.e. semi-symbolic).

3. The IRM – a module which uses a Prolog program to recognise objects via a full symbolic image-tree matching process. This is similar to syntactic pattern recognition.

The ECM is clearly the most important single aspect of our vision model, while the edge-evidence accumulator array [Jr] is a robust mechanism within the ECM.

Nevertheless, all concepts and software defining our vision model can be developed further, as a consequence of the essentially modular approach adopted.

The following chapter presents the results of applying the CA-based image processing mechanisms of the model to selected test images. It also shows the application of the complete vision paradigm to a real external scene.

CHAPTER 10

DEMONSTRATOR RESULTS AND DISCUSSION

A computational vision model embodying the principles described in Chapter 9 and elsewhere in this thesis is implemented in the C, and Prolog programming languages, in accordance with the details outlined in Chapter 8. This chapter presents the results of applying the model to a selected range of test images. The experiments also included the objective of discovering how our human vision model might be able to utilise aspects of the enigmatic concept of direct vision.

10.1 The Software Modules

Figure 10.1 diagrams the method of setting up software modules for the experiments. The writer has decided on the acronym "PROVIS" for the software part of the project—derived from PROlog VISion. Unfortunately, this title understates the powerful role played by cellular automata, and does not give any indication of the problems implicit in modelling the philosophical aspects of vision—especially those of direct vision.

As mentioned elsewhere in the thesis, a desirable feature of any model of human vision is the ability, wherever possible, to function autonomously—that is, there should be little direct intervention by the experimenter. This goal is rarely achieved in practice, and researchers often have to fine-tune ("tweak") their code in order that useful results can be logged. It is important that a model of vision that includes many distinct modules and processes is able to function as a cohesive whole. This, again, is rarely achieved in practice, the usual situation being a batch of routines each carefully tuned to demonstrate a specific vision concept.

The following is the list of custom programs (produced in both source and executable forms) which have been coded by the writer for use in the project demonstrations. No commercial or third party code is used anywhere in the project, other than the framegrabber driver software.

1. VIDEOC.EXE — grabs 64x64 or 128x128 colour still frames from a CCD video camera, converts these to a block-pixel format, and then stores the pictures as user-named test images on disk.

2. CAMDIS.EXE — allows synthetic images to be drawn, and custom CA image processing rules to be defined by the user. Stores the pictures as user-named test images on disk.

3. CAMVIS.EXE — demonstrates machine vision rules and procedures via the VPC command set. The latter includes both conventional C code, and CA lookup table versions of simplified common image processing algorithms. The range of algorithms is intended to work with a model-defined 128x128 image resolution.

4. PROVIS.EXE — implements a hybrid (Prolog plus VPC) model of intelligent vision, incorporating the writer's ECM-HRS-IRM paradigm as developed in Chapter 9 of this thesis. This is the main project demonstrator which will be used to supply the test results. The code includes CA routines for soap film formation, mechanisms for edge-detection, and direct vision mediation.

In general, only items 3 and 4 above are relevant to the results reported in this chapter, the first two utilities having been used earlier by the writer to prepare a range of test images, and to evaluate CA rules and algorithms.

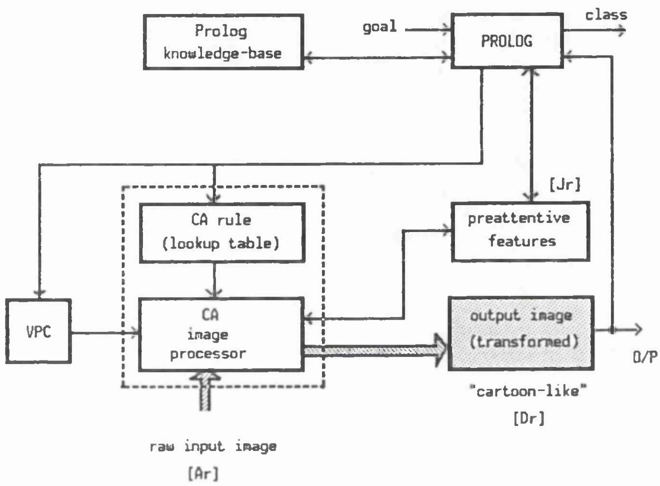


Figure 10.1 Illustrating the general concept of the PROVIS demonstrator.

A raw image is presented to the input buffer [Ar] and a goal is issued to start the Prolog IRM. The CA processor starts by looking for a set of preattentive features, the analysis of which is passed to Prolog. The Prolog IRM decides on the lookup table rule(s) and parameter(s) to be issued to the CA image processor. On completion of the sequence of image processes—via the ECM-HRS with direct mediation—the final grouped image emerges as a cartoon-like description. This is passed to the IRM for analysis and recognition. The IRM can call for further processing evidence if necessary, since the original (Gibsonian) raw image is always preserved.

Note how the two-way link shown between Prolog and the preattentive features array could be used later to produce EXPECTANCY features in an image.

10.2 Defining Test Images

A range of suitable test images is needed for the evaluation of any model of computer vision—see e.g. IEE (1982:31). Rather than use just arbitrary images, the selected images used here are intended to illustrate the specific capabilities of the PROVIS vision model, as follows:

- Basic CA processes: Demonstrate CA models and rules suited to conventional image processing. Demonstrate "soap films" as a method of realising Gestalt preferred image groupings.
- Natural and synthetic Illusory images: Demonstrate the handling of common types of illusions, including the colour perceptual stability ("discounting the illuminant") phenomenon, and the simple version of the COC brightness illusion.
- Direct vision: The processing of images with mediation via the original image maintained in array [Ar]. This will make use of semi-symbolic image region (object) masks.
- Simple image understanding: Demonstrate iconic to symbolic conversion—for example image centroids and connectivity to image-tree graph arcs and nodes. Demonstrate Prolog in use.

The following test/demonstration images are used in this project:

1. DO4.IMG — synthetic four corner dots image.
2. BRK.IMG — synthetic broken-edge test image.
3. KAN.IMG — synthetic Kanizsa-style Illusory Square image.
4. COC.IMG — synthetic Craik-O'Brien-Cornsweet illusory image.
5. TEX.IMG — synthetic 2-region textured images.
6. CAT.IMG — synthetic Computer-Aided Tomographic test scan.
7. VIL.IMG — natural external village scene.
8. EA1.IMG — natural external street scene.

10.3 The User Interface

Figure 10.2a shows the normal on-screen display, which consists of two frames. The left frame normally always shows the original input image (in keeping with the Gibsonian direct concept) while the right frame shows the on-going vision processing. The whole-screen display often includes the 3-state CA lookup table (LUT), represented at the bottom left corner as a three-colour matrix array as follows:

- BLACK — cell in CA state [0] (a defined [0]-value cell)
- LCYAN — cell in CA state [1] (a defined [1]-value cell)
- YELLOW — cell in CA state [2] (a defined [9]-value cell)

Figure 10.2b shows an enlarged screendump of the right frame that will normally be used to present the visual results of this chapter. The computer displays are usually represented in the IBM VGA 640x480 standard range of sixteen colours. Grey-scale output can be stipulated by the user, if this would be more meaningful. Optional use of the IBM 320x200 screen mode allows a maximum of 256 colours to be displayed simultaneously. This could be helpful when dealing with complex CA displays. The video images in this section were captured directly from the screen and printed using PINCH AND PUNCH—a proprietary utility designed for this purpose. All screen output is, of necessity, reproduced in this thesis as either black and white, or as a dithered grey-scale representation of the image PCX format.

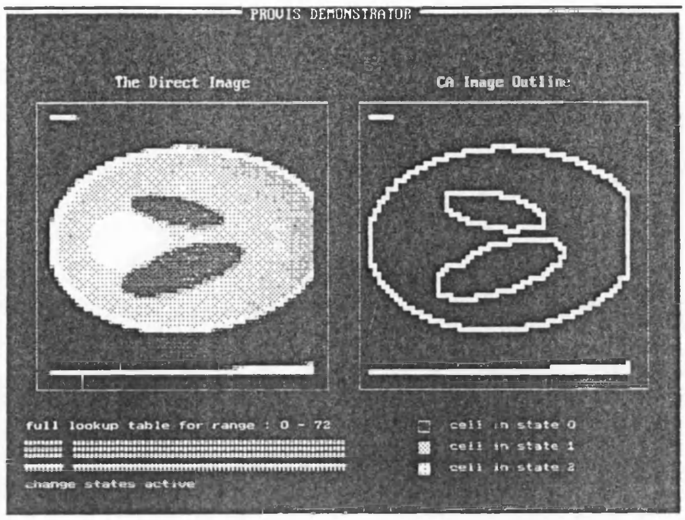


Figure 10.2a The PROVIS demonstrator screen can provide two frames for most processing. The left frame normally shows the original (Gibsonian) direct image, while the right side frame shows the results of the on-going vision processing.

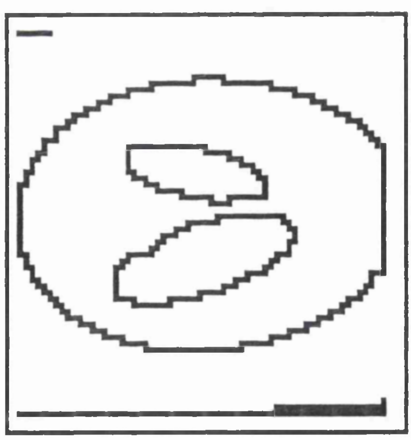


Figure 10.2b The right frame will normally be used for presenting the visual data and the experimental results of this Chapter.

10.4 Basic CA Processes

This section demonstrates three basic CA processes. The scope of CA rules and processes is essentially unlimited, but practicalities, and the limitations on project time, dictate that only some fundamental principles can be demonstrated here. However, it is hoped that this will be sufficient to fortify the writer's claim that virtually any conventional image processing algorithm can be programmed as a CA rule and associated lookup table. In addition, there seem to be many vision processes that can be **uniquely** defined by CA lookup table rules.

Before considering CA demonstrations it is necessary to review the mechanism of lookup tables and rules, first introduced in Chapter 4.

10.4.1 CA Lookup Tables and Rules

For the purposes of this thesis, there are TWO kinds of CA lookup table (LUT) rules: **totalistic** and **semitotalistic** types.

As mentioned in Chapter 4, totalistic rules are defined only on the sum of a cell's neighbours: the current cell state is switched to either a [1] or a [0] depending only on the 9-sum, which therefore includes the cell's own state. There are no intermediate states. An obvious example of a totalistic rule is the MAJORITY or VOTE rule, in which a cell becomes a [1] if the local 9-sum reaches 5 or more. The cell is set to the [1]-state irrespective of its current or previous states. As was mentioned in Chapter 4, it is possible to represent the rule as a convenient **totalistic code**—a positive integer obtained by reversing the binary bit-pattern of the lookup table's defined activation states.

Semitotalistic rules are more complicated because the cell is switched depending on BOTH the 8-sum of its neighbours AND the cell's previous history. In this thesis, semitotalistic rules are normally 3-state rules in which the intermediate state is the [1]-state. This is used to good effect in the SOAP film rule. For example "solid" objects can be characterised by the [2]-state cells, while the soap film is represented as the [1]-state cells. This enables soap films to be manipulated around and within "solid" objects, [1]-state cells subsequently being processed to yield [2]-state features, following stabilisation. The [0]-state normally represents "black" empty image background.

The functionality of 3-state CAs explains the need for a lookup table having 3x73 entries—mentioned in Chapter 4. If the totalistic sum of the [1]-state neighbours attains the numerical value of 8, then a further level of uniquely represented CA states requires a maximum numerical representation of 8x9 = 72 for any occurring [2]-states. Thus the lookup table index can range from 0 through 72 (i.e. a total of 73 defined LUT entries) for a tri-state CA image processor.

In addition, multi-state CAs can be arranged on levels, with 3-state CAs defined on a **Parameter Space** (McCafferty, 1990). This idea enables each array level to function independently, the final output being some amalgam of the CA hierarchy. This is an extension of array [Jr], discussed in the previous chapter. Further detailed research is needed to develop the concept.

The following abbreviated scheme will be used where required for showing the CA LUT rules. Only the relevant (i.e. changed) column entries above the 8-sum maximum are shown, the assumption being that the remainder of the lookup table retains its initialised rule-state entries, mentioned in Chapter 8. The third form shows an abbreviated re-setting of all entries in a LUT rule—for example, re-setting all [1]-state cells to [0]-state (except in the first column).

<u>Abbreviated LUT Form—Totalistic Rule</u>

0	1	2	3	4	5	6	7	8
0	0	0	0	0	1	1	1	1

<u>Abbreviated LUT Form—Semitotalistic Rule</u>

0	1	2	3	4	5	6	7	8	72
0	0	0	0	0	0	0	0	0	2
0	0	0	0	1	1	1	1	1	2
2	2	2	2	2	2	2	2	2	2

<u>Abbreviated Procedure—Reset LUT Definition</u>

0	1 $\rightarrow$ 72
0	0 $\cdots$ 0
0	0 $\cdots$ 0
0	2 $\cdots$ 2

CA rules are either single-pass or iterated, as required by the CA functionality. As deduced in Chapter 9, iterated rules are feedback systems that propagate LOCAL features to produce GLOBAL effects. In Chapter 9, an iterated CA rule was compared to the theory of digital filters. In any event, the CA iteration has to be terminated when a desired image effect, or CA status, has been achieved. This stopping condition is here taken to be a measure of the system "energy" —as defined on the rate of change of [1]-state cellular activity. When this reaches a minimum (or zero) the CA is considered to have entered its termination or halt state c_h as mentioned in Chapter 4. At this point, new or additional predefined CA vision processes can become activated, or else the PROVIS model may enter another phase, including shutdown.

10.4.2 CA LUT Rule—SOAP

The first demonstration of a CA process is the SOAP rule. Figure 10.3a shows four corner dots forming a square. This is what most people would describe as a "square"—despite the fact that there are only four dots present in the image! In order to be able to "see" a square, the system must create illusory edges or boundaries. This will be discussed in the next section, but for the present our aim is to demonstrate the soap film rule (search film, or search field—SF) in operation. Figure 10.3b–d shows a collapsing SF at different stages. It is seen that the SF disappears into the centre, leaving only a soap film on each of the four corner dots. Figure 10.3e–h is similar, except that a different rule threshold setting (τ) of the soap film (as discussed in Chapter 4) is used—that is, one which results in global self-organisation.

This time the SF is seen to stabilise on an illusory square. Since the small icons in these figures depict "soap" (cells in the [1]-state) it is possible to perform further image processing on the "square." For instance, the square can be made "solid" by a simple CA rule that resets all [1]-state cells to the [2]-state. Further processing using very simple CA rules can yield an outlined square—for example, by taking out the centre of the solid square.

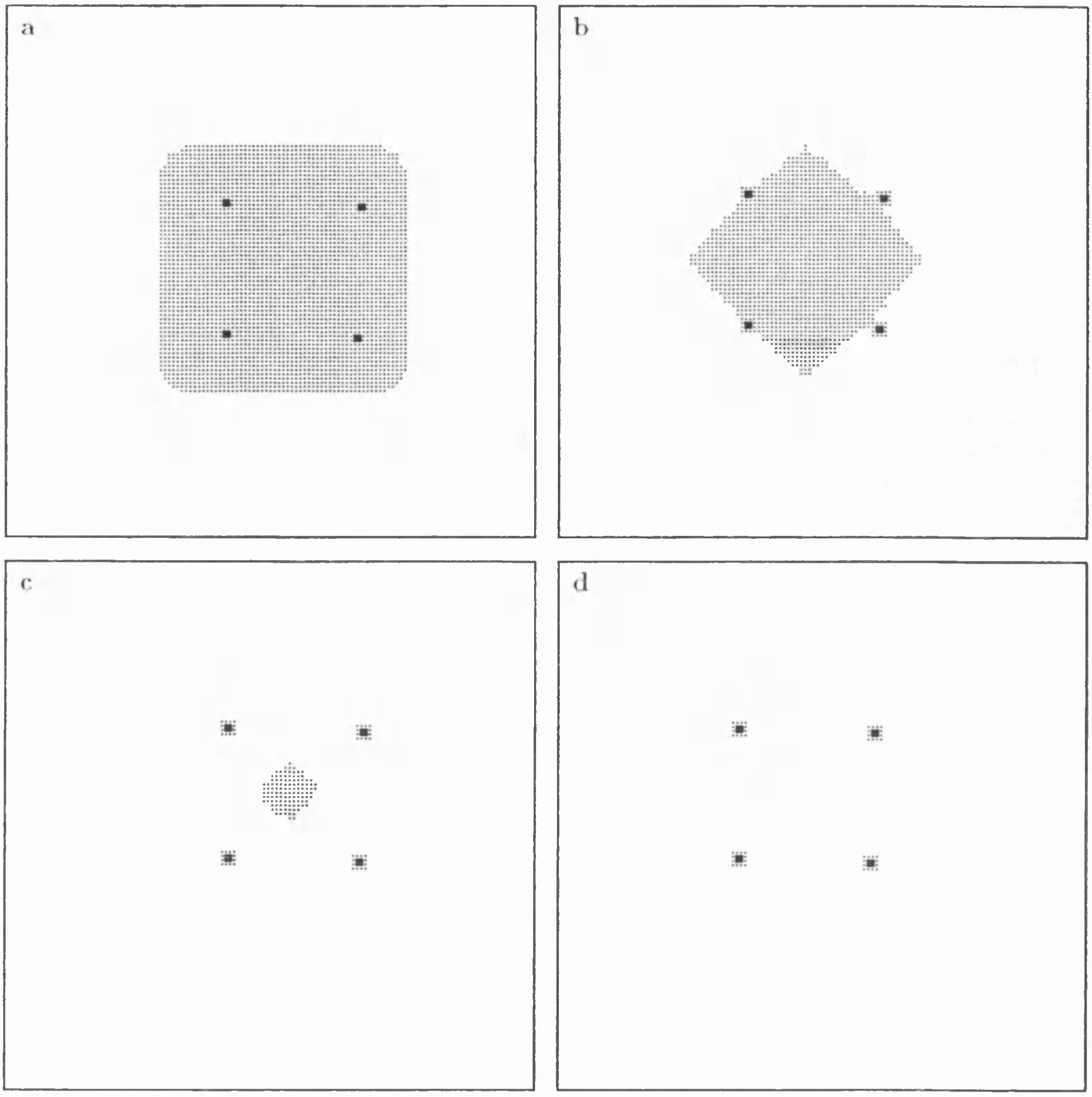


Figure 10.3 Test results for the four dots image.

(a) The initial image of four corner dots.

(b) The collapsing SOAP FILM (SF).

(c) The SF continues to collapse inwards.

(d) The final image as represented by a CA state field. The result is SOAP adhering to the four corner dots—but no illusion of a square is obtained using these parameters.

OPERATIVE CA LUT RULE — SOAP(0)

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1	1
2	2	2	2	2	2	2	2	2

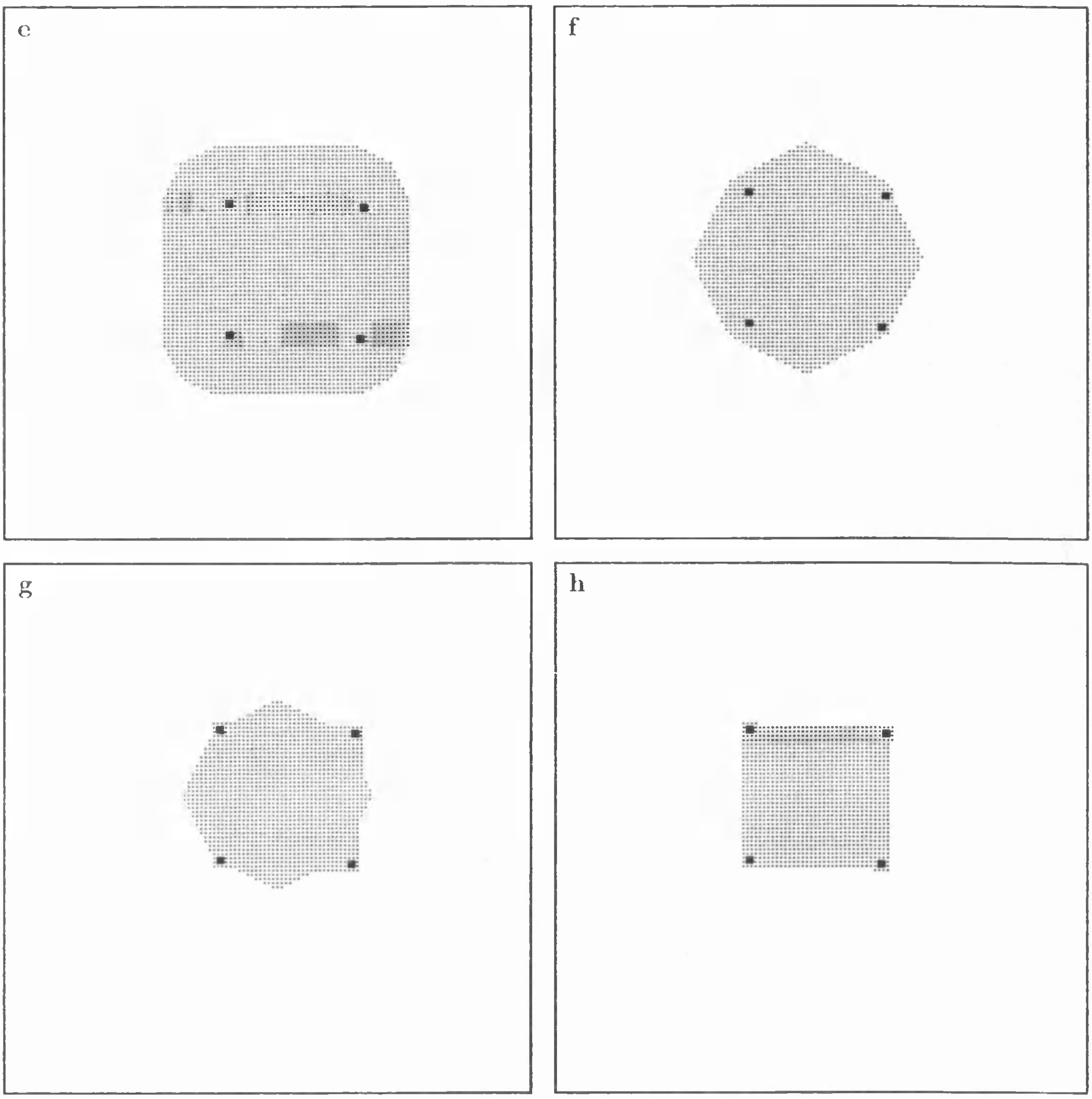


Figure 10.3 (contd.) Test results for the four dots image.

(e) The initial image of four corner dots.

(f) The collapsing SOAP FILM (SF).

(g) The SF continues to collapse inwards.

(h) The final image as represented by a CA state field. The result is an illusory square which can be further processed, if required.

OPERATIVE CA LUT RULE — SOAP(1)

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1
2	2	2	2	2	2	2	2	2

A similar kind of SOAP demonstration more clearly shows how global CA with self-organisation can result from purely local processes. The following demonstration produces results which are almost identical to those reported in Grossberg (1987b) and McCafferty (1990). This demonstration supports our contention that a suitable CA processor, plus a range supporting image processing rules, is equivalent (and possibly even superior) to the much more complex implementations of artificial neural networks (ANNs).

In figure 10.4, a sequence of three CA rules is used to:

1. Establish a global SOAP film, or search field, (SF).

2. Produce a linking SOAP web.

3. Convert all remaining [1]-state SOAP to solid [2]-state objects.

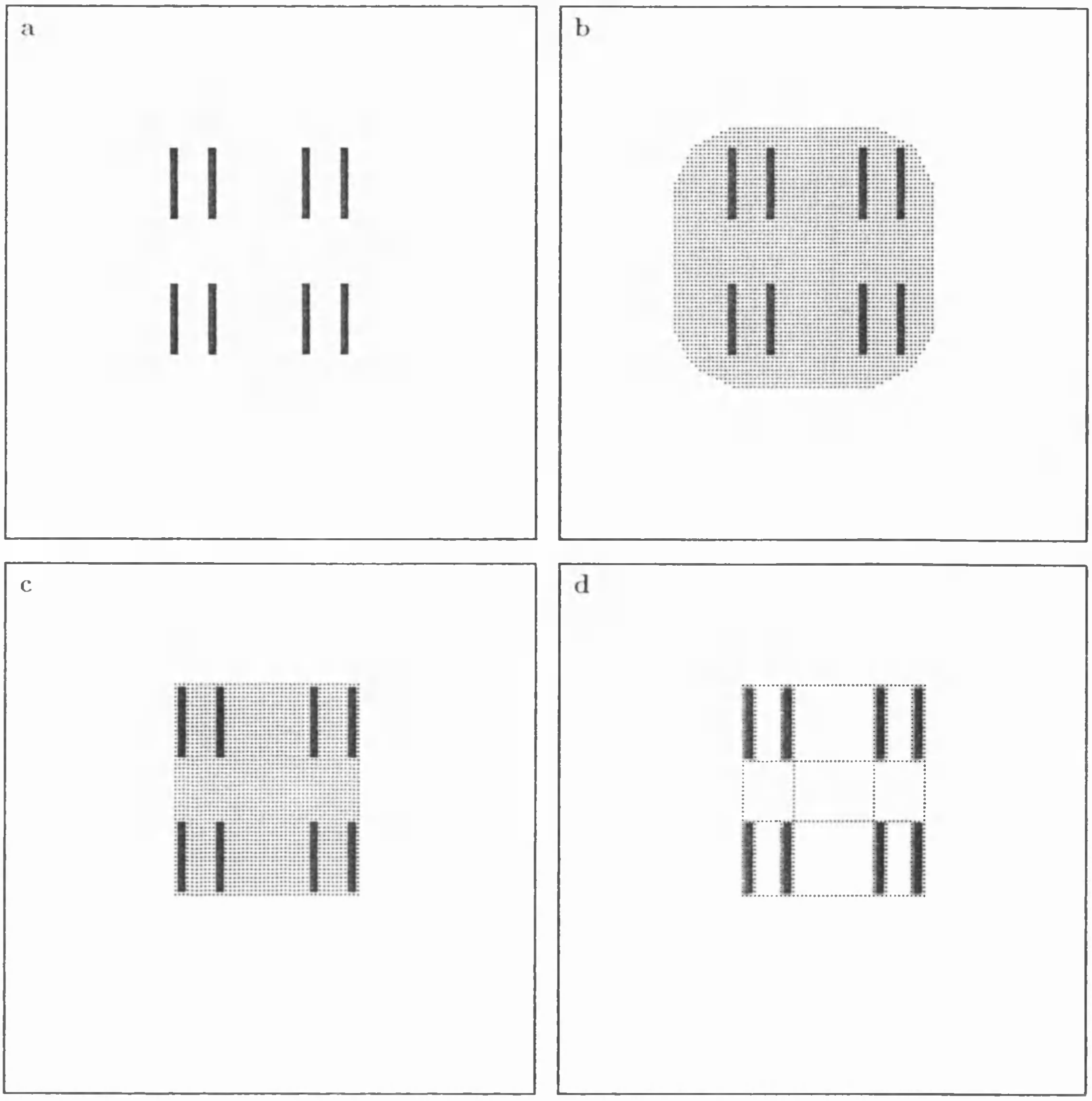


Figure 10.4 Test results for global features and self-organisation.

(a) The initial image of vertical bars.
(b) The collapsing SOAP FILM (SF).
(c) The SF deposits SOAP on the surfaces, and links adjacent bars.
(d) The final image of linked "solid" bars.

OPERATIVE CA LUT RULES — SOAP(0) + SOAP(1) + 1–2

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	1	1
2	2	2	2	2	2	2	2	2

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1
2	2	2	2	2	2	2	2	2

0	→	72
0	···	0
2	···	2
2	···	2

10.4.3 CA LUT Rule—EDGE

The second CA demonstration is that of edge detection on the CAT.IMG image, which is a standardised Computer-Aided Tomographic (CAT) test scan.

This is accomplished by setting all [0]-state and [1]-state entries in the CA lookup table to [0] in the range [0–71]. The final entry at table index [72] is set to [2]. Then a threshold value for CA image mapping is set. Finally, a single iteration of the rule yields the outlined regions shown in figure 10.5a and 10.5b. Notice how, with appropriate τ thresholds, very clean outlines can be obtained.

We can compare these results with conventional Sobel and Roberts edge detectors on the same CAT image, as shown in figures 10.5c and 10.5d respectively. It is seen that the output in both cases is a noisy outline and image. Of course, the CA threshold method has effectively converted the CAT colour (or equivalent grey-scale) image into binary form—which makes the edge detector's task easier. Nonetheless, the results demonstrate the convenience and elegance of the CA processor rule-based mechanism: a preattentively-derived parameter is used by the Prolog-based IRM to **automatically** determine and set the threshold by a LUT indexing mechanism.

Notice also that the above demonstrates a single-iteration rule: only a **single pass** over the entire image data is required in order to establish the bordered outline in this example image.

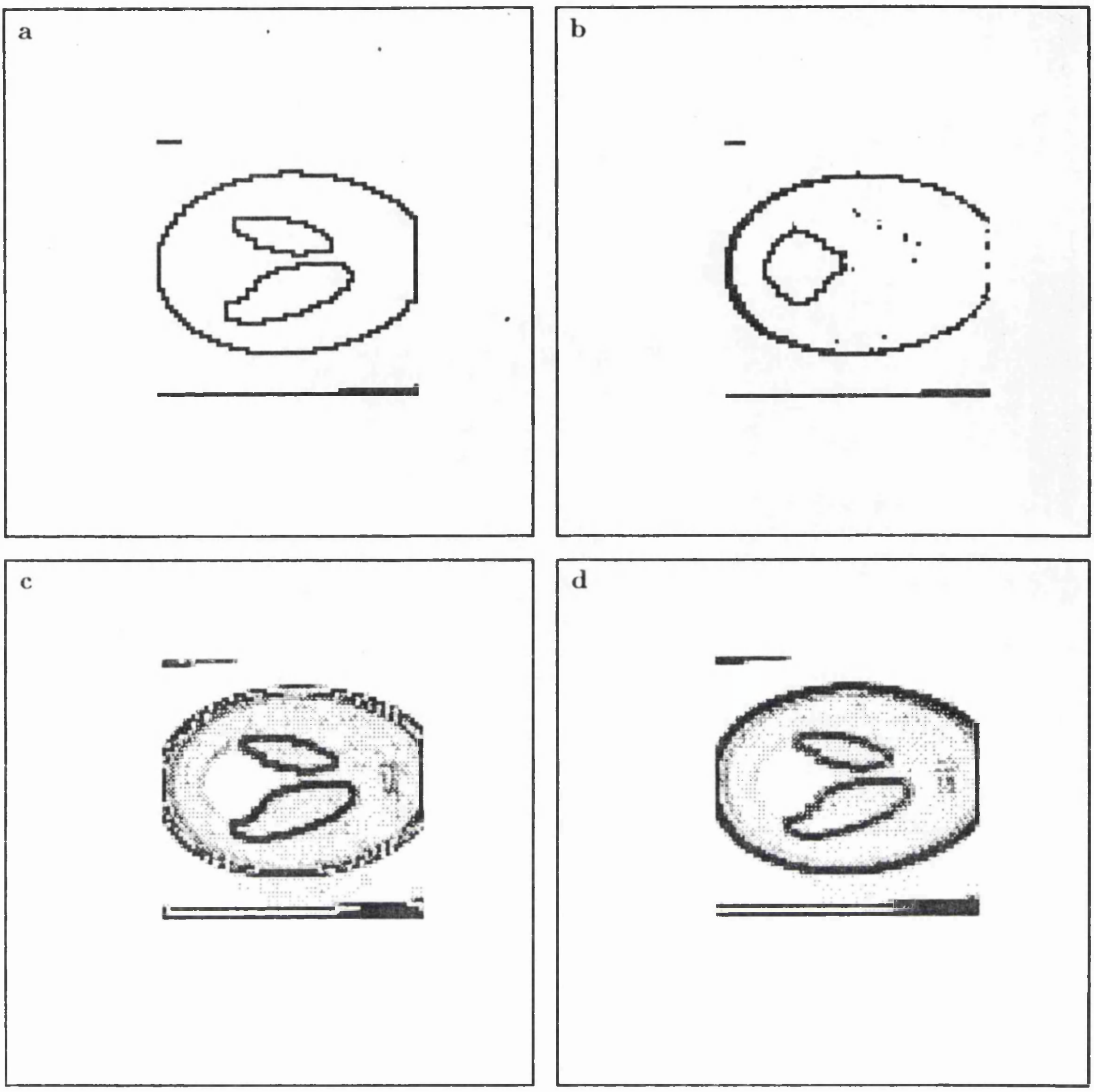


Figure 10.5 Test results for edge detectors.

(a) CA edge detector on CAT image—thresholded at [8].

(b) CA edge detector on CAT image—thresholded at [12].

(c) The Sobel edge detector on grey-level CAT image.

(d) The Roberts edge detector on grey-level CAT image.

10.4.4 CA LUT Rule—COMPLETE

The third demonstration of basic CA image processing is an iterated rule for completing boundaries across broken gaps, noisy edges, etc. This is a useful technical process in itself, but it also demonstrates a fundamental characteristic of natural vision.

The concept is to use SOAP to fill-in adjacent to gaps, the [1]-state soap cells subsequently being converted to a [2]-state solid outline. This is an iterated rule, which, according to Chapter 9, is a feedback system in that both local effects and global propagation interact to complete the boundary. This CA rule also validates the equivalence of Grossberg's neural circuit (G-M) model and CA processing as proposed in this thesis. The writer considers that the concept of rule-equivalence is crucial to the future understanding and development of our CA-based models of biological neural network circuitry.

Figure 10.6 shows boundary completion. The SOAP rule is used in the first instance to fill the entire image space with a soap film. Then, a rule causes soap to cling to the solid boundary [2]-state elements. Thirdly, the CA COMPLETE rule is invoked to complete the boundary by converting appropriate [1]-state cells into "solid" [2]-state cells. Finally, excess [1]-states (soap) are removed. Thus, in this example, three different CA rules require to be applied in succession. There are in fact several different rule sets that can achieve approximately the same net effect—so any specific CA solution is not necessarily **unique**.

The importance of the soap film adhering to the solid object boundary must be emphasised. This single-layer soap film enables one to CONTROL and RESTRAIN CA image processing by allowing the CA rules to be very specific about the LOCAL transformations that are needed. For example, the lookup table entry (rule-index) to complete an imaged boundary gap of **exactly** one pixel in width is different (due to the adhering [1]-state soap) from that needed for dealing with a "free" line-end. See below for more detail.

Observe the simplicity of the COMPLETE rule. This illustrates the power of the lookup table rule method, and the fact that complex global effects can result from simple local rules. This also reveals a problem with the "open-endedness" of CA rule development—the variety of potential rule types, and the time and effort expended in discovering and developing them are, in effect, boundless. The CA

COMPLETE demonstration is not claimed to be exhaustive. Our intention here is to illustrate the basic philosophy and mechanism of CAs: processes that do not extract specific numerical data from images.

10.4.5 Analysis of the COMPLETE Rule

It is instructive at this point to examine a CA rule sequence. Figure 10.7 shows in detail three typical edge conditions that may require a specific entry in a lookup table. The CA window (or receptive field, RF) positioned at (a) needs no transforming action, as it is already placed at a solid position having an 8-sum = 24. When positioned at (b), the requirement (in this example) will be to extend the shaded line (with a defined [9]-value) to the right. This is achieved by noting that the 8-sum is [13] (with soap) and that converting an axial [1]-state pixel (soap) to the solid [2]-state will produce the desired result on iteration. In order to maintain this cellular status on subsequent iterations of the COMPLETE rule, an additional entry is required to continually convert certain [0]-state (background cells) into [1]-state soap cells, as the edge traverses to the right. The net effect is that, on iteration, the solid ([9]-value) line is extended one pixel at a time to the right. It may then join up with other solid objects, or terminate in some manner.

The CA-based RF positioned at (c) is required to fill a boundary gap of exactly one pixel in width. The 8-sum is [24], as with the solid portion of the boundary, but the cell's own current state (the "X" in the centre of the RF) is set to the [1]-state. So this particular CA lookup table entry will transform any 8-sum [1]-state cell into a solid [2]-state cell.

By examining a range of potential situations, and gap sizes, it is possible to devise a LUT rule which has a number of entries preset to deal with specific instances of edge, or boundary, discontinuities. The following lookup table entries perform the boundary completion steps illustrated in figure 10.7. This is, of course, an idealised image: actual images will contain noise, and so further entries in the rule tables may be required to eliminate point noise, etc., and deal with the non-ideal edges. For example, the CLEANUP rule used in the final stage of figure 10.6 has all [1]-state entries set to [0] — to remove the final traces of soap. The [2]-states are maintained for the entire 73 indexed lookup table entries, but one can knowingly zero the table entry for (8-sum $\rightarrow$ 0) at the same time—thereby achieving point-noise removal as an added image processing bonus.

Recall that it is necessary, for practical and computing reasons, to re-define a cell's [2]-state temporarily as a 9-sum value. [9]-values are re-converted to [2]-states during the CA image processing cycle by using another LUT function.

Definition of The COMPLETE Rule

0	1	2	3	4	5	6	7	8	11	12	13	16	18	22	24
0	0	0	0	0	0	0	0	0	1	0	0	1	0	2	2
0	0	0	0	1	1	1	1	1	1	2	2	2	0	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2

The COMPLETE rule emphasises the usefulness of the CA rule-based approach in image and vision processing. While it would be possible to devise some algorithm for completing a *specific* example shape (by first inspecting the image—using our human direct vision) it is usually not easy to design a general-purpose utility of this sort. The coalescence of cellular states representing image elements and image data changes is an unusual approach.

Note that this basic CA process can also be regarded as an example of illusory image processing (discussed in the following section). The broken boundary can typically be caused by noise, or occlusion, of an actual physical image boundary; but the effect can also, in certain circumstances, be considered as an illusion—depending as always on the context in which the image occurs. A military tank obscured by hedgerows or trees is a real physical image, but completing four dots to form the concept and image of a square is an illusory process.

It would be instructive to investigate the performance of CA-based image noise removal rules on a range of exemplar images whose noise distributions and characteristics were mathematically-defined, and thereby controlled. However, space, and project time limitations do not permit this at present.

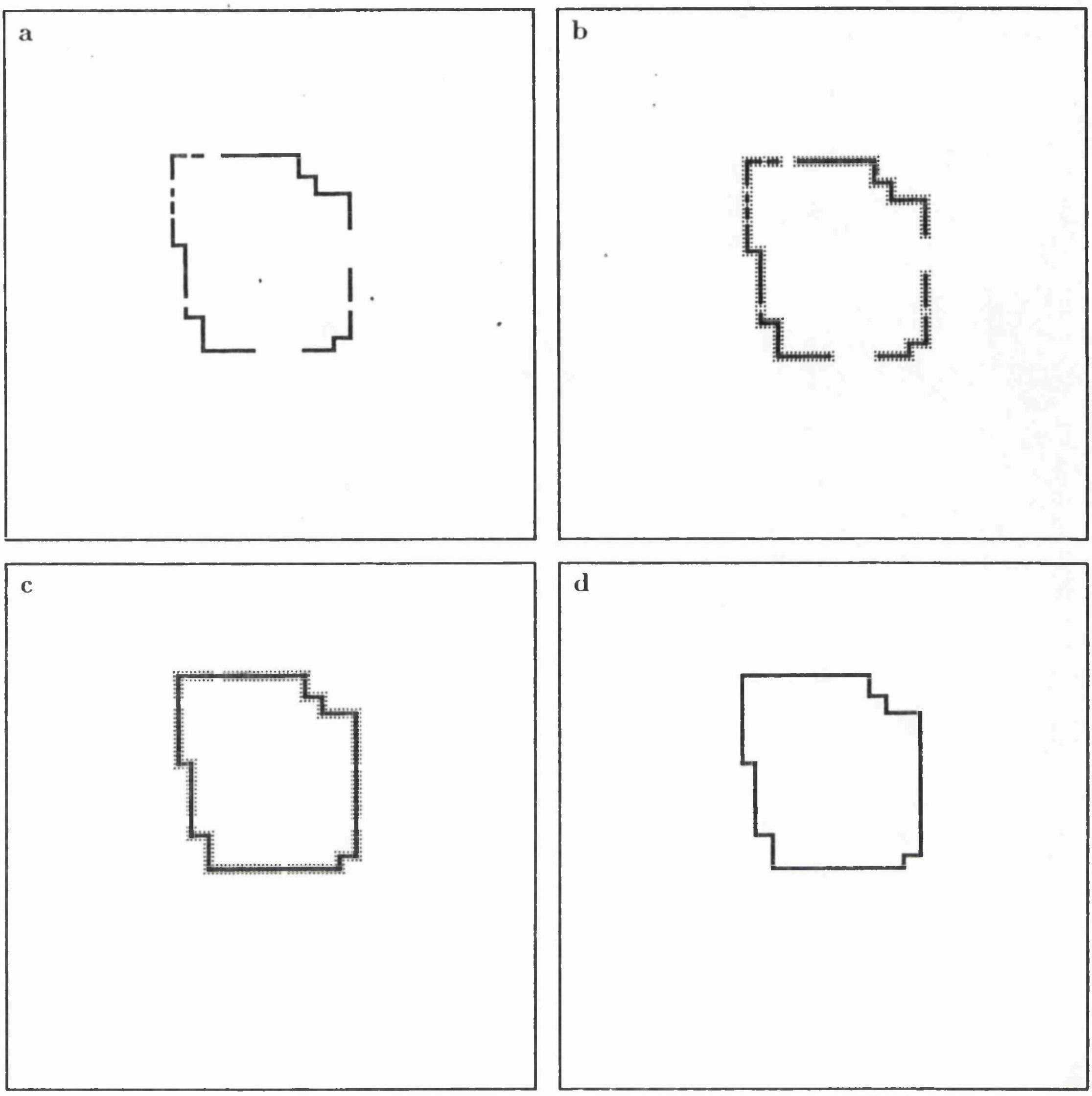


Figure 10.6 Test results for the COMPLETE rule.

(a) The initial image, including a broken boundary.

(b) The collapsed SOAP FILM (SF) adheres to the outline.

(c) The CA iteration solidly fills the gaps, leaving excess SOAP.

(d) The final cleaned-up image is obtained by removing any remaining [1]-state SOAP. The result is boundary completion to give a [2]-state "solid" edge and object outline.

<u>OPERATIVE CA LUT RULE — COMPLETE</u>

0	1	2	3	4	5	6	7	8	11	12	13	16	18	22	24
0	0	0	0	0	0	0	0	0	1	0	0	1	0	2	2
0	0	0	0	1	1	1	1	1	1	2	2	2	0	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2

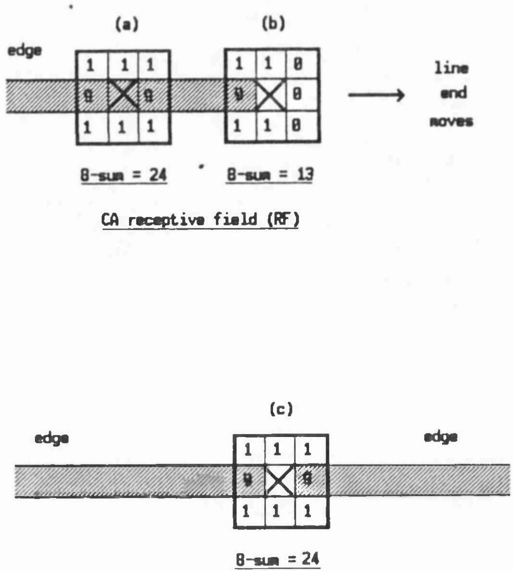


Figure 10.7 Illustrating the general concept of the CA rule COMPLETE, and the SOAP mechanism.

The CA receptive field at (a) is seen to produce an 8-sum value of 24, and the central RF cell is already in the [2]-state—so no action is required. The RF positioned at (b) produces an 8-sum of 13, and this can be used with SOAP to extend the boundary to the right.

The CA RF positioned at (c) is required to fill a gap in the boundary of exactly one pixel in width. The 8-sum is 24, as with the solid part of the boundary, but the cell's own current state (the "X" in the centre of the RF) is in the [1]-state. So this particular lookup table entry will transform an 8-sum [1]-state to a solid [2]-state.

The COMPLETE rule table shows the necessary entries required to complete the pixel-wide image edge. The potential of the CA method seems almost boundless.

The COMPLETE and CLEANUP LUT Rules

0 1 2 3 4 5 6 7 8	11	12	13	16	18	22	24
0 0 0 0 0 0 0 0 0	1	0	0	1	0	2	2
1 1 1 1 1 1 1 1 1	1	2	2	2	0	2	2
2 2 2 2 2 2 2 2 2	2	2	2	2	0	2	2

0	1 $\rightarrow$ 72
0	0 $\cdots$ 0
0	0 $\cdots$ 0
0	2 $\cdots$ 2

10.5 Synthetic Illusory Images

The ability of a machine vision model to correctly handle illusory images is regarded as an indication of biological plausibility. Some recent models have achieved this level of competence—for example Grossberg (1987b), McCafferty (1990). However, what one ideally would like is a model which can **automatically** detect the presence of illusory features within an image, and deal with them as a matter of routine. For example, the four-dot image discussed above is perceived as a square by humans. With colour vision, discounting the illuminant (i.e. colour perceptual stability) occurs in biological vision as a natural consequence of neural circuitry. Therefore, these are also desirable features of any advanced machine vision model.

One approach to this problem, developed in this work, is the use of preattentive features within an image. As mentioned elsewhere in this thesis, preattentive vision is considered by the writer to be a phenomenon, or manifestation, of direct vision.

The perception of the four-dot square illusion has already been demonstrated by means of the SOAP rule (Section 10.4) and need not be repeated. Similar illusions hold for other simple geometrical shapes, such as triangles and circles (see figure 6.1). The Kanizsa Square can also be demonstrated by the SOAP mechanism. All that is needed is to convert the four PACMAN shapes into four corner dots, by using the preattentive energy or other means of detecting the four right-angles (high-energy points), and then apply the previous SOAP rule. This is demonstrated in figure 10.8. It is assumed that this demonstration is sufficient to justify the validity of CA processing over the whole class of Kanizsa-style images (e.g. the Kanizsa Triangle challenge image mentioned in Batchelor, 1991).

The colour perceptual stability phenomenon can be illustrated by the well-known Craik-O'Brien-Cornsweet (COC) illusion. The PROVIS model's ECM first detects featural boundaries, and the HRS module fills-in the resulting webs to a level of brightness which is consistent with human brightness perception. Thus, the colour perception is decoupled from actual contrast differences, allowing the percept to be maintained within shadow, and so on (colour perceptual stability). Note that although one usually talks about *colour* perceptual stability, the phenomenon holds also for grey-level representation. Hence the COC illusion—even if it involves grey levels, or dithered images—is a valid demonstration of this important visual phenomenon.

In effect, the ECM defines the brightness perceptual limits, and the HRS relaunches the enclosing featural quality using the perception parameters with appropriate decay (diffusion). This is essentially the way the ECM-HRS system functions in delivering a semi-symbolic (cartoon-like) representation; so the colour perceptual stability capability comes as a bonus in the PROVIS vision model.

Figure 10.8 shows the derivation of a Kanizsa Square Illusion using the [1]-state CA SOAP rule, and preattentive corner detection. The original four PACMAN shapes have been superposed to form the illusion of a square floating in front of four black circles. There are other versions of this class of illusion, but the basic principle remains the same. Further processing of the [1]-states can reproduce the usual floating white square representation.

Figure 10.9 illustrates a one-dimensional (1D) demonstration of the COC illusion, even though the original image was a 64x64 synthetic 2D textual display. This is a CA-based demonstrator, the HRS in this case being due to a CA (particle-conserving) diffusion rule. It is seen that the normal human brightness perception—i.e. of the central band appearing darker—is easily reproduced by this purely **technical** means. The perceived "darker" central band, as represented in the graph of figure 10.9 (iv), is therefore the (simplified) perception that is passed on to the image recognition module (IRM)—not the complex signal whose profile is shown in figure 10.9 (ii). The COC image was derived with random dots whose statistical properties and distribution are known, and whose distribution is determined by the intensity profile curve shown in figure 10.9 (ii). In the above COC process the original (direct) image, as represented by image array [Ar], is used as a restraint.

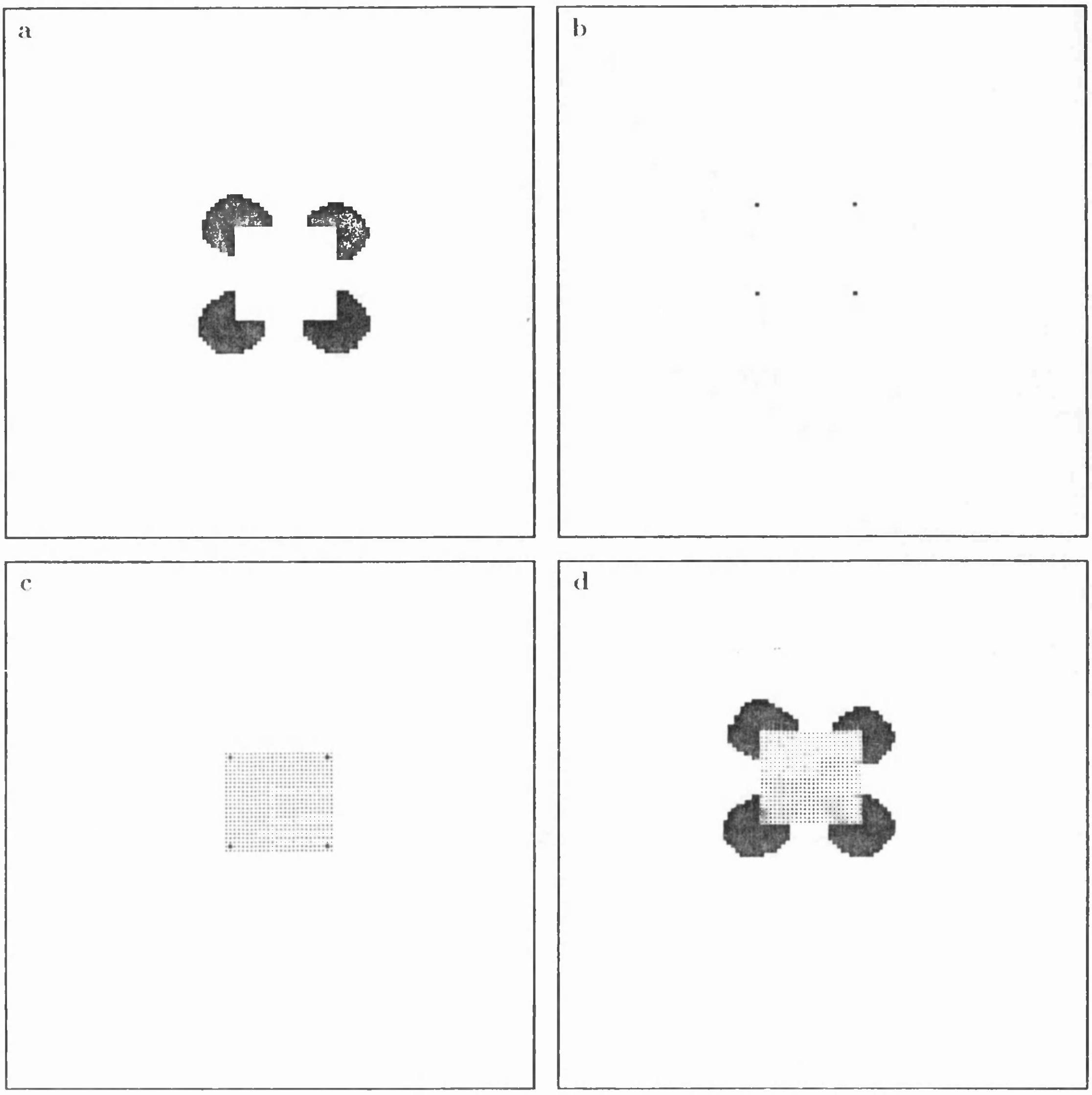


Figure 10.8 Test results for the Kanizsa Square illusion.

(a) The well-known Kanizsa Square (four-PACMAN) illusory image.

(b) The corners of the PACMAN shapes are preattentively detected.

(c) The SF completes the image with [1]-state SOAP.

(d) The final image as represented by a CA state field. The result is a [2]-state "solid" square perception. The original four PACMAN figures are superposed to complete the illusion of a square floating in front of four black circles.

OPERATIVE CA LUT RULE — SOAP

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1
2	2	2	2	2	2	2	2	2

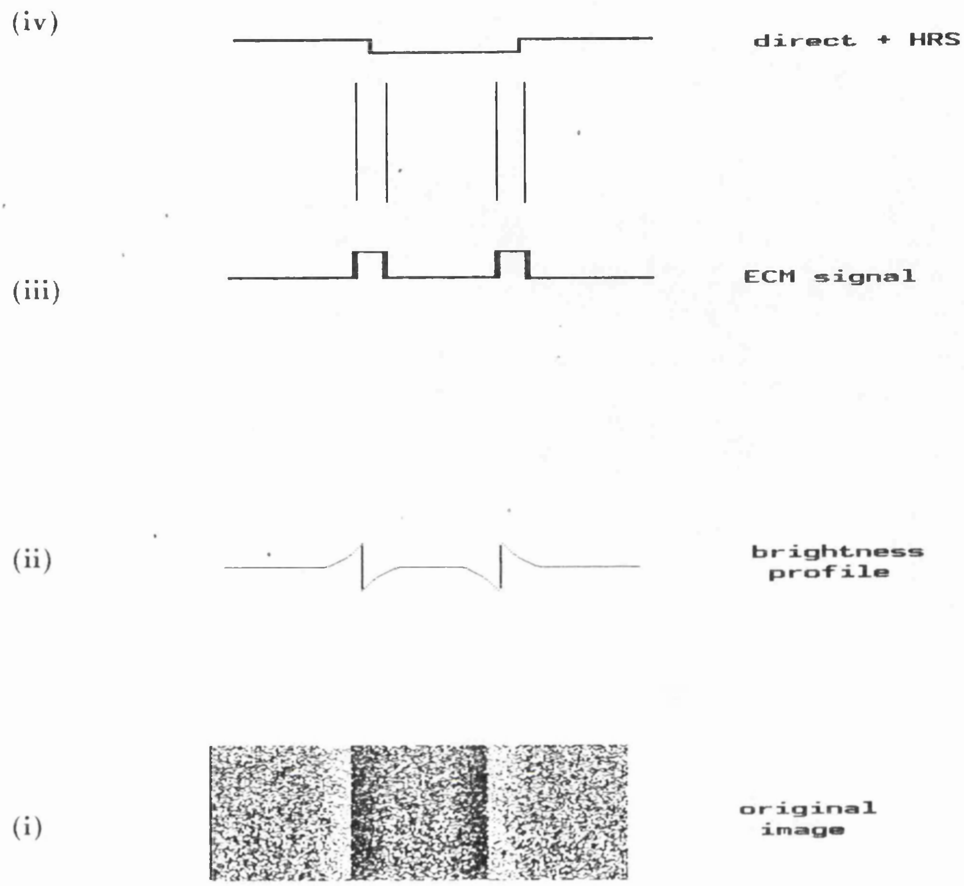


Figure 10.9 Demonstrating the 1D COLOUR PERCEPTUAL STABILITY phenomenon by means of the 1D Craik-O'Brien-Cornsweet Illusion, and the CA-based ECM-HRS.

(i) The original graded-intensity COC image.

(ii) The actual image intensity profile.

(iii) The ECM-derived boundary signal.

(iv) The particle-conserved redistributed HRS (brightness perception) signal.

OPERATIVE CA LUT RULES — SOAP(1) + DIFFUSE

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1
2	2	2	2	2	2	2	2	2

0	1	2	3	4	5	6	7	8	9
1	1	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
2	2	2	2	2	2	2	2	2	2

10.6 Demonstrating Direct Vision

This section demonstrates two aspects of direct vision within the technological processes of machine vision, as reproduced by our PROVIS model of vision.

10.6.1 Textured Images

The writer considers that the perception of image textural features or fields within an image is an aspect of direct vision—in the sense of the preattentive vision of Treisman (1985), discussed in Chapter 6 of this thesis. Detection of image texture is relatively straightforward using CA methods, provided that appropriate grouping parameters can be predetermined. In the method of McCafferty (1990), the user has to manually specify suitable parameters for texture segregation—using his or her natural direct vision. In the present work, the use of a range of preattentive (i.e. direct) vision methods for automatic rule selection will be emphasised wherever possible.

The technique is to set up appropriate CA lookup tables to provide SOAP LINKS. If suitable rule-table parameters are chosen, textural elements will, on iteration of the rule, tend either to fuse together by CA expansion to form "solid" regions, or else will disappear. By using a multi-layered CA method, different sets of parameters can be specified such that each hierarchical tree level contains a different segmented textural region of the image. This, again, is an aspect of the edge-evidence array [Jr] discussed in Chapter 9.

Much work has yet to be done in what is traditionally a difficult aspect of machine vision. The demonstration shown here is sufficient to emphasise the **potential** of our approach. Figure 10.10 shows a CA segmentation of a synthetic textured image. The required CA rule was selected by a preattentive metric, which *indexes* an array of rule pointers. This result is intended for demonstration purposes, and it is not possible to guarantee that a good textural segmentation will necessarily be achieved on arbitrary images.

Figure 10.10a shows the original textured image, consisting of a mixture of 45-degree lines (left) and random dots (right). This is a simple textural segmentation and, as before, the image resolution is only 64x64 pixels. Note that, although we humans can easily "see" that figure 10.10 contains two texture fields—because, of course, we have "direct" vision—this is certainly not apparent to a machine. And so here we have the hub of the AI vision problem—how to embody abstract concepts of direct perception within an inanimate machine. In this section, a preattentive mechanism is suggested as a useful starting point for further research and discussion.

Figure 10.10b shows the effect of the normal SOAP film rule of Chapter 4. It is seen that soap fills the spaces between the two regions, and almost separates them. In figure 10.10c, a simple alteration to the standard SOAP rule entries causes an image [1]-state separation into two distinct segmented fields. Finally, figure 10.10d shows the right field after it has been consolidated by the "1-2" rule—all soap [1]-state cells are transformed to solid [2]-state pixels. A similar process can be used to solidify the left-half image features (notice from figure 10.10b how the standard SOAP rule fills the spaces between the 45-degree lines).

Again, once the two fields have been separated, any relevant CA rule, or sequence of applied rules, can re-process the final image. This form of SOAP–ACTIVATION-CLEANUP cycle is typical of the CA methods developed here. Notice how the detection, the separation, and the subsequent solidification of the right textured field produces an effect not unlike the CONVEX HULL of conventional image processing. The use of CA principles involving ADAPTATION and GENETIC algorithms may lead to new concepts in CA computation: for example, what can be expected from the direct manipulation of the individual table entries within a CA LUT rule? The scope of CA computations and LUT rules is limited only by the imagination of the investigator.

As mentioned, this is only a very simple demonstration. It is hoped, nonetheless, that it is sufficient to illustrate the general principle of SOAP fusion, erosion, and expansion to define separate regions for segmentation. The important factor here is the preattentive mechanism which attempts to index a suitable rule for a textural segmentation.

Much work has still to be done in this connection, as well as in the investigation of new segmentation rules for various classes of real and synthetic diagnostic and clinical (image scans) textured images. Example image classes that come to mind include grass fields, vegetation, manufactured textiles, chest X-rays, and so on. The essential problem is orchestrating so many subprocesses.

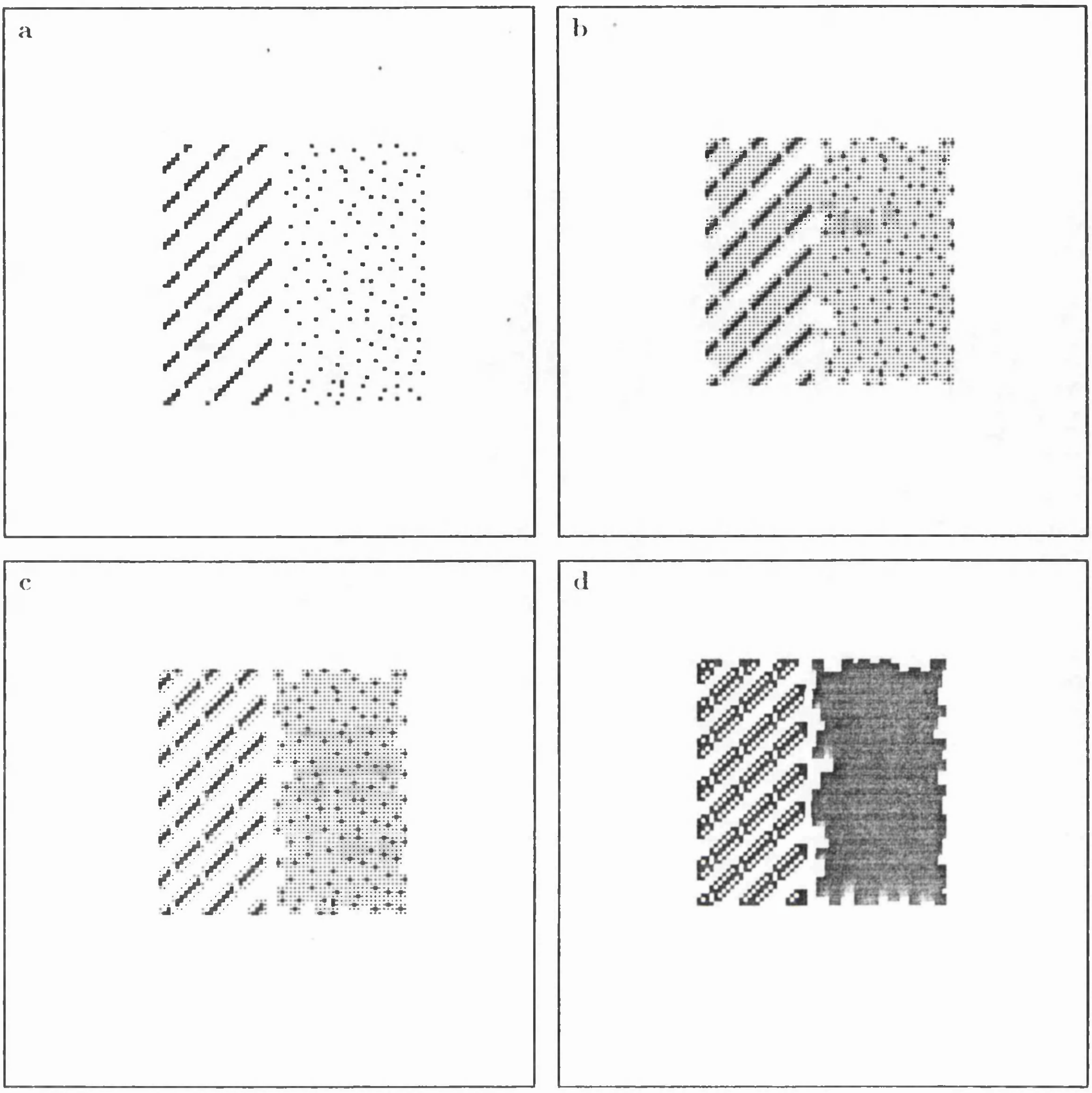


Figure 10.10 Demonstrating the segmentation of a simple two-region textured image.

(a) The original low-resolution textured image.

(b) SOAP fills the feature spaces, but still links the two regions.

(c) The modified SOAP rule successfully segments the image textures.

(d) The final segmented image. The right field is a "solid" perception obtained by the "1-2" transform rule, which simply changes all [1]-state (SOAP) cells to [2]-state solid pixels. Further (or alternative) CA processing can be specified at this stage.

OPERATIVE CA LUT RULE — TEX(5)

0	1	2	3	4	5	6	7	8	32
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	2
2	2	2	2	2	2	2	2	2	2

10.6.2 Direct Image Mediation

The second demonstration within this section concerns the role of direct vision in the mediation of the ECM-HRS mechanism. If, as in the earlier assertions of this thesis, there exists at least two distinct mechanisms of vision— "direct" and "processed" —then it can be expected that these two vision concepts will function in collusion. That is, the writer considers that as a scene is viewed "directly" and with high-acuity and colour perception, a simultaneous "image processing" paradigm is at work decoding the semantics of the scene for archival storage and retrieval.

It is asserted here that these two processes fuse or blend together seamlessly. Biological vision is normally unaware that two separate vision processes are occurring simultaneously. Nevertheless, this is a most significant feature of natural vision, and is indeed a tenet of this thesis.

The method of mediation is an inherent part of the Edge Constraint Map (ECM) mechanism. Earlier tests by the writer showed that several conventional averaging and segmentation methods failed to maintain the image semantics during segmentation. The merging of image regions, if allowed to continue unchecked, can ultimately result in meaningless segmentations. The ECM-HRS method in this work allows several factors to determine the best possible edge map—thereby providing the most likely candidate boundaries for image segmentation. The ECM module constrains any subsequent averaging of the HRS image semi-symbolic form to just these well-defined webs.

Figure 10.11 gives some idea of the mechanism applied to an actual picture (a village—VIL.IMG). Figure 10.11a depicts the original image. Figure 10.11b shows the resultant segmentation by one kind of conventional segmentation method, and figure 10.11c by another. Figure 10.11d is the segmentation by the PROVIS ECM-HRS modules. As in all of the demonstration in this chapter, iteration continues until the CA processor enters a predetermined halt state—which is normally the cessation of [1]-state (SOAP) cellular activity (the CA is then in equilibrium).

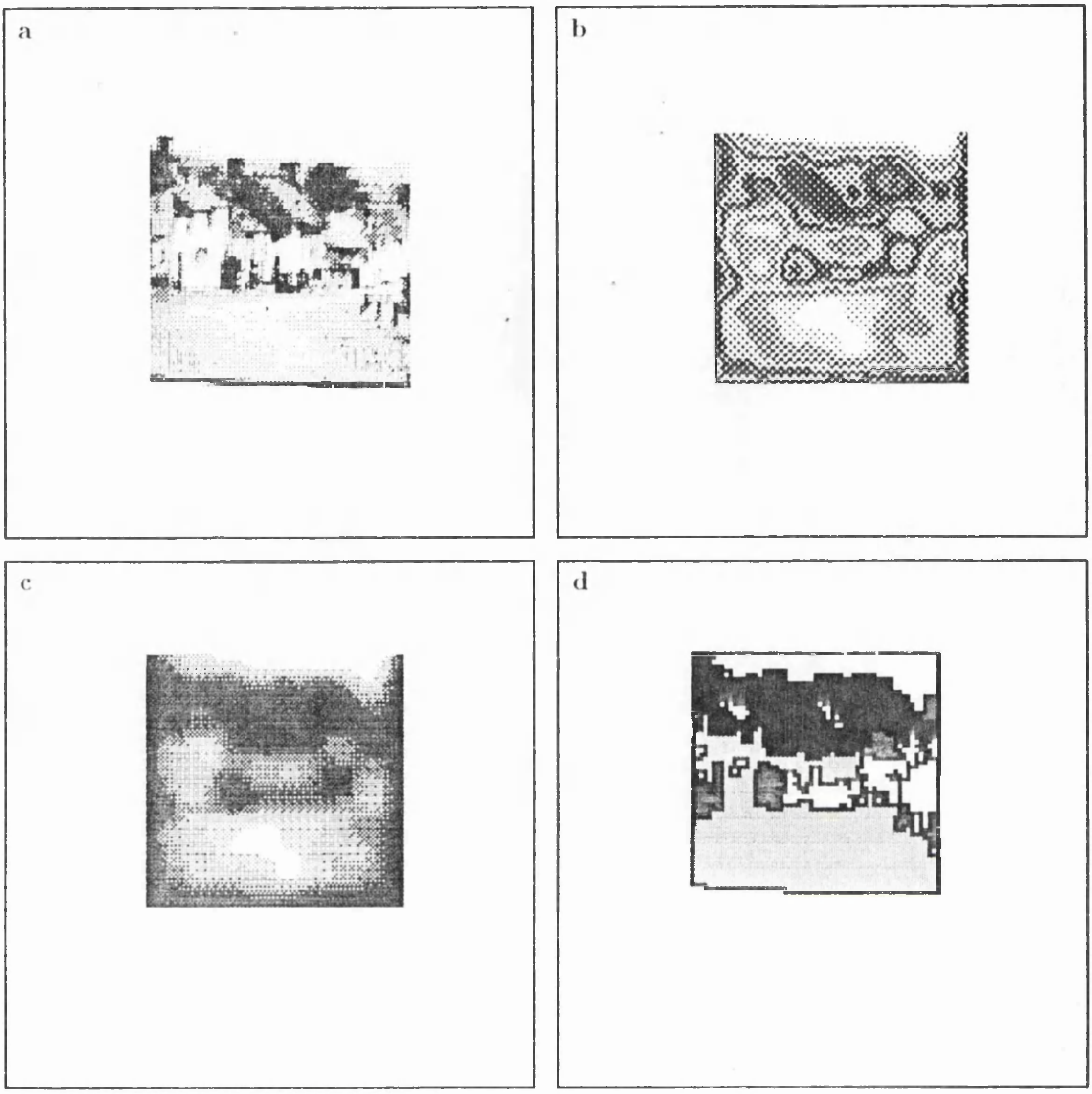


Figure 10.11 Demonstrating the use of very limited aspects of direct vision in the mediation of the ECM-HRS modules of the PROVIS demonstrator.

(a) The original low-resolution village image.

(b) Segmentation by a region growing method similar in concept to that described in Beaulieu and Goldberg (1989).

(c) Segmentation by a region growing method described in Ballard and Brown (1982).

(d) Segmentation by allowing the direct vision image (held in image array [Ar]) to mediate the ECM-HRS modules. This yields the required "cartoon-like" representation for transmission to the IRM.

10.6.3 Energy in CA Computations

Before leaving this section it is worthwhile briefly examining the role of "energy" within CA processes. As was mentioned elsewhere in the thesis, the use of some form of energy, or other appropriate metric, is required in order to determine when a CA process should be halted—either through programming (at the cessation of the current computation) or to enable another CA rule to take over at the relevant stage of the process. The CA has thus attained a state of equilibrium.

The method used here is the simplest technique of measuring cellular activity, as indicated by state changes. In many cases the soap film (SF) or [1]-state activity will suffice, while at other times it may be more appropriate to measure the [2]-state activity, total system energy—or perhaps all of these. The energy may represent some image global energy, or, more usefully, the rate of change of energy can be determined. The basic CA time-step is one iteration, so simple methods can be used in error-rate determination. When the energy-rate is zero the CA can either halt, or it can exchange rules as appropriate. When defining practical CA processes, one must examine the algorithms to establish processing points at which CA rules or processes are to be exchanged, or terminated. This, again, is an open-ended research topic and cannot be discussed in any depth here.

Figure 10.12 shows a graph of the [1]-state energy changes taking place during the TEXTURE rule (TEX(5)) invocation, as featured in figure 10.10 above. It is seen that, using the coarser 64x64 CA image pixel resolution, iterations cease on the count value [63], at which point the [1]-state cellular energy count has declined to zero (i.e. the CA has achieved equilibrium).

It may, in real images undergoing automatic examination, be more instructive to plot [1]-state, [2]-state, and total-energy changes throughout the several stages of a CA image processing computation. This could yield valuable statistical or research information on CA processes in particular, and on images in general. For example, it may in future be possible to derive meaningful image "signatures" from the further processing of cellular energy-change data.

The graph illustrates how the [1]-state energy change rate can be used to either terminate the ongoing CA process, or replace the current CA rule with some other processing rule. The system easily detects when the monitored [1]-state cellular energy-rate has attained zero.

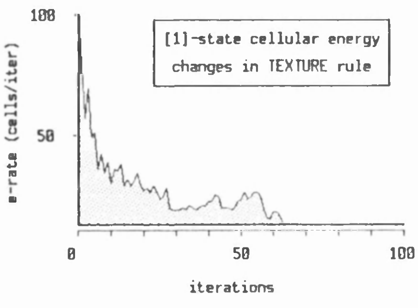


Figure 10.12 Graph showing typical [1]-state cellular energy changes taking place during the TEXTURE rule (TEX-5) demonstration. See text for details.

10.6.4 The Convex Hull

A very useful and important image processing function in conventional machine vision is the determination of the convex hull of an image. Figure 10.13 shows how this is easily achieved using appropriately thresholded versions of the previously defined SOAP rule.

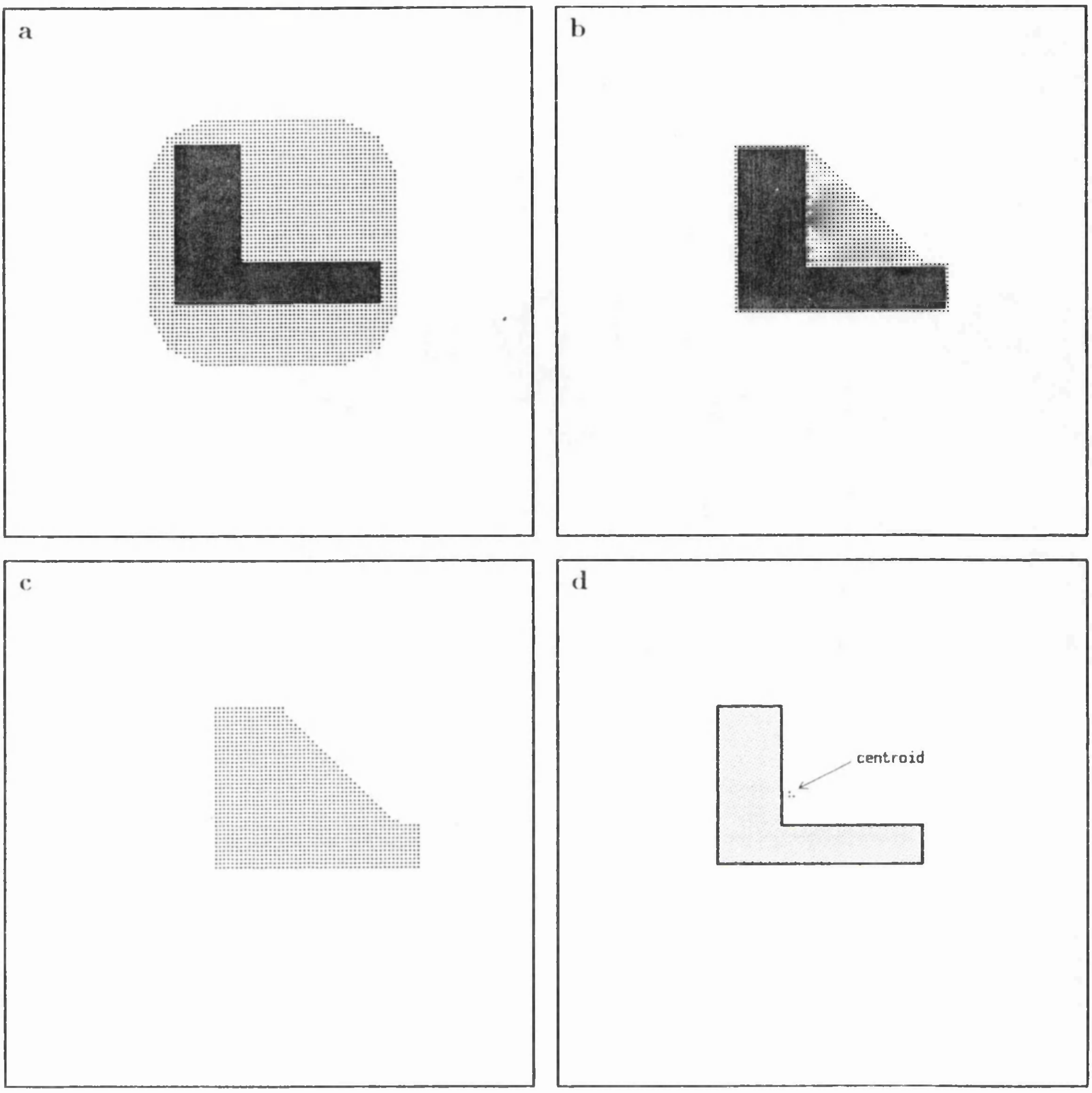


Figure 10.13 Test results for a typical convex hull determination of CENTROID.

(a) The original L-shaped test image, and SF.

(b) The stable L-shaped figure with SOAP-filled web.

(c) The convex hull re-defined in terms of [1]-state SOAP.

(d) Eroding to a stable energy state produces a 3-cell centroid. The original shape has been redrawn to show the position of the centroid. Any one of the three cells can be selected as the centroid—with reasonable accuracy.

OPERATIVE CA LUT RULES — SOAP(2) + CHULL

0	1	2	3	4	5	6	7	8	9 → 72
0	0	0	0	0	0	0	0	0	0 ··· 0
0	0	1	0	0	0	1	1	1	1 ··· 1
0	0	0	0	0	0	0	0	0	1 ··· 1

10.6.5 Demonstrating Preattentive Vision

Another role for cellular "energy" in the present work is the rapid determination of preattentive image features. As mentioned above, and elsewhere in the thesis, preattentive image features (in the sense of Treisman, 1985) are salient aspects of an image instantly[1] perceived. Examples of simple high-energy features are line-ends and right-angled corners. But the range of image preattentive phenomena is, for all practical purposes, boundless. It will be recalled that preattentive features are claimed to be image qualities that are detected without recourse to higher level knowledge. This should mean, for instance, that a "shock" photograph cannot be classified as a preattentive image in the sense defined by Treisman, because such photographs usually need to be interpreted.

A severe conceptual problem arises in the nature of preattentive vision itself. If image features are really perceived preattentively then there should be no requirement for computations whatsoever. This would appear to exclude even the simplest of CA processes. (In the writer's view, this problem arises because machine vision does not have direct perception. Thus, practically, every method of dealing with raw input images requires *some* form of processing in order to be able to detect preattentive features.)

For the purposes of this demonstration, only simple preattentive image concepts are used (line ends, corners, our Treisman-like figures), but the research must be ongoing, to discover ever new visual preattentive mechanisms. The multi-state array [Jr] is used as a Parameter Space to hold preattentive features, in addition to other data from orientated edge masks (Chapter 9). As discussed, these data are combined to form a part of the total evidence for the existence of robust edges and boundaries within an image.

Figure 10.14 shows a screendump depicting a simple image (left) and its corresponding preattentive features. As mentioned elsewhere in this thesis, preattentive featural data can be processed to yield a set of indices for CA rule selection. It can also provide contextual input and reinforcement for the Prolog-based higher functions. These possibilities will be discussed later.

Again, the direct image held in array [Ar] is invoked to establish conditional processing within the CA algorithm. This can modify the global result in quite

[1] In this case "instantly" means less than about 500 ms.

dramatic ways. Indeed, a characteristic of CA rule tables is that table entries can critically affect the outcome of the process, because of the sensitivity of table data. The above CA demonstration suggests how complex effects could also be induced by external **trigger** signals.

Figure 10.15 shows a Treisman-like "popout" image. In figure 10.15, (a) is the original input, (b) shows dot removal, leaving preattentive features intact, (c) removes 45-degree lines, also leaving the central preattentive figure. Figure 10.15d was obtained directly using an elementary one-pass CA rule (SOLID) that removes every cell except those embedded within a solid shape. The result is similar, except that the central figures are slightly reduced in area.

Notice how a simple one-pass CA rule is able to sweep across the image (in our serial simulation) eliminating all but the "popout" figures. In a real (i.e. massively-parallel) CA implementation this would be almost instantaneous. There is something intuitively satisfying about such a "direct" process: it appears to be getting very close to our goal of realising a technical implementation of natural vision.

This is admittedly a somewhat simplistic preattentive demonstration on an idealised image. Many different preattentive rules can be needed to search out preattentive features within an image, the results being stored within the layered arrays of parameter space [Jr]. As mentioned above, the orchestration of many such processes is a major challenge.

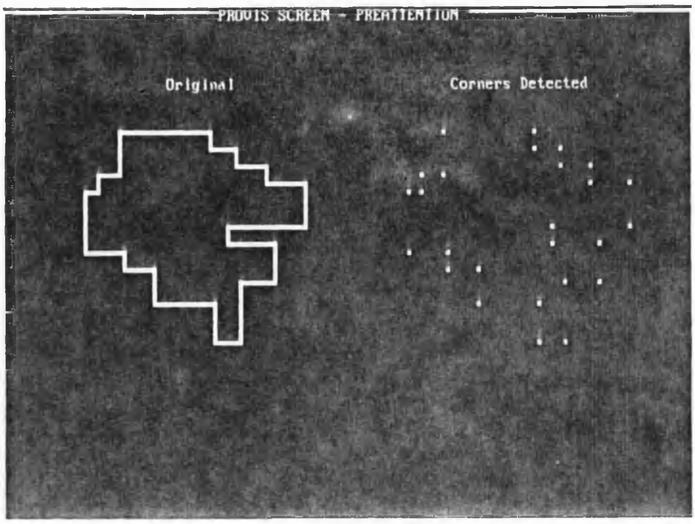


Figure 10.14a The PROVIS demonstrator screen showing an image (left) and its preattentive features (right). See text for details.

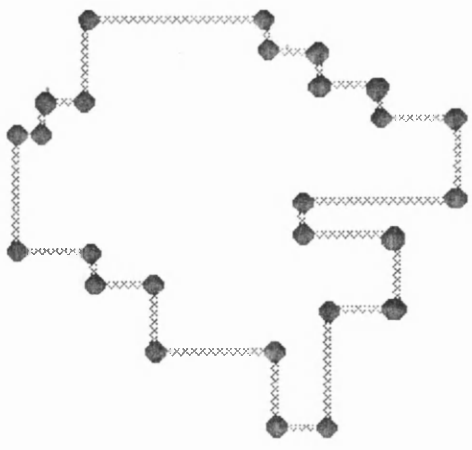


Figure 10.14b The right frame from the above is reproduced to show the image highlights. This may have some relevance to Attneave's highlights, discussed further in Appendix J.

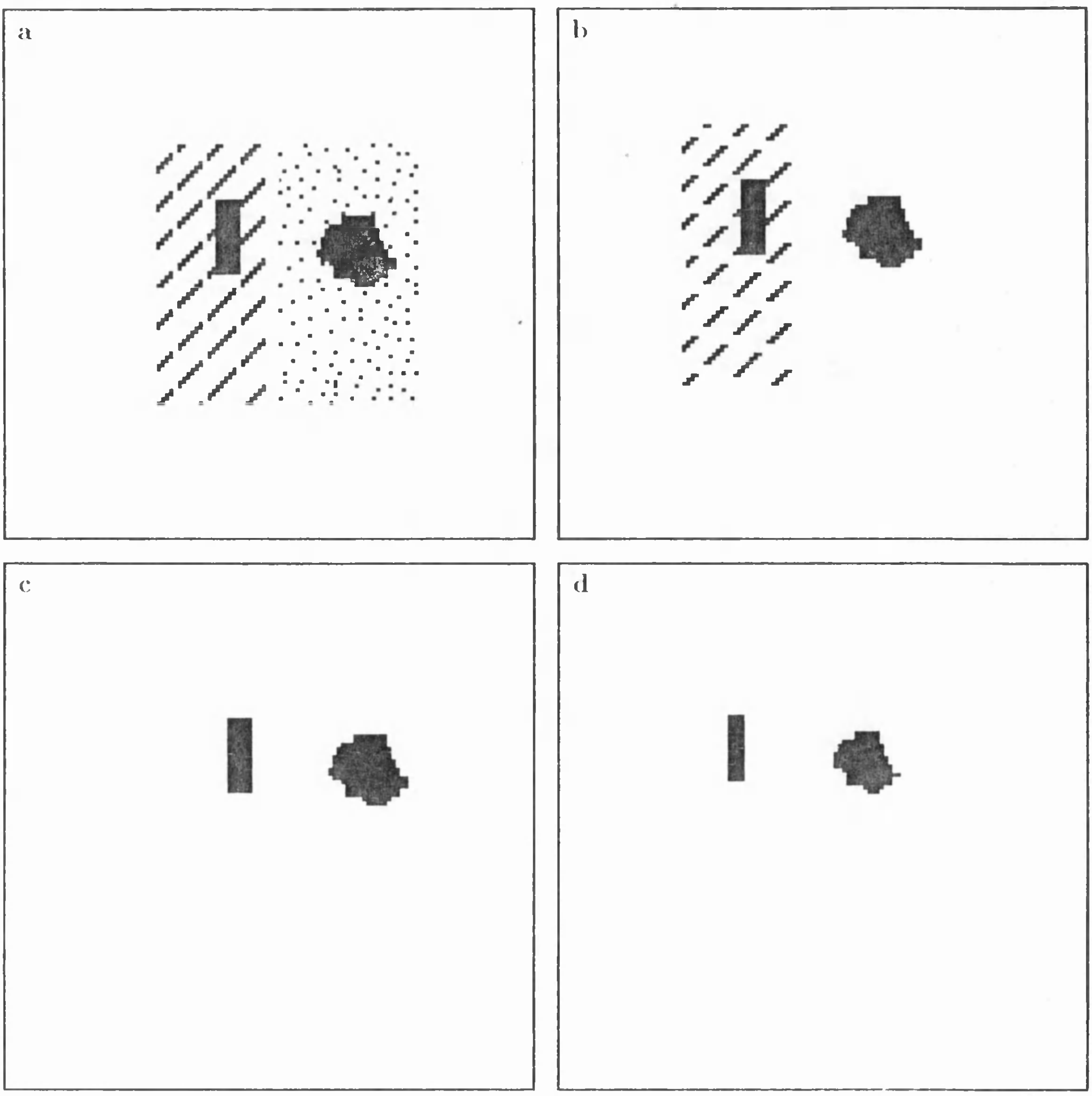


Figure 10.15 Test results for a Treisman-like preattentive ("popout") figure.

(a) The input test image with preattentive features.

(b) The central figures are preattentively detected; dots removed.

(c) The central preattentive features only remain.

(d) A simple method using the undernoted elementary CA rule.

<u>OPERATIVE CA LUT RULE — SOLID</u>

0	1	2	3	4	5	6	7	8	72
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2

10.7 Demonstrating Image Understanding

The previous demonstrations in this chapter have concentrated on the use of cellular automata (CA) rules, and their LUT definitions. These reported results are therefore concerned mainly with demonstrating the potential of CAs on a representative range of typical machine vision tasks. The novelty so far is our use of CAs as equivalents to ANNs, and their direct substitution for many conventional image processing algorithms. In addition, the relevance of CAs in the determination of Gestaltist (i.e. psychological) image features has been demonstrated. This can be verified by comparing these results for CAs with similar experiments performed on ANN models by Grossberg (1987). This is seen as a vitally important factor in our model.

This section shows how Prolog, the AI programming language, is able to call-up a series of VPC image functions directly, to perform a variety of low-level vision (LLV) image processing mechanisms. The VPC command set, as explained in Chapter 8, contains both conventional C language procedures and predefined CA rule functions. Most VPC calls return parameters that are used directly by Prolog.

Thus, a single specified VPC (low-level) function can be called directly from Prolog, or else a series of linked VPC routines can be implemented, following the general principles described in Batchelor (1991). Note, however, that unlike the Batchelor system, the PROVIS model is not functionally **interactive** at this level—machine autonomy is the preferred method of working. This also explains why the concept of direct vision (preattentive image analysis, etc.) is so important. Our use of a **hybrid** language approach allows Prolog to concentrate on the high-level vision (HLV) functions, leaving the LLV mechanics to the more efficient VPC (Turbo C) methods.

In this section, the mechanisms discussed previously are brought into play to provide a demonstration of image understanding. This requires the facilities of all three PROVIS modules—the ECM, the HRS, and the IRM. As discussed earlier in this thesis, an important by-product is that of image data reduction. A fine-grained, high-detail, image is segmented by the ECM-HRS modules to yield a semi-symbolic (cartoon) form, which is then offered to the image recognition module (IRM).

As discussed, real biological vision systems do not function with images alone. Object or scenic recognition requires a knowledge-base of factual and experiential information, in order that images can be placed within a context. In reality, this contextual data can come from the other senses (especially hearing), and (or) from associative memory recall—for example, as demonstrated by Grossberg's Adaptive Resonance Theory (ART). See Grossberg (1987a) for details.

A biological system is also able to adjust or constrain its image input by, for instance, changing its viewpoint. A sequence of images (or a continuous visual input stream) can be a great advantage in separating objects from background, or in eliminating image noise further by evidence-based reinforcement.

As mentioned, three-dimensional (3D) vision is of some importance in near tasks (but not necessarily so in image recognition *per se*) and in aiding the placement and location of objects by both humans and robots. However, 3D imaging is not implemented in our PROVIS model.

It is not possible here to consider all of the relevant points in detail, as they are subjects for future research. In any case, the primary goal of this project, as from Chapter 1, is to formulate a **paradigm** for a cogent model of human vision—which includes a concept of direct vision. In addition, the psychological tenet of the Gestalts must somehow be accommodated in any convincing model of biological vision, as should the distinct process of human facial recognition.

10.7.1 Brief Review of ECM-HRS

Before going on to consider recognition within the PROVIS system, it may be helpful to review the ECM-HRS approach. The starting point is the writer's insistence that a primitive semi-symbolic presentation to an image understanding system is adequate, when direct vision is able to provide the necessary original image in full detail. This is implicit in **every** conventional machine vision method that may resort to the stored **original** image to regain detail, or other basic image information. Recall that the idea of a cartoon-form is based on the writer's notion of simplified images: the tree-lined canyon of Kelvin Way is characterised by a simpler (semi-symbolic) image featural representation.

The essential notion from Gestalt theory is that objects are perceived as groupings of primitive tokens: a cluster of dots might be seen as a human form, for example. The Gestalt theory is applicable over various kinds of tokens, such as texture elements, and so on. The net result of Gestalt-inspired processing is that a typical scene (if indeed it has any meaning) is grouped into objects and background. In practice, the Gestalt (2D) spatial method effectively encloses psychologically grouped tokens or features with "rubber bands." Such a grouping can be significantly different from that obtained by a purely algorithmic and (or) conventional means.

In the present PROVIS application, the Edge Constraint Map (ECM) is effectively a Parameter Space (a multi-level image array) which is able hold a varied assortment of evidence relating to the probability of finding meaningful image edges. Thus the ECM may make use of some, or all, of the following stored iconic maps in its final probabilistic determination of meaningful image edges, or groupings:

$$\left.\begin{array}{l}\text{(1) edge contrast}\\ \text{(2) edge orientation}\\ \text{(3) preattentive highlights}\\ \text{(4) simple direct vision}\\ \text{(5) high} - \text{level expectancy}\end{array}\right\}\quad \text{in combination}$$

Once the ECM has provided a constraining edge-map (the outline of an object or region) or a web of edge-mapped elements, the Homogeneous Regional Segmentation (HRS) module is invoked to apply a mechanism of filling-in to reconstitute the original image as a "cartoon-form." Several different methods can be used in the HRS, including a CA-based (particle-conserving) diffusion, relaxation, or a re-launched colour CA diffusion (with exponential decay if required). In all cases, the original image (Gibsonian) held in image array [Ar] is used to assist in the process of mediation. This is but one instance of a technical interpretation of the psychological concept of direct vision.

The original image colours are meant to be used in the ECM-HRS, but this aspect of the project has not yet been fully developed. It will be recalled from Chapter 1 that colour, by itself, can be the single most important image recognition factor in symbolic vision.

10.7.2 The IRM Functions

When an image has been robustly segmented through the ECM-HRS, it can be presented to the IRM for recognition. The IRM is Prolog-based, but can optionally call-up any VPC (low-level) function directly. One such VPC function is the **image histogrammer**, which is the first stage of processing within the IRM.

The purpose of the histogram is to determine the number of distinct homogeneous regions within the cartoon-form image. The VPC histogram function call returns the image regional parameters, and can display the data on the computer (but it is not essential to do this). Unlike the ECM-HRS modules discussed above, the IRM is much less neuron-like, and rather more AI-like. That is, the IRM is a conventional image recognition module, since it uses image parameters and statistics to determine a classification. It is like a simple expert system.

Nonetheless, the IRM has the novelty of Prolog procedures, concentric contexts, and the beginnings of a scheme to deliver EXPECTANCY to the early vision processing level. Referring back to the system diagram of figure 10.1, it is noticed that a two-way link between Prolog (IRM) and the multi-level array [Jr] (holding the image Parameter Space and preattentive feature maps) could facilitate the placement within the evidence-accumulator array [Jr] of specific additional image features representing expectancy. Indeed, other kinds of edge evidence can be included within [Jr]. However, this aspect of the project too has to be developed.

It is the analysis of the image metrics, such as area, centre-of-area, perimeter, shape, and so on that make the IRM a seemingly conventional AI module for transforming iconics to symbolics. The writer believes that—ultimately—neural network (and hence CA) developments will deliver equivalent conventional numerical image processing. In other words, the routines currently carried out by VPC subroutine calls could be replaced by equivalent CA processes.

The following table shows the parameters returned by a typical VPC call **get_reg** to the IRM:

TABLE 10.1: PARAMETERS RETURNED BY GET_REG

VARIABLE	METRIC
R_id	The region ID (number)
R_area	The area of the region
R_perm	The perimeter of the region
R_Xc	The X-coordinate of region area
R_Yc	The Y-coordinate of region area
R_col	The region colour (or grey-value)
R_fac	The region shape factor
R_lis	The interconnectivity list

The parameters returned in GET_REG are sufficient for a simplistic, expert-like, approach to image understanding. More complex work can be done if additional, or enhanced, parameters are used—for example, a more capable version of R_fac. However, extending VPC is a relatively routine programming exercise and needs no further discussion.

In the next step, a Prolog VPC call returns the interconnection matrix to Prolog, which is then able to "understand" how the image regions are related to each other. At this stage, an image-tree has in effect been determined, from which the Prolog-based IRM may be able to deduce the **meaning** of the image from the current context.

If the IRM is unable to make an immediate identification, it may look for clues in order to set additional contexts—concentric contexts. It can do this by making further individual VPC calls, and examining the returned parameters. At the same time, the IRM can display its progress or decisions on the computer screen. Note that this is NOT a fully interactive process, since the user need not be involved in any further dialogue once a PROVIS task has started.

The IRM is able to use VPC function calls to establish certain additional (fuzzy) facts about the image regions it is processing. For example, it may need to know whether a certain bounding edge is **vertical** (within some fuzzy constraint), or if a specified region is to be regarded as being **very big** (an example **very_big** Prolog fuzzy clause was defined in Chapter 9). The end result is a justified symbolic recognition and interpretation of the input image, which may be a noisy, complex natural scene.

10.8 A Demonstration of the ECM-HRS

Figure 10.16 shows screendumps of the main steps in the demonstration of the ECM-HRS mechanism. A synthetic image was defined, having the edge-map in the shape of four rectangles. Random background noise was introduced, together with bias weights (colours) within each of the four regions. This was done to guarantee that each region stabilises on a different symbolic representation (colour). The edge map is normally represented as the (reserved) white pixels.

Figure 10.16a shows the original image; 10.16b the noisy kernels; 10.16c the filtered feature kernels **expanding** towards the constraining ECM; 10.16d the final 4-region symbolic image.

It is possible to convert the black space vacated by the ECM boundary to one of the image colours. This can be done by any of a number of simple methods, but will not be discussed further here. Figure 16 is a technical demonstration of the basic ECM-HRS mechanism, as used in our PROVIS model of human vision.

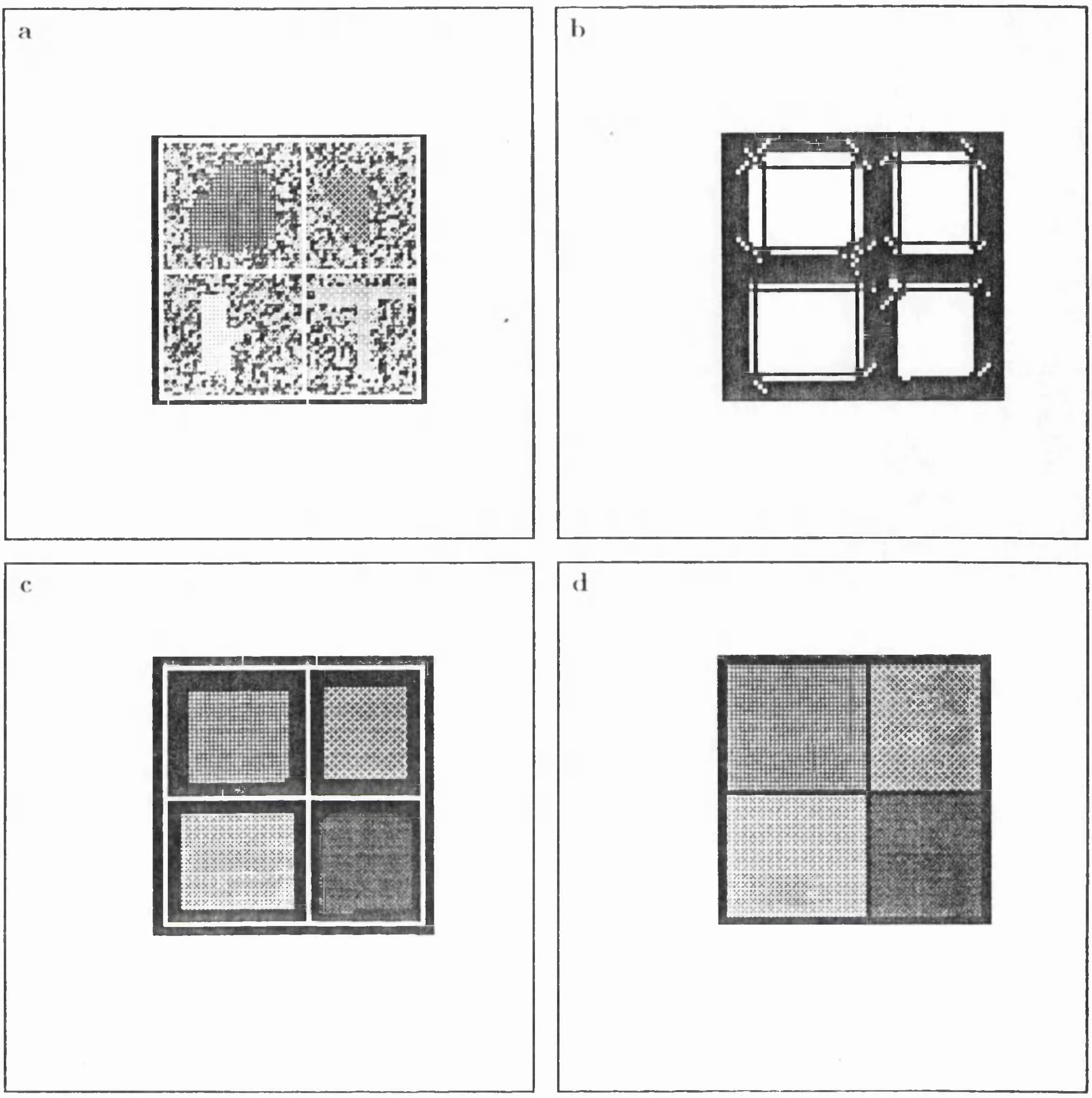


Figure 10.16 A demonstration of the basic ECM-HRS mechanism, as used in the PROVIS vision model.

(a) The synthetic input image with predefined edge map (white).

(b) Relaxation processing, in collaboration with the original image, yields noisy colour kernels. These are filtered to leave "clean" kernels.

(c) The colour kernels expand outwards to meet the ECM. This is quite a complex process involving the original image.

(d) The final clean, semi-symbolic, image of four homogeneous regions. This would normally be processed by the IRM.

10.9 A Synthetic Image Demonstration

Figure 10.17a shows the main PROVIS opening menu, and figure 10.17b is a selected VPC menu. The chosen demonstrator object is a 3.5-inch floppy disk, as used in the Batchelor (1991) discussion on high-level Prolog+ functions. This image is already in a semi-symbolic form, so PROVIS can deal with the recognition immediately. The image of the disk is loaded and analysed by VPC, as shown in figure 18. The data are passed back to the Prolog system which searches through stored features lists, looking for a match to the disk's attributes. A match is found, and PROVIS terminates with success. Figure 19 shows output from the PROVIS search log.

This is a very simple demonstration of the PROVIS vision model. The simplicity stems from the fact that the floppy disk image is already a noise-free, semi-symbolic representation. The problem is reduced to that of a simple Prolog match of robust features lists of prestored image models and categories.

Despite the model's simplicity, the advantage of the Prolog symbolic method has been demonstrated. A fully developed expert-domain system would be expected to handle much more complicated situations; and is, of course, readily available in current technology. A system based on the frame approach (Minsky, 1975) is one example.

The demonstration discussed in the next section shows how a natural external scene is analysed by the PROVIS system.

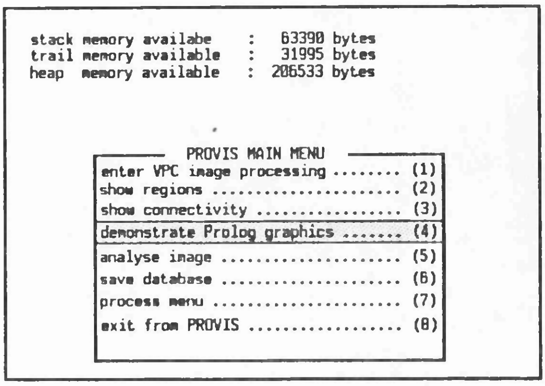


Figure 10.17a The PROVIS vision model's main menu. This is the user interface to the demonstrator.

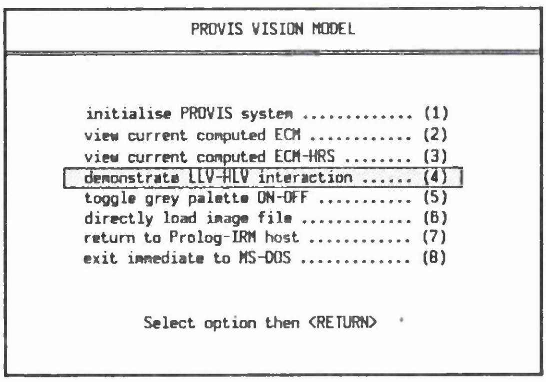


Figure 10.17b The PROVIS vision model's VPC menu. This is concerned mainly with low-level image processing options.

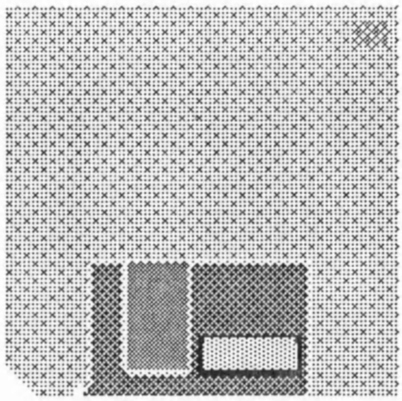


Figure 10.18a The floppy disk demonstration image is already in a pure symbolic form.

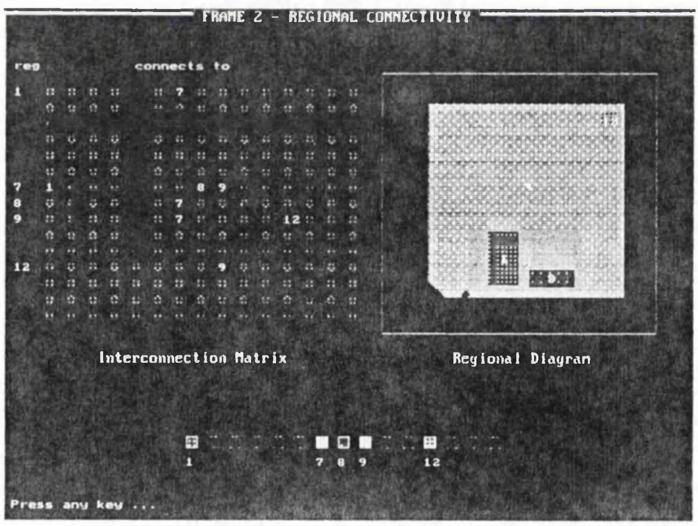


Figure 10.18b The floppy disk image connectivity map. This is used by PROVIS to create the full symbolic description for matching with the prestored database image feature lists.

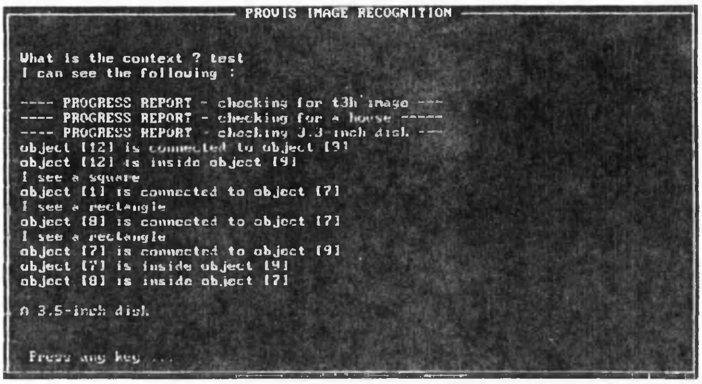


Figure 10.19a The PROVIS output report for the floppy disk image.

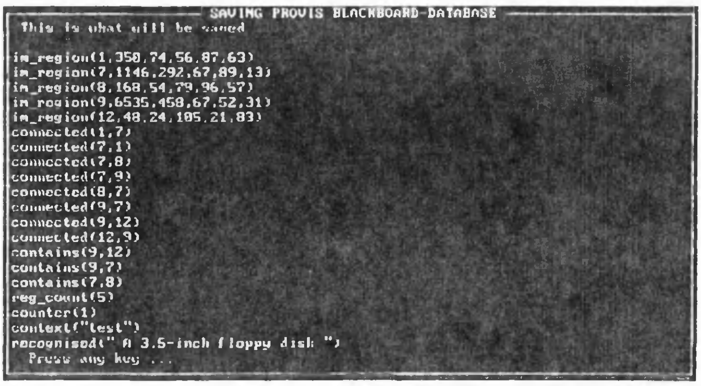


Figure 10.19b The information shown here can normally be used for updating the system, or for adding new entries to the PROVIS system database.

10.10 A Natural Scene Example

This section shows typical data obtained from a test run of PROVIS on a natural outdoor scene. The example image of a street was previously captured as a colour photograph, and subsequently digitized to produce a 128x128-pixel (16 grey-level) video image using the writer's VIDEOC video capture utility mentioned before.

Figure 10.20a shows the original grey-scaled image, and figure 10.20b is the ECM-determined edge-map. Figure 10.20c shows segmentation into semi-symbolic regions, as determined by PROVIS. The scenic mask-form representation as a cartoon-like image is sufficient to enable PROVIS to perform useful symbolic-level processing, calling VPC routines, as necessary. In this particular case, sufficient knowledge was prestored in the database to enable PROVIS to identify the scene. Direct human intervention was needed at two stages:

- To help in sealing-off corners of the edge-map. This is required to prevent featural qualities (colour) from flowing out into the adjacent regions (as in graphics "painting").
- To define the **meaning** of the image symbols to PROVIS: for example, region 11 is always SKY; region 7 is a ROAD; and so on.

Figure 10.21 shows screendumps of the interconnectivity matrix, and the image statistics, on which PROVIS's output decision was based.

To demonstrate simple feedback to the system input, an additional object—a motor-scooter rider—was introduced into the original scene. This new object creates an addition region within the image. The mask of this new object can be manipulated (together with the Gibsonian image) to produce a new image, as shown in figure 10.22.

This particular demonstration is meant to show how Prolog can easily manipulate images in symbolic form. A practical application of this could be a frame-based security surveillance system, where sensitive areas of an original image can be monitored. Indeed, a purely symbolic rendering of the background (empty) scene can assist in identifying new objects entering the camera's visual field. Further symbolic manipulation by a Prolog domain-expert can determine if the object is to be ignored (e.g. if it is an animal) or cause an alarm condition (an intruder is present). Similarly, in a robot workcell, a symbolic interpretation of the workcell's static background can reduce image processing load. As with biological vision, machine vision could concentrate only on the visual tasks in hand. However, further discussion of these interesting applications is beyond the scope of the present work.

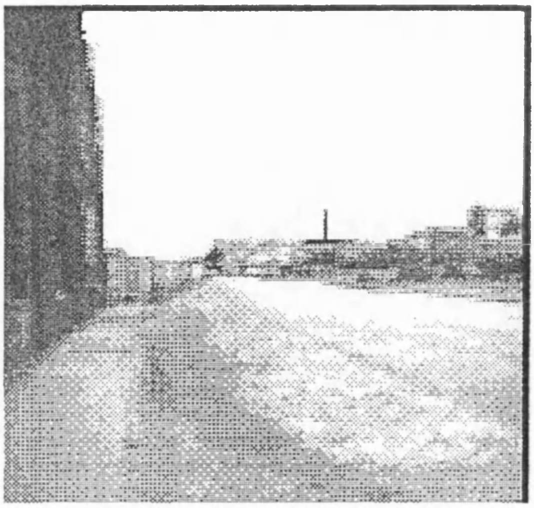


Figure 10.20a A natural street scene image, digitized to 128x128 grey-level pixels.

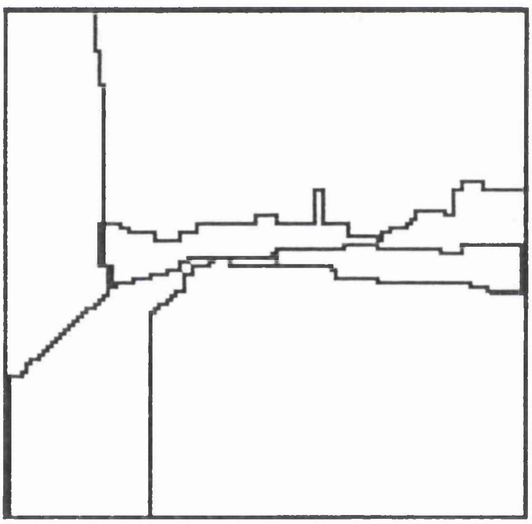


Figure 10.20b This figure shows the clean ECM-generated edge-map obtained from the above image. Clearly, the ECM-HRS process lies at the heart of the PROVIS concept.

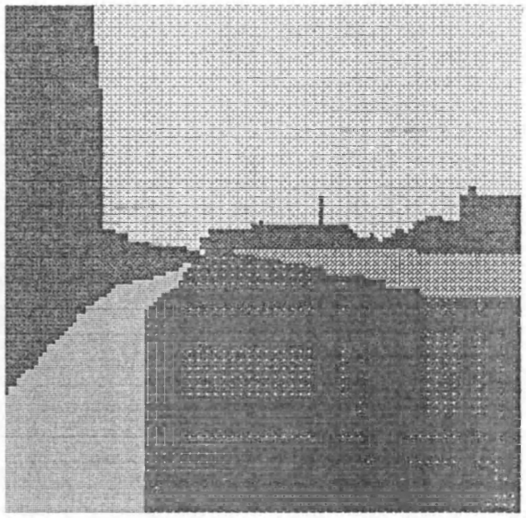
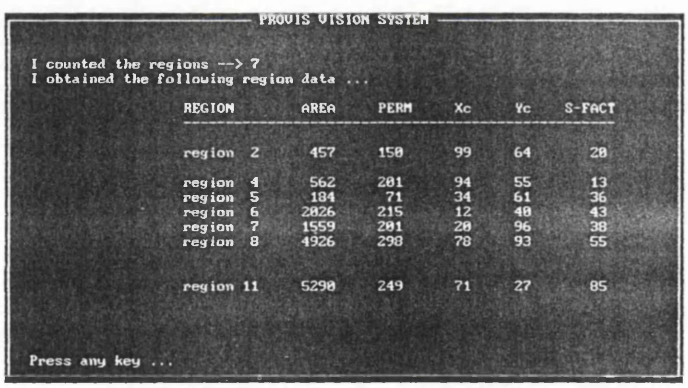


Figure 10.20c The symbolic image regions obtained by PROVIS.



PROVIS VISION SYSTEM

I counted the regions --> 7
I obtained the following region data ...

REGION	AREA	PERM	Xc	Yc	S-FACT
region 2	457	150	99	64	20
region 4	562	201	94	55	13
region 5	184	71	34	61	36
region 6	2026	215	12	40	43
region 7	1559	201	20	96	38
region 8	4926	298	78	93	55
region 11	5290	249	71	27	85

Press any key ...

Figure 10.20d Each image region has a distinct mask and identifier.

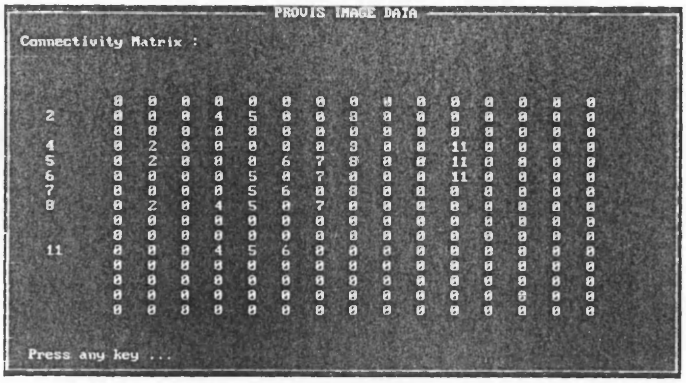


Figure 10.21a The natural street scene's interconnection matrix.

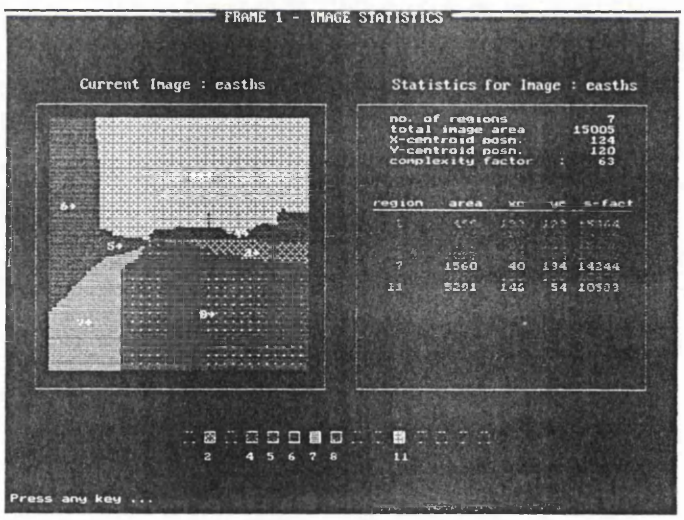


Figure 10.21b The node data relating to the symbolic interpretation of the test image.

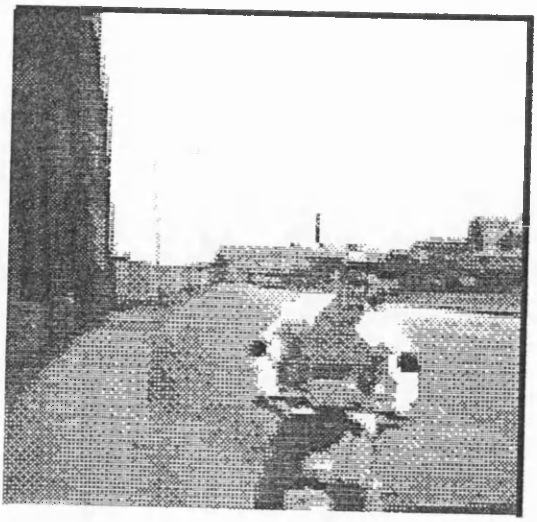


Figure 10.22a An additional object is introduced into the example street scene.

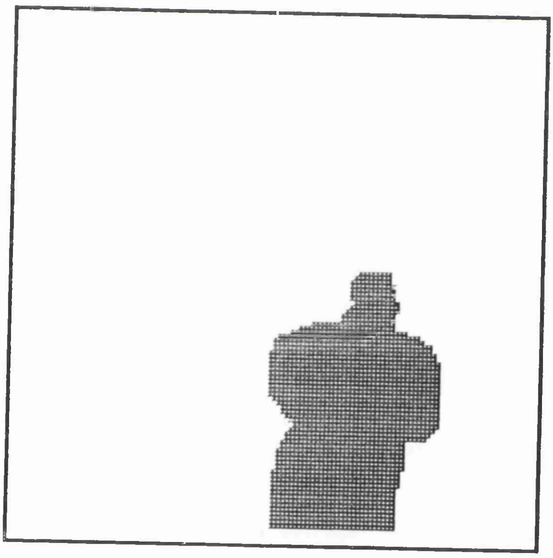


Figure 10.22b The new object creates its own symbolic mask.

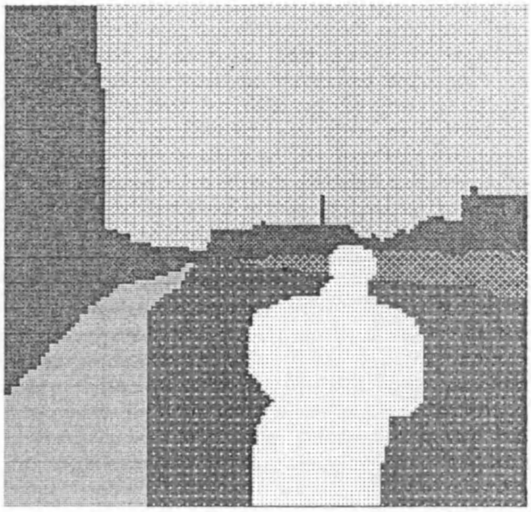


Figure 10.22c The object mask can be positioned anywhere within the scene by PROVIS.

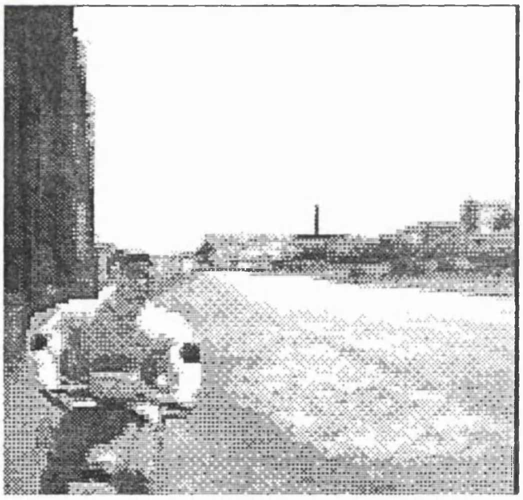


Figure 10.22d A new image can thus be created by manipulating an original image and the new object's symbolic mask. This illustrates how high-level feedback is able to influence low-level image processing. The effect is believed to occur in natural vision though LGN-C feedback interaction—refer to Chapter 7.

10.11 Discussion of Results

In summary, the PROVIS vision system's approach to a model of human vision employs a number of useful concepts, including:

- The use of CAs and rules, wherever possible, in place of conventional (i.e. algorithmic) image processing.
- A multi-featured edge-map (a form of parameter space), to help increase the probability of achieving robust and meaningful edge and boundary mappings.
- A search for preattentive features in images, in the belief that these can better guide subsequent image processing.
- The idea of a "cartoon-form" as an adequate representation to the higher level symbolic image understanding level. This is a semi-symbolic rendering of the original image, and is really a core concept of the PROVIS vision model.
- A concept, and novel mechanisms of direct vision (mediation) to assist in the formation of the semi-symbolic image form. This is believed to be a novel approach to machine vision.
- The use of Prolog in the control and organisation of the higher level AI vision functions (including Prolog-based concentric contexts, fuzzy metric definitions, and so on).

Thus, it is a combination of methods that can lead to a novel and robust model of human visual perception, for use in machine vision.

Although the image resolution in these demonstrations is relatively low it is considered adequate to justify the concepts, because the methods discussed are almost completely scale-independent.

It is impractical to demonstrate (even very simple) examples of every type of image process that can potentially be carried out by CAs, in our PROVIS vision model. The writer's hope is that the selection of results presented here is adequate to demonstrate their potential.

It is seen that CA methods are potentially very powerful in machine vision, particularly in defining processes—such as edge and boundary completion—that could be difficult to specify algorithmically using conventional computer programming languages.

The SOAP concept (active [1]-state cells) is an important feature of CA methods as [1]-state cells—once the CA has stabilised—can be converted to solid object features ([2]-state cells). The requirement for good stability is important, and the preattentive energy metric used here seems to be a sufficient critique for present demonstrator purposes. However, more research may reveal that rule iteration can terminate *before* a [1]-state energy minima is reached. This could, for instance, be due to feedback action by the Prolog-directed IRM. Note that, within this thesis, preattentive energy is regarded as a manifestation of direct vision (measured as a complexity factor, or in terms of some preattentive-feature aspects of an image).

Local-global interaction in CAs has been demonstrated. In particular, the equivalence (or the superiority?) of CA methods with the neural circuit models of Grossberg and Mingolla (1985) has been validated. In addition, CAs have been shown to be capable of dealing with the more complex textured image processes which normally would require complicated algorithmic methods for their solution—for example, as was shown in McCafferty (1990). The prospect of developing an **automatic** parameter selection demon was briefly mentioned.

The importance of the SOAP mechanism in relation to the Gestalt ideas has been stressed. This is one example of how the potential of truly parallel CA-based hardware may be exploited in the future; to yield psychological-based visual effects using purely technical means.

The relevance of illusory vision demonstrations is that they confirm, to some extent, the plausibility of a vision model. That is, they seem to be a necessary proof that a system is behaving in the same way as biological vision. But illusions are a *consequence* of neural circuit interactions; they are not necessarily useful in themselves. They are a result of both local and global interactions, where local edges (for example) are generated in image spatial positions where there are no localised physical contrast differences. These visual effects are forcefully demonstrated by the Kanizsa Square illusion. This phenomenon has been "explained" here on the basis of purely technical and procedural local effects.

At the same time, we know that mental image processes (for example, hallucinations) can account for the apparent perception of phenomena that are not physically present on the retina. This may be regarded as an argument favouring a multi-disciplinary approach to advanced machine perception—especially psychology.

An important feature of the PROVIS model is that the detection of illusory features can optionally be programmed into the system, and may not require the attention of the operator. Thus, these detected features—including all preattentive and edge-evidence clues—are eventually passed to the image recognition stage (the IRM) to be included within an object classification.

Direct vision has been demonstrated to the extent of its usefulness in preattentive parameter generation for CA procedures; and also in the mediation of the ECM-HRS processes for the production of the symbolic image form. However, more research is required in this subject.

Object recognition using Prolog symbolisms has been demonstrated. This is conventional in one sense (although Prolog has been used as a basis for the IRM), but the use of a cartoon was justified earlier in the thesis as a novel representation of the indirect mental processes as in the "Kelvin Way" concept. The image processing part of PROVIS has therefore to be regarded as an engine for the conversion of real noisy images into robust symbolic forms, which can then be interpreted within a knowledge-based domain.

In the COC demonstration, complex actual edge brightness differences are replaced by the simpler stepped-brightness profile based on our human perception of brightness levels. Thus, the IRM is able to benefit from both the image simplification itself, and an accompanying data reduction. It will be recalled from earlier chapters that the COC illusion exemplifies our human brightness perceptual stability: the "red" London Transport bus is always perceived as red, even when seen in dense shadow.

The elegance of CAs and associated lookup table rules has been shown to be comparable with image processing using conventional algorithms. In particular, the SOAP rule has provided a good demonstration of how simpler technical processes (CA lookup tables) can yield results which would require complicated conventional algorithms, and a considerably greater programming effort.

The essential link to Gestalt psychology has been achieved through the CA SOAP mechanism, even though this is a relatively simple procedure. More complex CA rules can be expected to be even more powerful, thus providing stronger links to the understanding of direct vision. This, in effect, means a better comprehension of vision in particular, and of AI issues in general. In the PROVIS demonstrator, some relatively complex psychological phenomena have been reduced to understandable technological processes.

Although not covered in any detail here, the prospect of invoking image EXPECTANCY is encouraged by the PROVIS model. The most likely method would be for Prolog to modify, directly or indirectly, the preattentive array [Jr] — which we use as a parameter space. For example, if PROVIS **expected** a SQUARE, then it could cause the placement of four corner dots as **additional** data within the image array [Jr] with scale determined by some fuzzy process. The four dots would then be processed as in the Kanizsa illusion. Notice that, in the idea of four dots representing a square, we already have defined a pure symbolic representation.

CHAPTER 11

CONCLUSIONS AND FUTURE WORK

This thesis has attempted to relate the writer's interpretation of natural vision to a technological demonstrator. It has very briefly traced the development of machine vision from its early beginnings, as an aspect of conventional image processing and pattern recognition, through the recent neural network revival, to our proposal for a psychology-inspired (Gestaltist) model in which the problem of direct vision is addressed. Direct vision—if indeed this phenomenon can be properly understood and captured in technology—is seen as a crucial additional factor in future machine vision processes; but it is not a replacement for existing successful methods.

Having postulated the existence of two entirely distinct vision systems; and having provided limited evidence in support of this notion, the next step in the present investigation was to devise a vision model. Demonstrating Marr-like image processing is relatively straightforward. A novelty aspect of this project is our use of simulated cellular automata to carry out basic image transformations, which we expect will be done ultimately by massively-parallel hardware systems.

The SOAP rule is an example of how a CA mechanism departs from the methods expected from traditional image processing. The SOAP rule demonstrates the means by which global properties (in this case of Gestaltist phenomena) emerge from purely localised CA interactions. This was demonstrated for a range of common vision algorithms.

The project also demonstrates the use of our modelled CA mechanisms in carrying out Gestalt-inspired image processing. The modelling of physical phenomena is something that CAs have been shown to perform successfully. Hence, the rationale is that once image regions can be physically "field encapsulated" and thereafter perceived as unique and distinct objects, there can be dialogue and symbolic interaction with artificial intelligence.

The present approach goes further than second-generation machine vision—by arguing for the involvement of psychology. It is only by replicating human mental processes that machine perception can hope to emulate human vision. However, this is likely to be a gradual process, in which novel technological techniques can match the discoveries in fundamental neurophysiology and neuroanatomy.

Consider the following quotation by Sullivan, writing in Jones (1982:359), with respect to the use of matched filters within the Roberts (1965) vision model:

> "It may seem that we have a vicious circle. In order to recognize something, we must first know what it is that we are looking at."

Does not this kind of puzzle provide a compelling reason for proposing a novel theory of natural vision—one that includes some notion of the mystical quality that we call direct vision?

11.1 Interpretation of the Results

The results reported in Chapter 10 provide demonstrations and a confirmation of the undernoted features of the PROVIS vision model. It is important that the novelty aspects of this model are emphasised.

- Cellular automata mechanisms, especially the SOAP rule, may provide image-processing functionality at least as good as conventional algorithms. However, CAs seem able to produce results that could be difficult to achieve algorithmically. An example is boundary completion. Unlike the conventional approach, CAs do not appear to need as much *a priori* information about an image. CAs also provide an important link to Gestaltist vision concepts.

- The PROVIS system is susceptible to the same kind of illusions experienced by humans. This may confirm the relevance of our present approach to the modelling of biological processes; that is, the mechanics of local-global neural interaction.

- The PROVIS model exhibits the colour perceptual stability characteristic of the human observer. Chapter 10 showed the usual version of the accepted Craik-O'Brien-Cornsweet (COC) illusion as a demonstration of this.

- "Direct vision" has been demonstrated to the extent of preattentive image processing. This appears to be as far as we can go with the present level of visual understanding. It is suggested that the existence of preattentive vision appears to provide a strong argument in favour of direct vision. Indeed, in some sense it is almost a definition of direct vision.

- The semi-symbolic image concept is a valid mechanism in the understanding of complex images by Prolog. It is also a model of the image data reduction inherent in natural vision. Some human intervention was found to be necessary in order to fully complete (seal-off) the ECM output. This is something that will require more research input since the iconic to symbolic translation process is the core problem of this research.

- Prolog is a sufficient representation of knowledge-based vision processing for our present purposes. It is able to understand and manipulate image regions when these can be represented in full symbolic form.

11.2 Final Conclusions

The following represent the writer's final conclusions, based on both the practical work and results supporting this thesis, and also as a consequence of the wider investigative research. It has to be borne in mind that, as concepts and technology can change so rapidly, much of what is said here can quickly become outdated.

1. Direct vision—if indeed it exists—is a philosophical concept that must somehow be solved practically to enable progress in future machine vision. The image processing paradigms of the past have significant limitations which require new models and new directions for their solution. In particular, there has been no single unifying percept within machine vision—a state of affairs that can be remedied by a theory which includes direct vision. Our model is able to demonstrate only a limited role for the concept.

   The demonstrated phenomenon of preattentive vision by Treisman (1985), and others, is almost a definition of direct vision. It is suggested that this could be a useful starting point for a technology-based approach to direct vision, using the concepts demonstrated in this project.

2. The difficulties increasingly being encountered in machine vision as higher performance is sought are considered to be due to a failure of researchers to understand and appreciate the philosophical nature of direct vision.

3. The acceptance of direct vision need not cause conflict, for the duality postulate allows both theories of vision. Marr is RIGHT in his computational and

memory models. Gibson is RIGHT when he claims that direct vision exists, and that it does not require any memory storage.

4. The PROVIS system has shown that a single model can cope with a range of phenomena, including various kinds of illusions, and the colour perceptual stability problem. This is very different from conventional approaches where a multiplicity of models and concepts are usually required to cover the range of phenomena. This feature is considered to be an important aspect of our model since several kinds of visuo-psychological phenomena, hitherto regarded as extremely difficult subjects, are more easily understood when cast in a technological mould.

   The concept of direct vision has been demonstrated within the PROVIS model—to the extent that it can mediate the ECM-HRS processing. However, the role of direct vision must assume a higher profile, by discovering more convincing applications of the phenomenon in machine perception. The Treisman experiments may provide a useful starting point.

5. The hybrid (top-down plus bottom-up) approach is likely to be the more successful technique in computer vision. This is because AI programming languages—like Prolog—are very good but cannot work efficiently at the basic pixel level, involving low-level image processing. The methods of list processing were found to be too slow for direct video work.

   On the other hand, conventional procedural languages are much better for this—but they, too, are part of a constantly changing scenario. An example of newer approaches is the use of object-oriented programming (OOP). This could offer some new insights to image processing by procedural languages.

6. CAs appear to be able to perform vision processing at least as efficiently as ANNs—because of the "rule-equivalence" in algorithmic processing. CA-based image processing algorithms have demonstrated a superiority over ANNs in certain kinds of visual processing. The iterated SOAP rule (or VOTE rule) can eliminate image noise, and produce a desired Gestalt grouping mechanism. CAs also seem able to implement almost any kind of conventional image processing algorithm—including some that might be difficult to implement using a conventional computer programming language. The CA approach is a

distinct advantage of our PROVIS vision model.

CAs have been shown to be similar in concept to digital filters (Chapter 9, and Appendix I). If this link can be strengthened, CAs could acquire a solid theoretical understanding derived from digital filter theory. This may be likened to the way that artificial neural network studies were able to benefit from the analogous properties of atomic magnetic dipoles, or *spins* (c.f. the Ising system of spin-glasses in the physics literature). Such a theoretical link could result in a more predictable CA model.

7. The Edge Constraint Map (ECM) is a vitally important feature of our model, as it is responsible for the determination of object boundaries within a context. Indeed, this is the single most important aspect of the PROVIS vision model. However, one important limitation is the need for human intervention at some stage to ensure that the edge-map is robust. Hence our goal of complete autonomy in the system has not yet been achieved.

8. The main limitation of the PC environment is memory capacity. Even with a reduced 128x128 image format, the RAM storage space requirement for multi-level arrays can be substantial. But this restriction does not affect the principle of our model, which can be scaled, or recast in an alternative processing platform. This is a distinct advantage of our modular approach.

The bottom line of the approach advocated here is that crude symbolic mask representations of complex natural images are an adequate method of interpretation—provided that we agree that direct vision exists to take care of the perceived detail. Two processes are at work.

We cannot pretend that our PROVIS model is able—at a stroke—to solve the long-standing problems of AI-based vision. The writer hopes that this thesis can serve as a contribution to the AI debate, and that it may inspire new directions in the thinking and approaches to advanced machine vision.

11.3 Future Work

Further work, directed towards developing a more robust machine vision paradigm within the concept of direct vision, could include any or all of the undernoted objectives. These subjects could be tackled individually, or else they could form part of some larger project. The proposals include the following activities:

1. Continuing with the development of hypotheses which can explain the nature of direct vision, in an attempt to involve the concept at all levels of technological vision. This could lead to AI insights of a general philosophical nature.

2. Continuing to maintain close contact with the fundamentals of biological neuroscience, as reported in the literature, in an on-going attempt to correlate the relevant neuroanatomical and neurophysiological discoveries with technical proposals for machine vision.

3. Continuing with the development of cellular automata means and methods, in the belief that CA methods can better compete with their more complex ANN equivalents. Such studies should also be directed towards the development of novel CA algorithms for machine vision, including advanced forms of massively-parallel computing. Investigating **adaptive** mechanisms in CAs.

4. Pursuing the concept of massive-parallelism in all of its hardware and software aspects, in furtherance of the goals mentioned above. This could initially include the Transputer model and its developments, but gradually moving into newer fields as processor technology advances.

5. Investigating the implications of direct perception within the wider sphere of AI. An obvious example is the direct perception of sound (music, etc.), but it can be speculated that there are equivalent sensorineural information processes.

6. Investigating dynamical models. For example, a vision system that is able to influence its working environment by physically panning and tilting its video input camera to bring potentially interesting objects within the frame of view. This would be of particular value in robot vision, and in novel approaches to image sequence analysis for security applications.

7. Investigating the informational aspects of both images and CA image processing. This could form part of a general movement towards a fuller understanding of the abstract qualities of information theory—as advocated in Stonier (1990).

8. Investigating practical applications (including vision concepts) resulting from our research. For example, a CA model could be used as a fire-spread predicter, using soap films. Detector and sensor points could act as reference images within the modelled space, thereby affecting the fire-spread.

LATEX-V2.09-PhD920912-TMcI

THESIS APPENDICES

LIST OF APPENDIX FIGURES AND TABLES

Figure C.1	Representing a digital image array	226
Figure C.2	A MONADIC operation on a digital image array	226
Figure C.3	A local window operation on a digital image array	227
Figure C.4	A large-area window operation	227
Figure G.1	Relationship between Turbo Prolog and Turbo C	243
Figure H.1	Basic neural-pulse signal features	249
Figure I.1	Functional form of rate of change of neuron firing-rate	253
Figure I.2	Local and global features of biological receptive fields	256
Figure I.3	Discrete-time representation of simple digital filter	260
Figure J.1	An elementary 3-region image and its image-tree	264
Figure J.2	The image-tree for three separated regions	264
Figure J.3	A multiplicity of similar image-tree forms	265
Figure J.4	An example of a more complex image-tree	265
Figure J.5	Illustrating "Attneave's Cat"	267
Figure J.6	Lights attached to the joints of the human frame	267
Figure J.7	Illustrating the concept of a Parameter Space	269
Table D.1	The Flynn Taxonomy	232

APPENDIX A

VPC PRIMITIVES AND PROLOG CALLS

The following is the list of the principal VPC image processing primitives and Prolog calls. These functions are all callable directly from Turbo Prolog. This list is periodically upgraded, and emphasis is placed on developing CA-style algorithms wherever possible. Most of the mnemonics require one or more calling arguments (parameters) and, as described previously, can be suffixed by _0, _1, etc.

VPC procedures can potentially work with any image resolution, but have been specifically coded by the writer to process the 64x64 and 128x128 block-pixel images intended for CA experimental applications. Section A.2 lists the current Prolog calling commands.

A.1 VPC Primitives

```
File and I/O

cpy          % copy image from array [Ar] to specified array
grb          % grab the current image in digitizer card
lda          % load an image from disk into array [Ar]
sta          % store a specified image array to disk

Grey-Level Image Processing

avg          % average intensities over image
avn          % average over local neighbours
cen          % compute image centroid
com          % compute image complexity
con          % compute connectivity of regions
crc          % convert to real colour-scale
his          % produce an intensity histogram
```

```
inv           % invert the specified image
msk           % output oriented-mask data
qnt           % quantise  the specified image
seg           % segment   the specified image
thr           % threshold the specified image
```

Grey-Level Filtering

```
bkp           % black-point filter
hpf           % high-pass filter
lnb           % replace each intensity with maximum value
lpf           % low-pass filter
red           % edge detector - Roberts
sed           % edge detector - Sobel
wkf           % white-point filter
```

Binary Image Processing

```
bed           % binary edge detect
ccc           % draw circumcircle around blob
cen           % compute centroid of a region
chu           % compute the convex hull
cmp           % complete an outlined shape
coa           % compute centre-of-area
con           % compute connectivity between regions
cop           % compute centroid of sparse-points region
cwp           % count white pixels
cxw           % expand white regions
dib           % compute dimensions of a blob
dis           % compute dimension of sparse-points blob
flh           % fill holes in image
iso           % isolate a blob or region
mid           % compute mid-point skeleton
```

```
mdl          % compute medial line transform
otl          % determine blob outline
sev          % compute edge evidence array (EEA)
shp          % compute blob shape factor
shr          % shrink entire image
shw          % shrink white regions
thn          % thin outline pixels
```

General Functions

```
cla          % clear VPC-CA system tables
clt          % clear VPC-CA lookup tables
set          % reset VPC-CA lookup tables
```

CA Rule Definitions (Reset lookup tables)

```
rul_cln      % CLEANUP  rule
rul_com      % COMPLETE rule
rul_otl      % OUTLINE  rule
rul_set      % RESET    rule
rul_sp0      % SOAP-0   rule
rul_sp1      % SOAP-1   rule
rul_sp2      % SOAP-2   rule
```

A.2 Prolog Functions

General Functions[1]

```
list_append(L1,L2,L3)
list_concat(L1,L2)
list_create(L)
list_del(L1,L2,L3))
list_insert(L1,L2,L3)
list_length(L)
list_lists(L1,L2)
list_member(X,L)
list_print(L)
list_sort1(L1,L2)
list_sort2(L1,L2)
```

Project Specific Functions

```
camd
CAM_sta

get_area(I)
get_imlist(L)
get_inf(I,DOM)
get_mat(I,I,I)
get_rct(C)
get_reg(R)
get_dta(T)
get_iml(L)
get_sta(I)
ini_graphics(I)
```

[1] These are expressed in terms of Edinburgh syntax Prolog. Turbo Prolog is a *typed* language variant, so there are usually additional constraints on the generality of predicate forms.

```
plt_bin
plt_pix(I,I,S)
plt_sta
```

APPENDIX B

BRIEF NOTES ON PROLOG

This appendix considers briefly some historical aspects of the Prolog programming language for artificial intelligence (AI). For further detail reference should be made to the many texts which have appeared in recent years. The standard work is by Clocksin and Mellish (1984). Good introductory material is provided in Bratko (1989), Stobo (1989), and Lloyd (1984). Batchelor (1991) demonstrates Prolog as a means of organising and calling low-level image processing routines in C code. Schildt (1987) provides useful Prolog code for more advanced topics.

B.1 Prolog Language Development

Prolog stands for PROgramming in LOGic—a proposal that emerged in the early 1970s to use logic as a programming language. The early developers of practical Prolog include Robert Kowalski at Edinburgh University (on the theoretical side), Maarten van Emden at Edinburgh (the experimental demonstrators), and Alain Colmeraurer at Marseille (implementation). The present popularity of Prolog is largely due to David Warren's efficient implementation at Edinburgh in the mid-1970s.

There are also some controversial views that have accompanied the practical introduction of Prolog as a programming language, mainly due to what some consider to be "mathematical impurities" of the language. We shall not discuss the controversial issues here.

Prolog achieved a considerable boost at the beginning of the 1980s when the Japanese Ministry of Research and Technology placed the language at the centre of its Fifth-Generation computer initiative. Although that 10-year programme has been much less of a success than originally hoped, Prolog remains the preferred language of researchers at ICOT (the Japanese institute for new-generation computer technology based in Tokyo). Japanese industrialists apparently still prefer the traditional LISP for many practical applications. In addition, Prolog was slow to gain acceptance in the USA where LISP remains dominant.

The European response to what was perceived as a Japanese threat was the setting up of the UK's Alvey Directorate, and the initiation of the ESPRIT programme. The latter continues with renewed vigour, and results from European research into Prolog applications and parallel computers have generally been superior to those ultimately achieved by the Japanese.

B.2 The Historical Roots of Prolog

Logic programming has its roots in the work on automated theorem proving and artificial intelligence. For well over fifty years there have been research efforts to construct automated deduction systems. One can even go back to the time of Liebnitz (1646–1716) who dreamed of a "calculator" which could solve philosophical problems by purely mechanical means.

The first serious work on automated deduction based on formal logic dates back to the ideas of Herbrand in 1930. This work was followed by others, and in 1965 Robinson introduced the Resolution Principle. The Resolution Principle is a sound and complete logical inference rule system, and represented an important breakthrough towards automated deduction systems. From an algorithmic perspective, the Resolution Principle and its underlying unification operation are very well suited to implementation on modern digital computers. All that is required is the repeated application of the unification operation on a set of clauses that have been put into a standard form, such as the Gentzen Normal Form:

$$A_1, \ldots, A_n \leftarrow B_1, \ldots, B_m \tag{B.1}$$

where $A_1, \ldots, B_m$ are literals that can contain atoms, variable, and structures as arguments.

The literals $A_1, \ldots, A_n$ are called positive literals, with the commas representing the disjunction. The literals $B_1, \ldots, B_m$ are the negative literals of the clause, and the commas here denote conjunction. For more detailed information, and a good introduction to formal aspects of logic programming, see Lloyd (1984).

The informal semantics of the above states "for each assignment of each variable, if $B_1 \wedge \ldots \wedge B_m$ is true, then one of the A_i $(0 < i \leq n)$ must be true."

Early attempts to use Robinson's Resolution Principle and unification algorithm as the "inference engine" of a logic-based computational model were not very successful—despite the algorithmic simplicity. The main problem was the inability to subject the Resolution Principle to a natural control mechanism—needed to restrict the search space in a meaningful way.

It was Kowalski who recognised the procedural aspect of a restricted class of logical theories—namely, Horn clauses. A Horn clause has the following form:

$$A \leftarrow B_1, \ldots, B_n \tag{B.2}$$

That is, Horn clauses have at most one positive literal, A, called the clause **head** (or consequent). The conjunction of the literals $B_1, \ldots, B_m$ (the antecedent) is called the clause **body**. In the context of logical programming, a Horn clause is also called a program clause. Of course, a program clause might have an empty body: that is, it consists only of a clause head. A clause of this form is called a *unit* clause. A logic program is simply a finite collection of program clauses.

According to this definition, it is legitimate for there to be more than one clause with the same clause head literal. Intuitively, one can regard program clauses as logical "rules" to solve the problem (or goal) statement. Program clauses with the same head literal are called **procedures**, and represent alternative ways of solving a problem statement. The *unit* clauses represent the "facts" of the relevant problem domain.

The declarative semantics of a logic program provide a logical specification of the desired output from a program, and a goal statement. As such, the logic program is completely implementation independent. Kowalski's idea was that Horn clauses of the form:

$$A \leftarrow B_1, B_2, \ldots, B_n \quad \{n \geq 0\} \tag{B.3}$$

can be given a procedural interpretation which is additional to their semantics. To solve a goal statement, A, execute procedure A, which consists of a conjunction of subgoals B_i; to solve the subproblems B_i execute the respective procedures B_i. When all subgoals have been solved, then the original goal A has been solved too.

It is this *procedural* interpretation that distinguishes logic programming from mechanical theorem proving. It allows one to perform deduction from the logical specifications of computable functions in a goal-directed manner.

In order to formalise the procedural interpretation, one needs to specify the search and computation rules. In general, mathematical ideals have to be sacrificed for the sake of efficiency and ease of implementation when considering practical logic programming systems. The first practical programming language based on Horn clauses was Prolog (PROgramming in LOGic). As noted in Section B.1 above, this was developed by Colmeraurer and his research Group at the University of Marseille, during 1972–73.

Conventional Prolog executes all subgoals sequentially from left to right, and considers candidate clauses sequentially in the textual order defined by the program. Recent research has sought to implement parallel forms of Prolog, both as a parallel language implementation and as "parallelised" versions of what is really sequential Prolog.

Since the declarative semantics of Prolog does not impose an execution order on the goals—or search order on the clauses—it is immediately obvious that logic programming provides an inherent parallelism. It was this observation that led to Prolog being selected by the Japanese Government as the preferred programming language for their Fifth-Generation parallel computer initiative. The stated goals of that initiative were massive parallelism embodying logic programming principles.

B.3 Turbo Prolog

Turbo Prolog—a popular dialect of Prolog produced originally by PDC (the Prolog Development Center in the USA), and marketed for several years by Borland Inc. before being reassigned to PDC—is a strongly **typed** Prolog. This requires that all data types and prototype clauses must be declared before they can be used in the Prolog program proper. This requirement has received a somewhat mixed response. Most traditional (that is, so-called "Edinburgh Syntax") Prolog programmers generally hate Turbo Prolog, but newcomers to the Prolog language seem better able to accept this imposition.

Although strong typing helps to clarify, document, and structure a Prolog program, it can lead to restrictions on the wider concepts of logic programming. This may or may not be important: it really depends on the target application. Turbo Prolog can implement a meta-Prolog which provides the looser form of conventional programming, but the cost is in terms of speed. Again, the importance of the speed factor depends on the application.

It is becoming common to compile Turbo Prolog code and link it with object code produced by other programming languages—such as Pascal or C. This hybrid approach (as used in the present project) allows the development of a very powerful computing environment for control and simulation systems, and modelling. A range of supporting calls and predicates is supplied by PDC for this purpose, but great care has to be taken in the development of hybrid systems—particularly with regard to system global memory allocation and de-allocation. Appendix G discusses these problems in detail.

APPENDIX C

BASIC IMAGE TRANSFORMATIONS

This appendix briefly considers some aspects of conventional machine vision techniques in relation to the present study. Full details of the topics mentioned are to be found in the many excellent texts on pattern recognition, image processing, and computer vision, including many recent books and papers. A review of current UK research can be found in BMVC-91 (Proceedings of the British Machine Vision Conference held at Glasgow University during September 24–26, 1991).

An excellent account of the mathematics of image transformation and image understanding is presented in the text by Kanatani (1990).

C.1 Conventional Image Processes

Figure C.1 shows the usual computer representation of an image as an $m \times n$ array of integer data elements which may be digitized to either two discrete levels (binary) or multiple-level (grey-scale). There also exists a close relationship between colour and multi-spectral images and their grey-level representation for computational purposes.

Let i and j denote two integers where

$$1 \leq i \leq m \qquad \text{and} \qquad 1 \leq j \leq n$$

In addition, let $p(i,j)$ denote an integer function such that

$$0 \leq p(i,j) \leq W \tag{C.1}$$

The array P, where

$$P = \begin{bmatrix} p(1,1), & p(1,2), & \cdots\cdots, & p(1,n) \\ p(2,1), & p(2,2), & \cdots\cdots, & p(2,n) \\ \vdots & \vdots & \cdots\cdots, & \vdots \\ p(m,1), & p(m,2), & \cdots\cdots, & p(m,n) \end{bmatrix} \tag{C.2}$$

is called a *digital image*, and an address (i, j) defines a position within P called a *picture element* or *pixel*. The integer elements in P denote intensity values within small rectangular (often square) samples of an optical image. The total $m \times n$ elements defines the image *spatial resolution*. The intensity values range from black (0) to white (W).

Conventionally, there are several classes of image processing functions that can be carried out:

1. Array–array mappings.

2. Array–scalar and array–vector mappings.

3. Vector–array and scalar–array mappings.

We cannot discuss these here, except to remark that (1) is the most useful for CA modelling. Usually, we wish to perform a mapping

$$[input\ image] \longrightarrow [output\ image] \tag{C.3}$$

the assumption being that such a process will yield meaningful and relevant image transformations in the context of the problem domain. Group (2) are data reduction processes, used mainly for deriving image parameters and measurements. Group (3) processes usually relate to CAD and computer graphics applications.

Figure C.2 depicts a *monadic* transformation in which a single element in array array $A(i, j)$ is transformed to a single element in array $B(i, j)$. In general image transformation, A will be an input image and B will be an output (transformed) image.

Figure C.3 depicts a *local* 3x3 window (or template) operator in which nine pixels (for a kernel operation) or eight pixels (for a nearest neighbourhood operation) are combined to yield a single new output element in array B. In simulated CAs this operation is equivalent to a CA rule, as discussed in Chapter 4. The B array then contains the updated cell in the same relative position as the A array. If a one-iteration delay is used, then the A cell can update itself via the B cell. Sometimes much larger input arrays are used to define *global* operations, as in figure C.4.

In *dyadic* operations, elements from two input arrays A and B combine to define the transformation of the output array C.

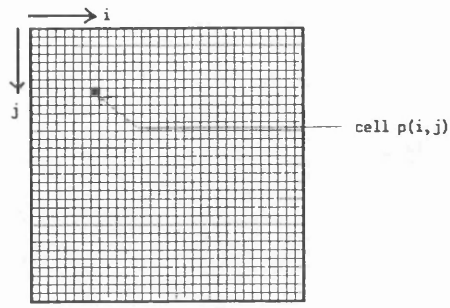


Figure C.1 A digital image consisting of an array of m × n pixels. The pixel in the ith row and the jth column has an intensity function value p(i,j).

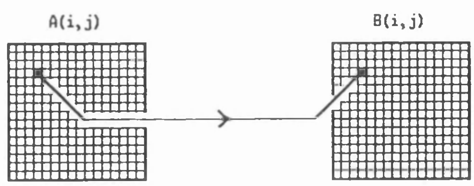


Figure C.2 A MONADIC or point-by-point operation. The (i,j)th pixel in the input image is transformed to the corresponding address pixel in the output image. That is,

$$A(i,j) \rightarrow B(i,j)$$

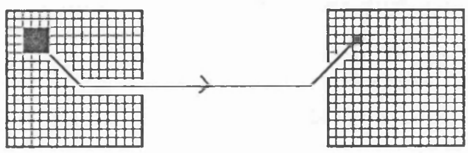


Figure C.3 A local window. In this case, the intensities of nine pixels arranged in a 3x3 window are processed to produce an output. This is also the mechanism used in the software simulation of CAs. See Chapters 4, 8 and 9.

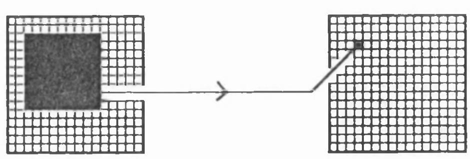


Figure C.4 A much larger neighbourhood of pixels can contribute to a definition of a global image transformation. This is useful in the emulation of ANN global properties by basically localised CA rules.

The array window used in kernel or neighbourhood operations is shown below. It is assumed here that i indexes horizontally and j indexes vertically.

(i-1, j+1)	(i+0, j+1)	(i+1, j+1)
(i-1, j+0)	(i+0, j+0)	(i+1, j+0)
(i-1, j-1)	(i+0, j-1)	(i+1, j-1)

The window can be abbreviated using the following simpler notation:

A	B	C
D	E	F
G	H	I

This is also seen to relate to the "compass" directions defined for square CA localised neighbourhoods, and for the image morphological operations mentioned previously in Chapter 4.

C.2 Convolution

Convolution is a fundamental neighbourhood operation used in filtering and related image transforming operations. Using a 3x3 image mask, similar to the kernel used in CAs, we can write the 2D image spatial convolution equation as

$$R_{i,j} = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} I_{k,l} h_{i-k,j-l} \tag{C.4}$$

where $R_{i,j}$ is the resultant pixel. It is seen from eqn. C.4 that each pixel is taken in turn from the neighbourhood of the image I and multiplied by one of the elements of the convolution mask h. There is clearly a relationship between the local neighbourhood concept of convolution, and the image processing carried out by cellular automata with defined rules. This is something that needs further investigation, in relation to the development of a general theory of machine vision involving CAs.

APPENDIX D

CELLULAR AUTOMATA AND PARALLEL COMPUTERS

This appendix considers the recent interest in parallel computing, and briefly traces the revival of cellular automata.

D.1 A Historical Perspective

Parallel processing is best defined by contrasting it with normal serial processing. During his 1947 Moore School lectures, John von Neumann and his associates propounded a basic design, or architecture, for electronic computers in which a single central processing unit (CPU) is connected to a single core of memory. In the von Neumann design, the processor fetches instructions from a program (programme) stored in memory, then fetches operands for those instructions from a different area of the memory, performs a calculation, and finally writes the results back into the memory. The John von Neumann architecture remains popular for the following reasons:

- it is conceptually simple—only one operation at a time
- it is relatively simple to program
- it is simpler to build than other architectures
- it is more economical than other architectures
- it is more reliable than other architectures

John von Neumann and his colleagues were well aware of the many limitations of his serial computer paradigm, as will be discussed later in this thesis. However, the hardware technology and the programming languages available at that time were not capable of creating or supporting parallel computing concepts.

In the late 1960s, the Digital Equipment Corporation (DEC) introduced their famous PDP series of minicomputers. This meant that the total dominance of the very large (and very expensive) "mainframes" was broken, and powerful computing facilities became available for much smaller commercial businesses, and—most importantly—educational establishments. The early 1970s saw the introduction of the first of the so-called "vector" computers—the legendary Cray-1—which was a development of DEC's earlier CDC 6600 Series mainframes.

In the vectored design concept, the arithmetic logic unit (ALU) circuitry which carries out the fast arithmetical operations is divided into manageable stages. Vectorisation fueled the demand for even greater computational power—particularly in educational and scientific research establishments—for simulation and modelling.

In the late 1970s, the following four things conspired to make parallel processing a practical reality:

1. Very Large Scale Integration (VLSI) semiconductor technology.

2. Advances in general programming techniques (flags, DMA, etc.).

3. Parallel prototype construction and evaluation.

4. Further vectorisation with the development of the Cray Series.

By the end of the 1980s, practical parallel computers, such as ICL's Distributed Array Processor (DAP) were appearing for applications use, rather than as purely novelty machines to be studied in themselves. In the academic research laboratories work began seriously to explore the possibilities of parallelism, because it was realised that performance returns and expectations for existing and projected microelectronics technology (including VLSI) were not encouraging.

Today, the number of parallel computer systems, and the number of companies developing and manufacturing them, are both rapidly increasing. Proponents of parallelism cite two main arguments in favour of parallel computers.

The first is economic: parallel computers tend to be much more cost effective than their serial counterparts, primarily due to the economics of scale and VLSI technology.

The second argument is based upon a fundamental physical law. Since information cannot travel faster than the speed of light, the only method of speeding up computers is by reducing the distance that signals have to travel, or by moving many more bits of information at once. Attempts to reduce distance within microcircuits are plagued by the limitations of quantum mechanics. See, for example, the arguments in Barker (1990), Randall et al. (1989), and the references contained therein. This suggests that speeding-up computers is only feasible by moving many more bits of information simultaneously—that is, by using **parallelism.**

Two kinds of parallelism need to be distinguished: fine-grained or dense parallelism, and multi-tasking. Dense parallelism requires a very large number of processing elements operating simultaneously, as in image processing. Multi-tasking in computing means that two or more distinct processes, or programs, can run concurrently. Multi-tasking can exist simultaneously with fine-grained parallelism to provide a very powerful computing platform.

The questions that remain to be resolved before mass-scale parallel computers become more widely applicable include the following:

- how many distinct processors should be used?
- how should they be designed, organised, and interconnected?
- how should they be programmed?

One promising—if more distant—hardware prospect is that of dense parallelism organised as very large arrays of individual processing elements with local memory and interconnectivity—that is, cellular automata (CAs) and artificial neural networks (ANNs).

Before leaving the topic of parallel computers, it is worth noting that a useful degree of parallelism comes "free" when implementing certain hardware systems by means of memory-mapped techniques. This is discussed further in Appendix F.

D.2 Computer Architecture—Flynn's Taxonomy

Michael J. Flynn's 1972 taxonomy of computer architectures is still the most generally accepted method of classifying computers. Flynn divided computers according to whether they use single or multiple "streams" of data, and single or multiple "streams" of instructions.

This classification is summarised in table D.1 below:

TABLE D.1: THE FLYNN TAXONOMY

Class	Single Instruction	Multiple Instruction
Single Data (SD)	SISD (von Neumann)	MISD
Multiple Data (MD)	SIMD	MIMD

An SISD computer carries out one instruction or datum at a time. This is the conventional (serial) von Neumann architecture. An MISD machine applies several instructions to each item of data it fetches. So far as the writer is aware, no examples of MISD architecture computers have yet been built.

The SIMD architecture uses many processors to simultaneously execute the same instruction, but on different data. These individual SIMD processors are very often identical in function, and may be called processing elements (PEs). Finally, MIMD computers are usually regarded as an evolutionary step forward from SISD designs. An MIMD contains several independent (and usually equipowerful) processors, each of which executes its own individual program.

Thus, in the terminology of the previous section, SIMD computers are examples of dense parallelism, while MIMD machines are multi-tasking. See, for instance, Hockney and Jesshope (1988). In the present work, cellular automata use fine-grained (dense) parallelism, and so would be classed as SIMD architectures. In the context of the overall vision model described in this thesis, an MIMD architecture classification could also be appropriate. Indeed, both classifications apply.

D.3 A Brief History of CAs

Cellular automata (CAs) were invented in the late 1940s in the USA by Stanislaw Ulam (1909–1984) and John von Neumann (1903–1957). One can say that the "cellular" part comes from Ulam, while the "automata" description was derived from the early work of von Neumann.

Ulam was primarily a mathematician. He was the inventor of the now well-known Monte Carlo simulation technique, and made contributions to mathematical analysis and number theory. He was also the co-inventor of the hydrogen bomb!

Von Neumann was a wide-ranging mathematician, and worked on set theory, quantum mechanics, economics, and game theory. He was also responsible, with Herman Goldstine, for the logical design of the first electronic computers. His 1948 conference paper "The General and Logical Theory of Automata," and his subsequent course of lectures in 1949, introduced the notion of designing self-reproducing machines. These were the first descriptions of modern CA ideas. Using techniques of mathematical logic, von Neumann was able to deduce that his DNA-like concepts of self-replication were feasible—or at least they did not seem impossible. However, it was not until he met Ulam in 1948 that the crucial mechanistic link to "cellular" modelling was eventually formed.

Ulam's contribution to CA was to force von Neumann to think in terms of an "idealised space of cells" that could hold finite-state numbers which then represented the different aspects of von Neumann's self-replicating theory. Ulam's "cells" replaced von Neumann's reservoir of mechanical components, from which the self-reproducing machine would assemble a clone of itself. Thus, Ulam's proposal was an infinite lattice, or a graph of points, each with a finite number of connections to defined patterns of its "neighbours." Each such point in the lattice is then capable of assuming a finite number of "states," where the states have some representative meaning in the system. It is the use of the word "finite" in these terms which leads to the alternative description of CA as particular forms of "finite automata."

As discussed in Chapter 4, a CA mechanism is iterative: the states of the local neighbourhood of any lattice point at a time t(n) induce, in a defined manner, the new state of that point at the next cycle time t(n+1). Thus, problems of synchronisation have to be considered. By 1952, John von Neumann had completed his model of the self-reproducing "cellular automaton" which, in this case, contained as many as 29 states to represent von Neumann's self-replicative concept of a reservoir of clone parts.

Nothing much seems to have happened in CAs until 1970, by which time John von Neumann had been dead for thirteen years. However, in 1970, a US mathematician named John Conway produced his now classic "Game of Life" which was based on CA principles. Conway had apparently selected his Game of Life cellular transformation "rules" on a trial-and-error basis—an approach which is still found in CAs to this day. (It is this seemingly non-deterministic method of deducing "rules" that still attracts considerable criticism of CAs from some quarters). Conway originally was attempting to find a CA rule that would enable simple patterns to grow to a very large size—but without growing towards infinity. This is a form of controlled cell mechanism, such as is found in normal living cells. Conway's "game" resurrected interest and enthusiasm in cellular automata mechanics, and the related mathematical ideas of chaos and fractal images.

During the 1970s, a number of people at the Massachusetts Institute of Technology (MIT) began seriously to study the Life Game, among them Edward Fredkin, William Gosper, Gerard Vichniac, Tommaso Toffoli, and Norman Margolus. In 1980, Fredkin, Vichniac, Toffoli, and Margolus formed the Information Mechanics Group at MIT. Fredkin's ideas were extended to the concept of CAs as "information packets," and the tenet of cellular mechanics which could suggest new forms of computers was conceived.

By the early 1980s, Toffoli and Margolus were demonstrating CAs as a novel modelling medium especially well suited the simulation of many kinds of phenomena in mathematics and physics. See the text by Toffoli and Margolus (1987). One very practical outcome of the Toffoli and Margolus partnership was the production of the so-named "CAM-6," a hardware CA adapter board that can be plugged into any IBM PC or generic computer to aid CA experimentation and demonstrations. This CA-based adapter board is extremely fast, but unfortunately it is also very expensive—because of its high specialism and the consequent low-volume production runs.

Another prominent CA researcher of this period (the early 1980s) was Stephen Wolfram, whose interest in wave particle physics led to his research into the mechanics of initially one-dimensional (1D) cellular automata. See Wolfram (1983,

1984) for a useful review of basic CA theory and associated mechanics.

By the mid-1980s, cellular automata research was becoming well established in the US and in a number of academic institutions in Europe. A relatively recent European venture was the Winter School at Les Houches, France, during the week of February 21-28, 1989. See the edited Proceedings by Manneville et al. (1989). The purpose of the Les Houches meeting was to discuss the use of CAs as models of complex physical systems, but the conference publications provide a source of basic CA theoretical papers, each citing further broad references.

In the UK, Michael Duff and his Group continued with the study of cellular logic machines—work which had been initiated as long ago as the late 1960s. Cellular logic principles are similar in concept to cellular automata, but a main difference is that the former are binary or two-state, whereas the latter are usually multistate. However, this difference is not as significant as might seem, because Michael Duff's cellular logic machines are multi-layered, such that any equivalent CA multistate process could be implemented. Duff's research over the years has resulted in a generic series of CLIP (Cellular Logic Image Processing) machines, since the Group's interests lay mainly in image processing. However, this interest apparently did not extend to the more advanced concepts of machine perception. Further details of the CLIP research Group's work can be found in Preston and Duff (1984), Duff and Fountain (1986), and Wood (1988).

The Duff-led research Group's CLIP series of image processors were mainly intended to interface with powerful minicomputers, but here again recent advances in IBM PC hardware design would allow a plug-in adapter card version of CLIP to be produced, in the same way as the CAM-6 coprocessor of Toffoli and Margolus. However, the writer is not aware of any such CLIP adapter card having been produced for sale outside Imperial College. The spectacular increase in speed and power of desktop microcomputers, and the availability of parallel computers as hosted subsystems (such as the Inmos-marketed Transputer platform), means that efficient and suitably fast CAs can be achieved by software simulation.

D.4 CAs: Some Applications and Possibilities

The following is a list of some current and potential CA applications:

- commercial graphics design, CAD
- artistic and decorative design
- synthetic 2D and 3D image generation
- synthetic 2D and 3D image processing
- medical imaging, CAT
- machine vision and robot perception
- artificial intelligence (AI)
- novel computer hardware and algorithms
- solving optimisation problems of all kinds
- modelling or simulating physical systems
- finite-element analysis
- fractal geometry and advanced mathematics

and, of course, others yet to be discovered.

As mentioned above, and elsewhere in the thesis text, a mechanism which is almost identical to CAs is that of cellular logic, pioneered Duff and associates in the UK, in collaboration with Preston et al. in the US. Their CLIP system has been used as an image processing engine in recent vision research, e.g. Ng et al. (1991). The explosive field of natural and artificial neural network research also shares much in common with the fundamental ideas and processes of CAs. Hence it can be expected that many advances in basic neural net theory will be applicable also to CAs—particularly where CAs are used in biological cell emulation.

APPENDIX E

CELLULAR AUTOMATA AND ARTIFICIAL NEURAL NETWORKS

This appendix briefly considers the differences between traditional cellular automata (CA) and artificial neural network (ANN) models. It should be stressed, however, that advances in these fields are occurring regularly, and so attempts to establish rigid boundaries will be fruitless.

The future development of both technologies is receptive to advances in device fabrication methods, theoretical and experimental studies, and new algorithm development: hence the writer's preference for the term "Artificial Neural Systems" (ANS) which can embrace both of these important disciplines.

E.1 CAs vs. ANNs

The field of cellular automata shares with neural networks a common interest in the computational properties of systems comprising large numbers of interacting processing elements (PEs). The main difference is that neural networks are more general and are biologically inspired and motivated.

The following is a listing of some specific differences:

1. Each node in a CA normally computes an identical transfer, or state transformation, function, while ANN cells are usually considered to be independent processors (although a common ANN transfer function is usually involved, for example, a **sigmoid**).

2. CA nodes are connected to a few (usually < 10) nearest neighbours, whereas ANN cells may have hundreds or possibly thousands of local interconnections.

3. CA nodes typically take on less than 20 discrete states, while ANN cells output a pulse analogue signal. (Natural neural networks are considered to output a pulse-compression analogue signal—similar to FM radio carrier-modulation).

4. CA nodes usually operate synchronously, and are run by a global clock. ANNs, especially electronic circuit analogues, usually run asynchronously. An exception is when ANNs are simulated on either conventional or parallel computers—in which case the computer's internal clock signal is the synchroniser.

5. The output state of a CA node on a clock cycle depends on the node's current and previous states, and on the current state of the nearest neighbour nodes. Also, unlike ANNs, a CA's update function is not merely a simple linear summation, but can be any relevant logical or nonlinear function.

6. Because a CA's update function is fixed (by a CA "rule") and is normally identical for every CA node or site, "memory" can only be held in the states of the CA nodes. In ANNs, memory is a function of both the interconnectivity mapping between cells and the synaptic weights, or plasticity, at the inputs (the dendrites)—that is, the memory capacity of ANNs is massively distributed.

Interest in cellular automata has been mainly theoretic in the past. They can be shown to be equivalent to Turing machines; they can be self-replicating. As discussed in Chapter 4 of this thesis, they are capable of forming complex patterns and thus exhibiting very complex "behaviour" given only a few simple "rules" and specific starting or initial conditions. Conway's game of "Life" is usually cited as the archetypical CA demonstrator, and fractal geometry shares a common theoretical background.

Research into CAs has recently been revived with the development of high-speed and parallel computers, and special-purpose hardware. Much recent work in this area has focussed on the ability of CAs to perform calculations equivalent to the modelling of complex physical systems, such as fluid flow, and electrical and magnetic fields. See, for example, Manneville et al. (1989). The recent interest in fractal geometry is likely to benefit CA studies—and artificial vision.

Many ANNs share characteristics with CAs. The Trion model of cortical information processing (Shaw et al., 1986) is one example of a hybrid system that has characteristics of both cellular automata and neural networks. It is like a CA in that the state of each unit is discrete and ternary. However, each unit represents a group of "neurons" such as the orientation column in the primary visual cortex, and the three states correspond to outputs above, at, or below the background firing rate. The update function of the Trion is CA-like, and the oscillatory and overall response is analogous to the stable states in a Hopfield artificial neural network memory.

APPENDIX F

THE CONCEPT OF MEMORY MAPPING

F.1 Memory Mapping

Practical computers of all types and sizes require read/write memory, commonly referred to as random access memory (RAM). Conventionally, in the so-called von Neumann architecture, the RAM contains the stored instructions and data for the computer to work on. The computer is "hard-wired" (or pre-programmed) to automatically fetch and execute instructions in a sequential manner, manipulating data in memory as it does so. The distinction between what constitutes instructions, and what is data, is context-dependent. Normally, computer instructions reside in different reserved areas of RAM than data, and therefore cannot be accidentally overwritten by the user, or a programmer.

The "stored program concept" means that a substantial portion of a conventional computer consists of random access memory, and hence fast and efficient **semi-parallel** (bytewise or wordwise) mechanisms are built-in to automatically scan the RAM areas—known as the memory address space. However, it generally does not matter to a computer whether this address space contains actual read/write memory (RAM), or fixed read-only memory (ROM). In fact, it does not matter what kind of mechanism resides within the address space, provided that any such device conforms to the logical signal and timing specifications of the system's memory address bus. This means that certain memory addresses, when generated automatically, or as a consequence of a stored computer program, may not actually address any memory cells. Instead, they may communicate with external devices—which can appear to the system as logical memory addresses. This is referred to as "memory mapping" and is a feature designed into every practical computer.

F.2 Sensors Within Memory Address Space

If, instead of an array of memory cells, a computer's address space consists of an array of photosites, then the computer will "read" the output of the photosites as though they were memory cells. And if the photosites are "memory mapped" into the areas normally reserved for the computer's video display refresh RAM, then a real-time image will be displayed **continuously** on the screen, corresponding to excitation of the photosites. This means that the computer "sees" the external world without needing any additional explicit scanning code program. Of course, in this specific example, the analogue outputs of the photosite array must be constrained to the logical signal levels appropriate to the computer memory hardware.

Similarly, if the memory address space contains elements corresponding to, say, the states of the cells of a binary cellular automata, then a computer system would be able to "read" or "see" the states of this array, without explicit programming. In the case of solid modelling CAs—a proposal by the writer—this could provide useful practical benefits, as well as giving an insight into the "mental imaging" concepts involved.

The significance of the memory mapping concept is that a considerable and very worthwhile degree of parallelism comes "free" — as an inherent design characteristic of conventional computers. In many cases at present (1992), this will provide 32-bit parallelism at no additional cost in design or programming complexity. The usefulness of memory mapping is limited only by the imagination, and should not be overlooked by researchers. Indeed, there is a sense in which such memory-mapped systems are similar to the concept of "direct" vision, as discussed in Chapter 5 and elsewhere in this thesis.

A final point on computers. The concepts mentioned here can apply even if, sometime in the future, "computers" take on unconventional roles and associated novel interpretations, such as could happen with dedicated vision or AI devices. Examples which come to mind include CAs and the unusual RAM-based artificial neural networks of Aleksander et al. — reported in Aleksander (1983) and elsewhere. Computers need not "compute" in the conventional sense.

APPENDIX G

INTERFACING TURBO PROLOG

The following notes show the method of interfacing Turbo Prolog with Borland's Turbo C. The same principles apply when linking Turbo Prolog with any other language, with the restriction that the linked languages must support 32-bit pointers ("large memory model") and be capable of producing Microsoft-compatible standard OBJ file format.

G.1 General Concepts

Figure G.1 shows the method of dual-language implementation. The host system **must** be Turbo Prolog, as Prolog is responsible for the global memory (heap) management, and the control of other processes.

Turbo C implements the VPC image processing procedures—see Appendix A for a full listing of the VPC commands. These involve image array manipulation, graphics, and other processes which are much easier to code in a procedural language. Turbo Prolog carries out the "smart" IRM functions much more efficiently than procedural Turbo C.

Procedures defined in C are callable from Prolog, using a convention established by Borland. The simple parameter types (integer, string, real, etc.) are passed to C functions via the STACK. Returned values are via memory reference. Compound objects must have space allocated on Prolog's global stack. This is achieved by using a Prolog library function "ALLOC_GSTACK". As we shall see, this causes problems with Turbo C's "underbar generation" option. The writer's solution was to change the name of this function from "alloc_gstack" to "_allocgstack" using a binary text editor on PROLOG.LIB. All of the user-defined VPC Turbo C functions can then be called directly from Prolog with leading underscores (_).

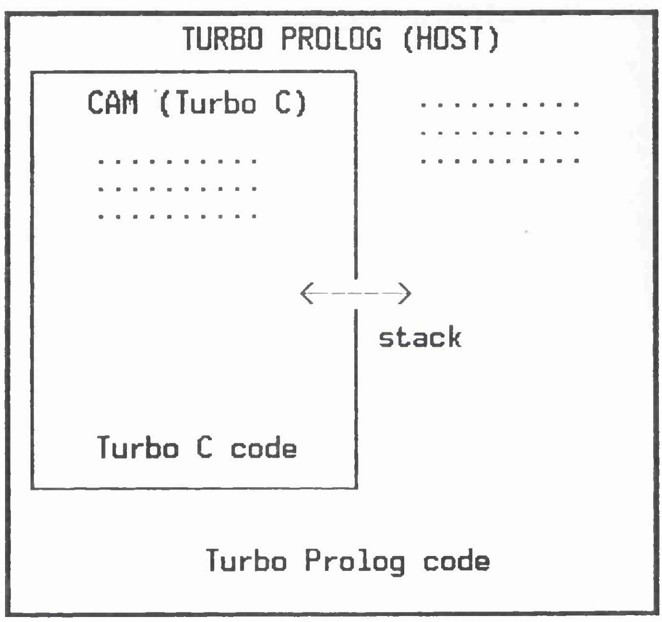


Figure G.1 Showing the relationship between the Turbo Prolog host environment and Turbo C (VPC) image processing subroutines.

In the present project, it is necessary to initialize the VPC system by calling the program by name from Prolog ("_vpc"). Thereafter, all of the separate C functions within VPC can be called from Turbo Prolog directly, i.e. without needing to call VPC (main) by its Turbo C name. (But see also the initialisation section below).

Special care has to be taken with memory which may have been MALLOCed within Turbo C modules. If this memory is not FREEed after any calls invoking C's MALLOC, there will eventually be a crash with MS-DOS reporting "memory allocation failure." This is only a problem in the present project because the PROVIS vision demonstrator system is fully re-entrant.

G.2 Turbo Prolog Compatibility Problems

Turbo Prolog has been discontinued by Borland, and has reverted to its former title of PDC Prolog (PDC stands for Prolog Development Center, in the USA). The most recent version of Turbo Prolog was version 2.0.; while at the time of writing (mid-1992) the latest version of PDC Prolog is 3.2. PDC Prolog is also at least four times as expensive as when marketed under the Borland "Turbo" banner.

Turbo Prolog (and by implication, PDC Prolog) is not compatible with any version of Turbo Pascal, although one is usually led to believe otherwise. This is because Turbo Pascal cannot produce the Microsoft standard OBJ file format for linking with other languages. The Turbo Pascal "UNIT" format is unique, but the internal structural details of Turbo units have never been published by Borland.

Turbo Prolog is likewise incompatible with most Microsoft languages, because of the latter's use of undocumented DOS system calls. Another problem is that Borland's OBJ linker, TLINK, cannot link any Microsoft language with Turbo OBJ files. This means that it can be expensive to depart from the Borland language product range. [2]

The effect of all this is that the best approach is to link Turbo C OBJ code with the Turbo Prolog OBJ modules. Development work can be carried out very quickly and easily in Turbo Pascal, the latter being translated to Turbo C. This is not very difficult. The idea is that "procedural" routines (e.g. for image processing) are

[2] Since completing this work, the writer has learned that STONY BROOK Pascal+ is fully Turbo Pascal 6.0 compatible, and can produce standard OBJ modules.

coded in Turbo C, while the "intelligent" (AI) functions are programmed in Turbo Prolog. This dual-language combination provides a very powerful programming environment on the IBM PC generic platform, and is the approach used in the present project. The only serious problem is memory demand.

G.3 Linking Prolog with C OBJ Modules

The minimum generic command line for Prolog-C linking is:

$$TLINK\ init\ \ < p.objs > < c.objs > < p.sym >,\ \ , Prolog + CL$$

(Note that with Turbo C version 2.0, and Turbo Prolog version 2.0 or PDC Prolog, there is now no need for the former "CPINIT.OBJ" file).

If stand-alone BGI graphics are to be linked-in (as opposed to being loaded automatically from disk at runtime) then one must use:

$$TLINK\ init\ \ < p.objs > < c.objs > < p.sym >,\ \ , BGI + Prolog + CL$$

In addition, the header:

bgidriver "_EGAVGA_driver_far"

must appear at the beginning of the Prolog source file.

Given the above generic forms, and assuming that the example source "VPC.C" has been compiled to "VPC.OBJ", and "PROVIS.PRO" has compiled to "PROVIS.OBJ" the following command line will link the project files:

$$TLINK\ provis\ vpc\ provis,\ \ , [BGI] + Prolog + CL$$

This results in the (default) executable "PROVIS.EXE". (The .sym file is generated automatically by the Prolog compiler). In the above line, "Prolog" is actually PROLOG.LIB, and CL is CL.LIB (the LARGE C library model). These must be pathed to the current directory.

For convenience, the linking process has been automated in a batch file called PCLINK.BAT, developed by the writer. This is desirable as the order of linking is important. The MALLOC, FREE and ALLOC_GSTACK functions needed for the

C modules are in the PROLOG.LIB—which must always appear as the first item in the linking list.

G.4 System Initialisation Call

The entire VPC program normally is called at least once to initialise global variables, declare structures, and so on. This global call need not achieve any image processing results as these are obtained in the subsequent calls to individual C-coded VPC functions. Initialisation is simply achieved by the call:

_init_vpc

(or similar) from the Turbo Prolog host.

"VPC.C" happens to be the name of the Turbo C program in which the various C routines are contained. It compiles to VPC.OBJ, which is the OBJ code linked with Prolog (as described immediately above). But there is no reason, of course, why the global program name could not have been "CINIT.C" or any other appropriate title.

The handling of "main()" also needs to be considered. One can include a main() within the Turbo C program if desired. This is what was done in the VPC.C demonstrator. Alternatively, the various VPC functions can be compiled without a main() call, provided that initialisation of the globals is handled properly. The advantage of retaining main() is that VPC primitives can be developed and tested separately from the calling Turbo Prolog host.

G.5 Underbar Generation in C Code

By default, Turbo C generates underbars, or underscores, (_) for all calls made to the Turbo C library functions. In the present project, this includes perhaps 70% of the code, because every graphics routine and almost every useful C function is a library call, e.g. line().

Borland stores these calls in the appropriate libraries with leading underscores, e.g. _line(), _outtextxy(), _getch(), _fopen(). This is why automatic underbar generation is the default. For some reason, the function ALLOC_GSTACK, which is a C function defined within the Turbo Prolog Library PROLOG.LIB, has not been stored with a leading underscore. This is why the Borland Turbo C and Turbo Prolog manuals require that automatic underbar generation be turned off. However, in practice, this requires that the programmer manually prefixes every C library function call with a leading underscore. Not only is this a very tedious procedure, but problems occur with certain C functions which have alternative definitions, for example fopen().

To avoid these problems, it is best to edit the PROLOG.LIB, and change the code text of the single reference to "ALLOC_GSTACK" to the underscored version "_ALLOCGSTACK" —or something similar.

G.6 Calling Convention for VPC Functions

Image-processing primitives defined in Turbo C are called from Turbo Prolog by merely prefixing the C function name by an underscore (_), provided that the C functions have been declared in Turbo Prolog as "global" predicates (see Borland guides).

Prolog, including even the fast compiled Turbo Prolog dialect, is far too slow for raw image processing work. The "PROVIS" demonstrator can redraw the original image on the Prolog side—but it takes several minutes, compared with a few seconds in C. Accordingly, Prolog must be reserved for any higher level image processing functions, such as inference, control and direction. With a vision processing command language written in C modules, there should be no need for Prolog to carry out basic image processing. Even a "small" 64x64 image list places a huge processing burden on Turbo Prolog.

One method of overcoming the speed problem developed by the writer is the use of list-of-lists to represent an image—rather than a single 4096 element list. Each 64-pixel scanned row is represented by its own list, and the entire image consists of 64 line-scan lists. The speed increase obtained in image list-processing is significant. It seems best, however, to avoid raw image processing in Prolog, unless it is desired only to reproduce a full image without processing.

APPENDIX H

NEURAL SIGNAL COMPRESSION

The following notes, due to Duggan (1991), illustrate how extremely simple mechanisms can provide a robust and nonsaturating mechanism for artificial neural analogue signal implementation and transmission.

H.1 Introduction

In a highly complex signal processing and control system, such as a biological nervous system, many different kinds of signal have to be processed rapidly. In the case of an animal, there will be sensory signals, memory information signals, interneuronal signals, dynamic control signals, and muscle (effector) actuation signals. In order to minimise transmission and propagation delay in such systems it is necessary to adopt a standard and reliable signal encoding scheme throughout. This is necessary to avoid conversion between different signal types. The type of coding must be analogue, have unlimited resolution, and must withstand transmission over axons of up to several metres in length. In addition, the signals must not saturate, and they should be as immune from interference and noise as possible. There are probably many possible solutions to these requirements, but Nature has evolved what is probably the simplest scheme possible: a pulse duty-ratio analogue system. This is described below.

H.2 The Signal Form

The signal form is a pulse waveform in which information is carried in the ratio of pulse width to pulse period. This signal is illustrated in figure H.1. below:

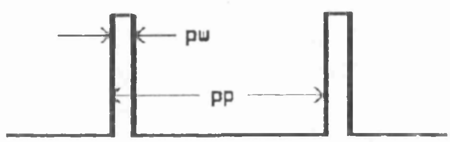


Figure H.1 Basic neural signal pulse features.

It can be seen that the pulse signal is always positive-going (Duggan calls this "posinomic") with a dynamic range of zero (no pulses) to unity (consecutive pulses with no gaps between). The numerical value of the information carried is not, however, simply a linear function of the duty ratio. The nonlinear function is explained below.

Firstly, however, note that from Figure H.1 the linear duty ratio, p, is given by the ratio of pulse width, pw, to pulse period, pp:

$$p = pw/pp \tag{H.1}$$

In biological nervous systems the pulse width, pw, is nominally fixed, and the pulse period, pp, changes. The mechanism is highly tolerant, however, to pulse width variation. Most processing elements in the system (neurons) provide nominally a pulse width of approximately one millisecond: but this can (and does) vary considerably between cell types.

In real (i.e. biological) neurons the period of "switch off" is usually called the **refractory period**, and may be several times longer than the output pulse width. This does not normally affect the compression function, other than to limit the maximum signal level (measured as the output duty ratio) to a fraction of unity.

H.3 The Processing Elements

In order to provide a nonlinear compression function that guarantees freedom from saturation and yet provides global stability, Nature uses a very simple device: during the generation of each output pulse the neuron's inputs are effectively switched off. In this way, as the output increases, there is less and less time between incoming pulses for the inputs to influence the cell body. If the output pulse ratio signal from a cell is "p", and the sum of all inputs is "q", then the net input to the cell is given by the product of the time between pulses $(1 - p)$ and the input q. Thus:

$$p = (1 - p) \times q \tag{H.2}$$

which transposes to:

$$p = q/(1 + q) \tag{H.3}$$

This is called the "posinomic compression" or "natural compression" function. It can be seen that as the input, q, approaches infinity, the output, p, approaches unity. Thus signal saturation cannot occur, and the gain across the element (from input to output) is always less than or equal to unity:

$$gain \;=\; output/input \;=\; p/q \;=\; 1/(1 + q) \tag{H.4}$$

H.4 Discussion

The fact that biological nervous systems successfully employ the natural compression function is evidence of its effectiveness. The method is simple, yet fulfils the three essential requirements for complex adaptive neural signal processing, namely:

1. Nonlinear transfer function for adaptive learning.

2. Signal compression to avoid saturation.

3. Global stability in mass shunting feedback loops.

Computer simulations have shown that the following conclusions are valid for this simple coding mechanism:

1. Since each compression element (neural cell) has a gain of less than unity, any required number of units can be used in a feedback multiplicative loop without loss of system stability.

2. Although each element is essentially nonlinear, the design and analysis of neural control systems is extremely simple. This follows from the approximation to unity gain near zero signal level. This means that approximate loop dynamics can easily be calculated, and that there is no potential instability problem.

3. The compression function is compatible with, and provides a natural extension to, posinomic theory. See Duggan (1984).

APPENDIX I

MATHEMATICAL NOTES

The following notes derive some useful mathematical results relevant to the current vision model—supporting the material of Chapter 9.

I.1 Neural Cell Processes

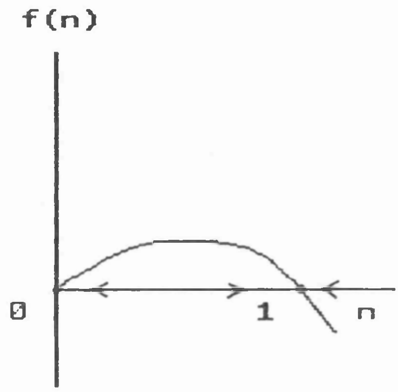
This section considers the derivation of a model of a neural cell's **receptive field**, an important aspect in the development of both the GM and CA models of long-range interactive effects. This leads to the typical "mexican hat" shape for the receptive field, which can be realised in both the ON-CENTRE, OFF-SURROUND and the alternative inverted OFF-CENTRE, ON-SURROUND forms.

Neurons can fire spontaneously; that is, they show a sudden burst of activity. They can also fire repeatedly at a constant rate (Murray, 1989). Whether or not a cell fires depends on its autonomous firing rate, and the excitatory and inhibitory input it receives from neighbouring neurons. This input can be from other than the cell's nearest neighbours; that is, long-range interaction. Such input can be positive, which induces cell activity, or negative, which inhibits it. These effects account for the classical mexican hat profile of a neuron's **receptive field**.

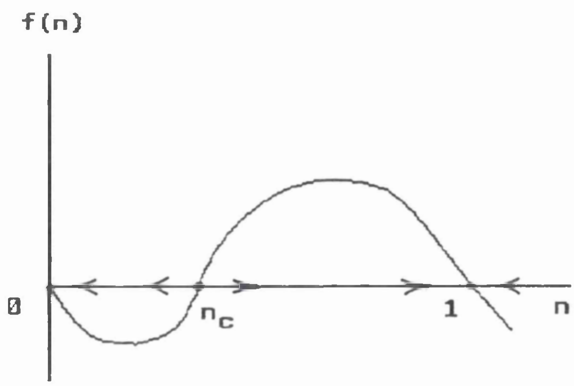
Consider the one-dimensional (1D) case in which the neural cells are functions only of x and t. Let $n(x,t)$ be the firing rate of the cells. In the absence of any neighbourhood influences we assume the cells can be in either a quiescent state, or fire autonomously at a uniform rate—which we normalise to unity. If the cells' firing rate is perturbed, we assume it evolves according to

$$\frac{dn}{dt} = f(n) \tag{I.1}$$

where $f(n)$ has zeros at $n = 0$ and $n = 1$ (the steady states) with a functional form shown in figure I.1a. Here the only steady state is $n = 1$. $f(n)$ might exhibit threshold bistable kinetics whereby there is an unstable threshold steady state, n_c say, such that if $n > n_c$, $n \to 1$, and if $n < n_c$, $n \to 0$. A typical bistable form for $f(n)$ is shown in figure I.1b.



(a)



(b)

Figure I.1 Illustrating the functional form for the rate of change of neuron firing rate f(n).

(a) Situation having a single stable positive steady firing rate.
(b) Situation having bistable threshold kinetics.

The kinetics of the firing dynamics in eqn. (I.1) determines the subsequent firing rate, given an initial rate n. In the case of bistable kinetics, if $n(x,0) = 0$, input from neighbouring cells could temporarily raise the firing rate to $n > n_c$, in which case the cells could eventually fire at constant rate $n = 1$.

Let us now include spatial variation, and incorporate the effect on the firing rate of cells at position x, of neighbourhood cells at a position x'. We assume that the effect of close neighbours is greater than that from more distant ones; the spatial variation can be incorporated in a weighting function w which is a function of $|x - x'|$.

We must integrate the effect of all neighbouring cells on the firing rate, and we model this by a convolution integral involving an **influence kernel**. Specifically, take $f(n)$ as in figure I.1a where the positive steady state is $n = 1$. We now modify eqn. (I.1) so that if $w > 0$ there is a positive contribution to the firing rate from neighbours for $n > 1$, and a negative one if $n < 1$.

Thus, a simple model of the mechanism is the integro-differential equation

$$\begin{aligned}\frac{\partial n}{\partial t} &= f(n) + \int_D w(x - x')[n(x', t) - 1]\, dx' \\ &= f(n) + w * (n - 1)\end{aligned} \tag{I.2}$$

where D is the spatial domain over which the influence kernel $w(x)$ is defined, and $*$ denotes the convolution defined by this equation. The form (I.1) ensures that $n = 1$ is a solution. The influence kernel, w, is symmetric, and is specified as

$$w(|x - x'|) \;=\; w(x - x') \;=\; w(x' - x) \tag{I.3}$$

A non-symmetrical kernel could arise as the result of some kind of superimposed gradient. To ensure that n is always non-negative we must ensure that $n_t > 0$ for small n. This requires

$$\int_D w(|x - x'|)\, dx' \;<\; 0 \tag{I.4}$$

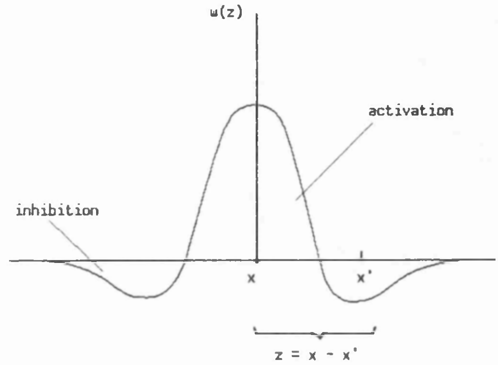
Since $f(n) \to 0$ with n, eqn. (I.2) reduces to

$$\frac{\partial n}{\partial t} \sim -\int_D w(|x - x'|)\, dx' \;>\; 0 \tag{I.5}$$

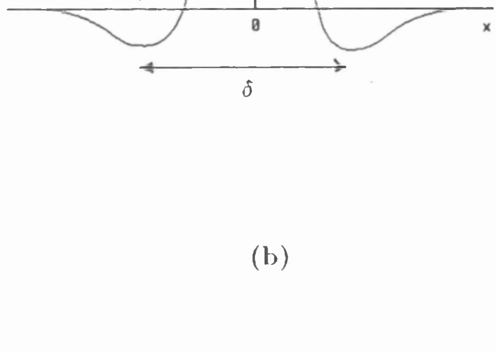
This condition is satisfied with practical kernels, the form being the "mexican hat" shape where cells have a short-range activation effect and a long-range inhibition effect. This is the more common ON-CENTRE, OFF-SURROUND model, incorporated in the kernel of figure I.2a.

On the infinite domain, $w(z)$ is a continuous symmetrical function of the variable $z = x - x'$ such that

$$w(z) \rightarrow 0 \quad as \quad |z| \rightarrow \infty; \qquad z = x - x'$$



(a)



(b)

Figure I.2 Relating to local and global features of biological receptive fields.

(a) A typical exponential kernel.
(b) The parameters affecting the shape of a receptive field.

It is possible to quantify the features of the kernels produced by eqn. (I.2). Consider the infinite domain and linearise (I.2) about the positive steady state, $n = 1$ by setting

$$u = n - 1, \qquad |u| \ll 1$$

$$\Rightarrow u_t = -au + \int_{-\infty}^{\infty} w(|x - x'|)\, u(x', t)\, dx' \qquad a = |f'(1)| \tag{I.6}$$

We can now look for solutions of the form

$$u(x,t) \propto exp[\lambda t + ikx] \tag{I.7}$$

where k is the wavenumber and λ the growth factor. Substituting into eqn. (I.6), setting $z = x - x'$ in the integral, and cancelling $exp[\lambda t + ikx]$ gives λ as a function of k, that is, the dispersion relation from eqn. (I.6) as

$$\lambda = -a + \int_{-\infty}^{\infty} w(z)\, exp[ikz]\, dz \quad = \quad -a + W(k) \tag{I.8}$$

where $W(k)$ is simply the Fourier transform of the kernel $w(z)$.

A simple symmetric kernel of the shape shown in figure I.2a can, for example, be constructed from a combination of exponentials of the form

$$exp[-bx^2], \qquad b > 0 \tag{I.9}$$

which has a Fourier transform

$$\int_{-\infty}^{\infty} exp[-bx^2 + ikx]\, dx \quad = \quad \left(\frac{\pi}{b}\right)^{\frac{1}{2}} exp\left[\frac{-k^2}{4b}\right] \tag{I.10}$$

By the appropriate use of parameters, it is possible to define and control the shape of the exponential kernel. The taller and narrower the defining function is in x-space, the shorter and broader is its transform in k-space. The typical "mexican hat" can be obtained as the difference of two separate kernels of different shape (as in the so-called DOG—difference of Gaussians—neuron receptive fields). This is possible by the use of appropriate parameters to define the receptive field: usually A for the height and δ for the width, as shown in figure I.2b.

A positive short-range activation with long-range inhibition produces the typical ON-CENTRE, OFF-SURROUND form, while short-range inhibition with long-range activation gives the alternative form of the upside-down OFF-CENTRE, ON-SURROUND type of receptive field.

I.2 A Simple Digital Filter

In order to realize digital filter theory it is necessary to convert continuously varying **analogue** signals into representative discrete-time signal forms. Such signals are Nyquist sampled-signals, often—but not always correctly—called **digital** signals.

Figure I.3a shows the usual circuit of a simple low-pass analogue filter. This filter is represented approximately by the system diagram as shown in figure I.3b.

From figure I.3a we can write

$$y + RC\frac{dy}{dt} = x \tag{I.11}$$

from which we obtain the following *difference equation*

$$\frac{dy}{dt} \simeq \frac{y(nT) - y[(n-1)T]}{T}$$

Substituting this into the *differential equation* (I.11) gives

$$y(nT) + \frac{RC}{T}\, y(nT) - \frac{RC}{T}\, y[(n-1)T] = x(nT)$$

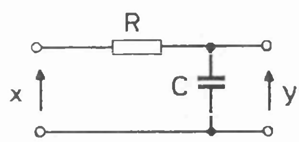
This can be solved explicitly for the response $y(nT)$ as follows

$$y(nT) = \frac{1}{1 + RC/T}\, x(nT) + \frac{RC/T}{1 + RC/T}\, y[(n-1)T]$$

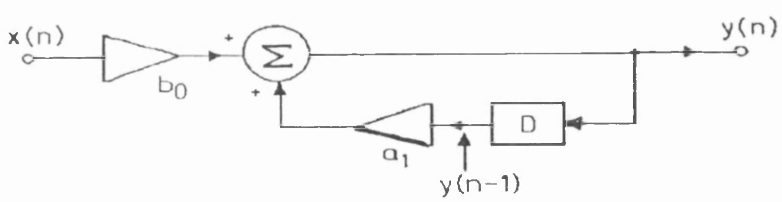
from which we can obtain, by setting $T = 1$ and rewriting

$$y(n) = b_0 x(n) + a_1 y(n-1) \tag{I.12}$$

The above is a **recursion** formula. Each calculated value $y(n)$ provides a time-delayed value $y(n-1)$ for the next calculation in the sequence. We see that this is in fact realised by the feedback system circuit depicted in figure I.3b.



(a)



(b)

Figure I.3 Derivation of the discrete-time (digital) representation of an RC low-pass filter.

(a) The circuit diagram of an analogue RC low-pass filter.
(b) The discrete-time (feedback) representation of the RC filter's iteration formula.

APPENDIX J

GRAPHICAL REPRESENTATIONS

This appendix describes some useful graphical representation methods that were were not developed in the main thesis text. The mechanisms discussed here are associated with ideas of using minimum storage for image data—a subject of ongoing research in several disciplines.

J.1 Simple Image-Tree Graphs

This section discusses a hierarchical image-tree representation as used in the present work. This simplified graphical method enables the Prolog-based Image Recognition Module (the IRM) to classify objects or scenic images that have been processed through the ECM-HRS module. It is emphasised that this method, like many of the topics described in this thesis, is subject to ongoing development and modification. The approach described has significant limitations, but is believed to be adequate for the purposes of the thesis project.

The idea of hierarchical image-tree graphs is discussed in a number of recent sources, particularly Ballard and Brown (1982), Arbib (1989), and Rich (1983). [3] Such graphs can be used to represent different kinds of visual information, to an arbitrary degree of complexity.

The graphs used here represent processed image regions—and their relationships to each other. This information is compiled within the IRM system, from an image histogram scanned by relevant VPC routines called from the Prolog host. This code translates C structures and image records into Prolog functors. These image regional structures are then imported into the IRM system proper, and are subsequently assembled into the appropriate Prolog database.

[3] The writer understands that a second edition of Elaine Rich's excellent and near-standard textbook *Artificial Intelligence* has recently been published.

The VPC routine produces image region records containing the following information:

- The allocated ID number of each distinct image region.
- The normalised AREA count of each region.
- The normalised CENTROID (or C-of-G) of each region.
- The normalised X–coordinate of each region.
- The normalised Y–coordinate of each region.
- The normalised SHAPE-FACTOR of each region.

Further image region metrics can be added at any time. The above listed parameters are considered sufficient for present purposes.

The basic concept is that image-trees and graphs are a hierarchical representation of the information contained in an image. This latter statement can, however, have multiple interpretations and important implications—particularly in view of the current awareness of the entropy aspects of information. See, for example, the paper by Chang (1984), and the thought-provoking monograph by Stonier (1990).

The lowest level of the hierarchy represents the smallest elements present in an image—that is, **pixels**. The second lowest level is a meaningful grouping of pixels into regions. This, of course, is a nontrivial matter, and is the *raison d'être* of intelligent image segmentation, region growing, and object recognition—including the arguments developed in the present thesis.

The concept of regions here includes discrete objects, to which we can normally attach a meaning. However, the question of how many distinct objects are contained in a given image is not easy to answer. There is generally no requirement to attach image labels to individual pixels, and so the pixel tree level—that is, the "leaves" of the image tree—can be normally be set at level 0.

Once region or object grouping has been achieved, a graphical tree can be constructed showing the relationship between the regions at level 1 and upwards. As mentioned, image pixels, once grouped, cease to have a useful role in graphical methods. Regions are represented in graphs as NODES, and the relationships between the regions are shown as lines or ARCS. Both the nodes and the arcs can be labelled. For example, a node might contain the parameters represented in the above coded record structures, with arcs denoting regional interconnectivity.

Figure J.1 shows an elementary 3-region image and its corresponding image-tree representation. If there is only one distinct object within an image then there is a single root node at the highest level; otherwise the graph contains a number of nodes at the highest graph level. [4] Using a mesh of nodes and arcs, it is possible to devise a graphical scheme in which vertical arcs link region nodes on different levels, while horizontal arcs indicate directly connected regions on the same level. Figure J.2 illustrates the representational difference between three separated and three connected image regions.

Using the above protocol, one can build a list of primitive graph elements from which any image-tree of arbitrary complexity can be constructed. However, there can be problems with such a simplistic representation. Figure J.3 shows a basic three-connected image-tree and just a few of the many permutations of image components that this tree structure can represent. Nevertheless, useful generalisation properties can result from this.

The writer's solution to this multiple-representation problem is to rely on an adequate shape-factor for regions. This simple approach can enable an **overall** shape-factor to be applied to the root object or region, thereby alleviating at least part of the problem. The method used in the present project is to assign shape parameters according to image roundness, squareness, and so on. Triangles and rectangles are thus represented by parameters of intermediate value.

An analysis of the image-tree, in association with Prolog and the methods discussed in Chapter 9—especially context—may give acceptable object recognition-identification success rates. The fuzzy logic approach within Prolog rules can utilise a system of recognition confidence factors to assist in the process. The paper by Ng et al. (1991), and the references cited therein, contains a useful discussion of the evidence-based approach. Figure J.4 is a more complex example of an image-tree.

[4] It is also possible to represent the image "frame" as a single root node at the highest level—but this is not done here.

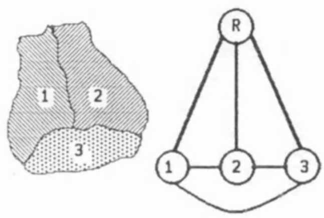


Figure J.1 Illustrating an elementary 3-region synthetic image and its corresponding image-tree representation.

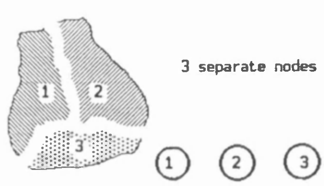


Figure J.2 This diagram shows the representational difference between three separated regions and three connected regions, both image sets having the same basic image component shapes.

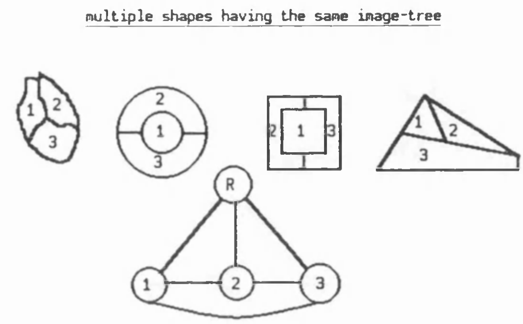


Figure J.3 Illustrating the multiplicity of images that can be represented by the same basic image-tree structure. This can be useful for generalisation.

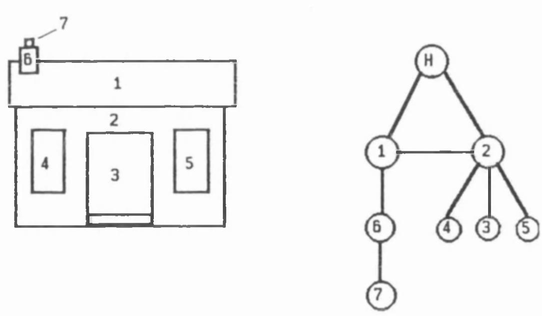


Figure J.4 An example of a more complex synthetic image and its corresponding image-tree, or graphical representation. A problem is the limited number of colours available for representing all image regions as discrete objects.

J.2 Attneave's Image Highlights

The work described in Attneave (1954), and mentioned in recent sources such as Grossberg (1987b), Arbib (1989), and Bruce and Green (1985) has relevance to the present thesis. It is also a very interesting subject in its own right, unusual aspects of the phenomena having been demonstrated in public interest television programmes in recent years. [5]

Briefly, Attneave showed that when an outline drawing of a cat is replaced by a drawing in which the points of maximum curvature in the original are joined by straight lines, the new drawing still resembles a cat. This is illustrated in Figure J.5. The question then arises: why are the points of maximum curvature such good indicators of the entire image form? Alternatively, one could ask: why bother to produce a good and accurate facsimile if an approximation can suffice?

The Attneave observation can have profound technological as well as psychological implications. Technically, we can reduce image storage requirements by storing only points of maximum outline curvature, reconstructing the boundaries, and then finally filling-in sufficient detail. This is somewhat similar to the ECM-HRS concept described in this thesis. Psychologically, the Attneave experiment demonstrates yet again the propensity of biological—especially human—vision to "see" details that are not present in the retinal image.

The writer considers that another example of this phenomenon is the classical demonstration of the moving human form in total darkness when indicated only by the presence of lights fixed to the joints of limbs—see figure J.6. Even when a limb is held perfectly straight, the presence of the lights marks the joints as always being points of maximum curvature. In this case, we can also say that these are the points of maximum information about the image form. Yet another (and extremely amusing) example of this phenomenon is the animated cartoon sequences of desk lamps representing the human form of mother and child—again discussed in a television documentary programme.

[5] An example is a 1987 edition of the BBC2 "Horizon" series entitled *The Blind Watchmaker*.

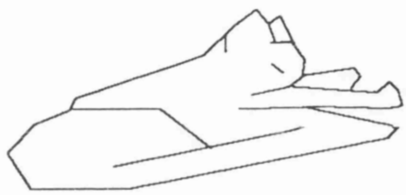


Figure J.5 "Attneave's Cat": Connecting points of maximum curvature by straight lines yields a recognisable caricature of a cat. Adapted from Attneave (1954).

Figure J.6 Lights are attached to the joints of an athlete who walks or runs in the dark. The changing pattern of the lights is immediately interpreted as a human figure in motion.

The concept of Parameter Space (McCafferty, 1990) may be useful in the processing of images that can be represented by only their points of maximum curvature. For example, in terms of an energy metric (and also possibly as a preattentive response) straight lines have zero energy, gentle curves have small energy, while corners and line endings are points of maximum energy. Parameter space is able to both maintain spatial relationships (iconics) and hold energy values (symbolics) within a multi-layered cellular automaton, or an artificial neural network. The possibilities are most encouraging, and are as always limited only by the imagination.

Figure J.7 suggests how simple geometrical shapes can be represented, and parameterised, using only their points of maximum curvature. The possible connection between this phenomenon and the notion of direct perception, as expounded in this thesis, requires further and much more detailed investigation.

Figure J.7 Illustrating the general concept of Parameter Space representation of simple image outlines by their points of maximum curvature (i.e. maximum energy points).

REFERENCES

Aleksander, I. (ed.) (1983)
ARTIFICIAL VISION FOR ROBOTS, Kogan Page, London.

Aleksander, I. and Morton, H. (1990)
AN INTRODUCTION TO NEURAL COMPUTING,
Chapman and Hall, London.

Arbib, M.A. (1989)
THE METAPHORICAL BRAIN 2, John Wiley, New York.

Attneave, F. (1954)
"Some informational aspects of visual perception"
in Psychological Review, 61. pp 183-193.

Ballard, D.H. (1981)
"Generalising the Hough transform to detect arbitrary shapes"
in Pattern Recognition, vol. 13, 2. pp 111-122.

Ballard, D.H. and Brown, C.M. (1982)
COMPUTER VISION, Prentice-Hall Inc., New Jersey.

Ballard, D.H. (1984)
"Parameter nets" in Artificial Intelligence, vol. 22, pp 235-267.
Elsevier Press (North-Holland).

Barker, J.R. (1990)
"Novel logic and architectures for molecular computing"
in PARALLEL PROCESSING IN NEURAL SYSTEMS AND COMPUTERS,
Proc. 10th ICNC, (ed.) Eckmiller et al., North-Holland.

Barrow, H.G. and Tenenbaum, J.M. (1978)
"Recovering intrinsic scene characteristics from images"
in COMPUTER VISION SYSTEMS, (ed.) E.M. Riseman.
pp 3-26. Academic Press, New York.

Batchelor, B.G. (1978)
PATTERN RECOGNITION : IDEAS IN PRACTICE,
Plenum Press, New York.

Batchelor, B.G. (1991)
INTELLIGENT IMAGE PROCESSING IN PROLOG,
Springer-Verlag, London.

Beale, R. and Jackson, T. (1990)
NEURAL COMPUTING : AN INTRODUCTION,
Adam Hilger (IOP), Bristol.

Beaulieu, J.-M. and Goldberg, M. (1989)
"Hierarchy in picture segmentation : a stepwise optimisation approach"
in Pattern Analysis and Machine Vision, vol. 11, no. 2. pp 150-163.
IEEE, New York.

Beck, J. (1982)
ORGANISATION AND REPRESENTATION IN PERCEPTION,
LEA, London.

Besl, P.P. and Jain, R.C. (1988)
"Range image segmentation"
in Freeman, H. (ed.), MACHINE VISION, Academic Press.

Boden, M. (1988)
COMPUTER MODELS OF MIND, University of Cambridge Press, New York.

Borland Inc. (1988)
TURBO C REFERENCE AND PROGRAMMERS GUIDES

Borland Inc. (1989)
TURBO PROLOG REFERENCE AND PROGRAMMERS GUIDES

Braitenberg, V. and Schuz, A. (1991)
ANATOMY OF THE CORTEX, Springer-Verlag, Berlin.

Bratko, I. (1990)
PROLOG : PROGRAMMING FOR ARTIFICIAL INTELLIGENCE,
Addison-Wesley, Wokingham, England.

Brown, A.G. (1991)
NERVE CELLS AND NERVOUS SYSTEMS, Springer-Verlag, London.

Bruce, V. and Green, P. (1985)
VISUAL PERCEPTION : PHYSIOLOGY, PSYCHOLOGY AND ECOLOGY,
LEA, Hillsdale, New Jersey.

Chang, S.K. (1984)
"Image information measures and encoding techniques" in DIGITAL IMAGE ANALYSIS, (ed.) S. Levialdi. pp 3-17. Pitman, London.

Churchland, P.S. (1986)
NEUROPHILOSOPHY, MIT Press, Cambridge, Massachusetts.

Clocksin, W.F. and Mellish, C.S. (1984)
PROGRAMMING IN PROLOG, Springer-Verlag, New York.

Cohen, G. et al. (1986)
MEMORY : A COGNITIVE APPROACH,
The Open University Press, Milton Keynes.

Cotter, J.R. (1990)
"The visual pathway : an introduction to structure and organisation" in THE SCIENCE OF VISION, (ed.) K.N. Leibovik, Springer-Verlag, New York.

DARPA (1988)
NEURAL NETWORK STUDY, AFCEA International Press, Virginia.

Domanay, E. et al. (eds.) (1991)
MODELS OF NEURAL NETWORKS, Springer-Verlag, Berlin.

Duff, M.J.B. and Fountain, T. (1986)
CELLULAR LOGIC IMAGE PROCESSING, Academic Press, London.

Duggan, M. (1991)
Personal communication.

Duggan, M. (1984)
POSINOMIC PIECEWISE CONTROL, PhD Thesis, University of Nottingham.

Eggermont, J.J. (1991)
THE CORRELATIVE BRAIN, Springer-Verlag, Berlin.

Ellis, W.D. (1955)
A SOURCEBOOK OF GESTALT PSYCHOLOGY, Routledge and Kegan Paul, London. (Translated from the 1923 original German text).

Frisby, J.P. (1990a)
"Computational theory of perception" in A DICTIONARY OF COGNITIVE PSYCHOLOGY, Blackwell, London.

Frisby, J.P. (1990b)
"Direct perception" ibid.

Fryer, R.J. and Miller, J. (1991)
"Imaging polarimetry for industrial inspection" in BMVC-91, Springer-Verlag, London. pp 347-350.

Fu, K-s. (ed.) (1984)
VLSI FOR PATTERN RECOGNITION AND IMAGE PROCESSING, Springer-Verlag, Berlin.

Fukushima, K. (1988)
"Neocognitron : a hierarchical neural network capable of visual pattern recognition" in Neural Networks, vol. 2, no. 1. pp 119-130. Pergamon Press.

Fujita, M. (1989)
"A model for oblique saccade generation and adaptation" in DYNAMIC INTERACTIONS IN NEURAL NETWORKS : MODELS AND DATA, (eds.) Arbib and Amari, Springer-Verlag.

Gibson, J.J. (1950)
THE PERCEPTION OF THE VISUAL WORLD, Houghton Mifflin, Boston.

Gibson, J.J. (1966)
THE SENSES CONSIDERED AS PERCEPTUAL SYSTEMS, Houghton Mifflin, Boston.

Gibson, J.J. (1979)
THE ECOLOGICAL APPROACH TO VISUAL PERCEPTION, Houghton Mifflin, Boston. Republished by LEA, New Jersey, 1986.

Gonzalez, R.C. (1977)
DIGITAL IMAGE PROCESSING, Addison-Wesley, Reading, Massachusetts.

Grossberg, S. (ed.) (1987a)
THE ADAPTIVE BRAIN I, Elsevier Science Publications, North-Holland.

Grossberg, S. (ed.) (1987b)
THE ADAPTIVE BRAIN II, Elsevier Science Publications, North-Holland.

Grossberg, S. and Mingolla, E. (1985a)
"Neural dynamics of form perception : Boundary completion, illusory figures, and neon color spreading" in Psychological Review, vol. 92, pp 173-211.

Grossberg, S. and Mingolla, E. (1985b)
"Neural dynamics of perceptual grouping : Textures, boundaries and emergent segmentations" in Perception and Psychophysics, vol. 38. pp 141-171.

Haken, H. (ed.) (1990)
SYNERGETICS OF COGNITION, Springer-Verlag, Berlin.

Hayes, P.J. (1979)
"The naive physics manifesto" in EXPERT SYSTEMS IN THE MICROELECTRONICS AGE, (ed.) D. Michie, Edinburgh University Press, Edinburgh. pp 168-201.

Hochberg, J. and Brooks, V. (1960)
"The psychophysics of form : reversible perspective drawings of spatial objects"
in American Journal of Psychology, vol. 73. pp 337-354.

Hockney, R.W. and Jesshope, C.R. (1988)
PARALLEL COMPUTERS 2, Adam Hilger (IOP), Bristol.

Holland, J.H. (1975)
ADAPTATION OF NATURAL AND ARTIFICIAL SYSTEMS,
University of Michigan Press, Ann Arbor.

Hough, P.V.C. (1962)
"Method and means for recognising complex patterns". US Patent No. 3069654.

IEE (1982)
ELECTRONIC IMAGE PROCESSING, IEE Conference Publication No. 214,
The Institution of Electrical Engineers, London.

Isenberg, C. (1978)
THE SCIENCE OF SOAP BUBBLES AND SOAP FILMS,
Tieto Limited, England. ISBN 0-905028-02-3.

Kanatani, K. (1990)
GROUP-THEORETICAL METHODS IN IMAGE UNDERSTANDING,
Springer-Verlag, Berlin.

Kirkpatrick, S. et al. (1983)
"Optimisation by simulated annealing"
in Science, vol. 220, no. 4598. pp 671-680.

Koenderink, J.J. (1986)
"Optic flow" in Vision Research, vol. 26, No. 1. pp 161-180.

Kohonen, T. (1987)
CONTENT-ADDRESSABLE MEMORIES, Springer-Verlag, Berlin.

Kohonen, T. (1988)
SELF-ORGANISATION AND ASSOCIATIVE MEMORY,
Springer-Verlag, Berlin.

Krebs, W. and Krebs, I. (1991)
PRIMATE RETINA AND CHOROID, Springer-Verlag, New York.

Land, E.H. (1977)
"The retinex theory of colour vision"
in Scientific American, Vol. 237, No. 2. pp 108-128.

Lewis, R. (1990)
PRACTICAL DIGITAL IMAGE PROCESSING, Ellis Horwood, London.

Little, J.J. (1988)
"Integrating vision modules on a fine- grained parallel machine" in COMPUTER VISION, (ed.) H. Freeman, Academic Press, New York.

Lloyd, J.W. (1984)
FOUNDATIONS OF LOGIC PROGRAMMING, Springer-Verlag, Heidelberg.

Loebner, E.E. (1987)
"Concurrency assurance in vertebrate retinas" in ICNN-87, vol. 4. pp 147-159. IEEE, New York.

Manneville, P. et al. (eds.) (1989)
CELLULAR AUTOMATA AND THE MODELLING OF COMPLEX PHYSICAL SYSTEMS, Springer-Verlag, Berlin.

Marr, D. (1976)
"Early processing of visual information" in Transactions of the Royal Society of London, Series B, 275. pp 483-524.

Marr, D. (1982)
VISION : A COMPUTATIONAL INVESTIGATION INTO THE HUMAN REPRESENTATION AND PROCESSING OF VISUAL INFORMATION, W.H. Freeman, San Francisco.

Marr, D. and Hildreth, E. (1980)
"Theory of edge detection" in Proceedings of the Royal Society of London, Series B, 211. pp 187-217.

Marr, D. and Nishihara, H.K. (1978)
"Representation and recognition of the spatial organisation of three-dimensional shapes" in Proceedings of the Royal Society of London, Series B, 200. pp 269-294.

McCafferty, J.D. (1990)
HUMAN AND MACHINE VISION : COMPUTING PERCEPTUAL ORGANISATION, Ellis Horwood, London.

McIndoe, T. (1988)
SOME ASPECTS OF ARTIFICIAL NEURAL NETWORKS IN THE CONTROL OF NON-LINEAR SYSTEMS, unpublished MPhil Thesis, University of Strathclyde, Glasgow.

Mead, C.A. (1988)
"A silicon model of early visual processing" in NEURAL NETWORKS, vol. 1, no. 4. pp 91-97.

Meuller, P. et al. (1987)
"Neural computation of visual images" in ICNN-87, vol. 4. pp 75-88. IEEE, New York.

Minsky, M.L. (1975)
"A framework for representing knowledge" in THE PSYCHOLOGY OF COMPUTER VISION, McGraw-Hill. pp 211-272.

Mowforth, P. (ed.) (1991)
BRITISH MACHINE VISION CONFERENCE (BMVC-91), Glasgow University, September 1991. Springer-Verlag, London.

Muller, B. and Reinhardt, J. (1990)
NEURAL NETWORKS : AN INTRODUCTION,. Springer-Verlag, Berlin.

Murray, J.D. (1990)
MATHEMATICAL BIOLOGY, Springer-Verlag, Berlin.

Ng, I. et al. (1991)
"Supervised segmentation using a multi-resolution data representation" in BMVC-91, (ed.) P. Mowforth, Springer-Verlag, London.

Olsen, R.K. and Attneave, F. (1970)
"What variables produce similarity grouping?" in American Journal of Psychology, vol. 83. pp 1-21.

Pratt, W.K. (1991)
DIGITAL IMAGE PROCESSING (Second Edition), Wiley-Interscience, New York.

Pellionisz, A.J. (1986)
"Tensor network theory and its application in computer modeling of the metaorganization of the sensorimotor hierarchies of gaze" in NEURAL NETWORKS FOR COMPUTING, (ed.) J.S. Denker, AIP Conf. Proc., No. 151. pp 339-344.

Pieroni, G.G. (ed.) (1990)
ISSUES ON MACHINE VISION, Proc.CISM 1989, Springer-Verlag, Berlin.

Preston, K. and Duff, M.J.B. (1984)
MODERN CELLULAR AUTOMATA, Plenum Press, New York.

Pugh, A. (ed.) (1983)
ROBOT VISION, IFS Publications UK, and Springer-Verlag, Berlin.

Randall, J.N. et al. (1989)
"Nanoelectronics : fanciful physics or real devices?" in Journal of Vacuum Science and Technology, Section B7, vol. 6. pp 1398-1403.

Ranka, S. and Sahni, S. (1990)
"Parallel algorithms for image template matching" in PARALLEL ALGORITHMS FOR INTELLIGENCE AND MACHINE VISION, (ed.) Kumar et al., Springer-Verlag, Berlin.

Rich, E. (1983)
ARTIFICIAL INTELLIGENCE, McGraw-Hill International, Singapore.

Roth, I. and Frisby, J.P. (1986)
PERCEPTION AND REPRESENTATION : A COGNITIVE APPROACH, The Open University Press, Milton Keynes.

Rothstein, J. (1988)
"Bus-automata, brains, and mental models"
in IEEE Trans. SMC, vol 18, no. 4. IEEE, New York.

Sanz, J.L.C. (1988)
RADON AND PROJECTION TRANSFORM-BASED COMPUTER VISION, Springer-Verlag, Berlin.

Schildt, H. (1987)
ADVANCED TURBO PROLOG, Osborne McGraw-Hill, Berkeley, California.

Shaw, G.L. et al. (1986)
"Trion model of cortical organization : toward a theory of information processing and memory" in BRAIN THEORY, (eds.) Palm and Aertsen, Springer-Verlag, Berlin. pp 177-191.

Siebert, J.P. et al. (1991)
"The active stereo probe : dynamic video feedback" in BMVC-91, Springer-Verlag, London. pp 383-386.

Stobo, J. (1989)
PROBLEM SOLVING WITH PROLOG, Pitman Publishing, London.

Stonier, T. (1990)
INFORMATION AND THE INTERNAL STRUCTURE OF THE UNIVERSE, Springer-Verlag,. London.

Sullivan, G. (1982)
"Filtering in image understanding" in DIGITAL SIGNAL PROCESSING, (ed.) N.B. Jones, IEE (PPL), London.

Tennant, H. (1981)
NATURAL LANGUAGE PROCESSING, Petrocelli, New York.

Toffoli, T. and Margolus, N. (1987)
CELLULAR AUTOMATA MACHINES, MIT Press, Cambridge, MA.

Treisman, A. (1985)
"Preattentive processing in vision" in Computer Vision, Graphics and Image Processing, 31. pp 157-177.

Ullman, S. (1980)
"Against direct perception" in The Behavioral and Brain Sciences, vol. 3. pp 373-415.

White, I. (1992)
"Some new directions in AI" in The Computer Bulletin, vol. 4, part 2, BCS, London.

Wolfram, S. (1983)
"Statistical mechanics of cellular automata" in Rev. Mod. Phys., vol 55, pp 601-644.

Wolfram, S. (1984)
"Computational theory of cellular automata" in Commun. Math. Phys., vol 96, pp 15-57.

Wood, A. (1988)
"Intermediate-level vision, relations, and processor arrays : an application of CLIP4 to image sequence analysis" in PARALLEL ARCHITECTURES AND COMPUTER VISION, Clarendon Press.

Yager, R.R. et al. (eds.) (1987)
FUZZY SETS AND APPLICATIONS : SELECTED PAPERS BY L.A. ZADEH, Wiley-Interscience, New York.

GLOSSARY

The defining terms presented here cover a wider field than the immediate goals of the present research. The terminology may refer to biological or artificial vision systems—or both. Human vision has the advantage that natural visual phenomena can be better understood and described, but is also highly subjective.

It is hoped that this glossary will provide useful reference and background material, but original sources should always be consulted for detail—or in case of doubt.

ACCOMMODATION

Adjustment of the optics of the eye to keep an object of interest in focus on the retina as its distance from the eye varies. In the human eye, this is achieved by varying the thickness of the natural lens.

ACUITY

See Visual Acuity.

ADAPTIVE COEFFICIENTS

In ANS, values of previous computations (weights), stored in local memory, which modify subsequent computations.

ADAPTIVE CONTROL

A control system which is capable of altering its output in response to sensory input from its environment.

ALGORITHM

A specified procedure for solving a problem—especially by an electronic computer.

ARTIFICIAL INTELLIGENCE (AI)

Machine or computer capability to carry out humanlike functions, such as learning, reasoning, and self-control or motivation. In control terminology AI is regarded as being one step beyond adaptive control. AI is often a computational approach based on symbolic representation and manipulation of human knowledge. (Note that the concepts and definitions of current AI are liable to change, and often in quite subtle ways).

ARTIFICIAL NEURAL NETWORK (ANN)

A network, usually a computer simulation or hardware implemented, of a very large interconnected group of artificial neurons, or processing elements (PEs), which is intended to reproduce some of the processing capability of biological neural networks.

ARTIFICIAL NEURAL SYSTEMS (ANS)

The rapidly evolving science of computer concepts applied to the simulation of biological intelligence and brain function.

ASSOCIATIVISM

A school of perceptual psychology, prevalent in the late 19th century, based on the notion that perception could be explained solely in terms of component sensations, and that it was the association of simple sensations which resulted in complex perception.

AXON

The output process of a nerve cell (neuron) leading away from the cell body (soma) and terminating at synapses with other cells. The axon normally carries signals to other neurons although in some cases it may receive input.

BANDWIDTH

The defined range of frequencies or values over which a particular system responds. The higher the bandwidth, the more detail is represented and preserved, but the more susceptible the system becomes to noise.

BIT

The smallest individual element of computer storage and manipulation.

BOTTOM-UP

In vision systems, means that image processing and interpretation are data-driven (or low-level) rather than model-driven (or high-level).

BOUNDARY

A connected set of edge points, or pixels.

BOUNDARY DETECTION

A computational process which attempts to determine image boundaries, based on the implicit assumption that these correspond to boundaries of objects in the real world. See also Segmentation.

BOUNDARY TRACKING

A method of boundary detection. It is based on the search and grouping of edge points which share some common property, such as intensity or orientation.

BOUNDING ENVELOPE

Peripheral tokens (or nodes) connected by arcs.

BRIGHTNESS

The subjective, intensive attribute of a light source or sensation, which is unaffected by context. Compare with luminance.

BYTE
A group of 8 bits. The most common and practical unit of computer storage and manipulation.

CELLULAR AUTOMATON (CA)
An arrangement of tightly-coupled processing elements (q.v.) or cells in which a cell's present state is determined by both the cell's previous state and the states of the nearest-neighbour cells.

CENTRAL NERVOUS SYSTEM (CNS)
The system of neurons and axonal fibres (nerves) which include both the brain and the spinal cord, but not peripheral neurons. See also Peripheral Nervous System (PNS).

CHAIN CODE
A boundary representational schema. It starts at an initial point, and records a chain of directions to successive points on a grid.

CHARGE-COUPLED DEVICE (CCD)
A solid-state device operating on the bucket-brigade charge-passing principle, usually for detecting light. Such devices are often used as the image sensor in modern video camera technology.

CLASSIFICATION
An image processing technique which attempts to divide an image into regions, according to the pixel values in one or more bands.

CLOSED-LOOP CONTROL
A control system in which the output is continuously monitored by feedback from the controlled environment.

CODONS
A set of six primitives used for canonical curve and shape representation.

COLOUR CONSTANCY
The tendency of the perceived chromatic colour of a surface to remain constant despite changes in the spectral distribution of the illuminant. The effect is often referred to as "discounting the illuminant."

COMPUTER-AIDED DESIGN (CAD)
The use of computer-based graphics and printing technology in the design process—especially electronics systems. The design can be that of physical printed-circuit board layout, or complex electronic circuits, including simulation studies.

COMPLEX CELL
Orientation-selective cell in the visual cortex—without separate ON and OFF regions in its receptive field.

COMPUTATIONAL THEORY
In the sense used in this thesis, is a term introduced by David Marr to explain how, in principle, relevant object visual information can be extracted from images.

COMPUTER MODELLING
The numerical characterisation of certain aspects of a system, capable of computational representation and simulation.

COMPUTER VISION
See Machine Vision.

CONCENTRIC FIELD
A receptive field divided into an inner circular region and an outer ring-shaped region.

CONE
The photopic (q.v.) photoreceptors for daylight colour vision, which are of three types with overlapping spectral sensitivities.

CONNECTION
In neural networks, a signal transmission pathway between processing elements (PEs or neurons).

CONTRAST
A measure of the difference in intensity (brightness) between the lightest and the darkest regions of an image or scene.

CONVEX HULL
The smallest polyhedron which bounds an imaged object, such that all vertices are convex relative to the object. The convex hull of a 2D shape can be imagined as the stretching of a rubber band around the object. Likewise, the 3D convex hull is imagined as the enclosing of the object by a rubber-sheet or membrane.

CONVOLUTION
An image processing technique which can be used to provide a wide variety of different filtering functions, using a single algorithm. Optimisation of this one algorithm provides high speed implementations of all filters implemented in this way.

CORTEX
The outer layer of an organ, especially the brain.

COVARIANCE MATRIX
A matrix which shows how a set of variables move together.

CUMULATIVE HISTOGRAM
An image histogram in which each entry is the number of pixels having a numerical value less than, or equal to, the current value.

CURVATURE PRIMAL SKETCH
A representational schema for curves, based on a set of curve primitives.

DATA-DRIVEN
A forward-reasoning, or bottom-up, problem solving approach.

DATA REDUCTION
The representation of a set of data points by a smaller number of parameters. The resultant reduced data set is easier to manipulate than the original points themselves. The parameter set provides a model of the behaviour which gave rise to the original data points.

DENDRITES
The processes of nerve cells which carry slow electrical potentials from the synapses to the cell body or soma.

DEPOLARISATION
A change in the membrane potential of the nerve cell (neuron) such that the interior becomes less negatively charged relative to the exterior. If the membrane of an axon is depolarised, action potentials are generated with increased (pulse) frequency.

DIFFRACTION
The scattering of rays of light by collision with particles of matter as they pass through a medium such as air or water.

DIGITAL-TO-ANALOGUE CONVERTER (DAC)
An electronic device which converts some physical quantity into an equivalent numerical form, such as a voltage level. See also the entry for Transducer.

DILATION
In image processing, a morphological operation during which an imaged object is increased in size by the addition of pixels to its boundary. It also refers to equivalent cell state changes in cellular automata imaging mechanisms.

DISPARITY
A term used to describe the differences between two or more images, when the objects in view are positioned at different distances from the viewer. This forms the mathematical basis for stereo vision. (Disparity is usually measured in minutes of arc).

DISPERSION
The variable refraction of light, depending on its wavelength.

DISTANCE TRANSFORM
An image transform in which pixels are replaced by their distance to the imaged object boundary. See also Hough Transform. The term may also include transforms in which distance measures (such as Hamming distance) replace the pixels.

DOMAIN TRANSFORM
A transform producing a representation which, although completely different from the original data, is equivalent to it in terms of information content. See also Hough Transform and Parameter Space.

DUPLICITY THEORY
Holds that vertebrate retinas are mediated by two different types of photoreceptors—RODS and CONES—which function under different levels of illuminance.

DYNAMIC PROGRAMMING
A mathematical programming approach to the solution of combinatorial optimisation problems involving sequential decision making processes.

DYNAMIC RANGE
In image processing, the range between the maximum and minimum value of the pixels forming an image.

EDGE
Linked points of substantial intensity change, usually between two or more regions of relatively uniform intensity values.

EDGE DETECTION
An approach to image segmentation, based on the assumption that discontinuities in pixel values often correspond to the edges of imaged objects.

EDGEL
An edge map pixel.

EDGE MAP
An intrinsic image in which each pixel represents the likelihood of an imaged object edge at that point.

EDGE STRENGTH
The numerical value of an edgel.

EMERGENT PROPERTY
An image property resulting from perceptual groupings, but which is not evident in the component image tokens. For example, a single dot or point has no orientation. Two dots or points possess the emergent property of orientation (that of a line passing through them).

EMULATOR
A (computer) program or hardware circuit that simulates another system, program or circuit, often in real-time.

END EFFECTOR
The gripper, or working extremity of a robot arm or manipulator.

END-INHIBITION
A property of some cells in the visual cortex causing them to respond strongly to an edge, bar, or slit that ends within the visual field.

ENTROPY
A measure of the amount of disorder in a system. Entropy can be used in image and vision processing to estimate the information content of an image or scene.

EFFECTOR CELL
A nerve cell (neuron) which excites muscular contraction. See also Motor Neuron.

EROSION
In image processing, a morphological operation during which an imaged object is decreased in size by the removal of pixels from its boundary. It also refers to equivalent cell state changes in cellular automata imaging systems.

EXPERT SYSTEM
An AI-based system which "reasons" like a human expert. It normally requires a knowledge-base for the specific domain of operation, and has the property of being able to trace (or explain) its reasoning.

FAN-IN
The number of input signals to an electronic circuit or logic gate. Useful, in the context of ANS, for describing weighted dendritic input to artificial neurons.

FAN-OUT
The number of output signals that a logic gate is able to support; limited usually by the power capability of the circuit. Useful, in the context of ANS, for describing the axonal fibre distribution of artificial neurons.

FAST FOURIER TRANSFORM (FFT)
An algorithm for computing the Fourier transform requiring many fewer operational steps than the direct method. It is particularly suited to digital computational methods.

FIBRE
See Axon.

FIGURE-GROUND SEPARATION
A term describing the human vision system's capacity for separating objects or visual stimuli into foreground and background.

FIRING RATE
In the context of this work, the frequency at which action potentials pass down the axon of a nerve cell (neuron).

FIRST-GENERATION MACHINE VISION
An early approach to computer vision, involving:

1. Image capture.
2. Feature extraction.
3. Feature matching against limited object lists.

Unlike more recent methods, no 3D imaging or high-level knowledge is normally involved.

FLUX
In the context of this thesis, the amount of light falling on a given area in a given time.

FOVEA
The pit-shaped depressed area of vertebrate retinas having the maximum possible visual acuity—comprises mainly CONE type photoreceptors.

FOVEATION
Controlled muscular movement of the eyeball—in such a manner and direction as to image scenic areas of visual interest on the fovea.

FOURIER ANALYSIS
A mathematical method of decomposing arbitrary signals into a series of pure sinusoidal components. See also Fourier Transform.

FOURIER DESCRIPTORS
The sinusoidal components obtained by Fourier Analysis. In image processing, factors due to scale and position can be filtered out.

FOURIER DOMAIN
Another term for the Frequency Domain (q.v.).

FOURIER SPECTRUM
In image processing, an image in which each pixel is the magnitude of the corresponding element of the Fourier transform.

FOURIER TRANSFORM
A domain transform, as a result of which an image is represented as a set of spatial frequency components. Used in Fourier analysis.

FREQUENCY DOMAIN
In image processing, the domain in which an image is represented as a set of spatial frequency components.

FRAME
In the context of image processing, one complete image or picture, usually referring to original or pre-processed images.

FRAME-GRABBER
An electronic device which can digitize and store a complete frame of a video camera image—or a television picture—in real time.

FRICTION
A resistive force due to the molecular attraction of two bodies moving in contact with each other. Together with inertia, it constitutes a source of positional inaccuracy in servo control systems.

FRONTAL PLANE
The plane perpendicular to the line of sight.

FULL PRIMAL SKETCH
See Primal Sketch.

FUZZY LOGIC
A logic system in which fractional weight is given to the full range of the logic states between 0 and 1. That is, probability values, such as 0.7, are meaningful within a fuzzy logic system.

FUZZY SET THEORY
The theory of operations, such as union, intersection, etc. defined on sets with fuzzy logic values, or probabilities. Fuzzy Logic is linked with the pioneering work of Lofti A Zadeh—see Yager et al. (1987).

GANGLION CELL
A type of cell in the vertebrate retina. The axons in the ganglion cells bunch together to form the optic nerve, which carries neural signals and visual information from the retina to the distinct regions of the brain.

GENERALISED CONE
See Generalised Cylinder.

GENERALISED CYLINDER
A method of 3D shape representation which defines shape by sweeping a predefined cross-section along a predefined axial path. The sweeping cross-section may evolve and change as it sweeps along the path, which is normally a spline (q.v.). An ordinary cylinder is a circular disc swept along a straight line axis.

GENERALISED HOUGH TRANSFORM
Based on the standard Hough transform, this technique allows the correlation of 2D and 3D shapes which have no analytic description. It maps a spatially indexed feature space into a non-spatially indexed parameter space. See Ballard (1981).

FIRING RATE
In the context of this work, the frequency at which action potentials pass down the axon of a nerve cell (neuron).

FIRST-GENERATION MACHINE VISION
An early approach to computer vision, involving:

1. Image capture.
2. Feature extraction.
3. Feature matching against limited object lists.

Unlike more recent methods, no 3D imaging or high-level knowledge is normally involved.

FLUX
In the context of this thesis, the amount of light falling on a given area in a given time.

FOVEA
The pit-shaped depressed area of vertebrate retinas having the maximum possible visual acuity—comprises mainly CONE type photoreceptors.

FOVEATION
Controlled muscular movement of the eyeball—in such a manner and direction as to image scenic areas of visual interest on the fovea.

FOURIER ANALYSIS
A mathematical method of decomposing arbitrary signals into a series of pure sinusoidal components. See also Fourier Transform.

FOURIER DESCRIPTORS
The sinusoidal components obtained by Fourier Analysis. In image processing, factors due to scale and position can be filtered out.

FOURIER DOMAIN
Another term for the Frequency Domain (q.v.).

FOURIER SPECTRUM
In image processing, an image in which each pixel is the magnitude of the corresponding element of the Fourier transform.

FOURIER TRANSFORM
A domain transform, as a result of which an image is represented as a set of spatial frequency components. Used in Fourier analysis.

FREQUENCY DOMAIN
In image processing, the domain in which an image is represented as a set of spatial frequency components.

GENETIC ALGORITHM
An approach to optimisation based on a direct analogy with natural adaptation and evolution. Each solution state is coded as a genetic string, and a population is evolved. See Holland (1975).

GESTALT PSYCHOLOGY
A system of psychological theory originating in Germany in the 1920s, which emphasises organisation, wholes, and the field properties of elemental precepts. The central tenet is the notion that the whole is greater than the sum of the parts.

GIBSONIAN
In this thesis, refers to an original, fine-grained, unprocessed visual image. Also may refer adjectively to the ideas and concepts associated with the vision researcher James J Gibson.

GLIAL CELL
A type of cell which is understood to support or nourish natural neurons—in a metabolical sense.

GOAL-DRIVEN
Used of a backward chaining, top down, problem solving strategy.

GREY-SCALE IMAGE
A retinotopic (q.v.) representation of image intensities. It is discrete and tesselated, and involves only sub-sampled intensity values—usually in the integer range 0 to 255.

GROUPING
The formation of meaningful sets of data from otherwise unorganised data, which share some characterisation or features.

HEURISTICS
A term used to describe "rules-of-thumb," or pragmatic knowledge used to guide a problem solver.

HIGH-LEVEL
In image processing, the interpretative processing stages of vision involving tasks such as object recognition or scene interpretation, as opposed to low-level processes which involve only image manipulation.

HISTOGRAM
A graphical representation of the number of pixels of each discrete value occurring in an image. Normally refers to grey-scale images. The vertical axis plots the number of occurrences, and the horizontal axis represents pixel numerical values.

HORIZONTAL CELL
A type of intercommunicating cell in the vertebrate retina.

HOUGH TRANSFORM
A global parallel method of finding straight or curved lines in an image. All points lying on a curve in image space map into a single point in transform parameter space.

HYPERCOLUMN
Characteristic regular areas on the visual cortex in which all cells have receptive fields mapped to an equivalent single area of retina. In particular, the hypercolumn cells exhibit a preference for retinal lines of a specific orientation.

HYPERFIELD
An array of hypercolumns distributed across the cortex, which map to equivalent receptive fields in the retina.

ICONIC
Image-like. A representation in which the image pixels share a correspondence with the original scenic image.

ILL-POSED
Used of a problem which is not well-posed (q.v.); meaning that usually only approximate—and often unsatisfactory—solutions can be found.

ILLUSORY CONTOURS
See Shape Completion.

IMAGE PROCESSING
A collection of transformation techniques and computer algorithms which can be applied to an input image to yield an output image with altered properties. It is used for enhancing, transforming, encoding, decoding, and transmitting images. A classic applications example is image noise reduction.

INDUCTION
A mechanism which involves the assimilation of large sets of data in order to originate one or more rules which characterise each of the individual data sets. The evolved rules can subsequently be used for the future interpretation of unknown data.

INERTIA
The tendency of a mass at rest to remain at rest, and of a mass in motion to remain in motion.

INTEGER
A whole number.

INTERNEURON
A nerve cell in the central nervous system which is neither a receptor nor an effector (eg. a so-called "processing" neuron).

INTERFACE

A shared boundary between two devices, software programs, or human to device interaction over which some kind of operation or communication is transferred. In computer systems, this term usually refers to a standardised method of data exchange (or protocol).

INTRINSIC IMAGES

A set of representations of the image array in which each individual representation corresponds to a particular intrinsic characteristic of the array. The intrinsic characterisations include intensity, surface reflectance, and surface orientation. See Barrow and Tenenbaum (1976).

INTROSPECTION

A technique used by visual psychologists when investigating the human visual system.

INVERSE PROBLEM

A problem which involves the recovery of a process, or of data, which has undergone distortion by some operation when only the output and the nature of the operation are known. The inverse imaging problem involves the recovery of an original scene from the computed image data—that is, "image understanding."

ISOMORPHISM

A doctrine of the Gestalt psychologists, which states that underlying every perceived sensation there exists some "brain event" which is structurally similar to the sensation. For example, the notion of magnetic "force fields" was believed to account for perceptions in which minimum field energy shape is perceived. An analogous concept is that of the physics of soap films and soap bubbles.

JOINT

In robotics, a mechanical device providing for motion in one rotational or translational degree of freedom (DOF).

KINETIC DEPTH EFFECT

The term used to describe the human vision system's capacity to infer structural information from the movement of objects in silhouette, even when the static view of the object is unrecognisable.

LAG

A delay in time or motion between an input and an output, or response.

LATERAL GENICULATE NUCLEUS (LGN)

The part of the mammalian brain where the axons of retinal ganglion cells terminate, and from which additional axons project to the visual cortex. Its primary processing function is as yet unknown.

LEARNING LAW
An equation or algorithm that modifies all or some of the adaptive coefficients (weights) in a neural network (or a local memory) in response to input signals and a transfer function. The learning law enables the network to adapt itself, when given examples of required responses or results.

LEAST SQUARES
A set of techniques for fitting models to data. The parameters of the model are adjusted until the sums of the squares of differences between the data and the model's prediction are minimised.

LINEAR
Used to describe an input-output relationship in which the output is directly proportional to the input.

LINEAR PROGRAMMING
A mathematical programming technique for solving optimisation problems where the cost functions and constraints are strictly linear.

LOCAL SYMMETRY
Symmetry which exists only locally, between different parts of the boundary of a shape.

LOOKUP TABLE
A mechanism for converting pixel values into values which are used to drive a video display. A means of providing fast precomputed values for use in display or computation, thereby avoiding a repetitive time delay at every discrete computational step, cycle, or iteration.

LOW-LEVEL
In image processing, the early stages of processing involving only simple image transformation and feature detection tasks. Normally considered to be image data-driven.

LOW-PASS FILTER
A system or structure of any kind in which slow, low-frequency signals are carried faithfully but fast, high-frequency signals are largely attenuated or reduced in magnitude.

LOW-RESOLUTION
A term used to describe an image digitized with relatively few pixels. It is a relative term since the adequacy of a resolution depends upon the intended application.

LUMINANCE

The luminance flux of an extended surface as measured at a point at which a unit of the projected area of the surface subtends a unit solid angle. It is related to radiance (q.v.) by the photopic luminous efficiency function, which takes into account the spectral sensitivity of the eye.

MACHINE VISION (Computer Vision)

Computer perception of visual data, in which a concise description is developed of a scene depicted in one or more images. It encompasses a variety of different disciplines and techniques, including electronics and signals, mathematics, computer algorithms. The typical processes include image processing, feature detection, perceptual and other relevant groupings, object recognition, and scene analysis. The mechanization of visual tasks and processes—especially for application to industrial and information processing operations.

MASK

A binary bit-image defining an area in which an operation is to be performed on another image of the same size and resolution.

MATCHED FILTER

A filter which maximizes signal-to-noise ratio so that a waveform of known and predefined shape can be separated from random noise.

MEDIAL AXIS

The medial axis of an imaged object is a set of lines joining those locations which are local maxima in terms of their distance from the object's boundary.

MEDIAL AXIS TRANSFORM (MAT)

A transformation which reveals the medial axis of an imaged object.

MEMBRANE POTENTIAL

The difference in electrical potential between the interior and the exterior of a neuron. In the resting state, the interior is always negative relative to the exterior.

MESOPIC

Levels of illuminance at which both ROD and CONE photoreceptors can operate. See also the entries under Photopic and Scotopic.

MONOCHROME

Refers literally to single colour, but is often used synonomously with grey-scale images.

MOTOR NEURON

A nerve cell (neuron) which synapses with a muscle cell. The action potentials passing down the axon of a motor neuron cause muscle fibres to contract and produce mechanical force.

NATURAL VISION
Biological visual perception. Used synonomously in this thesis with human vision.

NEIGHBOURHOOD OPERATION
An image operation involving a pixel and a set of its neighbours. Often the neighbour set will include the eight nearest neighbours, referred to as a 3x3 neighbourhood. The term is also used in the definition of Cellular Automata operations, or rules.

NEURAL NETWORK (NN)
A network of massively interconnected and parallel nerve cells (neurons) responsible for all biological intelligence. Includes the brain, and the central and peripheral nervous systems.

NEURON
A nerve cell with all its fibres. It may be located in the brain, the spinal cord or in the peripheral nervous system. UK spelling: NEURONE. In artificial neural systems (ANS) the neuron is often referred to as a Processing Element (PE).

NEUROTRANSMITTER
A chemical agent, such as Acetylcholine or Dopamine, released from nerve endings and transmitting impulses across synapses to nerve, muscle or other cells (neurons).

NOISE
In image processing, the random variation in the pixel values of an image resulting from the acquisition process, rather than by the image or scene itself. Noise originates in all electronic equipment, and its detection and elimination is a major aspect of image and vision work.

NORMALISATION
An operation applied to the result of a computation or process to bring the numerical values obtained into some consistent range, in order that they can be compared with other values.

OCCLUSION
The effect of concealing part of one object by another.

OCTAVE
One octave corresponds to a change in frequency by a factor of 2.

OPEN-LOOP CONTROL
A control system in which the output is not continuously modified by feedback from the environment.

OPTIC ARRAY
A term introduced by J J Gibson, to refer to the instantaneous pattern of light reaching a point in space from all directions. In different regions of the optic array, the spatial pattern of light will differ, according to the nature of the surfaces from which it has been reflected, or transmitted.

OPTIC FLOW (FIELD)
The fluctuating pattern of light intensity reaching an observer, caused by any relative movement between observer and environment.

OPTIC NERVE
The name given to axonal fibre bundles running from retina to brain.

ORIENTATION COLUMN
A region of the striate cortex within which neural cells respond preferentially to lines having the same orientation.

PALETTE
The definition of the set of colours which can be displayed or manipulated, and their relationship to the individual pixel values in an image.

PARALLEL DISTRIBUTED PROCESSING (PDP)
A term used to describe massively-parallel computation which bears some resemblance to the processing within natural neural networks. It is characterised by relative immunity from damage and graceful degradation—due to the distributed nature of the subsystems.

PARAMETER SPACE
A term used to describe a 3D representational schema generated by a 2D (x,y) topological map, having a third, or "parameter" axis—z, say. A place token in a cell (x,y,z) denotes the existence of a datum, value z, at the corresponding (x,y) location of the map.

PARAMETRIC
A description in terms of a model and some characteristic values. The values are known as PARAMETERS.

PATTERN RECOGNITION
A set of techniques for classifying data (not necessarily visual) into predetermined classes or categories. See Statistical and Syntactical Pattern Recognition.

PERCEPTION
An active process in which hypotheses are formed and tested, concerning the nature of the environment. The input is sensory data and information.

PERCEPTUAL ORGANISATION

The term used by visual psychologists to describe the low-level organisational capability of vertebrate visual systems towards structuring stimuli. The earliest work in connection with human Perceptual Organisation was carried out by the Gestalt school of psychology.

PERIPHERAL NERVOUS SYSTEM (PNS)

The sensory neurons and axons at the extremities, eg. the surface of the skin. See also Central Nervous System (CNS).

PHOTOPIC

The brighter levels of illuminance over which only CONE photoreceptors are believed to operate.

PIXEL (Pel)

The smallest individual element of a digitized image array, or the smallest part of an image which can be assigned a single colour or brightness value. An acronym for "picture element."

POINT-TO-POINT CONTROL

In robotics, a control method in which the input commands specify only a relatively few discrete points along a proposed path or trajectory.

POLYGONAL APPROXIMATION

A means of representing a curve as a series of polygons which bound the curve.

POSINOMIC CONTROL

A novel method of robotics control described by Duggan (q.v.) which attempts to replicate unidirectional muscular control using two opposing electrical solenoids or other actuators. The associated simplified control concepts and signals are in keeping with the parallel distributed processing philosophy of ANS.

POWER SPECTRUM

In image processing, the spectrum derived from the Fourier transform of an image. It is an estimate of the power in each component of the transform, and is a real quantity.

PRAGNANZ

A basic law of the Gestalt psychology, which states that of several geometrically possible organisations, that perceived will be the best, simplest, and most stable shape.

PREATTENTIVE VISION

A term which describes how the human visual system is capable of discriminating between various visual stimuli, without a focus of attention. Also refers to visual image processing requiring less than about 600 ms—see e.g. Treisman (1985).

PRESTRIATE CORTEX
In terms of the cortical map, refers to cortical areas V2, V3, V3A, V4, MT(V5). Consult referenced texts for details.

PROCESSING ELEMENT (PE)
In this thesis, an artificial neuron in a neural network, or a processor in an array or automaton—however implemented.

PRIMAL SKETCH
A primitive representation of the intensity change in a grey-level image. It was used by David Marr to make the properties of changes in an image explicit. The Raw Primal Sketch makes explicit only very localised properties, such as size, position, or orientation. The Full Primal Sketch results from grouping together elements of the raw primal sketch, thereby making explicit image global properties, such as alignment. The primal sketch is derived independently of higher level processes.

PROPORTIONAL CONTROL
A control method in which the actuator drive signal increases monotonically and proportionally with difference (or error) between the actual output response and a desired output.

PROPORTIONAL INTEGRAL (PI) CONTROL
A control method in which the actuator drive signal is generated by a weighted sum of: the difference between the actual output and the desired output; the time integral of the difference (or error).

PROPORTIONAL INTEGRAL DERIVATIVE (PID) CONTROL
As PI control, but includes the time derivative of the difference or error signal. PID control is generally faster and more accurate than Proportional or PI control. Note that in some US literature this form of control may be referred to as the Predictive Integral Derivative.

PULSE
A brief surge of voltage, current or electromagnetic wave energy.

RADIANCE
Radiant intensity per unit projected area. Related to luminance (q.v.) by the photopic luminous efficiency function which takes into account the spectral sensitivity of the eye.

RANDOM-DOT STEREOGRAM
A pair of images containing no significant monocular cues which, when fused stereoscopically, yield 3D depth information.

RAW PRIMAL SKETCH
See Primal Sketch.

RECEPTIVE FIELD
The active field of a neuron cell, within which an appropriate stimulation modifies the cell's firing rate (causes a response).

RECEPTOR CELL
A nerve cell (neuron) responsive to external stimuli (energy).

RECEPTOR POTENTIAL
The change in membrane potential within a receptor cell caused by external energy impinging upon it.

REDUNDANCY
The provision of duplicate mechanisms, data, or information such that a system can continue to function even if a part of it fails.

REGION
A set of connected pixels which possess a common attribute, such as grey-level image intensity.

REGION GROWING
A process of region segmentation which uses successive merging of adjacent regions, which have sufficiently small differences between them.

REGULARISATION
An approach to the solution of ill-posed (q.v.) problems by the use of additional qualitative information.

RELAXATION
An iterative problem solving approach in which initial conditions are propagated using *a priory* constraints, until certain goal conditions, such as convergence on a unique solution, are satisfied.

REPRESENTATIONAL SCHEMA
A data storage plan which makes certain visual information explicit. This may be either an iconic or a symbolic description.

RESOLUTION
The smallest measurable increment or change in a measuring device or transducer. May have other meaning in a computer context. A similar concept is that of GRANULARITY.

RETINEX THEORY
A scheme that attempts to predict the colour appearance of test colours in complex situations where colour context has a significant influence.

RETINOTOPIC
Preserving the topological layout of the retina.

ROD
The scotopic (q.v.) photoreceptors for monochrome, low-illuminance vision.

ROTATION
In image processing, means the rotation in the plane of the image.

SACCADE
Rapid movements of the eyeball to fixate a target.

SCALAR
A quantity which has magnitude only.

SCHEDULING FUNCTION
In neural networks, a function that determines if, and how often, a processing element (PE) should apply its transfer function.

SCOTOPIC
The dimmer levels of illuminance over which only ROD photoreceptors are believed to operate.

SEGMENTATION
The process of grouping data points, in a representation having some mutually compatible attributes, into data sets. This can be complex, and is always context-dependent.

SENSOR
A device whose input is some physical phenomenon, and whose output is an electrical quantity or some signal reproducibly—and usually proportionally—related to the input. See also the entry under Transducer.

SERVO
A control system (not necessarily mechanical) which measures an output quantity and compares it to an input quantity through a feedback loop in order to improve control capability.

"SHAPE-FROM" PROCESSES
The perception or calculation of 3D shape from 2D image data.

SIMULATED ANNEALING
A stochastic optimisation technique for solving large, multi-variate, multi-modal problems. It is based on statistical mechanics techniques for determining the equilibrium energy of molecules in solids at a constant temperature. See Kirkpatrick et al. (1983).

SPATIAL FREQUENCY
A measure of the rate of change of intensities within an image. Usually measured in cycles of sinewave intensity modulation per degree of visual angle.

SPLINES (B-Splines)
Piecewise continuous polynomials, used to approximate curves.

STATISTICAL PATTERN RECOGNITION
The application of statistical decision theory to the discrimination of data patterns. The data measurements are interpreted as points in a vector space, where each measurement is an axis in the space. The matching problem is then to find the nearest neighbour cluster, or point, (obtained during training) for each pattern presented. This is a geometrical approach, and requires an appropriate distance metric.

STEREOPSIS
The process of recovering the three-dimensional (3D) structure of a scene from two different views. It requires the measurement of the disparity of corresponding points in the two images, and the further interpretation of these disparity measurements to recover the range and orientation of the surfaces in the scene.

STOCHASTIC
Random. Conjecture.

STOCHASTIC OPTIMISATION
An approach to optimisation involving randomness. The use of randomness is intended to reduce the likelihood of converging on a local minimum.

SUBIMAGE
A set of adjacent pixels within an image. Often square or rectangular in shape.

SUPERIOR COLLICULUS (SC)
An area in the mid-brain (thalamus) which appears to be responsible for the collation of sensory signals, reflex action, and eye movements associated with foveation.

SYMBOLIC
Relating to the substitution of abstract representations, using tokens or symbols, for real objects or phenomena.

SYNAPSE
A point where the membranes of two neurons almost touch, and where the electrical activity in one neuron influences the membrane potential of the other neuron. Can be either sign-conserving or sign-inverting, that is, excitatory or inhibitory.

SYNERGISM
The action of two or more substances, organs, organisms, or systems to achieve an effect greater than the sum of their individual effects.

SYNTACTIC PATTERN RECOGNITION
An approach to pattern recognition in which complex patterns are recursively decomposed into simpler, primitive elements. Sometimes called Linguistic Pattern Recognition—due to its connection with grammars and parsing.

TEXTON
A texture element, as proposed by Julesz and Bergen (1983).

TEXTURE ANALYSIS
A set of techniques used for the visual discrimination between different textures, and also for calculating shape from texture.

THETA-AGGREGATION
An algorithm described by Marr (1976) which attempts to group similarly oriented edge tokens in directions which may differ from their intrinsic orientation.

THINNING
One technique for retrieving the medial axis of an imaged object, based on removing pixels from the object boundary.

THRESHOLD
The transition level between two states. In neural networks, the point at which a cell or neuron "fires" when a membrane potential is reached. The level at which a decision is made as to the value of an output.

THRESHOLDING
A fundamental image processing operation which converts a grey-level representation into a binary representation. Pixel values above the threshold are set to state "1" and those below set to state "0". The method has implications for the subsequent interpretation of the resulting binary images.

TOKEN
A symbol used as part of a representational schema, to mark a point of interest in an image. A token thus has a position, and may also posses other relevant properties.

TOP-DOWN APPROACH (Goal-Directed)
An approach in which the interpretation stage is guided in its analysis by trial or test, and *a priori* models. It is sometimes referred to as Hypothesise-and-Test.

TOPOLOGY
In the sense used in this work, a system network or arrangement of interconnecting pathways. It is also a branch of geometry which describes the properties of forms and shapes.

TORQUE
A turning force. Equal in magnitude to the force applied multiplied by the distance of its point of application from the centre of rotation.

TRANSDUCER
A device for converting energy from one form to another, usually from physical forms (light, heat, position, etc.) to electrical potentials or currents.

TRANSLATION
Motion in the plane of the image which does not involve rotation.

TRANSFER FUNCTION (TF)
In neural networks, a mathematical law that determines the response of a processing element (neuron) as a function of the set of the most recent input signals and their associated weights. This will normally include summing and thresholding.

VECTOR
A quantity that has both magnitude and direction.

VIDICON
A type of video camera tube, operating on the cathode-ray tube principle.

VIRTUAL LINES
Imaginary lines linking nearby independent visual tokens, such as dots.

VISUAL ACUITY
The ability to perceive fine detail. Measured by the angle between the stripes in the highest frequency grating which an observer can distinguish from a plain field of the same average brightness as the grating.

VISUAL CORTEX
The region of the mammalian cortex (layer) receiving visual input signals from a retinotopic map of retinal neurons. That is, the visual cortex area of the brain.

VOXEL
A unit of volume in a 3D representation.

WEIGHT
In neural networks, an adaptive coefficient within a processing element (neuron) associated with a single input connection or synapse. The weight determines the intensity (or strength) of the connection or synapse. The term can also refer to one element of a convolution mask.

WELL-POSED
By definition, a problem is mathematically well-posed if and only if the following conditions are met:

1. A solution exists.
2. The solution is unique.
3. The solution is continuously dependent on the data.

WHITE NOISE
Noise whose amplitude is constant across the range of frequencies of interest.

WHOLISTIC
Global or general, especially with respect to perception and the Gestalt psychology.

YAW
Angular displacement of a moving body about an axis perpendicular to the line of motion, and through the apparent vertical of the body.

ZERO-CROSSING
A point at which numerical values of a function change sign.