Spiral Architecture for Machine Vision

by

Phillip Sheridan

Thesis submitted for the degree of

Doctor of Philosophy

UNIVERSITY OF TECHNOLOGY, SYDNEY

School of Computing Sciences

February 29, 1996

Preface

In a time not too long ago and in a land not too far away, there lived a simple people who sought evolvement by application of their sacred myths and symbols. They held that Art and Science were servants of nature and that the medium though which the energies of these two servants flowed and mixed was a third servant known to folks of other lands as games. This manuscript is, in part, a record of selected conversations between two of its inhabitants which took place in the unfolding of one such game.

The story begins one fine morning when Pekanini goes in search of her mentor, Kimo. She is full of excitement with the desire to tell Kimo of the announcement of this year's Challenge of the Third Servant and to begin their assault on it.

As Pekanini stepped onto the beach which had been left smooth by the receded tide, she began her spiral dance. This dance, an art form unique, by all accounts, to this Land, is performed with strong and decisive movements over a carpet of sand. The movements must deliver bird-like flight while simultaneously sketching a pattern of artistic expression by dragging a combination of toes over the surface of the sand.

Half way up the beach, Pekanini broke off the dance to survey her etchings glistening in the moist sand. Only partially pleased with her efforts, she turned and out of the corner of her eye, she saw sitting cross legged in the sand the familiar figure of Kimo, who had been watching her dance from its beginning.

Acknowledgements

The ideas expressed within this thesis are largely a result of a life long exercise in *learning to see*: one in which many people have exerted important influences. It is with considerable pleasure that I take this opportunity to thank those people who have contributed to the exercise. I gratefully acknowledge the valued contributions that resulted from the numerous conversations held with my supervisor, Tom Hintz. This thesis also contains influences that range from those of a subtle nature with a long gestation period to those that are obvious and had an immediate consequence. Although this is not the place to specify the particular contribution of each person listed below, each in their own way has made an input of significance which I value. I have two regrets in attempting to make individual acknowledgements to the following people. The first is that some will never come to know of this fact; the other is that I will undoubtedly think of others too late to be included.

Mildred Asato, Terry Bossomaier, Jo Burns, Ed Burtram, Robert Chen, Errol Chopping, John Coen, Kathleen Collard, Ken Court, John Crossley, Brad Curry, Alex Depodalinsky, Bernard Ellem, Jim Geyer, Orm Gould, Jean Haakenson, Tim Harata, Carel Harris, Burtram Hinke, Jerry Huff, Ann Ito, Ron James, Harry Kenaman, Bruce Keolani, Landy Kia, Mike Low, Hans Lausch, Glenn Lean, Ken Lodge, Graeme MacDougall, Valerie MacDougall, Shauna McKenna, John McKenzie-Low, Xain Ma, Sue Madeley, Wayne Moore, Adele Morey, Anant Nachimuthu, Ed Niske, Vivian Niske, Ed Niven, Ron Niven, Sue Niven, Nancy Pang, Nigel Pearson, Jim Pratley, Gorden Preston, Yan Rek, Ted Rocavert, Emma Sheridan, Mary Jane Sheridan, Walter Sheridan, Edward Silva, Stanly, Ury Szewcow, Linda Tada, Kees Tesselaar, Ed Toma, Marilyn Van Dyke, Lary Walin, Laison Watai, Ed Watanabe.

I would also like to extend a special thanks to my wife, Manon.

Contents

Abstract 1

1 Introduction 3

1.1 The Perplexities of Vision 4

1.2 Machine Vision 5

1.3 Goal of this Thesis 6

1.4 Observation of this Thesis 8

1.5 Significance of this Thesis 11

1.6 Organization of this Thesis 12

2 A Detailed Look at the Theory 14

2.1 Introduction 15

2.2 Related Other Works 17

2.3 The Spiral Honeycomb Mosaic (SHM) 19

2.3.1 Elements of SHM 20

2.3.2 Definitions and Terminology 22

2.3.3 Spiral Counting 25

2.3.4 Operations on SHM 30

2.3.5 Algebraic Properties of SHM 35

2.3.6 Algebraic Significance of SHM 36

2.4 Spiral Honeycomb Lie Algebra (SHLA) 37
2.4.1 Transformations on SHM 38
2.4.2 Lie Multiplication 39
2.5 Interpretation of SHLA 40
2.5.1 Meaningful Transformations 40
2.5.2 SHLA is Computational 50
2.5.3 Invariant Properties 51
2.6 Summary 52

3 Implementation of the SHLA **53**
3.1 Introduction 54
3.2 Transputers 55
3.3 Related Other Works 56
3.4 Data Structure 57
3.4.1 Elements of the Data Structure 57
3.4.2 Operations 60
3.5 Algorithms 63
3.6 Configuration 65
3.7 Simulation 66
3.7.1 Description of the Simulator 67
3.7.2 Simulation Results 68
3.8 Interpretation of Results 72

4 A Segmentation Algorithm for SHLA **74**
4.1 Introduction 75
4.2 Related Other Works 78
4.3 Definitions and Terminologies 78

4.4 Informal Statement of 2DSA 81
4.4.1 The Local Actions 82
4.4.2 The Global Actions 84
4.5 Formal Statement of 2DSA 88
4.6 A Transputer Implementation of 2DSA 91
4.7 Simulation Results 93
4.7.1 Synthetic images 93
4.7.2 Natural image 102
4.7.3 Timing Results 110
4.8 Higher Level Segmentation Algorithms 112
4.9 Discussion 112

Perception of Time and Space **115**

5.1 Introduction 116
5.2 Related Other Works 118
5.3 Perception of Depth 118
5.3.1 A 3-D View of SHLA 121
5.3.2 Segmenting the 3-D SHM 127
5.4 Perception of Motion 133
5.4.1 A Representation of Time 133
5.4.2 Transformations of Time and Space 134
5.4.3 Perception of Depth from Parallaxis 137
5.5 Implementation of 3DSA 138
5.6 Error and Illusions 139
5.6.1 Error from Spatial Mismeasurement 140
5.6.2 Error from Time Mismeasurement 142
5.7 Discussion 146

6 Future Directions **148**

6.1 Akamai 149

6.1.1 Overview of Akamai 150

6.1.2 Image Acquisition 151

6.1.3 Image Analysis Unit 157

6.1.4 Locomotion Unit 162

6.1.5 Hardware for Akamai 162

6.1.6 Methodology 163

6.2 Biological Considerations 164

6.2.1 Understanding Biological Vision 164

6.2.2 The Analytic Component of SHM 166

6.2.3 The Retinal Cone Mosaic 169

6.2.4 Binocular Vision 170

6.2.5 Irregularities in the Cone Mosaic 171

6.3 Summary 173

7 The Next Higher Octave **174**

7.1 Summary 175

7.2 Conclusion 176

Appendix A **178**

Glossary **184**

Other Publications by the Author **187**

Bibliography **188**

Bibliography **190**

List of Tables

1.1 Burt's list of elements that make up a machine vision system 7

3.1 Spiral Addition and Multiplication Tables. 60

3.2 Time (measured in ticks) required to complete transformations of translation, rotation and scaling on transputer networks configured in a pipeline of various lengths. 69

4.1 Cluster Computations. 82

List of Figures

1.1 Distribution of cones on a primate's retina, taken from [23]. 9

1.2 Distribution of hexagonal shaped petals on the flower of a chrysanthemum. 10

1.3 Nerve fibre around the fovea and point of exit at the optic disc showing the spiralling effect, taken from [90] page 63. 10

1.4 The fundamental unit of vision. Seven hexagons arranged such that a centre hexagon is adjacent to six other hexagons. 11

2.1 A fundamental unit of vision. A collection of seven hexagons where the centremost is labelled with address 0 and each neighbour is labelled consecutively. 21

2.2 Step 1. Each of the original addresses from 2.1 are multiplied by 10. . 21

2.3 Step 2 . The cluster from Step 1 is dilated so that six additional hexagons can be placed about each of the addressed hexagons. 22

2.4 Step 3. Each new group of seven hexagons from Step 2 is labelled consecutively. 23

2.5 Collection of $7^3 = 343$ hexagons with labelled address from 0 to 666 in base seven. 24

2.6 An octave. 25

2.7 Octave established by key of 1. 27

2.8 Octave established by key of 10. 27
2.9 The octave established by the key of 15. 29
2.10 Three examples of Spiral Addition and two examples of Spiral Multiplication. 31
2.11 (left) Pal can of dog food represented by 384x288 pixels, (right) shows the relative position of the 7^6 hexagon SHM to the rectangular figure. 41
2.12 Transformation by A_{1000}. The effect is one of translating the image a small distance along the line segment joining addresses 0 and 1000. . 41
2.13 Transformation by A_{10^5}. The effect is one of translating the image along a line segment joining the addresses 0 and 100000. The amount of translation makes visible a *wrap around* effect. 42
2.14 Transformation by M_2. The effect is one of rotation by 60 degrees. . . 43
2.15 Transformation by M_3. The effect is one of rotation by 120 degrees. . 43
2.16 Transformation by M_4, The effect is one of rotation by 180 degrees. . 44
2.17 Transformation by M_5. The effect is one of rotation by 240 degrees. . 44
2.18 Transformation by M_6. The effect is one of rotation by 300 degrees. . 45
2.19 (upper left) original image, (upper right) application of M_{100000} to the original image, (middle left) application of M_{10000} to the original image, (middle right) application of M_{1000} to the original image, (lower left) application of M_{100} to the original image, (lower right) application of M_{10} to the original image. The observable effects in these transformations are 1. rotation, 2. scaling, 3. production of multiple near copies of the original image. 46
2.20 The inverse transformation of M_{15}. The observable effects are 1. rotation, 2. seven contractions of the original image of which six are partial representations of the original image. 48

2.21 The inverse transformation of M_{12}. The observable effects are similar to those of M_{15}^{-1} differing in the amount of rotation, degree of contraction and the number of multiple copies. 48

2.22 A compound transformation, inverse transformation of M_{15} followed by A_{100000}. The partial copies observable in Figure 2.20 have been wrapped around by the translation. 49

3.1 The unwinding of the Spiral Honeycomb Mosaic indicates; i) the one-to-one correspondence between SHM and the integers; ii) the spiral pattern contained within it. 58

3.2 Processors in a pipe, ring and grid. 65

3.3 A flow diagram indicating the communication between the Application Handler and the $n \times H_i$ processes. 71

4.1 A collection of 7 pixels each of which is initialized with four pieces of information as indicated within the detached hexagon. 79

4.2 A flow diagram of the segmentation program. 84

4.3 The diagram indicates the relationship between the active and passive processes of 2DSA for a 49 hexagon SHM. The active processes are marked with a $\star$. 85

4.4 A synthetically generated circle. 94

4.5 The 49 patterns generated by 2DSA for the representation of the circle (Figure 4.4) represented at the lowest resolution. 95

4.6 The 7 patterns generated by 2DSA for the representation of the circle (Figure 4.4) represented at the middle resolution. 96

4.7 The single pattern generated by 2DSA for the representation of the circle (Figure 4.4 represented at the highest resolution. 97

4.8 A synthetically generated annulus. 98

4.9 A synthetically generated trapezoid. 99
4.10 The 7 patterns generated by 2DSA for the representation of the trapezoid (Figure 4.9) represented at the middle resolution. 100
4.11 The single pattern generated by 2DSA for the representation of the trapezoid (Figure 4.9) represented at the highest resolution. 101
4.12 (upper left) original image with 7^6 pixels, (upper right) application of M_{100000} to the original image producing 7 unique copies each of which contains 7^5 pixels, (middle left) application of M_{10000} to the original image producing 49 unique copies each of which contains 7^4 pixels, (middle right) application of M_{1000} to the original image producing 7^3 unique copies each of which contains 7^3 pixels, (lower left) application of M_{100} to the original image producing 7^4 unique copies each of which contains 49 pixels, (lower right) application of M_{10} to the original image producing 7^5 unique copies each of which contains 7 pixels. 103
4.13 The image represents a scaling up of one of the copies from 4.12 (middle right). 105
4.14 The collection of images represent six iterations of the segmentation algorithm as applied to the image from Figure 4.13; (upper left) first iteration at highest δ grey-value producing 2 conjugacy classes, (upper right) second iteration, (middle left) third iteration, (middle right) fourth iteration, (lower left) 8th iteration, (lower right) 12th iteration. 106
4.15 The image represents a scaling up of one of the 49 copies in Figure 4.12 (middle left). 108

4.16 The collection of images represents six iterations of the segmentation algorithm applied to the image in Figure 4.15; (upper left) first iteration, (upper right) second iteration, (middle left) third iteration, (middle right) fourth iteration, (lower left) fifth iteration, (lower right) sixth iteration. 109
4.17 The image represents a scaling up of one of the 7 copies in Figure 4.12 (upper right). 110
4.18 The collection of images represents four iterations of the segmentation algorithm as applied to the image of Figure 4.17; (upper left) first iteration, (upper right) second iteration, (lower left) third iteration, (lower right) fourth iteration. 111

5.1 The projection of two images at different distances from the observer onto the retina. 116
5.2 A point P at a distance Z from the base line joining points E_l and E_r. 119
5.3 The geometry of stereo vision. 120
5.4 The figure indicates the relationship between the positions of seven clusters of the SHM to the seven units of 3-D space that they represent. The optical axis intersects each of the seven clusters. 123
5.5 The optical axis intersects the centre of each hexagonal end of a hexahedron which is a component of the visual world. 124
5.6 The figure indicates the relative movements of the seven clusters from Figure 5.4 when a transformation of rotation is applied to the SHM. . 128
5.7 Horizontal cross section of a simplified left eye viewing a patch P in the visual world. The three photo receptors are labelled 1, 2 and 3. . 129
5.8 Cross section of the right eye's view of the same patch P displayed in Figure 5.7. 130

5.9 The combined effects of Figure 5.7 and 5.8 shows the patch locked in the intersection of two corridors. 131

5.10 (upper) Trapezoid in a plane which is perpendicular to the optical axis of the observer. (lower) Same trapezoidal shape as in (upper) but with shading, giving the illusion of a rectangle in space tilted away from the perpendicular to the optical axis. 132

5.11 The figure indicates (right) the motion of an object along a circular path and (left) the representation of the position of the object in SHM at seven units of time. 135

5.12 The figure indicates (right) the motion of an object along the path of a vertical line and (left) the representation in SHM of the object's position at seven units of time. 135

5.13 The figure indicates (right) a stationary object and (left) the representation in SHM of the stationary object over seven units of time. . . . 136

5.14 The figure shows an object (darkened hexagon) in the centre of the visual field. 137

5.15 The curvature of the can produces an occluded strip from E_r which is visible to E_l. 141

5.16 Binocular view of swinging pendulum with both eyes receiving unfiltered light. 143

5.17 One eye, E_r, receives filtered light. The perceived motion of the pendulum is along the path of an ellipsoid. 144

5.18 Showing the actual position of a swinging pendulum at two points in time and the perceived position of a single point in time. 145

6.1 Architectural overview of Akamai. 151

6.2 Cone distribution along the horizontal meridian in humans. Adapted from [23]. 153
6.3 Distribution of a non-uniform sampling scheme. The shaded hexagons of the combined three fundamental units of vision in regions, 1, 2 and 3, indicates the distribution of one such scheme. 154
6.4 A collection of hexagons is partitioned into two equivalence classes. The unshaded hexagons constitute an open class and the collection of shaded hexagons is an example of a closed equivalence class. 158
6.5 Viewing this figure induces the perception of a rectangle due to the presence of a corner etched in each of the four blobs. 159
6.6 Two sinusoidal curves with the same amplitude and different frequencies appear as aliases when sampled by the array of photo receptors of the figure. 169

Abstract

Spiral Architecture for Machine Vision

Phillip Sheridan

This thesis presents a new and powerful approach to the development of a general purpose machine vision system. The approach is inspired from anatomical considerations of the primate's vision system. The geometrical arrangement of cones on a primate's retina can be described in terms of a hexagonal grid. The importance of the hexagonal grid is that it possesses special computational features that are pertinent to the vision process. The fundamental thrust of this thesis emanates from the observation that this hexagonal grid can be described in terms of the mathematical object known as a Euclidean ring. The Euclidean ring is employed to generate an algebra of linear transformations which are appropriate for the processing of multidimensional vision data. A parallel autonomous segmentation algorithm for multidimensional vision data is described. The algebra and segmentation algorithm are implemented on a network of transputers. The implementation is discussed in the context of the outline of a general purpose machine vision system's design.

Chapter 1

Introduction

Shouting to him as she ran, "Kimo, Kimo, this year's Challenge of the Third Servant is to create a machine endowed with vision."

After considerable conversation on the issue, the pair fell into silence before Kimo spoke. "We might be well served by commencing with a search for the simplicity that emanates from those things we wish to see."

1.1 The Perplexities of Vision

One of the grand challenges in computer science is to produce a machine that is capable of emulating certain aspects of intelligent human processes. An example of such a process that has remained particularly elusive is *vision*. Fundamentally, vision is the process of recognising objects of interest from an image. The perplexities that surround this process are immense not the least of which is the perplexity of the nature of the objects themselves. Martin Gardner, in [32] put it as follows:

> If the cosmos were suddenly frozen, and all movement ceased, a survey of its structure would not reveal a random distribution of parts. Simple geometrical patterns, for example, would be found in profusion - from the spirals of galaxies to the hexagonal shapes of snow crystals. Set the clockwork going and its parts move rhythmically to laws that can often be expressed by equations of surprising simplicity. And there is no logical or *a priori* reason why these things should be so.

Given this powerful statement describing the structure of the *objects of interest*, how then does a biological vision system recognise these objects? The perplexities surrounding the recognition of objects have been addressed by many people from many quarters [77]. These range from the epistemological concerns of the Gestalt[1] *laws of visual perception* formulated at the turn of this century [57] to the analytical tools of the discipline of modern pattern recognition[2] [9].

Finally, one comes to the perplexity of the *process* itself. The perceptual psychologist J. J. Gibson [34] described this process as follows:

[1]Gestalt is a school of psychology whose origins go back to the beginning of the 20th century. A principal tenet of this school is that an organism must be viewed as an organised whole in which every part affects each of its other parts. Their work in this regard is largely qualitative.

[2]The theory of pattern recognition is a highly mathematical discipline. This quantitative approach is in contrast to the Gestalt school.

> When no constraints are put on the visual system, we look around, walk up to something interesting and move around it so as to see it from all sides, and go from one vista to another.

Gibson describes this *active* process in terms that suggest *it is more than the sum of its parts.* That is, if the process of vision is to be fully understood, not only must the various components of vision, eye [53], brain [72, 110], light, objects being viewed [41], be understood, but all of the components must be understood as a totality in relationship to one another. Therein lies the difficulty. Biological vision systems have evolved in response to a multitude of subtle influences some of which may no longer be evident today. Consequently, the act of now trying to deduce the effects of these influences on the *totality* of vision may well be impossible.

1.2 Machine Vision

The emergence of the pursuit to create a machine capable of vision is a rather recent event. Given the *nature of vision*, it is easy to appreciate Barrow's [8] view of the progress made in this endeavour. Although this statement was made in 1978 it is equally pertinent today:

> Despite considerable progress in recent years, our understanding of the principles underlying visual perception remains primitive. Attempts to construct computer models for the interpretation of arbitrary scenes have resulted in such poor performance, limited range of abilities and inflexibility that, were it not for the human existence proof, we may have been tempted long ago to conclude that high performance, general purpose vision is impossible.

This dire statement was used by Charniak in [18] to introduce a discussion of an overview of both the *vision problem* and the classical approaches taken in its *solution.*

Barrow's statement also explains why the path to the solution of the vision problem has generated two schools of thought on the matter: the *reconstructive* and the *purposive*. The former works from the belief that the organising principles inherent in biological systems have optimised the mechanism for processing visual information [51, 96]. The latter group believes that the silicon based machine is so fundamentally different from the carbon based biological systems that the task is better served by attempting to solve specific vision application problems independent of biological considerations. However, some researchers [45, 97] including this author suggest that there is a productive middle ground between these two schools. This middle ground might also be well served to incorporate the principles of an inter-disciplinary approach; one which involves the expertise of electrical and mechanical engineers, neuro human anatomists and neurophysiologists, computer scientists, perceptual psychologists, physicists and mathematicians. In any case, there is at least some general agreement of the elements of a powerful machine vision system [19, 76, 98]. These elements can be classified as *representation* issues and *control* issues. The issues are summarized in Table 1.1 and have mostly been taken from [16] where they are discussed in detail.

1.3 Goal of this Thesis

The goal of this thesis is to unveil a new approach to machine vision; one with sufficient power required for the development of a general purpose machine vision system. The approach will traverse a path between the *reconstructive* and *purposive* schools of thought on machine vision. It will also embrace two further essential principles:

i) A vision system is an organized whole in which each individual part affects every other.

Table 1.1: Burt's list of elements that make up a machine vision system

Representation :

i) **Controlled Resolution** Use the lowest resolution for the task

ii) **Restricted Windows** Restrict high resolution analysis to windows of interest

iii) **Integrated Measures** Represent feature measures at reduced positional resolution

iv) **Compressed Ranges** Represent samples with few bits

Control :

i) **Peripheral Alerting** Monitor the full field at low resolution to detect critical events

ii) **Model-Based Guidance** Direct foveal analysis to resolve ambiguities and verify hypothesis

iii) **Coarse-to-Fine Analysis** Compute rough estimates at low resolution, then refine at higher resolutions

iv) **Concurrency** Independent tasks should be processed concurrently

ii) A general purpose machine vision system must operate without *a priori* information about the environment that it explores.

The first of these principles simultaneously embodies the spirit of concurrency in parallel processing and is compliant with the fundamental tenet of the Gestalt view of vision. The second principle expresses the essence of the vision process.

The approach initiates from a key observation which was made by the author regarding the shape and distribution of the cones on the retina of a primate. It is the properties of this shape and its arrangement that provides a crucial component of the approach to machine vision taken in this thesis. The key observation leads to the development of a new data structure and the development of new algorithms which are capable of supporting the two principles of this approach to machine vision.

1.4 Observation of this Thesis

Theodor Schwenk [89] has investigated in great depth the formative forces of water and air and in the process he came to certain conclusions regarding living forms and their origins. In particular, he suggests that the retina of the human eye, as mediator between the visual world and the brain, embodies a form which is compatible with the structure of the visual world and the function of the brain. To illustrate this, consider the shape and arrangement of the petals of a chrysanthemum. They bear a striking resemblance to the shape and distribution of the rods and cones on the retina of a primate's eye. See Figures 1.2 and 1.1. According to Schwenk, this structural resemblance is not merely a curious coincidence; both entities emerge from the archetypal pattern of the spiral which is associated with the growth of an organism.

One of the evolutionary effects on the human vision system pertains to the retina which is the photo receiving device at the back of the eye. On the top layer of the

retina are rods and cones which are the photo receiving elements of the retina. There is no longer any doubt as to the shape of the cones on the primate's retina. They have the shape of hexagons and are arranged as a triangular mosaic [106]. It is the observation of the thesis that the rods and cones are arranged in spiralling *rosette like* clusters of seven. Figure 1.1 is a highly magnified view of the retina and displays this geometrical arrangement.

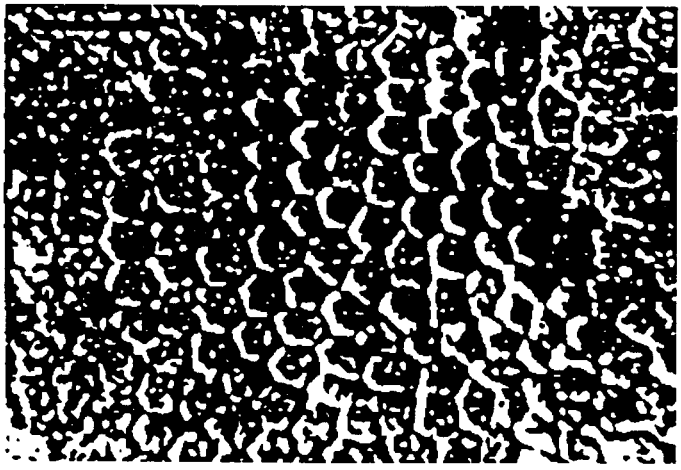


Figure 1.1: Distribution of cones on a primate's retina, taken from [23].

This observation pertains to functional implications of the spiralling clusters of cones on the primate's retina. The geometry of this distribution has inherent organization and is endowed with powerful computational features. The cones have the shape of hexagons and are arranged in spiralling *rosette-like* clusters. Figure 1.3 displays the spiralling pattern of the nerve fibre connecting the rods and cones in the retina. The *rosette-like* cluster is composed of seven hexagons and is an organizational unit of vision. It will be referred to as *the fundamental unit of vision* (see Figure 1.4) and it will play a critical role throughout this thesis. This disarmingly simple observation gives rise to three geometrical properties of the hexagons:

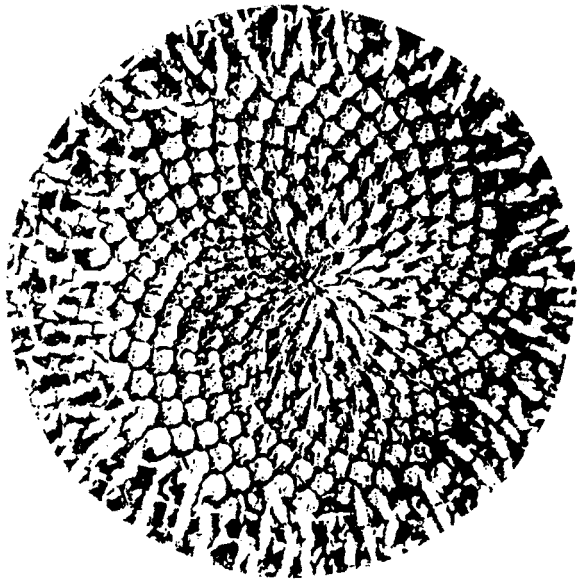


Figure 1.2: Distribution of hexagonal shaped petals on the flower of a chrysanthemum.

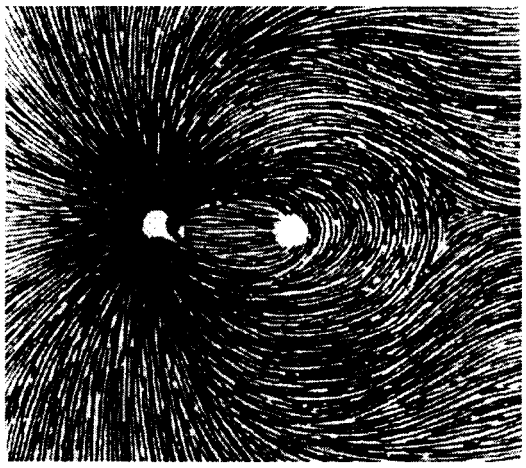


Figure 1.3: Nerve fibre around the fovea and point of exit at the optic disc showing the spiralling effect, taken from [90] page 63.

i) The distances from the centre hexagon of the *fundamental unit of vision* to each of its six surrounding hexagons are all equal.

ii) Each of the six cusps of the centre hexagon on the *fundamental unit of vision* is a junction where exactly three hexagons join.

iii) Any hexagon on a uniform grid of hexagons may be considered as the centre hexagon of a *fundamental unit of vision.*

There is no other geometrical object which possesses these three properties.[3] This observation will be exploited to the extent that the development of all of the original work presented in this thesis can be traced back to it.

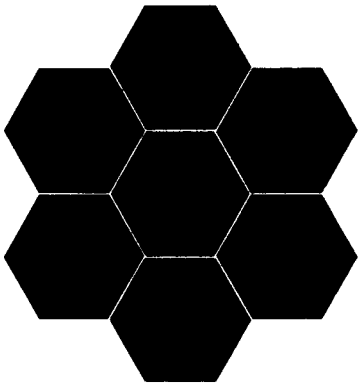


Figure 1.4: The fundamental unit of vision. Seven hexagons arranged such that a centre hexagon is adjacent to six other hexagons.

1.5 Significance of this Thesis

The most significant aspect of this thesis is that it provides access to the computational power required for machine vision that hitherto has not been available. This

[3]Although no mathematical proof of this statement is given, it is a simple matter to check its validity with regards to the classical forms of spatial data structures which are discussed in [86].

access is a direct result from the *observation of the thesis* which is:

> **The three geometrical properties of the hexagons can be employed to describe the spiralling clusters of cones on a primate's retina in terms of the mathematical object known as a Euclidean ring.**

The existence of this Euclidean ring makes possible the construction of a rich collection of other mathematical objects that take the form of image transformations. The collective properties of these transformations results in the formulation of a new data structure that possesses considerable computational utility. This data structure can be described in terms of the mathematical object known as an algebra. This data structure provides access to computational features inherent in the organization of the primate's retina. It is important for two reasons:

- The data structure provides the means by which a solid theoretical foundation that unifies the wide range of image transformations in a manner consistent with considering the process of vision as a totality can be formulated.
- The data structure can be practically implemented as a fundamental component of a general purpose machine vision system.

1.6 Organization of this Thesis

Chapter 2 provides the theoretical foundations on which the rest of the thesis is based. A new data structure for machine vision, named Spiral Honeycomb Lie Algebra (SHLA), will be described. This data structure is inspired from the geometry of the retina of primates and is consistent with the requirements of machine vision as described in Table 1.1. Chapter 3 will describe an implementation of SHLA on

a parallel architecture of a network of transputers. The main result of Chapter 4 describes an algorithm to perform one of the essential components of machine vision, known as segmentation. It will be shown that the segmentation algorithm can be implemented on a network of transputers that is compatible with the implementation of SHLA. Chapter 5 will extend the results of Chapter 4 to multidimensional visual data. In particular, it will outline an algorithm that can be employed in the perception of motion in three-dimensional space. Chapter 6 discusses the future directions of this work. It explores the implications of SHLA from two perspectives. The first describes the design of a general purpose machine vision system which is based on SHLA and is in keeping with the principles of Table 1.1. The second speculates on some open questions in the field of biological vision systems. Chapter 7 provides the concluding remarks to this thesis. There are two appendices. The first will provide the mathematical details of the proofs to the theorems stated in Chapter 2. The second contains a glossary of a few of the more peculiar terms that have invaded the field of machine vision.

Chapter 2

A Detailed Look at the Theory

On the next morning, Pekanini and Kimo described the patterns they each had seen on their journey though the mountains the previous day.

Reflectively, Kimo then said, "Now we must describe these patterns with numbers and to do this one must learn to count."

2.1 Introduction

The theoretical foundation on which the results of this thesis are based is presented in this chapter. The primary goal of this chapter is to describe a new data structure for machine vision in both qualitative and mathematical terms. The name given to this data structure is Spiral Honeycomb Lie Algebra (SHLA). The first part of the name, Spiral Honeycomb, is derived from a description of the structure from which it was inspired - a primate's retina. The arrangement of cones on the primate's retina forms a pattern of spiralling clusters of hexagons.

The importance of this arrangement of spiralling clusters, apart from its intriguingly beautiful geometry, is that it possesses powerful computational advantages to machine vision. This computational power associated with this arrangement of hexagons emerges from the algebraic structure which will be unveiled in this chapter. This structure is a Lie Algebra and provides the second part of the name, SHLA.

In essence, SHLA is a rich collection of image transformations on a space which is composed of hexagons. The hexagons hold the picture elements of an image. These transformations are visually meaningful because they achieve rearrangements of an image in a manner compatible with the way an object is perceived to move in the visual world. A crucial characteristic of this collection of transformations is that for any given sequence of transformations applied to an image there is a unique transformation that achieves the same final result of the sequence when applied to the original image. Thus, any visually meaningful rearrangement of the hexagons can be obtained directly from any given arrangement. It is for this reason that SHLA can be considered as a *clustering structure*[1] as opposed to a hierarchical structure such as a Quadtree, which has a definite starting and ending point in its representations.

[1]The author is unaware of the existence of the term *clustering structure* in the machine vision literature. As such it is anticipated that the term will connotate certain geometrical qualities which will contrast it with the geometrical qualities of other structures.

In Chapter 1, reference was made to Table 1.1 presenting the criteria for an active machine vision system. One of the criteria, multiple resolution representation, relates to the problem of extracting information about a scene for which no *a priori* information is known. The recognition of many objects is related to their size. For example, the representation of a person makes sense only over a range of sizes. If that representation is too small, it might be classed as a doll. On the other hand, if it were too large, it may be perceived as a monster. Consequently, a vision system must be able to reconcile the scale of an image. This problem amounts to the selection of an appropriate scale to start the recognition process. One strategy is to start the process at all possible scales concurrently and let the data be the driving force of the recognition process. SHLA provides the means by which such a strategy can be implemented. With regards to the representation of the image at different scales, SHLA has a number of important features that result from its algebraic structure:

i) all possible scales are achievable with equal computational complexity,

ii) all scales, contractions or magnifications, can be obtained from any single scale without intermediate transformations,

iii) any number of scalings can be performed without degrading the input image, and

iv) a change of scale never results in the loss of information contained in the representation.

It is important to emphasise the point that, although the SHLA was inspired from the organising principles inherent in the retina of primates, no claim is made that SHLA is a model of the primate vision system. Rather, SHLA is explored from the point of view of a computational tool for use on a computer. The speculation, relating SHLA to the psychophysics of human perception, will be discussed in Chapter 6.

Section 2.2 of this chapter places SHLA in the context of other works; indicating similarities and differences to other data structures for machine vision. Section 3 describes a structure called the Spiral Honeycomb Mosaic, (SHM) which is then employed to impart the main result of the chapter, in section 4. The chapter concludes with a summary and discussion of the main points presented in the chapter. The main results of the chapter are presented in the form of theorems. However, the proofs of the theorems do not lead to a greater understanding of the ideas expressed so the proofs will be relegated to Appendix A.

2.2 Related Other Works

The facility of SHLA to produce multiple scale representations has certain similarities to the scale space concept as introduced by Witken. The literature on the subject is now extensive and Lindeburg provides an excellent treatment of it in [60]. However, one of the major differences between SHLA and the scale space is that SHLA does not *smooth* the data to obtain the scale; it obtains the effect by merely re-grouping the data. The use of hexagons as the elements of an image representation is certainly not new. This structure has been described by Burt [15]. It has been used in the context of Generalized Balanced Trees (GBT) by Gibson and Lucas [36]. Samet in [86] discusses this structure in the context of spatial Quadtrees in which certain considerations of the shape of the elements (hexagon) leads to the conclusion that the data structure has severe limitations for its practical use. This is an assertion which this thesis refutes. According to Samet, the primary limitation results from the requirement that the edges of certain elements would need to correlate to the edges of objects in the image. This in turn has implications for the segmentation process. The segmentation process of SHLA, which will be treated in Chapter 4, does not have the requirement of edge correlation. The covering properties of the hexagon have been

exploited in the design of an optical telescope to improve focusing as described in [59]. Ahuja in [2] discusses the role of the hexagon in the context of hierarchical structures.

It is important to realize that the hierarchical representations are fundamentally different from a clustering representation such as SHLA. Hierarchical approaches attempt to fit recursively subdividible elements optimally into regions. The decomposition of the elements takes place at lower levels in order to obtain a better *fit* of the elemental shapes (squares, triangles, hexagons, etc). It has a starting level. Any given level of the hierarchy is derived from the immediate predecessor. The process is generally not reversible.

By contrast, the role of the pixel's shape in SHLA is immaterial. It is equally consistent to consider these elements as *points* at a location rather than the shape of a particular two dimensional object. Also the multi-scale representations of SHLA have arbitrary starting points but not necessarily ending points. Any scale can be derived directly from any other scale and the process is reversible. Most importantly, the transformations on SHLA enable multiple scale representation of an image without the inefficiencies of a hierarchy.

The use of Lie Algebras in both the modeling of the human vision system and as a tool in machine vision is not new. Hoffman in [46] uses Linear Transformation Groups as elements of a Lie Algebra to model the human vision system. Although there are some striking parallels of SHLA and Hoffman's model, the discussion of this will be deferred to Chapter 6. Gu in [39] reviews the use of Lie Algebras in robotic vision for the purpose of perceiving motion in three dimensional space. All of the Lie Algebras mentioned by Gu operate on a real or complex-valued space and consequently suffer the *rounding* effects of reconciling a continuous space to a discrete machine. SHLA, on the other hand, is a purely discrete data structure. Its elements are discrete and the operations performed on its elements are discrete.

There have also been a wide ranging number of approaches that attempt to capture the scale invariance property of the primate vision system. A few of these are: Ebrahimi in [29] discusses the application of the Gabor Transform to data compression; Rojer and Schwartz in [80] employ a complex logarithmic geometry to achieve a space invariant transform. One of the problems with the logarithmic geometry is that what it gains with the scale invariance it loses in the translation process [70, 17].

The use of wavelet analysis has been explored in [47]. It employs data smoothing to fit curves to the internal boundaries of an image. This smoothing process results in the discarding of information from the image.

Spiral functions applied to image processing representations is not new. Such issues are discussed by J. Bigun in [11]. In contrast, the important feature of the spiral function associated with SHLA is that it is a discrete function.

2.3 The Spiral Honeycomb Mosaic (SHM)

This section will describe the fundamental mathematical structure which underpins the main theoretical results to be presented in this chapter. The name given to this structure is Spiral Honeycomb Mosaic (SHM). It turns out that SHM has the structure of the important mathematical object known as a Euclidean ring.[2] It is this rich algebraic structure that provides the tool to create the transformations that will be presented later in this chapter.

The development of SHM will unfold in the following five steps. The first step is to describe the elements of SHM. A collection of definitions and terminology will be developed for use in subsequent steps. The third step will reveal the symmetries of a process called Spiral Counting. The fourth step will formalise Spiral Counting to

[2]The concept of a Euclidean ring is of fundamental importance to mathematics. By analogy, it is to many branches of mathematics what electricity is to the light bulb. It is also of crucial importance to this thesis.

obtain the algebraic operations on SHM. This section culminates with a summary of the results.

2.3.1 Elements of SHM

Intuitively, the elements of SHM may be considered as the collection of pixels (picture elements) of an image. These elements correspond to the position of the photo receiving cells of the image capturing device. In the case of the human eye, these elements would represent the relative position of the rods and cones on the retina.

Assume that each pixel is in the shape of a hexagon with its own unique "address" and light intensity. This location corresponds to a point in the visual world, and the light intensity corresponds to the amount of light emanating from that point in the visual world. Ultimate knowledge of the visual world is derived from the relative positions of the light intensities captured at the hexagons.

The most important aspect of each hexagon is not its six sides but that it has six neighbouring hexagons. This establishes the property that for all hexagons in a grid of hexagons, the centre of each hexagon is a constant distance from every one of its six neighbours[3]. An algorithm is used to determine the address of each hexagon. The property of *equal distance* between adjacent hexagons is the source of a deeper reason for the importance of this data structure. The hexagonal shape of the elements has relevance only at the photo receiving stage of the process[4]. In order to reinforce the idea of a particular grouping of seven elements, the addresses of these elements (centres of hexagons) will be referred to as hexagons.

The first task at hand is to label each of the individual hexagons with a unique

[3]It is a simple matter to confirm this property of the fundamental unit of vision by performing the following exercise. Use a compass to draw a circle of some desired radius on a sheet of paper. At an arbitrary point on the circle use the compass to construct another circle of the same radius. At all points where two circles intersect construct another circle. Each point of intersection represents the centre of a hexagon.

[4]A group of seven polygons achieves optimal covering when each polygon is a hexagon.

address. This will be achieved by describing a process that begins with a collection of seven hexagons.

Each of the hexagons is labelled consecutively in base seven with address 0, 1, 2, 3, 4, 5, 6 as displayed in Figure 2.1.

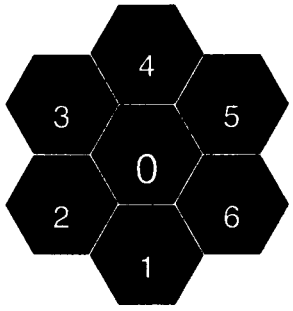


Figure 2.1: A fundamental unit of vision. A collection of seven hexagons where the centremost is labelled with address 0 and each neighbour is labelled consecutively.

Repeatedly apply the following three steps as illustrated in Figures 2.2, 2.3 and 2.4.

Step 1 Multiply the address of each hexagon by 10. See Figure 2.2

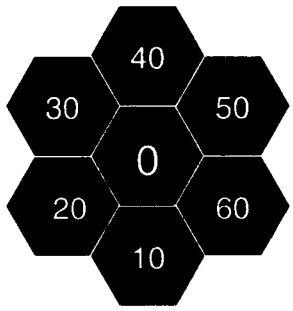


Figure 2.2: Step 1. Each of the original addresses from 2.1 are multiplied by 10.

Step 2 Dilate the structure so that six additional hexagons can be placed about each addressed hexagon. See Figure 2.3

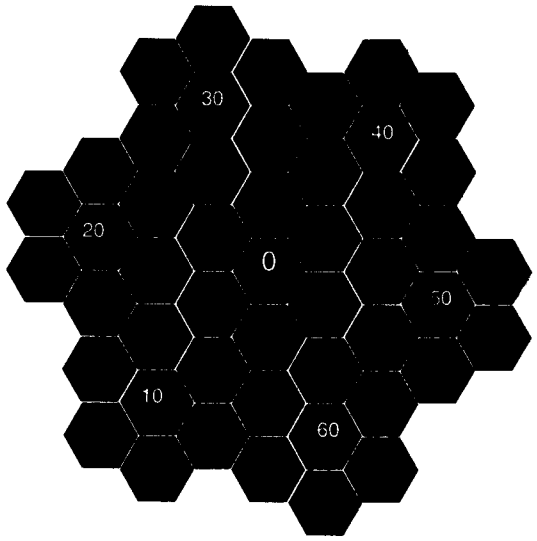


Figure 2.3: Step 2 . The cluster from Step 1 is dilated so that six additional hexagons can be placed about each of the addressed hexagons.

Step 3 For each new cluster label each of the six enclosing hexagons consecutively from the centre address. See Figure 2.4

The repetition of the steps permits the collection of hexagons to grow in powers of seven with uniquely assigned addresses. It is this pattern of growth that generates the spiral. Further, the addresses are consecutive in base seven. Figure 2.5 displays the results of applying the inductive step for the third time.

2.3.2 Definitions and Terminology

In order to facilitate both the development of the data structure in this chapter and the narrative throughout the thesis, the following definitions and terminologies will be employed. Some of the terms presented here originate in what might seem to be

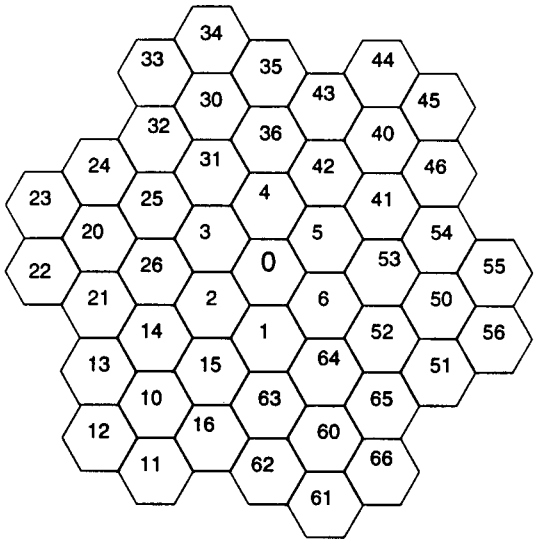


Figure 2.4: Step 3. Each new group of seven hexagons from Step 2 is labelled consecutively.

unrelated contexts. However, it is intended that the intuition associated with the common usage of such terms will enhance the understanding of the concepts being presented.

Definition 2.1 A **cluster** is an arrangement of six hexagons around a centre hexagon with each hexagon a constant distance from the centre hexagon. See Figure 2.1

Definition 2.2 An **octave** is a sequence of eight hexagons each of which has associated with it one of the notes *doh, ray, me, fah, soh, lah, te, doh.* The first seven notes correspond to the addresses of a cluster. The eighth note (second *doh*) corresponds to the address of the centre hexagon of a cluster determined by the following rule: move from *te* to *ray* and then in the same direction twice again. See Figure 2.6

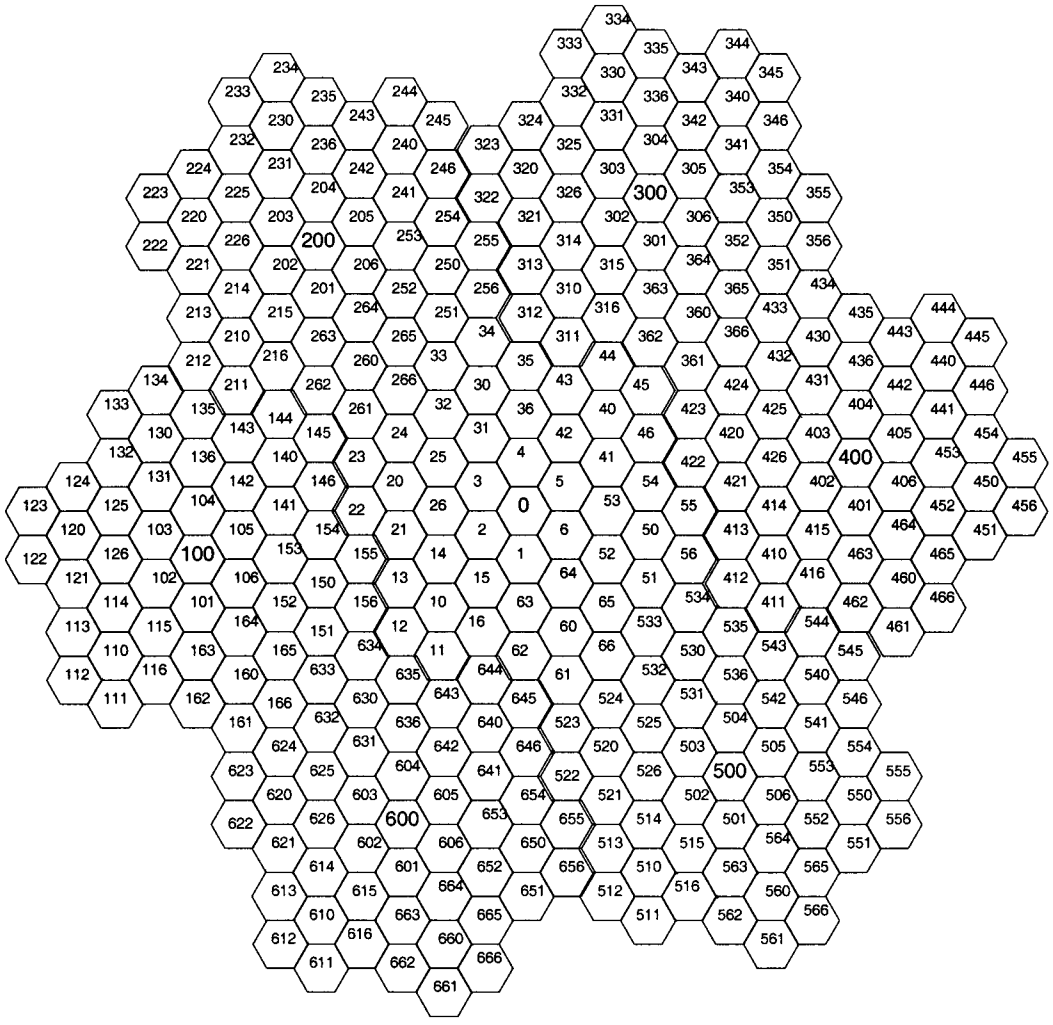


Figure 2.5: Collection of $7^3 = 343$ hexagons with labelled address from 0 to 666 in base seven.

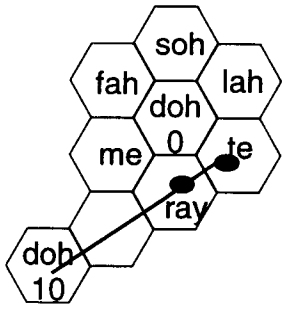


Figure 2.6: An octave.

Associated with an octave is the concept of the key.

Definition 2.3 The **key** of an octave is the address of *ray* in the octave. It has two numbers associated with it. The first number is the *metre* which is the length of the line segment joining the centre of the hexagon at the first *doh* to the centre of the hexagon at *ray*. The second number is the *orientation* of the line segment relative to some fixed position. The key completely determines the octave.

The next definition facilitates the concept of linking two distinct octaves.

Definition 2.4 A **next higher octave**, of a given octave, is that octave obtained by taking as its *ray*, the second *doh* of the given octave.

With these definitions in hand, the fundamental concept underpining the *visually meaningful* transformations that constitute SHLA can be described.

2.3.3 Spiral Counting

Spiral Counting is an algorithm that designates a sequence of hexagons in SHM. It may be thought of in qualitative terms as a process of movement through the SHM, tracing a pattern prescribed by a particular octave. This is a spiralling movement that has a commencing hexagon, counts a pre-determined number of hexagons and

terminates at a distinct hexagon. This is accomplished by recursively applying the pattern of a given octave. The recursive process works as follows:

Given the three inputs - key, commencing address and the number of hexagons to be counted-

i) Establish the pattern of a primary octave from the key. Let the commencing address coincide with the first *doh* of the octave. Call this the present octave.

ii) Specify the sequence of address of the SHM which corresponds to the notes of the present octave. Call this the established pattern.

iii) Specify the next higher octave and at each one of the notes *ray* to *te* employ the established pattern to continue the count.

iv) Let the pattern, which was determined by the counting to this point, be the established pattern and let the present octave be the next higher octave.

v) Repeatedly apply steps iii) and iv) until the number of hexagons to be counted has been reached.

To illustrate how the algorithm is applied, consider the following three examples of spiral counting.

First Example Spiral counting in the key of 1 commencing at address 0:

i) The present octave is established by the key of 1 as displayed in Figure 2.7.

ii) The first seven counts associated with the pattern of the present octave produces visits at the following sequences of addresses: 0, 1, 2, 3, 4, 5, 6.

iii) The next higher octave is established by the key of 10. See Figure 2.8. The addresses associated with this octave are
0, 10, 20, 30, 40, 50, 60, 100. Employing the established pattern at each

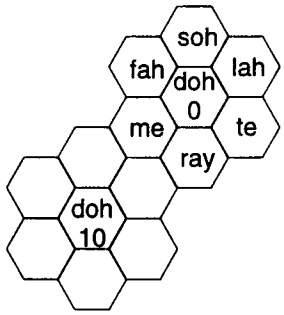


Figure 2.7: Octave established by key of 1.

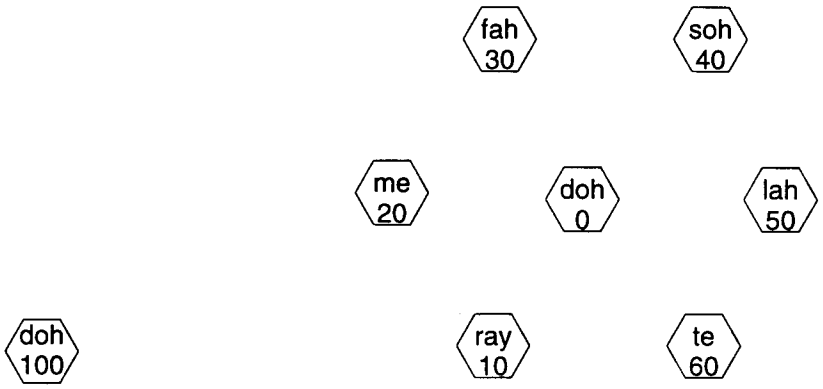


Figure 2.8: Octave established by key of 10.

of the notes *ray* through *te* results in visits at the following sequence of addresses:

10, 11, 12, 13, 14, 15, 16,

20, 21, 22, 23, 24, 25, 26,

30, 31, 32, 33, 34, 35, 36,

40, 41, 42, 43, 44, 45, 46,

50, 51, 52, 53, 54, 55, 56,

60, 61, 62, 63, 64, 65, 66

A close inspection of Figure 2.5 will reveal the pattern of this sequence of addresses.

iv) The address associated with the last note of the octave, 100, once again establishes the key for the next higher octave.

Second Example Consider the process of Spiral Counting in the key of 1 from a non-zero commencing address. Spiral Counting in the key of 1 from address 6 proceeds as follows (the octaves are the same as those displayed in Figure 2.7 and Figure 2.8):

i) The first octave is the same as the previous example.

ii) The addresses associated with this octave are 6, 64, 1, 0, 52, 65, 16.

iii) The addresses associated with the second octave are 6, 16, 26, 36, 46, 56, 66, 106. The counting associated with this octave yields visits at the following sequences of addresses:

ray: 16, 644, 11, 10, 15, 63, 62,

me: 26, 14, 21, 20, 25, 3, 2,

fah: 36, 4, 31, 30, 35, 43, 42,

soh: 46, 54, 41, 40, 45, 423, 422,

lah: 56, 524, 51, 50, 55, 413, 412,

te: 66, 524, 61, 60, 65, 533, 532

iv) 106 establishes the key of the next higher octave

Third Example Consider Spiral Counting in the key of 15 from address 0.

i) The key of 15 establishes the pattern of the primary octave as displayed in Figure 2.9.

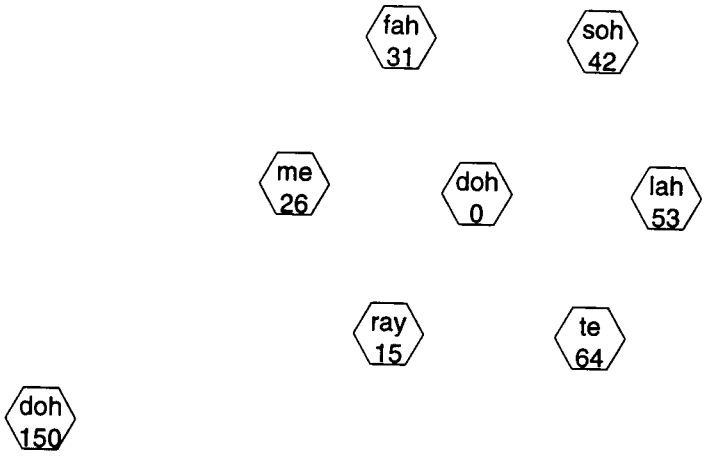


Figure 2.9: The octave established by the key of 15.

ii) The sequence of addresses associated with the primary octave are 0, 15, 26, 31, 42, 53, 64, 150

iii) The key of the next higher octave is 150. The addresses associated with the next higher octave are 0, 150, 260, 310, 420, 530, 640, 1500

The sequence of addresses to be visited are

ray: 150, 165, 106, 141, 22, 13, 634,

me: 260, 216, 201, 252, 33, 24, 145,

fah: 310, 321, 302, 363, 44, 35, 256,

soh: 420, 432, 403, 414, 55, 46, 361,

lah: 530, 543, 504, 525, 66, 51, 412,

te: 640, 654, 605, 636, 11, 62, 523

iv) The second repetition of steps iii) and iv) commences with the key of the next higher octave which is 1500.

2.3.4 Operations on SHM

SHM, as it stands at this point, is merely a collection of elements endowed with a certain symmetry. Spiral Counting will now be used to define two operations on SHM that will give it the desired structure of a Euclidean ring.

Define Spiral Addition of address a and b as:

$$a + b = \textit{that address found by spiral counting b addresses in the key of } 1 \textit{ from } a. \tag{2.1}$$

Define Spiral Multiplication of a by b as:

$$a \times b = \textit{that address found by spiral counting b addresses in the key of } a \textit{ from } 0. \tag{2.2}$$

A few examples will serve to illustrate the processes of Spiral Addition and Spiral Multiplication before considering the arithmetic modular form of these operations. The following examples would be most easily understood by tracing the particular octave pattern through the sequences of address on Figure 2.10.

Examples of Spiral Addition :

i) 2 + 3 = that address found by counting 3 addresses in the key of 1 from 2. The first address from 2 is 15. The second is 14 and the third is 26.

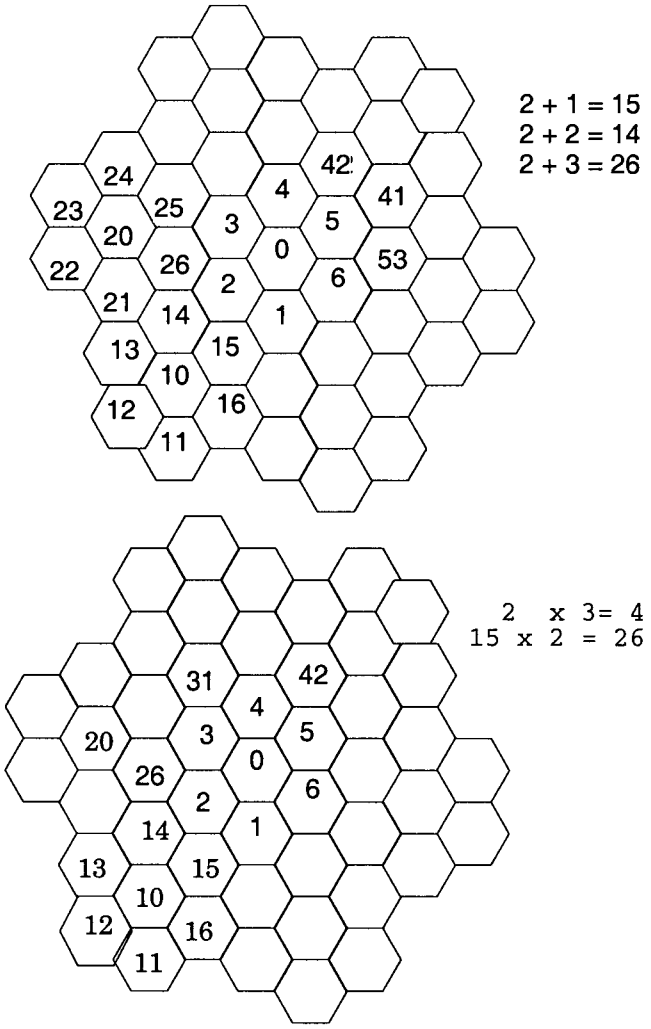


Figure 2.10: Three examples of Spiral Addition and two examples of Spiral Multiplication.

Thus 2 + 3 = 26.

ii) 5 + 12 = 10, which is the 12th address in the sequence 6, 0, 4, 42, 41, 53, 15, 16, 10.

Examples of Spiral Multiplication :

i) 2 x 3 = that address found by Spiral Counting in the key of 2 from 0 for 3 counts. The sequence of three addresses is 2, 3, 4.

Thus $2 \times 3 = 4$.

ii) $15 \times 2 = 26$, which is the address found by Spiral Counting in the key of 15 from 0 for 2 counts. The sequence of addresses is 15, 26.

Although the cardinality of SHM is infinite, in practice the use to which the SHM will be put is that of a finite space with modular forms of the arithmetic defined. The modular arithmetic, in turn, results from a modular form of Spiral Counting. Intuitively, the modular operations are a *pruning and grafting* process. That is, the collection of elements that would be moved outside of the designated range (the *prune*) by an operation, must be mapped back into the range in such a way as to preserve its relative position to the other elements in the collection (the *graft*). In other words, the modular arithmetic must possess a *Euclidean-like* property.

The following examples illustrate the process.

Example 1: Addition modulo 10. From Figure 2.3 it can be verified that

1 + 1 = 63 , which is greater than 10.

In order to determine the position of 63 in the range 0 to 6, the process commences at 0 and counts through the first seven hexagons and returns to 0 any time the pattern departs from these seven hexagons. The sequence of counts would be:

0, 1, 2, 3, 4, 5, 6, (for each of 6 repetitions)

0, 1, 2, 3 (for counts 60, 61, 62, 63 respectively)

Thus $(1+1)\,mod\,10 = 3$.

This tedious process is *short circuited* by performing remainder division in base 7. In general, modular addition of two addresses $a\,and\,b \in SHM$ is defined as follows:

Step 1: Perform Spiral Addition of a and b as defined in Equation 2.1.

Step 2: Perform remainder division of the result from Step 1 using the modulus.

Symbolically, the algorithm is expressed as:

$$(a+b)\,mod\,(modulus) = result \tag{2.3}$$

The remainder on division of 63 by 10 is 3. These two methods are equivalent. The first method embodies the geometrical significance of Spiral Counting but the second method is the one most easily implemented for computational purposes.

Example 2: Addition modulo 100. $1 + 62 = 645$

Addition modulo 100 amounts to dropping all but the two least significant digits.

Thus $(1+62)\,mod\,100 = 45$.

The general case of modular multiplication is motivated by the following observation: All of the addresses in SHM which are multiples of the modulus are, in effect, zero elements. That is, each acts as an additive unit element. In order to create multiplicative inverses for all of the non-zero addresses, these addresses

must not be mapped to zero under modular multiplication. Otherwise SHM would contain zero-divisors and the possibility for achieving a multiplicative inverses for all non-zero addresses would be lost. The following two step algorithm that will define modular multiplication on SHM insures that the operation of multiplication is compatible with the criteria of a multiplicative inverse.

Modular Multiplication

Step 1 Let p be the product of two elements a and b $\in$ SHM. $p = a \times b$ as defined in Equation 2.2.

Step 2 If $p \geq (modulus)$, then if the element a is a multiple of 10 map p to

$$(p + (p \div (modulus)))\, mod\, (modulus); \tag{2.4}$$

Otherwise, map p to

$$p\, mod\, (modulus). \tag{2.5}$$

Example 3: Multiplication modulo 100. Observe that $10 \times 15 = 150$

As the value of a, 10, is a multiple of 10, the hexagon whose address is in the range 0 to 66 and corresponds to 150 is determined as follows:

$(150 + (150 \div 100))\, mod\, 100 = 51.$

Example 4: Multiplication modulo 100. $15 \times 10 = 150$

As 15 is not a multiple of 10, the product is mapped to

$150\, mod\, 100 = 50.$

These last two examples illustrate an important property of the multiplication operation. The non-modular form is commutative while the modular form is not commutative. The other algebraic properties of the operations are presented in the next section.

2.3.5 Algebraic Properties of SHM

The SHM together with the operations of Spiral Addition and Spiral Multiplication possesses some important algebraic properties. These are:

The operations are associative :

$a + (b + c) = (a + b) + c$

$a \times (b \times c) = (a \times b) \times c$

The operations are distributive :

$a \times (b + c) = a \times b + a \times c$ and

$(b + c) \times a = b \times a + c \times a$

0 is the additive identity :

$0 + a = a + 0 = a$

1 is the multiplicative identity :

$1 \times a = a \times 1 = a$

Each address has a unique additive inverse :

$a + (-a) = (-a) + a = 0 \forall a \in SHM.$

$-a$ denotes the additive inverse of a.

Each non-zero address has a unique multiplicative inverse :

$a \times a^{-1} = a^{-1} \times a = 1 \forall a \neq 0.$

a^{-1} denotes the multiplicative inverse of a.

Commutative property :

$a + b = b + a$

$(a + b)\, mod\, (modulus) = (b + a)\, mod\, (modulus)$

$a \times b = b \times a$

$(a \times b)\, mod\, (modulus) \neq (b \times a)\, mod\, (modulus)$

Non-existence of zero-divisors There are no elements $a \neq 0$ and $b \neq 0 \in SHM$ such that $a \times b = 0$.

Euclidean property For every a $\in SHM$ there is defined a nonnegative integer d(a) such that:

(1) For all $b \in SHM, d(a) \leq d(a \times b)$

(2) For any a, b $\in SHM$ there exist t, r $\in SHM$ such that $a = t \times b + r$ where either r = 0 or d(r) < d(b).

Every address of SHM can be obtained by repeatedly adding the address 1.

2.3.6 Algebraic Significance of SHM

The algebraic properties of Spiral Counting developed in the preceding subsections are summarised in the following theorem. The mathematical proof of the theorem is found in Appendix A.

Theorem 1 *Suppose that the SHM is defined with the operations of Spiral Addition and Spiral Multiplication, as defined in Equations 2.1 and 2.2. Then,*

i) SHM is an abelian cyclic 7-group under the operation of Spiral Addition.

ii) SHM is a Euclidean ring.

Let the number of hexagons in SHM be

$K = 7^n$ *for* $n \geq 0$ *and let the arithmetical operations be modulo* K *as defined in Equations 2.3, 2.4 and 2.5.*

Then,

iii) The subset of SHM given by {0, 1, 2, 3, 4, 5, 6} is a finite field.

iv) The collection of addresses of SHM which are not a multiple of 10 form an abelian group under the operation of Spiral Multiplication and has order $6 \times 7^{n-1}$.

The principal algebraic significance of this theorem emerges from the fact that SHM is a Euclidean ring. The algebraically rich structure of a ring is well known [42]. The immediate significance of the Euclidean ring in this context is that geometrical properties associated with images represented on the SHM can be dealt with in purely algebraic terms. More specifically, it will facilitate the creation of a collection of image processing transformations that include translation, rotation, dilation and contraction. These transformations will be developed in the next section.

2.4 Spiral Honeycomb Lie Algebra (SHLA)

In this section, the algebraic properties of SHM (a Euclidean ring), will be employed to describe the data structure referred to at the beginning of the chapter. This data structure, Spiral Honeycomb Lie Algebra (SHLA) has, as its name implies, the properties of a Lie Algebra. This mathematical structure will be described in detail. This eloquent algebraic structure is not a theoretical end for its own sake. Rather, this theory has two important features. Firstly, it provides the unification for the development of a collection of image processing transformations. Secondly, it provides a technique for producing other computational image processing tools. Its importance to this thesis emerges from the use to which the tools are put; namely, to the creation of invariant transformations required for an active machine vision system. The transformations observable in Figures 2.12 to 2.22 are examples of the transformations referred to and that which will now be described in formal mathematical terms. In particular, it will be demonstrated how to achieve the primitive operations of translation, rotation and scaling. The theory will also be used to produce non-primitive

operations from the primitives.

2.4.1 Transformations on SHM

Consider the following two sets of transformation on SHM.

Define for each $x \in SHM$ the mappings:

$$A_x\colon SHM \to SHM \text{ by } aA_x = (a+x)\, mod\, K\, \forall\, a \in SHM. \tag{2.6}$$

$$M_x\colon SHM \to SHM;\ \text{by } aM_x = (a \times x)\, mod\, K\, \forall a \in SHM. \tag{2.7}$$

Let $\mathcal{A} = \{\mathrm{A}_x \,|\, \forall x \in SHM\}$ and $\mathcal{M} = \{\mathrm{M}_x \,|\, \forall x \in SHM\}$.

The sets $\mathcal{A}$ and $\mathcal{M}$, at this point are without algebraic structure. The definition of operations on these sets will provide them with the desired structure.

Let T_x be a transformation from either of these sets: Then, define

$$\textbf{addition} \quad T_x + T_y = T_{x+y} \tag{2.8}$$

$$\textbf{multiplication} \quad T_x \times T_y = T_{x \times y} \tag{2.9}$$

$$\textbf{scalar multiplication} \quad bT_x = T_{bx} \tag{2.10}$$

Note that the operations of addition and multiplication on the left side of the equal sign refers to the operations on the transformations (the ones being defined) and those on the right refers to operations on the elements of SHM, as defined in the previous section. It also follows directly from the definition that T_x is a bijective transformation.

Theorem 2 With regards the transformations on SHM:

i) Each of the sets $\mathcal{A}$ and $\mathcal{M}$ forms an algebra under the operations of addition, multiplication and scalar multiplication (equations 2.8 through 2.10).

ii) The external direct sum of $\mathcal{A}\oplus\mathcal{M}$ forms an algebra under the pairwise operations of addition, multiplication and scalar multiplication. Let this algebra be denoted by T.

2.4.2 Lie Multiplication

A fourth operation, called bracket multiplication or Lie multiplication, which has the effect of intertwining the other three operations is defined as: For $\mathcal{X}$, $\mathcal{Y} \in$ T and $\alpha \in$ SHM let

$$\alpha[\mathcal{X},\mathcal{Y}] = \alpha\mathcal{X}\mathcal{Y}\mathcal{X}^{-1}\mathcal{Y}^{-1}. \tag{2.11}$$

In words, the bracket multiplication is that transformation which is equivalent to the successive applications of $\mathcal{X}$ followed by $\mathcal{Y}$ followed by the inverse of $\mathcal{X}$ followed by the inverse of $\mathcal{Y}$.

Bracket multiplication is a rather special operation which possesses the following properties:

Multiplication is left and right associative with respect to addition :

$$[\mathcal{X}+\mathcal{Y},\mathcal{Z}] = [\mathcal{X},\mathcal{Z}] + [\mathcal{Y},\mathcal{Z}] \text{ and } [\mathcal{X},\mathcal{Y}+\mathcal{Z}] = [\mathcal{X},\mathcal{Y}] + [\mathcal{X},\mathcal{Z}] \tag{2.12}$$

The identity transformation is :

$$[\mathcal{X},\mathcal{X}] = I \tag{2.13}$$

The third property is known as the Jacobi identity:

$$[[\mathcal{X},\mathcal{Y}],\mathcal{Z}] + [[\mathcal{Y},\mathcal{Z}],\mathcal{X}] + [[\mathcal{Z},\mathcal{X}],\mathcal{Y}] = I \ \forall \mathcal{X},\mathcal{Y},\mathcal{Z} \in \mathcal{A},\mathcal{M}. \tag{2.14}$$

Let SHLA denote the algebra described in Theorem 2 above plus bracket multiplication. Then, the following theorem summarises the main algebraic result of this chapter.

Theorem 3 *SHLA is a Lie Algebra and is a subspace of the dual of SHM.*

The elements of SHLA are ordered pairs of elements from $\mathcal{A}$ and $\mathcal{M}$, respectively. The identity element is (A_0, M_1) and the inverse of element (A_x, M_y), denoted by $(A_x, M_y)^{-1}$ is $(A_{x^{-1}}, M_{y^{-1}})$.

2.5 Interpretation of SHLA

The elements of SHLA derive their relevance to machine vision for three reasons:

- The transformations are visually meaningful
- SHLA is computational
- The transformations possess invariant properties

2.5.1 Meaningful Transformations

The term *meaningful transformations* is used in the sense of Gibson's meaning in [35]. When an object moves in the visual world relative to an observer, the visual field of the observer undergoes a transformation. The fact that the objects in the transformed image preserve critical properties of the objects is an essential component of the recognition process. The relationship of the elements of SHLA to the collection of *meaningful* transformations can be readily appreciated by considering a few examples of the effects of these transformations on an image.

The image of the can of Pal dog food in Figure 2.11 is represented with six octaves, 7^6 pixels.

Figures 2.12 and 2.13 shows the effect of applying A_{1000} and A_{100000} respectively

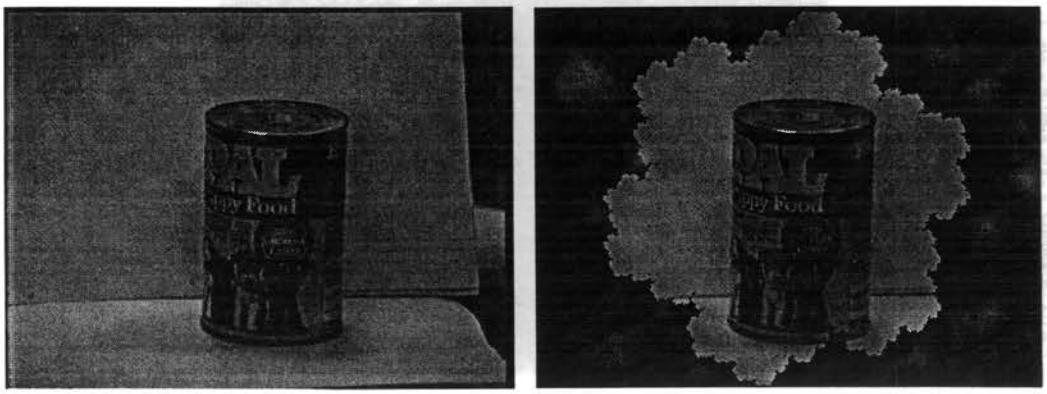


Figure 2.11: (left) Pal can of dog food represented by 384x288 pixels, (right) shows the relative position of the 7^6 hexagon SHM to the rectangular figure.

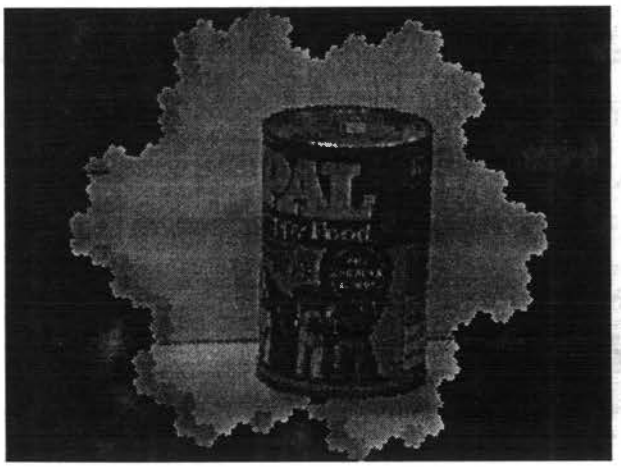


Figure 2.12: Transformation by A_{1000}. The effect is one of translating the image a small distance along the line segment joining addresses 0 and 1000.

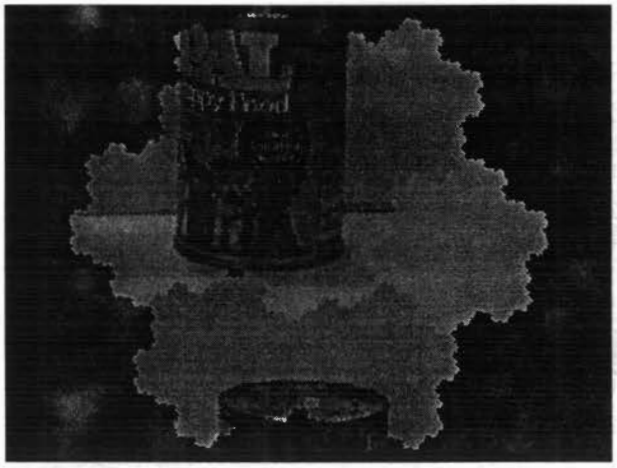


Figure 2.13: Transformation by A_{10^5}. The effect is one of translating the image along a line segment joining the addresses 0 and 100000. The amount of translation makes visible a *wrap around* effect.

Observe that the first of these two transformations is a translation of the input image by a small amount. However, the second is a translation of a larger amount in the direction of *5 o'clock* from the centre. In this translation, part of the can has been shifted out of view. However, it is not lost as it has been *wrapped* around and is visible in the *11 o'clock* direction from the centre. The original image in 2.11 is recoverable by applying the inverse transformations, A_{4000} and A_{400000} to 2.12 and 2.13 respectively.

Figures 2.14, 2.15, 2.16, 2.17 and 2.18 display the effect of applying M_2, M_3, M_4, M_5, and M_6 respectively. These images represent rotations of 60, 120, 180, 240 and 300 degrees about the address zero.

The results displayed in Figure 2.19 are the effects of applying M_{10}, M_{100}, M_{1000}, M_{10000}, M_{100000}, and $M_{1000000}$ and are somewhat more complex and interesting illustrations of the transformations of SHLA.

[5]The transformations A_{1000} and A_{100000} result from replacing x in Equation 2.6 with 1000 and 100000, respectively

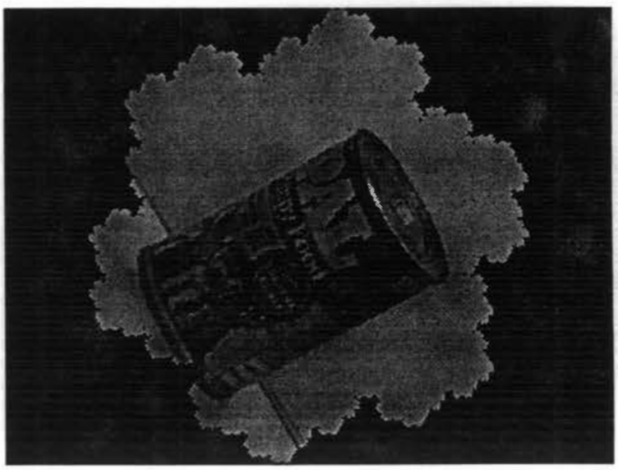


Figure 2.14: Transformation by M_2. The effect is one of rotation by 60 degrees.

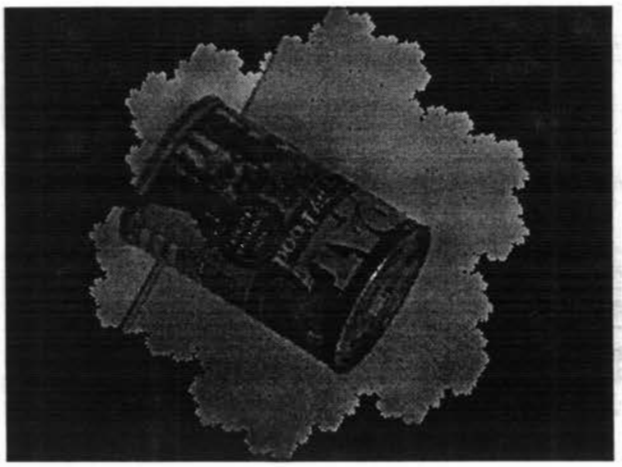


Figure 2.15: Transformation by M_3. The effect is one of rotation by 120 degrees.

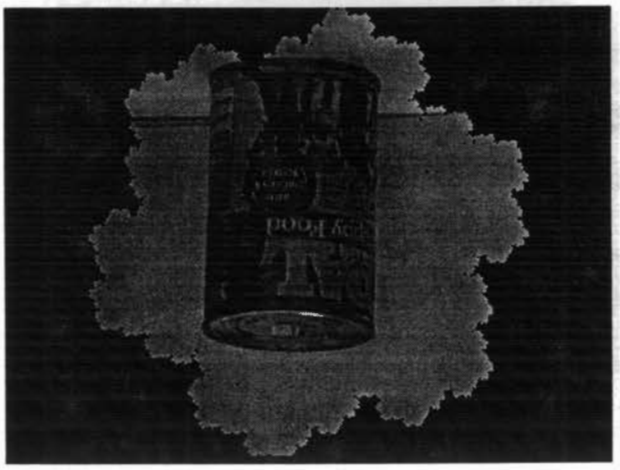


Figure 2.16: Transformation by M_4, The effect is one of rotation by 180 degrees.

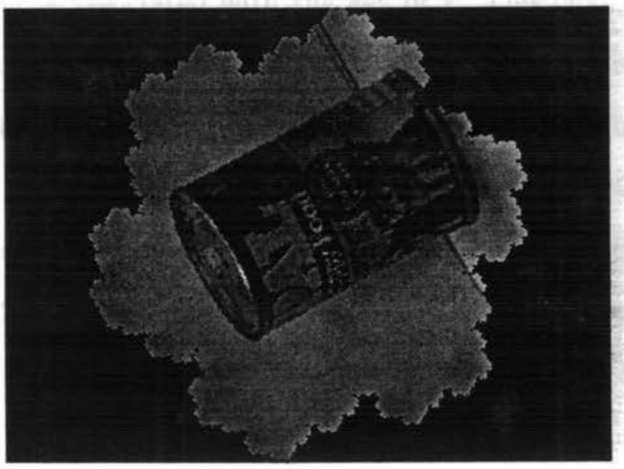


Figure 2.17: Transformation by M_5. The effect is one of rotation by 240 degrees.

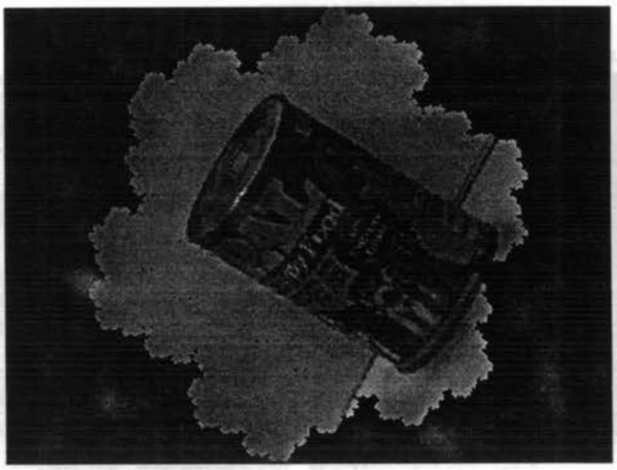


Figure 2.18: Transformation by M_6. The effect is one of rotation by 300 degrees.

In these figures, four effects on the input image are observable: **1. contraction**, **2. dilation**, **3. rotation**, and **4. apparent production** of multiple copies. Recall that the key of an octave has associated with it a metre and an orientation. The application of M_x to the input image has the effect of changing the metre and orientation of the primary octave to those associated with the key of x. This change in metre induces a simultaneous contraction and dilation. Observe that each of the seven can's image in 2.19 (upper right) are approximately one third the size of the can in the input image (upper left). Likewise, each of the images represented in 2.19-b to 2.19-f is a contraction of approximately one third of the preceding image. The exact contraction factor is the ratio of the distances between the addresses 0 to 1 and 0 to 10.

The dilation effect is observable in only the lower right image of 2.19. Although the figure represents the 5th contraction $(M_{10})^{-5}$, it also contains a magnification of the figure in (upper left) by the same factor. Observe that the dog on the can in the upper left is magnified and has a *ghost-like* appearance in lower right. The magnification effect is also present in each of the other five figures. However, due to

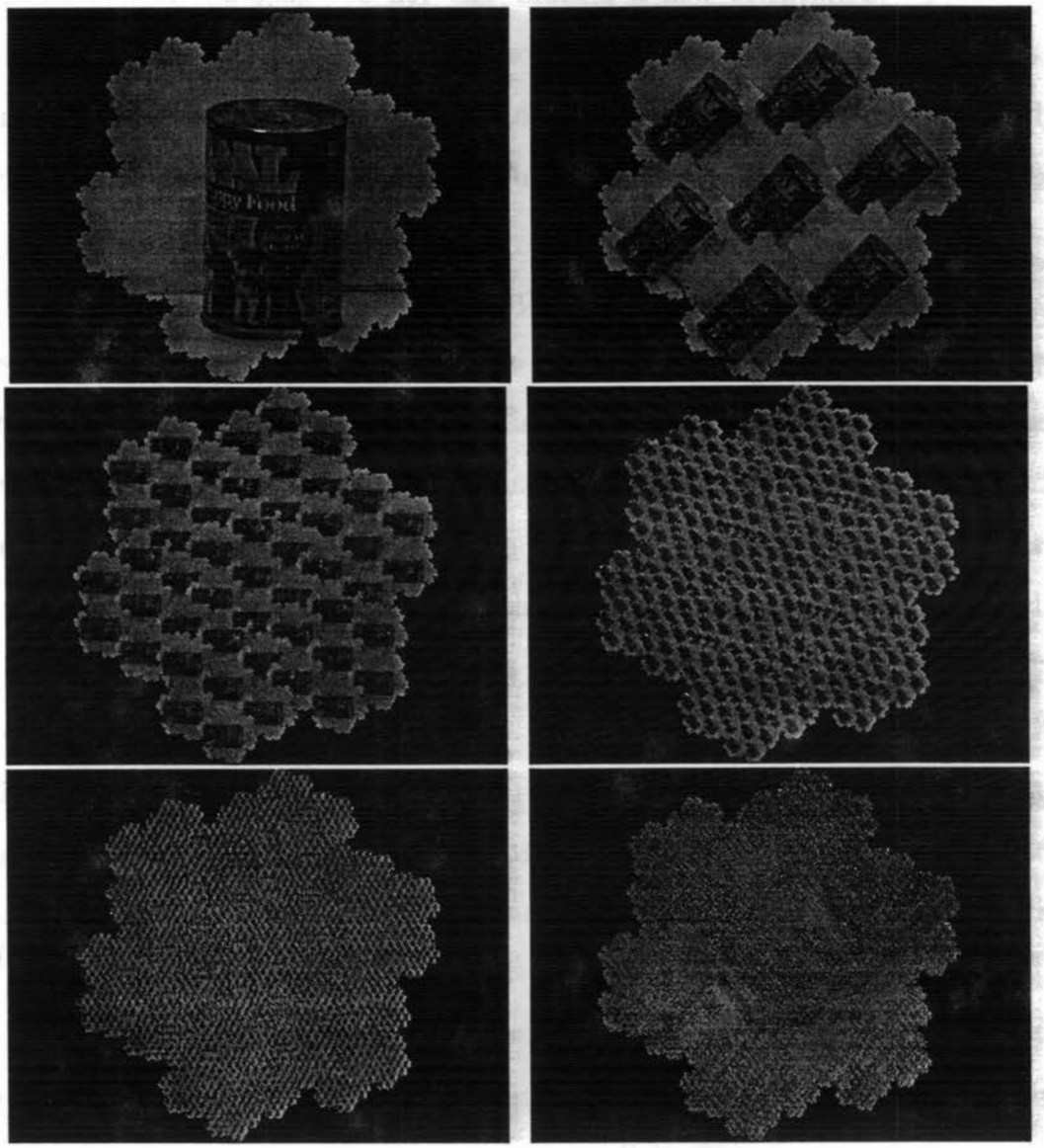


Figure 2.19: (upper left) original image, (upper right) application of M_{100000} to the original image, (middle left) application of M_{10000} to the original image, (middle right) application of M_{1000} to the original image, (lower left) application of M_{100} to the original image, (lower right) application of M_{10} to the original image. The observable effects in these transformations are 1. rotation, 2. scaling, 3. production of multiple near copies of the original image.

the magnitude of the magnification, it is not apparent.

The effect of changing the key also induces a new orientation. Observe that the rotation in Figure 2.19 corresponds to the angle subtended by the straight line segments joining centres of the cells at address 1, 0 and 10. This is approximately 25 degrees. Again, each of the remaining figures in Figure 2.19 represents a rotation of the previous image by this same amount.

The fourth effect of producing multiple copies is somewhat more curious and possibly a bit more difficult to appreciate. First, an effect that is difficult to observe is that the copies are only *near* copies. This is due to the fact that each copy results from a unique sampling of the input image. Each sample is mutually exclusive and the collection of all such samples represents a partitioning of the input image. Secondly, observe the number of *near* copies in each successive figure. This number is related to the degree of scaling of the image (the metre). In this case, the number grows by a factor of seven in each successive image. To further illustrate this effect of multiple copies, consider the images represented in Figures 2.20 and 2.21. These two images result from the application of $M_{15^{-1}}$ and $M_{12^{-1}}$ respectively[6]. Observe that the copies in the periphery of each are incomplete and that the number of copies in the latter are not a power of seven.

The immediate consequences of this fourth effect, (multiple near copies), are three fold. The first is an implication for concurrency. That is, there is the possibility of pursuing different recognition strategies in parallel. This issue is treated in some detail in Chapter 4. The second consequence is the potential for handling images that can be represented in a multidimensional space involving depth and time. This

[6]There are at least two computational methods that can be used to find the inverse of a transformation in SHLA. One method is provided by raising the transformation to a power one less than is required to produce the identity transformation. Another method would employ the algorithm derived in the proof given in Appendix A that established the existence of the multiplicative inverse of all non-zero elements in SHM. The inverse of M_{15} is M_{456123}.

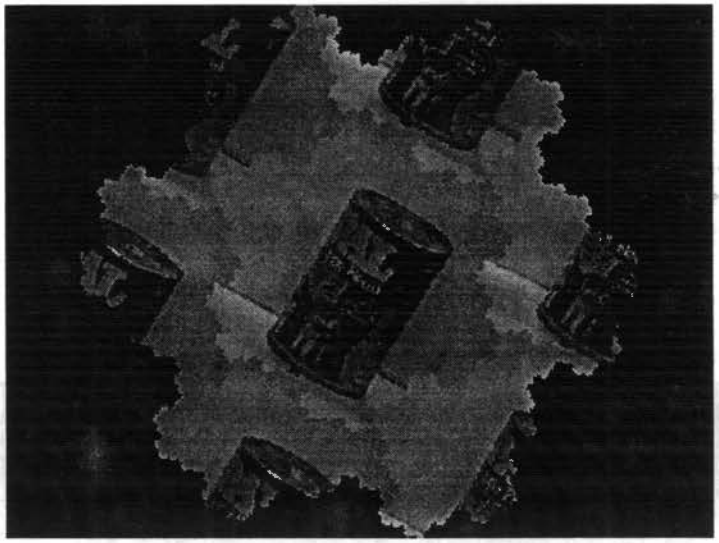


Figure 2.20: The inverse transformation of M_{15}. The observable effects are 1. rotation, 2. seven contractions of the original image of which six are partial representations of the original image.

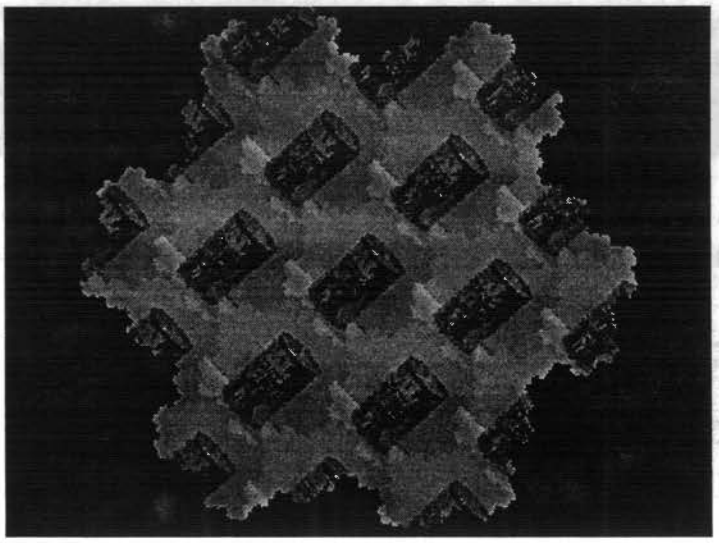


Figure 2.21: The inverse transformation of M_{12}. The observable effects are similar to those of M_{15}^{-1} differing in the amount of rotation, degree of contraction and the number of multiple copies.

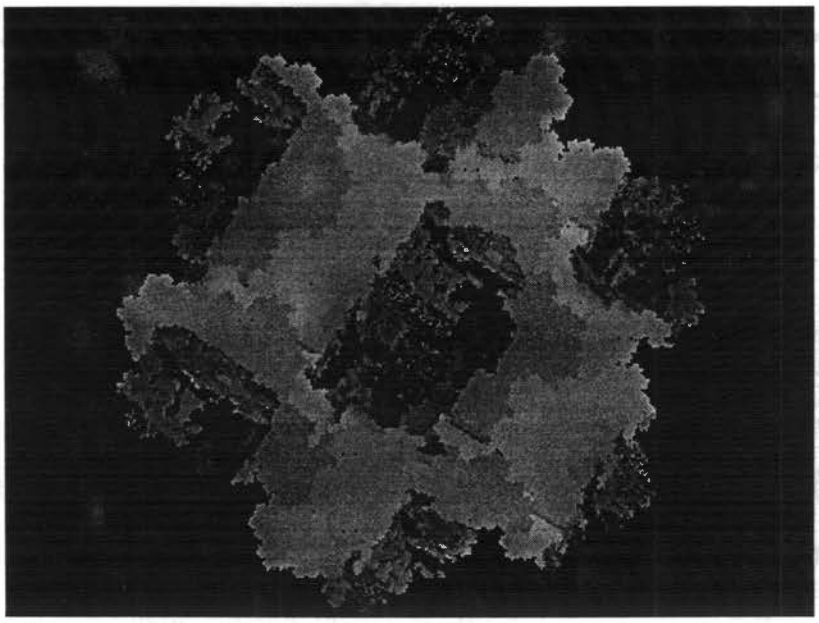


Figure 2.22: A compound transformation, inverse transformation of M_{15} followed by A_{100000}. The partial copies observable in Figure 2.20 have been wrapped around by the translation.

will be explored in Chapter 5. The final consequence to be considered is possibly the most important. As the scaling in effect represents the viewing of the image at a lower resolution, each copy has *less* information. However, as none of the individual light intensities have been altered in any way, the scaled image still holds *all* of the information contained in the original. This means that the computational complexity has been both reduced and nicely partitioned without giving away any information. The reduction of computational complexity results from the fact that the object is represented with fewer pixels. The production of the multiple copies results in the effect of partitioning the computations in a manner compatible with certain parallel computer architectures. These issues will be taken up again in Chapters 3 and 6.

2.5.2 SHLA is Computational

The claim that SHLA is computational partly results from the fact that each element of $\mathcal{M}$ has associated with it a particular scaling and rotation. An arbitrary rotation on the SHM is achieved by a trade off in scale, not by incurring rounding error. Likewise, an arbitrary scaling is achieved by a trade off in orientation. The computational consequence of not incurring any rounding error when performing a transformation is that the image will not suffer any degradation no matter how many transformations are performed on it. This has important consequences for the creation of the non-primitive transformations which can be arbitrary long sequences of primitive transformations. The deeper reason for the occurrence of this desirable property is due to the fact that SHM has the structure of a Euclidean ring and that all non-zero addresses possess unique inverses. When the SHM is restricted to a finite set of hexagons and the arithmetic operators are defined modularly a *Euclidean-like* property is induced. Technically the modular multiplication does not have the Euclidean property globally. Rather, the modular effect of the operation imposes an

inverse Euclidean property. That is, the points are transformed closer together. The net effect of multiplication on the image represented in the SHM is that a simultaneous dual Euclidean effect is induced. The result of this dual Euclidean effect is that the image is both dilated and contracted. An example of this effect can be observed in Figure 2.19 where the image of the dog on the original figure is dilated and overlaid on the contracted image.

It is also worth noting that the Euclidean property of SHM has a computational implication. Although the concept of Spiral Counting can be applied to a rectangular grid (where the cluster is composed of a centre square with 8 neighbours and whose integer base is 9), the structure cannot be made to have the Euclidean property. For example, a rotation of 45 degrees would distort the original image. This loss of the Euclidean property results from the fact that each of the neighbours of the centre square are of different distances from the centre. This observation also represents the first significant difference between the hexagonal and rectangular grids.

The fact that the transformations are bijective also adds to the computational claim. Intuitively, one would expect a bijective transformation to be easily paralleliseable. It turns out that this is so and will be demonstrated in Chapter 3.

2.5.3 *Invariant Properties*

In its own right, the Lie (bracket) multiplication has relevance to the SHM for two reasons. Firstly, as a mathematical tool it can be employed to study the structure of SHM. There are techniques which use Lie multiplication to obtain the automorphisms of a p-group [38] p209 and [12]. Secondly, according to the equations of Forbinious [46], all elements of the Algebra are representable as linear combinations of the basis (primitive) transformations. However, the Lie multiplication coupled with the Euclidian property of Spiral Counting provides an additional powerful tool for the

analysis of images. One of the *ecologically meaningful* [7] transformations on an image relates to the image appearing the same before and after the transformation. A simple example of this situation is the rotation of a circle about its centre. Such a rotation, by any amount, would be unnoticeable to an observer. This transformation, which holds the circle invariant, can be used to characterise the circle. At present there are no techniques for finding an invariant transformation for an arbitrary image. Hoffman's Linear Transformation Group approach has made a considerable thrust in this direction. This issue will be taken up in greater detail in Chapter 6.

2.6 Summary

The unique geometrical properties of the *fundamental unit of vision*, as displayed in Figure 2.1, has the algebraic properties of a *finite field.* It was shown how the *fundamental unit of vision* could be employed as the atomic units of a larger structure, SHM, while retaining the important algebraic properties of the Euclidean ring. The SHM was employed to create SHLA which is a rich collection of image transformations and has the algebraic structure of a Lie Algebra. These transformations possess three important properties relevant to machine vision:

- The transformations are visually meaningful
- SHLA is computational
- The transformations posses invariant properties.

The above properties of SHLA which are possessed by no other data structure result directly from the structure of the *fundamental unit of vision.*

[7] This term was coined by J. J. Gibson in [34] to distinguish the types of transformations performed by biological vision systems from image transformations in general.

Chapter 3

Implementation of the SHLA

On the third day, after hearing Pekanini's tale of counting, Kimo said, "Counting accurately is one thing. But, one must also count swiftly."

3.1 Introduction

This chapter will demonstrate how the Spiral Honeycomb Lie Algebra (SHLA) can be implemented on a parallel processing architecture. The significance of the results presented in this chapter is that it provides further evidence to support the claims made in Chapter 2 about the relevance of SHLA to machine vision. In particular, it will be demonstrated that SHLA is computational.

The mathematical structure of SHLA and its implementation evolved together. A synergistic relationship arose between the theory and the practice where at times the theory drove the implementation and at other times the reverse was the case. The basis of this relationship nurtured the belief that a real-time machine vision system must possess the property of concurrency. Consequently, the development of the theory was pursued only in the event that it supported this paradigm.

Recall that SHLA is a collection of transformations of a space on which an image is represented. The nature of these transformations possesses both global and local expressions. Their global nature has the effect of re-arranging the picture elements of the image while preserving some critical properties of the image, such as shape, orientation and size. However, these global effects are achieved by purely local actions. The local action is an operation which is performed on an address which determines a unique address to which information is sent. The derivation of the new addresses is dependent on the process of Spiral Counting. Consequently, the implementation will amount to the coding of this process. The resulting transformations may then be considered as a collection of independent operations that may proceed concurrently.

The intensive computational demands of machine vision appears to be well suited to parallel architectures in general [66]. The transputer was selected as an appropriate parallel architecture on which to implement the SHLA for three primary reasons. These are:

i) As the transformations of the SHLA will be performed at many levels of the vision process, the Multiple Instruction Multiple Data (MIMD) capability of the transputer was considered to be desirable.

ii) The transputer has the potential for the parallel input of an image.

iii) The *lego* like flexibility enables the addition/substraction of transputers to meet specific timing requirements according to the particular application.

This chapter is organized as follows: Section 2 provides a brief description of the general architecture of the transputer and the particular network of transputers employed for the simulation tests. Section 3 discusses related work in this area. Section 4, 5 and 6 address three key issues associated with the development of the software implementation of SHLA: Theses are: data structure, algorithms and configuration. Section 7 provides a report on the simulation trials of the implementation. The chapter concludes with a discussion of some of the problems encountered with the implementation.

3.2 Transputers

A brief description of the salient features of the software and hardware follows. The reader is referred to [44] for more details of these matters.

OCCAM[1] is the language of the transputer. It is a high level language designed so that parallel algorithms can be implemented on a network of processing components. OCCAM is highly structured and is similar to Pascal in many ways. The two distinguishing features from Pascal-like languages are *concurrency* and *communication.*

[1]OCCAM was named after the 14th century philosopher William of Occam and captures the spirit of the now popular term, "Occam's razor".

Concurrency is expressed in a number of ways. The most important of these are the PAR and ALT construct. PAR facilitates the replication of processes that may progress concurrently. ALT provides the mechanism for outputs from a variety of parallel processes to be input to a single process.

Communication takes place through channels which are software constructs that may be mapped to hardware links. The channels connect processes at both logical and physical levels.

David May, one of the developers of the transputer, gives a long term vision for the prospects of this architecture in [68]. The language is tightly coupled to its intended hardware, the transputer. This device is a microcomputer that has been placed on a single chip. The individual transputers may be connected in the form of a network by physical communication links. The networks may be configured in a variety of ways; the most popular of which are the pipeline, ring and grid.

3.3 Related Other Works

There have been many worthy applications of transputers to the field of machine vision. See [95, 103] for details of a few of these. The Eastman Kodak Co. [20] has developed a parallel image processor capable of performing a variety of imaging tasks, including an imaging editing workstation, an address block location finder for mail parcels, and a special effects film editing workstation. A team at The National University of Singapore has employed the transputer to develop a multi-resolution scene analysis system [95]. This work configured the network as a pyramid structure to handle a resolution of an image at each layer. Sandon [87] has developed a method of dynamically reconfiguring an array of processors in a pyramid arrangement. Groups based at Monash University are applying the transputer to robotic vision and are

developing parallel image processing algorithms and implementing them on transputers. BHP has a research group working on the implementation of dynamically reconfiguring the processors [28].

SIMD and other parallel architectures are also being employed extensively in computer vision [7, 43, 58, 84]. For a review of these developments see [24, 66].

Yasrebi [108] gives a review of the different philosophical approaches to the design and implementation of parallel based vision systems.

3.4 Data Structure

This section commences the analysis of the three components of the software implementation of SHLA: data structure, algorithms and configuration. In the introduction to this chapter, it was suggested that a belief in the MIMD paradigm was a driving force in the development of SHLA. At the outset, the target of the implementation was to maximize its parallelism. A natural strategy for achieving this target manifested when the *one-to-one* correspondence between the SHM and the integers was considered. The parallelism should occur at the most local level, the hexagon. The correspondence is illustrated by *unwinding* the SHM into a one dimensional representation. Each hexagon has associated with it an integer address. See Figure 3.1.

3.4.1 Elements of the Data Structure

The elements of the data structure are the hexagons each of which is identified by its address in the SHM. Each element will be associated with a concurrent process. The primary benefit to be gained by assigning each hexagon to a unique concurrent process is that the transputer can handle either the logical or the physical parallelism

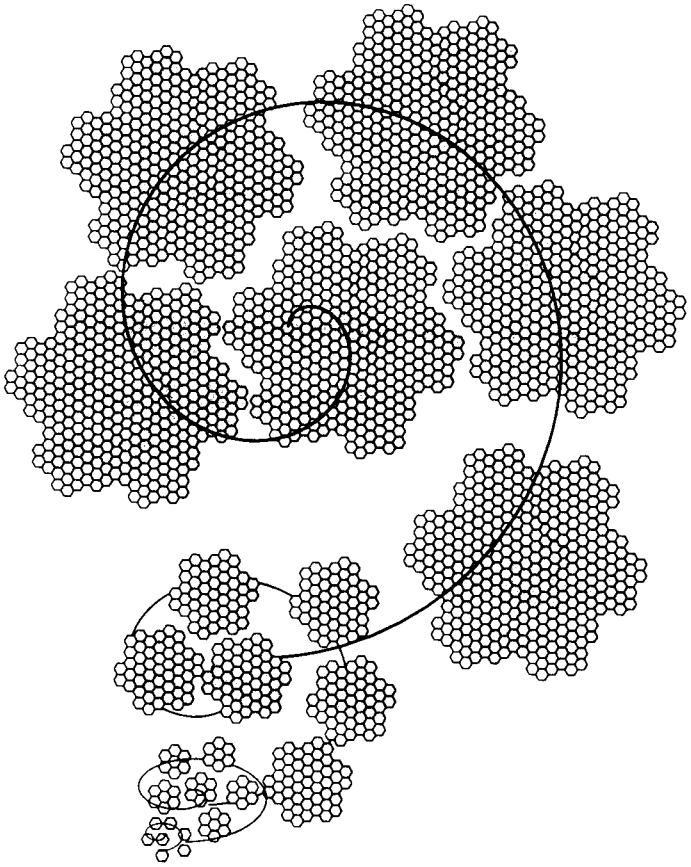


Figure 3.1: The unwinding of the Spiral Honeycomb Mosaic indicates; i) the one-to-one correspondence between SHM and the integers; ii) the spiral pattern contained within it.

without the apparent need to consider efficiency issues[2]. At a logical level each process (hexagon) can operate in parallel with all other processes, while at the physical level the processes may be grouped into as many physical processors as is desired. The number of physical processors will be constrained by such factors as the availability of transputers or the speed requirements of the particular application. Consequently, each of the elements of the data structure is associated with a logical process.

The arrangement of the elements is a function of the adjacency properties of the hexagons of SHM. The method of addressing the hexagons of the SHM amounted to tracing a spiral pattern from the centre of the SHM and designating the addresses sequentially. In general, hexagons with similar addresses tend to be in close proximity even when the spiral is *unwound* into a one-dimensional representation. It is worth making two points with regard to the parallelism taking place at the hexagon level. Although it is not imagined that there will be a physical processor for every logical process at present[3], the convenience gained by considering the logical and physical levels as equivalent did simplify the development process. The second point stems from a remark made by David May at the 1991 WATUG Conference. He suggested that by the end of this decade, it would be possible to have 10^9 processors in a single network. If and when this prediction eventuates, parallelism at the hexagon level is feasible.

[2]The developers of the transputer had intended that the design of an Occam program should be to achieve optimal parallelism with minimal consideration of the physical resources. Although this is largely achievable, the design approach employed for this implementation did experience some difficulty by ignoring the hardware issue at the design stage.

[3]The CM5 might have sufficient processors to accomplish this.

Table 3.1: Spiral Addition and Multiplication Tables.

Addition

	0	1	2	3	4	5	6
1	1	63	15	2	0	6	64
2	2	15	14	26	3	0	1
3	3	2	26	25	31	4	0
4	4	0	3	31	36	42	5
5	5	6	0	4	42	41	53
6	6	64	1	0	5	53	52

Multiplication

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	3	4	5	6	1
3	0	3	4	5	6	1	2
4	0	4	5	6	1	2	3
5	0	5	6	1	2	3	4
6	0	6	1	2	3	4	5

3.4.2 Operations

The Lie multiplication of the SHLA assures that an arbitrary number of the transformations of SHLA are representable as linear combinations of the primitive transformations from $\mathcal{A}$ and $\mathcal{M}$, which in turn result from the operations of Spiral Addition and Spiral Multiplication. Each of these arithmetic operations is considered in turn.

Spiral Addition

Spiral Addition was defined in terms of Spiral Counting. This process, as introduced in Chapter 2, is largely a *pattern tracing* exercise. That is, Spiral Counting commences at a particular address and then traces an *octave* pattern from that address for a given number of counts. The task at hand then is to derive an algorithm to emulate this tracing process. This is achieved in two steps. First, the addition of single digit numbers is obtained by the coding of values listed in Table 3.1. The table provides the *atomic* operations of addition. For example, from the table 1 + 1 = 63 and 3 + 4 = 31. Secondly, the simple *carry rule* is implemented to handle the addition of numbers composed of more than one digit. To illustrate the algorithm, consider the example of adding the addresses 6 and 21. The *tracing* method from Chapter 2 indicates that their sum is 14. Now employing Table 3.1 and the carry rule, the steps are:

i) 6+1 = 64; put down the 4 carry the 6

ii) 2+6 = 1 put down the 1 carry nothing

iii) sum = 14

Spiral Multiplication

Spiral Multiplication was defined in terms of Spiral Counting which was described as a visual task of tracing a pattern. The difficulty associated with the implementation resulted from the need to express the algorithm in non-visual terms. As Spiral Counting in a given key is just a generalization of Spiral Counting in the key of one, the algorithm for M_x will employ and build on the algorithm developed for A_x. Its development will also proceed in a similar manner. The first requirement is the implementation of scalar multiplication. This is achieved by the coding of the scalar multiplication table of Table 3.1. The next step is to establish the scalar multiplication of numbers (addresses) composed of more than one digit. Recall that any address from SHM can be represented as an n-tuple of digits from the set (0,1,2,3,4,5,6), where n is the number of octaves of SHM. More formally, let an address $a \in SHM$ be represented as,

$$a = a_n, a_{n-1}, ..., a_1 \, \forall \, a_i \in (0, 1, 2, 3, 4, 5, 6) \tag{3.1}$$

Multiplication of address a by the scalar *alpha* is obtained by applying scalar multiplication to the components of a. That is,

$$\alpha(a) = (\alpha a_n, \alpha a_{n-1}, ..., \alpha a_1). \tag{3.2}$$

For example,

$$\begin{aligned} 3 \times 426 &= (3 \times 4, 3 \times 2, 3 \times 6) \\ &= (6, 4, 2) \\ &= 642 \end{aligned}$$

The last step is to achieve the multiplication of an address by a non-scalar. Given two addresses $a, b \in SHM$, let

$$\begin{aligned} b &= (b_n, b_{n-1}, \cdots, b_1), \text{ then} \\ a \times b &= a \times b_n \times 10^{n-1} \\ &\quad + a \times b_{n-1} \times 10^{n-2} \\ &\quad \vdots \\ &\quad + a \times b_1 \times 10^0 \\ &= \sum_{i=1}^{i=n} a \times b_i \times 10^{i-1} \end{aligned}$$

For example, consider the multiplication of 15 by 23. As defined in Chapter 2, this would amount to Spiral Counting from zero for 23 counts in the key of 15. The answer is 201, which can be verified from an inspection of Figure 2.5. Performing the same multiplication employing the algorithm just described yields:

$$\begin{aligned} 23 \times 15 &= 2 \times 15 \times 10^1 + 3 \times 15 \times 10^0 \\ &= 26 \times 10 + 31 \times 1 \\ &= 260 + 31 \\ &= 201 \end{aligned}$$

This example illustrates that although the operations of addition and multiplication as defined in SHM are quite different from their more commonly known counter parts, the relationship between addition and multiplication is analogous. In essence, the relationship in both systems is that multiplication is equivalent to repetitive addition.

The final task to complete the multiplication algorithm is the implementation of the modular form of the operation. The product is first checked to see if it is within the specified range (the modulus). If it is in the range, no further work needs to be done. Otherwise, the *key* is checked to see if it is a multiple of 10. If the *key* is a multiple of 10 Equation 2.4 is employed; otherwise, Equation 2.5 is used.

For example, suppose the modulus is 100. Then there are 49 addresses ranging from 0 to 66 in base seven. Consider the multiplication of 10 by 15.

$$15 \times 10 = 150.$$

As the *key*, which is 15, is not a multiple of 10, Equation 2.5 will map it to 50.

Consider multiplication of 10 by 20.

$$10 \times 20 = 200.$$

As the *key*, which is 10, is a multiple of 10, Equation 2.4 will map it to 2.

3.5 Algorithms

In this section, the primitive image processing transformations of translation, rotation and scaling are discussed. As these transformations form the building blocks for many of the other transformations of SHLA, it is imperative that they can execute in real-time[4].

[4]real-time: This term refers to the performance of a computing system. If the desired outputs of the system can be delivered at a rate commensurate with handling subsequent inputs to the system, it is said to be operating in real-time. By necessity, a biological vision system must be a real-time system.

A translation can be thought of as a shift that re-centres the image to a particular distinct point while holding the relative position of all pixels fixed. This operation, in terms of the SHM, is exactly the transformation A_x. In other words, A_x transforms an image centred at the address zero to one centred at the address x. Likewise, the transformations of $\mathcal{M}$, namely M_x as defined and discussed in the previous chapter, are determined by Spiral Multiplication.

The achievement of the global effects of the transformations of SHLA amounts to the collective local effects of the operations of Spiral Addition and Spiral Multiplication. The algorithms for the transformations can be described as follows:

i) Send a message to each process (address) indicating the operation to be performed; for example, add 123

ii) At each process, perform the arithmetical operation, thus determining the new address of the grey-value held at that address

iii) Concurrently at each address, send its grey-value to the new address and receive a grey-value from another address

The algorithm is described in Occam pseudo code. The reader is alerted to the significance of indentation in the expression of the code. All lines of code at the same indentation are in the block of code governed by the construct SEQ, PAR or ALT, at the beginning of the block.

```
PAR
    send operation to be performed at each address.
    PAR i = 0 for number of address
        PAR
            SEQ
                receive operation to be performed
                compute new address from operation
                send grey value to new address
            receive new grey-value for this address
```

3.6 Configuration

In absence of the T9000 series transputer[5] channel routing capabilities, use of the T800 always begs the question of how the arrangement of the transputers will affect the performance. Three different configurations were employed to test the performance of the various transformations, (1) pipeline, (2) ring, and (3) grid; see Figure 3.2.

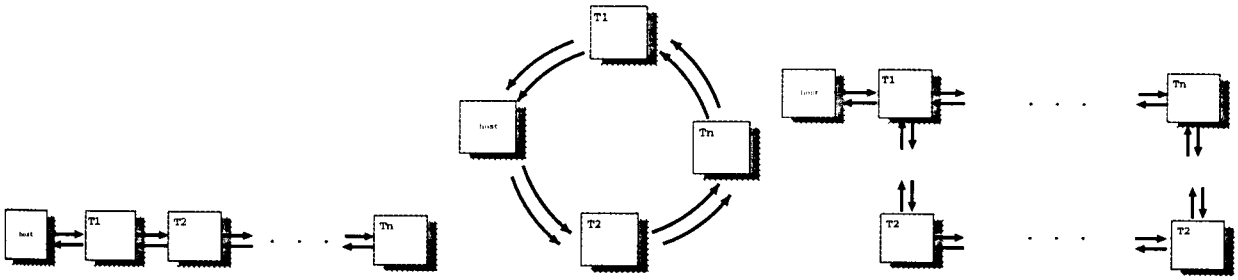


Figure 3.2: Processors in a pipe, ring and grid.

All three configurations required the use of a software multiplexing tool because there can be as many logical channels connecting processes on the same processor as is desired. However, there is only one physical link connecting two processors and there

[5]Access to the T9000 eventuated near the completion of the writing of this thesis and, to date, its possibilities have not been explored for SHLA.

can be only one pair of channels mapped to this link. As a transformation algorithm requires the passing of data from any given address to any other address, there would need to be n! (n being the number of addresses in SHM) if every process had a channel to every other process. If all the processes existed on the same processor, this would, in theory, be feasible. However, spreading the processes over more than one processor necessitates the grouping of channels which would logically cross processors into a single virtual channel. This is achieved in Occam with the use of the ALT construct which allows the inputs from parallel processes to be output to a single process.

In addition to the multiplexer, two other algorithms were required. One was to achieve two-way communication between adjacent nodes on the network of transputers. The other algorithm was required for the ring and grid configurations to find the shortest path between non-adjacent nodes. The Occam code developed to achieve the desired multiplexing, communications and shortest path finder was adapted from programs presented in [21].

3.7 Simulation

One of the practical problems to overcome in the implementation of SHLA as part of a machine vision system is the building of a device that will act as an interface between the camera and the network of transputers. The building of this device is in progress and is described in Chapter 6 Section 6.1.2 under the name of Pixel Processor. It is intended that this pixel processor will receive input from a camera and then, after dividing it into smaller packets of data, deliver it in parallel to a network of transputers at a speed commensurate with image capture. The packets of data will be in the form of ordered sequences of addresses. Figure 3.1 shows how this grouping of ordered sequences of addresses is achieved. As this device as well as some of the other components of the proposed vision system have not yet been built,

their effects must be simulated. A description of the simulation and the results of performing the simulation over a network of transputers follow.

3.7.1 Description of the Simulator

The simulator is a software package implemented on a network of transputers. The root transputer was assigned the task of handling all of the input/output between the host and the rest of the network. A number of images were captured with a video camera and stored on the host and made accessible to the root transputer. The root then accessed a given image, partitioned it and passed the pieces to designated processors. At that point, the time was noted and a command indicating the transformation was passed from the root to each of the processors in the network. The individual processes, upon receiving the command from the root, would perform its designated operations. When it received its new grey-value, it then passed this new grey-value along with its address back to the root. When the root received responses from all of the processes, it noted the time. The difference in the two times was then taken to be a measure of the time required to complete a given transformation on the image.

The time required to complete the transformations should be a function of two factors. One factor is the amount of computations to be performed on any one processor. The other factor is the amount of communication between processors. There would be an expected trade off between these two factors. On the one hand, if all the processes resided on the same processor, all the computations would in effect be performed sequentially but there would be no communication time penalty. On the other hand, if all the processes could be uniformly spread over a large number of processors, the amount of computations performed sequentially would be minimal. This would be at the expense of having to pass the grey-values over physical links to

other processors. It is of interest to determine how the addition of processors into the network affects the overall time required to complete a given transformation.

The size of the images stored on disk was 384 × 288 square pixels. As there is no possibility of gaining a one to one correspondence between the square pixels from the rectangular image and the hexagons from the SHM, an appropriate algorithm had to be determined to optimize the correspondence between the addresses of the two coordinate systems. The number of octaves chosen to represent the images was taken to be six. Thus the number of hexagons is:

$$7^6 = 117,649 \simeq 100592 = 384 \times 288$$

The metre of the first octave is approximately equal to the distance between two adjacent pixels in the rectangular image. The relative regions represented by the 7^6 hexagons and the 384 × 288 square pixels of the rectangular image is displayed in Figure 2.11.

3.7.2 Simulation Results

In this section evidence is provided that:

- The transformations of SHLA produced the expected affects when applied to an arbitrary image.
- It is possible to exploit the inherent parallelism of the transformations by implementing them on a network of transputers.

The evidence that the transformations were performing as expected on the network of transputers was provided by conducting the following experiment. An input image was sampled using the SHM. Various transformations were performed. The resulting transformed images were printed and visually compared with the theoretical

Table 3.2: Time (measured in ticks) required to complete transformations of translation, rotation and scaling on transputer networks configured in a pipeline of various lengths.

no. of nodes	A_0	A_{1000}	M_{10}	M_{15}	M_{115}	$A_{100}M_{15}A_{100^{-1}}$
1	215327	220113	217125	224419	230791	23121
2	55674	57623	56200	60211	63329	64925
4	15890	17001	16432	18117	19208	20380
8	7945	9498	9450	9834	10500	10827
16	7727	9056	7961	9104	9419	9829
32	5227	7789	7135	9073	10145	9937
41	6160	9036	9246	11023	12670	11709

predictions. Selected examples of the empirical results obtained from this experiment have been presented in the Figures 2.12 to Figure 2.22 in Chapter 2. The conclusion is that the transformations, as desired, are taking place without any degradation as evidenced by the visual presentation.

With regards to the time required to perform the transformations, three questions were posed:

i) How do the execution times of the various types of transformations compare?

ii) Is there an optimal number of processors that can be used to perform the transformations?

iii) How do the execution times vary on different configurations?

The data presented in Table 3.2 represents the timing results of performing a sample of transformations on various networks of transputers. The data suggests the following answers to the three questions that were posed. The transformations, as indicated across the columns, are arranged in order of computational costliness and the degree to which data moved between transputers. That is, A_0 is an example of a transformation which requires one arithmetical operation and all of the data moves within the transputer it resides on. In contrast, M_{115} requires three arithmetical

operations and the data will, in general, move to a different transputer. $A_{100}M_{15}A_{100^{-1}}$ is a non-primative transformations which has the effect of rotating the image by 25 degrees about the address 100. The conclusion is that the various transformations require a similar amount of time for a given number of transputers. However, it is observable that the amount of variation in times increases with the number of transputers employed.

It would appear that for this implementation of SHLA expressed at the fourth octave, 32 is about the optimal number of transputers for a pipeline configuration. The rate of decrease in time is greatest over the first eight transputers and then gradually falls away as the pipeline is increased to 32 nodes. For a pipeline of 41 nodes, the time measured to complete all the selected transformations was greater than for a network of 32 nodes.

The rate of decrease in time required to complete the transformations over the first 4 transputers in the pipeline exceeded expectations. In general, if the number of nodes in the pipeline doubles, then under perfect conditions one would expect at most a 50 percent reduction in execution time. However, for this implementation, there was a 75 percent reduction in execution time over the first four nodes. Although this four fold decrease is surprising at first glance, an analysis of the implementation provides an explanation.

There would appear to be a discrepancy between the semantic expression of OCCAM and its realization in hardware. In particular, the ALT construct in OCCAM provides the facility of reading from a number of distinct channels from within a single process. ALT is supposed to read from the first channel that receives an input. However, the way in which this is achieved in hardware is that each input channel is assigned a priority based on its textual order in software and before a channel with a given priority can be read, all channels with a higher priority must be checked for

possible inputs.

In order to see how this design characteristic of the ALT construct has impacted on the implementation of SHLA, consider the flow diagram of Figure 3.3 In the diagram,

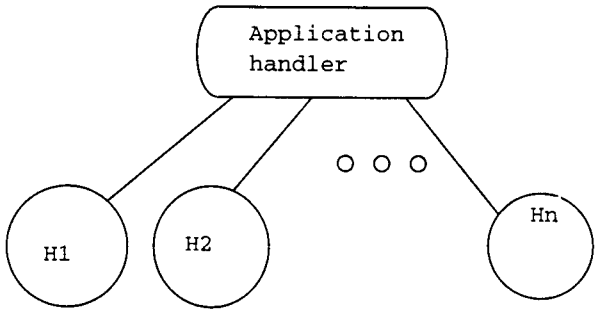


Figure 3.3: A flow diagram indicating the communication between the Application Handler and the $n \times H_i$ processes.

each H_i represents a process which is associated with a hexagon of SHM. All of these processes are concurrent and must be able to communicate with every other process. The Application Handler is a process that facilitates this communication by managing the n input channels from each of the H_is with an ALT. For the nth channel to be read, the first n-1 channels must be checked for inputs. Thus, for all n channels to be read, the number of channel checks is calculated as follows:

Channel read	*Number of checks required*
1	0
2	1
3	2
.	.
.	.
.	.

$$
\begin{array}{lcl}
 & n \qquad n-1 & \\
\text{---------- ----------} & & \\
\textit{Sum of the checks for the n channels} & = & 1+2+3+\cdots+(n-1) \\
 & = & n(n-1)/2 \\
 & = & n^2 - n/2
\end{array}
$$

This means that ALT is $O(n^2)$. In the case of this implementation of SHLA, each time the number of transputers doubled, the number of input channels to the Application Handler halved. This in turn is compatible with the *speed up* experienced by the implementation of the first four nodes.

Unfortunately, a comparison of the implementation's execution time on different configurations has not yet been realized. This is due to the occurrence of a hardware fault in the transputer network at the time of conducting the experiment. Although the code is in place for the *ring* and *grid* configurations for the implementation, the results of testing the implementation will probably have to wait until the hardware fault is corrected.

3.8 Interpretation of Results

The parallel implementation of SHLA discussed in this chapter has revealed that considerable *speed up* in execution time can be realized with increased numbers of transputers. One point made abundantly clear by the test results is that there is considerable scope for improvement in the coding of the algorithms. Over the period of development of the implementation, many design considerations were discovered, not the least of which was provided by experimentation with the ALT construct. It is anticipated that a second pass at an implementation of SHLA will result in improved

performance of execution time.

Chapter 4

A Segmentation Algorithm for SHLA

Kimo: "Can you now group that which you have counted, each with its own kind?"

4.1 Introduction

Segmentation is the process of subdividing an image into its constituent parts. The resulting subdivisions must be meaningful and the process should embody an organizing principle that reflects the innate organization of the visual world. The overriding purpose of the process is to reduce the complexity of the image. For purely economic reasons, it is a crucial component of most image analysis systems.

An image may be regarded as a collection of pixels (light intensities, each of which is associated with a unique location). If a pattern had to be recognized based on the analysis of pixels alone, the problem would be intractable for all but images composed of the smallest number of pixels. Mathematically, the number of possible patterns detectable within a given image is equal to the number (number of possible light intensities) raised to the power (number of pixels). For example, consider an image composed of a trivial number of pixels, as indicated in Figure 4.1. If each of the seven pixels had one of a possible 256 light intensities, then the number of possible patterns to be considered would be $256^7 = a\ big\ number$. An image of moderate size might contain one million pixels. If the number that represents the number of possible patterns from such an image were to be included in this thesis, it would require about 400 pages. Suffice to say there is no machine ever imagined that could cope with that kind of computational complexity. Consequently, the need for a process that will reduce the complexity of pixel data is overwhelming. The name given to the process is *segmentation*. Apart from the need for an effective segmentation algorithm for machine vision, there still remains a major challenge in biological vision research - to understand this process.

There have been many different types of segmentation algorithms devised to operate under a variety of conditions. Most of the segmentation algorithms derived to date require *a priori* information to be effective. That is, information about the

operational environment is required prior to the commencement of the segmentation process. This constraint of *a priori* information for many industrial applications is quite acceptable because information about the working environment is available. However, a general purpose machine vision system operating without *a priori* information requires an autonomous segmentation algorithm. That is, one which provides consistent and reliable results that is largely independent of the environment. For example, this would require the algorithm to produce consistent partitioning regardless of the object's orientation and source of illumination. Although a fully general purpose segmentation algorithm has yet to be described, it would appear that most biological vision systems possess such a segmentation facility. For example, the human vision system would perceive a typical kitchen fork irrespective of whether the fork is lying on a table or being held in a person's hand. A human would also experience these perceptions in a wide range of lighting conditions. Despite the ubiquitous occurrence of autonomous segmentation algorithms across the spectrum of biological vision systems, it has proven to be a tall order to produce one for a machine vision system.

Most traditional approaches to the problem of creating a segmentation algorithm considers the process of interpretation of an image as a separate and distinct process to that of segmentation. In contrast, segmentation, as considered in this work, is a dynamic process. It is composed of two levels of processing; a low level process that partitions on the basis of light intensity and locality alone, and a high level process which operates in accordance with learned rules of association. By analogy with the human, the rules of association are best thought of as the Gestalt rules of proximity, similarity, closure, continuation, symmetry and familiarity [46]. The segmentation process is then a concurrent interplay between these two levels of processing.

This chapter, for the most part, concentrates on the low level aspect of segmentation. The usual classification of segmentation algorithms places them in either the category of partitioning on the basis of grey-level discontinuities or on the basis of grey-level similarities. For a detailed account of the traditional methods, see [37]. The segmentation algorithm presented here falls into both categories, as it determines boundaries on the bases of abrupt grey-level changes and grows regions on the bases of similarities.

The algorithm is named 2DSA, (short for Two Dimensional Segmentation Algorithm) and has a number of distinctive features:

- The algorithm embodies a high degree of parallelism. That is, pieces of the image reside on a number of distinct processors and the segmentation process begins concurrently on all processors employing purely local (to the processor) information. The processors communicate the results of their local segmentation with other processors as the process evolves.

- The segmentation proceeds concurrently on all resolutions. Information gained at each resolution is passed to the high level process for interpretation

- The segmentation evolves in a top-down fashion. It commences with the partitioning based on the most abrupt and therefore most significant grey-level change. It then repeatedly produces finer partitionings based on the next most significant grey-level change. The process logically terminates when the number of equivalence classes equals the number of pixels in the original image. This is, in effect, a convergence property of complexity.

- The process of repeated refinements of the partitioning is concurrent with the interpretation process. Each refinement is interpreted. Consequently, the interpretation may conclude before the segmentation logically finishes.

- The low level process is completely autonomous. It is data driven; based only on the grey-level changes.
- The algorithm is intrinsically linked to SHLA. It is dependent on the adjacency properties of the hexagonal grid. That is, any vertex of a hexagon can be adjacent to, at most, two other hexagons.

4.2 Related Other Works

The literature abounds with descriptions of segmentation algorithms possessing the *a priori* information constraint and [81] provides an extensive review of the techniques. There have been a number of approaches that exploit the possibilities of representing the input images at various resolutions [62, 54, 105, 56, 22, 102, 109]. Other researchers have tried to utilize computational features adapted from biological vision systems. An aspect of the frog's vision system is treated in [5]. It is argued in [100] that vision must be understood at the complexity level and concludes that a bottom-up approach to segmentation is ineffective.

A paper that investigates segmentation in a spirit close to that of the one presented in this chapter is found in [3]. The paper presents a method of first representing the image in a particular manner and then employs a variety of modules to generate a description that satisfies certain Gestalt criteria.

4.3 Definitions and Terminologies

Before describing the algorithm in detail, it will be convenient to establish a few definitions and notations.

The *cluster* is an arrangement of seven hexagons, as displayed in Figure 4.1. It is characterised by the fact that the distance between the centres of adjacent hexagons

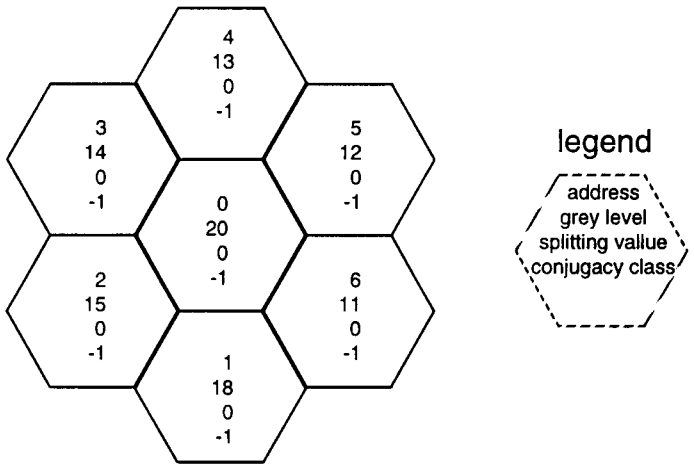


Figure 4.1: A collection of 7 pixels each of which is initialized with four pieces of information as indicated within the detached hexagon.

are equal. The term image will be used to denote a collection of pixels that can be partitioned into a collection of clusters. Thus, the clusters can be considered as the atomic elements of an image. Let $\mathcal{S}$ represent the entire image space (the collection of all hexagons, each of which is identified by a unique address). The segmentation algorithm will partition $\mathcal{S}$ into sub-spaces $S_1, S_2, ..., S_n$ in accordance with a *connectedness* relationship. This relationship, denoted by C, forms equivalence classes. That is, C has three properties:

reflexive property :

$\alpha C \alpha \, \forall \, \alpha \in \mathcal{S}$

symmetric property :

$\alpha C \beta \Rightarrow \beta C \alpha \forall \alpha, \, \beta \in \mathcal{S}$

transitive property :

$\alpha C \beta \, and \, \beta C \gamma \Rightarrow \alpha C \gamma \forall \, \alpha, \, \beta, \, \gamma \in \mathcal{S}$

The reflexive property insists that any element is connected to itself (intuitively trivial). The symmetric property insists that the connection applies in both directions. The transitive property states that any element in a chain of connected elements is connected to every other element in the chain. This property, in relation to the implementation of the algorithm, is the important one. The equivalence relation properties of C on $\mathcal{S}$ establishes the following three conditions:

$$\begin{aligned} \bigcup_{i}^{n} S_i &= \mathcal{S} \\ S_i \bigcap S_j &= \emptyset \forall i \neq j \\ \text{All elements} \quad S_i &\in \mathcal{S} \text{ are connected} \end{aligned}$$

The establishment of these three properties represents the first step in fulfilling the requirement that the partitioning is meaningful, both mathematically and visually. That is, each equivalence class A_i corresponds to an entity in the image. However, this does not guarantee that any particular entity is of value or interest at the interpretation stage. The following notation will denote the negation of the *connected-ness property*:

Let (i D j) symbolise the condition that the pixel at address i is not connected to the pixel at address j (reads address i has been disconnected from address j). Note that D is not an equivalence relation and is required to facilitate the execution of the algorithm.

A disconnection between adjacent addresses will be made on the basis of the significance of the differences in grey-levels at the addresses concerned.

Let δ denote the significance of the change in grey-level across two adjacent pixels, g_1, g_2 which are in the range 0 to 255[1].

$$\delta(g_1, g_2) = 256 - max(g_1, g_2) + (((513 - |g_1 - g_2|) \times |g_1 - g_2|) \div 2) \qquad (4.1)$$

[1]The δ function can be generalized to any range of light intensities.

For example, $\delta(255, 255) = 1$ the smallest and least significant change in grey-level.

$\delta(0, 255) = 32895$, the most significant change in grey-level.

The δ function was derived in such a manner so as to embody the following heuristic: distinct pairs of grey-levels give rise to a unique δ and that for a given absolute difference in grey-level, the δ is inversely proportional to the magnitude of the individual grey-levels. This approach imitates a characteristic of human vision. That is, grey-level changes at low light intensities are more noticeable than equal changes at higher intensity values. More specifically, a change in grey-level from 0 to 1 is more noticeable than a change from 254 to 255 even though the difference in grey-levels for both is one.

With this notation in mind consider the hypothetical image as represented in Figure 4.1. The image is composed of seven hexagons (pixels) each of which has associated with it four values, an address (number between 0 and 6), a grey-level (number between 0 and 255), a conjugacy class value (initially zero indicating that it is connected to address 0) and a splitting coefficient (initially the value is zero, indicating that it has not been disconnected) indicates the δ value used to determine a disconnection from another pixel.

4.4 Informal Statement of 2DSA

The notation and terminology of the previous section will now be employed to describe a segmentation algorithm named 2DSA (Two Dimensional Segmentation Algorithm). The description of this low level algorithm will unfold in a *bottom up* fashion. The first section will treat the three sub-procedures that operate at the cluster level. The second sub-section will demonstrate how these sub-procedures interact to handle images of arbitrary number of clusters.

Table 4.1: Cluster Computations.

iCj	$g_1 - g_2$	δ
$0C1$	$20 - 18$	747
$0C2$	$20 - 15$	1506
$0C3$	$20 - 14$	1757
$0C4$	$20 - 13$	2007
$0C5$	$20 - 12$	2256
$0C6$	$20 - 11$	2495
$1C6$	$18 - 11$	2009
$2C1$	$15 - 18$	1003
$3C2$	$14 - 15$	497
$4C3$	$15 - 16$	496
$5C4$	$16 - 17$	495
$6C5$	$17 - 18$	494

4.4.1 The Local Actions

Atomic Initialization :

Step 1: Compute the δ values for each of the twelve adjacency relations in the cluster. Table 4.1 illustrates the computations associated with the cluster of Figure 4.1.

Step 2: Initialize the Splitting Value to the largest δ value in the cluster. In this case, the splitting value is 2495. Set all twelve adjacency relations to *connected* and set the seven conjugacy class values to minus one.

Atomic Fission The second procedure to consider is a process of disconnection called *Atomic Fission*, which operates at the cluster level. The process commences with the input of two addresses that have been *disconnected*. Then

further *disconnections* are made to resolve any contradictions to the laws governing equivalence relations.

To illustrate the process consider the data from Figure 4.1 where 0C6 is the input *disconnection*. If this were the only disconnection to be made, the equivalence relationship criteria of transitivity would be violated. That is, 0 is still connected to 6 through both 1 and 5. The violations must be resolved. First, through 1 by disconnecting either 0C1 or 1C6. Disconnect the one with the greater δ (both, if they are equal). In this case 1C6 is the larger so re-label it as 1D6 and the violation is resolved. The second violation, through 5, is resolved in a similar manner. Here 0C5 is disconnected as its δ is greater than that of 5C6. Re-label it as 0D5. However, this creates a violation through 4. Thus, resolve this violation and continue the process of *disconnecting and resolving* until the equivalence relationship is restored. In this case the following sequence of *disconnections* will be made: 0D4, 0D3, 0D2 and 2D1. At this point, the *fission* process is complete.

Atomic Fusion: The third procedure to be described is a *fusion* process which establishes a pattern for a cluster that is independent of grey-levels. This process has a hexagon address and a conjugacy class value as its inputs. The process assigns this conjugacy class value to all hexagons which are *connected* to the hexagon of the input address.

To illustrate how this works, continue with the example from Figure 4.1. In this case, the inputs are (0, 0) which indicates the address and conjugacy class value of the centre hexagon. The assignments will be conjugacy class value, 0, to addresses 0 and 1. Further, if the inputs were address 2 and conjugacy class value 2, the addresses 2, 3, 4, 5 and 6 would be assigned the conjugacy class value 2. The fusion is complete and the segmentation of the cluster is also

complete. This partitioning generated two equivalence classes. One contains the hexagons at addresses (0, 1) and the other has the addresses (2, 3, 4, 5, 6). Observe that the relative positions of the hexagons in each conjugacy class, as designated by the address of each hexagon, establishes a pattern which is now independent of the grey-level of the individual hexagons.

4.4.2 The Global Actions

The three *local* processes just described have *global* counterparts that extend the process so that images composed of more than one cluster can be segmented. The *global* activity is governed by a synchronization process that coordinates the *atomic* activity. This synchronizing process communicates with and controls the processing at each of the 7^n (one for each hexagon) processes, H_i. In the present example, the processes are $H_0, ..., H_{48}$. The processing associated with each hexagon is the combined processing of the three *atomic* processes. Observe that the *atomic* processes are essentially sequential. The *global* processing, on the other hand, is a collection of processes that operates concurrently at all resolutions. Figure 4.2 is a diagram of the aforementioned processes.

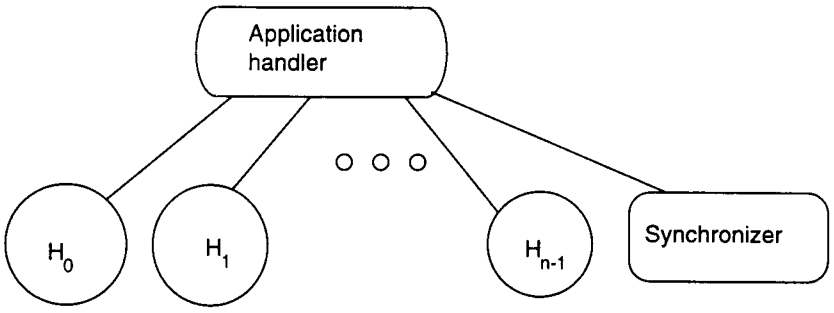


Figure 4.2: A flow diagram of the segmentation program.

Once again, each hexagon which now may be identified by the process H_i, has

associated with it four values: an address (in this case a number between 0 and 66 in base 7), grey-level (number between 0 and 255), a conjugacy class value (all minus one) and a splitting value (initially zero). In addition, each H_i is classified as being either *active* or *passive*. The *active* H_i's are identified by *Spiral Counting* in the key of 15. There are 19 such addresses in a 49 hexagon SHM. They are the addresses 0, 15, 26, 31, 42, 53, 64, 11, 13, 22, 24, 33, 35, 44, 46, 55, 51, 66, 62 and are marked with a $\star$ in Figure 4.3

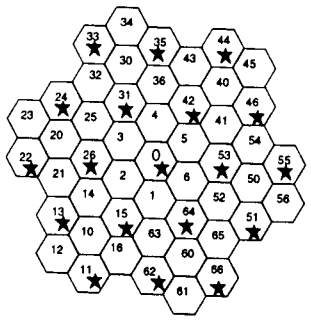


Figure 4.3: The diagram indicates the relationship between the active and passive processes of 2DSA for a 49 hexagon SHM. The active processes are marked with a $\star$.

Each of the above active ($\star$) processes is associated with a hexagon that represents the centre hexagon of a cluster which has the following properties:

- Each cluster overlaps with at least one other active neighbouring cluster.
- An overlapping pair of clusters has two hexagons in common.

The *global* processes will be illustrated by considering a synthetic image represented on the 49 hexagons of Figure 4.3. Let the first seven hexagons, addresses 0 to 6, be initialized with grey-values as in Figure 4.1 and let the remaining hexagons all have a grey-value of 13.

Synchronization This process governs the three processes of *global* Initialization, Fission and Fusion. It communicates with each of the 7^n processes where *atomic* initialization, fission and fusion occur. The process passes appropriate messages that initiate and terminate the three *global* processes.

Global Initialization This is the collective term for initialization at each of the H_i's. Each *passive* process passes its grey-level to each of its neighbouring *active* processes and awaits a response. Each *active* process receives a grey-level from each of its neighbours and performs *Atomic Initialization* on this data. Each then passes its *Splitting Value* to the *Synchronizer* which selects the largest of these *Splitting Values*. The Synchronizer then passes this largest splitting value back to each of the H_i's, so that the *fission* process may commence.

In the example, the largest δ value is, once again, 2495 and is associated with 0C6. This value emanates from H_0 and is assigned to the *Splitting Value* by the *Synchronizer*.

Global Fission The process commences when the *Synchronizer* passes the *Splitting Value* to each of the *active* H_i's. At each *active* H_i, perform *Atomic Fission* with the *Splitting Value* which has been passed to it. Thus, a disconnection will occur at each H_i that possesses a δ equal to the *Splitting Value*. If a disconnection has occurred between any two non-centre hexagons of a H_i, this will create a possible violation to the equivalence relation for some other H_j, $i \neq j$. In this case, the *disconnection* information is passed to the appropriate H_j; otherwise an *inactive* message is passed to the *Synchronizer*. If any process receives *disconnection* information, it performs another *Atomic Fission* and the appropriate communication. This *Global Fission* terminates when the *Synchronizer* receives the *inactive* message from all of the H_i's.

In the example, H_0 will produce *disconnections* as described in the previous Section. This will generate *disconnection* information for H_{15} and H_{64}; namely, 2D1 and 1D6, respectively. *Atomic Fission* at H_{15} will make disconnections at 1D15 and 1D64 while H_{64} will disconnect 1D64 and 1D63. The two processes will pass the *disconnection* information 1D63 to each other. When this information is received, no further violations of the equivalence relation will have been generated. At this point, the *Synchronizer* will have received *inactive* messages from all processes and it can then commence the *Global Fusion* process.

Global Fusion When H_0 receives the appropriate message from the *Synchronizer*, *Atomic Fusion* with inputs (0,0) commences at this address. This *seeds* the fusion process and a *wave* of effects passes to other H_i's in the following manner: $H_{15}, H_{26}, H_{31}, H_{42}, H_{53}, H_{64}$ receive conjugacy class assignments from H_0 about the hexagons that each has in common with H_0. *Atomic Fusion* is initiated on any of the hexagons that are *connected* to the common ones. If any of these processes produce effects on other H_i's, they pass this *fusion* information to the appropriate H_j, and so on. When this first *wave* of conjugacy class labelling subsides, the *Synchronizer* starts the second *wave* at the next *active* process. The *Synchronizer* repeats this process until all of the hexagons have been labelled. When *Global Fusion* terminates the *Synchronizer* then passes a message to all of the H_i's to send their conjugacy class information to the higher level segmentation process and selects the next largest δ value and re-starts the *Global Fission* with this as its *Splitting Value.* The *Synchronizer* repeats this process until either it completes the smallest δ value or it receives a *stop* message from the higher level segmentation process.

In the case of this example, the *Global Fusion* will be completed in the first *wave* as follows: as previously described, H_0 will generate two conjugacy classes; one with

addresses (0, 1) and labelled with conjugacy class value 0 and the other with addresses (2, 3, 4, 5, 6) labelled with conjugacy class value 2. H_0 then passes the appropriate *fusion* information to its six active neighbours where the following assignments take place:

H_{15} address 1 is labelled 0 and the rest of its addresses, (15, 16, 10, 14, 2, 63) are labelled 2. The *fusion* information is passed on to its active neighbours where similar labelling occurs.

H_{26} labels all of its hexagons with 2 and passes this information to $H_{15}, H_{13}, H_{22}, H_{24}$ and H_{31}. Likewise, each of H_{31}, H_{42} and H_{53} label all of their hexagons with 2 and pass this information to each of their active neighbours.

H_{64} labels address 1 with conjugacy class value 0 and addresses (64, 60, 63, 6, 52 and 63)) with 2 and passes this information on to its active neighbours.

All remaining processes label the rest of their hexagons with 2 and the *wave* will subside.

The *Synchronizer* passes a message to the $H_{i's}$ to pass the conjugacy class information to the higher level process and commences the *Fission* process at the next highest δ value.

4.5 Formal Statement of 2DSA

The description of 2DSA from the preceding section is summarized in the following more formal statement of the algorithm. Here Pascal-like pseudo code, with an added parallel construct is employed.

Let the term PAR be used as a replicator of a block of statements that are to be executed concurrently over some index. Let an image I be of size 7^n. The highest level block has four concurrent procedures.

```
(* begin main body of program*)
PAR over all resolutions of image
  PAR
    Synchronizer
    Global Initializer
    Global Fission
    Global Fusion
(* end main body*)
(* procedures called from main body *)
procedure Synchronizer
  REPEAT (synchronizing)
    PAR
      Output message to commence Global Initialization
      Input message from each process indicating its largest delta value
      Output message to commence Global Fission
      Input messages from all processes indicating the result of their Atomic
         Fission
      Output message to commence Global Fusion when Global Fission has completed
UNTIL there are no more delta values
```

```
procedure Global Initialization
  PAR at all addresses
    PAR
      Input message of commencement
      IF address is identified in Spiral counting in the key of 15 for each
         near copy, then the address is Active; otherwise the
         address is Passive.
      IF Address is Passive then Output grey-level; otherwise Input grey-levels.
      Atomic Initialization
      Output message to Synchronizer, indicating the
        maximum delta value for the cluster

procedure Global Fission
  PAR at all addresses
    PAR
      Input fission information from Synchronizer
      WHILE fission information indicates that there is more fission to perform
         Atomic Fission
         Output messages to Synchronizer and all affected processes,
             the results of Atomic Fission
procedure Global Fusion
  PAR
    Input fusion information
    WHILE fusion information indicates that there is more fusion to perform
      Atomic Fusion
      Output messages to Synchronizer and all affected processes the results
        of Atomic Fusion
```

```
procedure Atomic Initialization
   Compute delta values for each of the
       twelve adjacency relations in the cluster
   Set conjugacy class value for each hexagon to -1
   Set Splitting Value for each hexagon to 0

procedure Atomic Fission
  Input disconnection information
  Resolve any equivalence relationship violations
 procedure Atomic Fusion
   Input an Address and a Conjugacy class value
   Assign Conjugacy Class value to any hexagon connected to Address
```

4.6 A Transputer Implementation of 2DSA

This section describes the implementation strategy adopted for 2DSA. In particular, the treatment will include a discussion of the more difficult parts of the algorithm, the problems faced and the compromises made in the coding process.

The algorithm was designed to be fully compatible with SHLA as presented in Chapter 3. Consequently, the implementation follows a similar line of development. All the transformations referred to in the previous section, M_x, are fully described in Chapter 3 and as such will not be mentioned again here.

The first issue to be considered is the degree of parallelism achieved by the implementation. Although in theory one should design for optimum parallelism in a program, in practice one has to make a decision regarding the amount of parallelism that is achievable in the implementation. The reason for this is a pragmatic one. Sequential constructs tend to be less complicated than concurrent ones. One may

observe that the *Atomic* procedures possess considerable parallel potential but it was decided to implement these procedures as sequential segments. The parallelism sought was derived from the association of each hexagon with a concurrent process. This decision alone meant that in practice very little of the potential parallelism could be obtained due to the limited number of transputers available. More specifically, the real[2] parallelism, as opposed to the functional[3] parallelism, was achieved by grouping approximately equal numbers of processes on individual processors.

The processors were then configured in either a pipeline, ring or grid.

The big programming task of this implementation emerged from the requirements of the *synchronization* process. A *Synchronizer* was placed at each processor in the network. Its job was to coordinate the three processes of *Initialization, Fission* and *Fusion* on its own processor and to communicate with the *Synchronizers* on the other processors. Upon completing the *Initialization* process, each *Synchronizer* would send a message of *completion* to the other *Synchronizers.* Upon receiving a message of *Initialization complete* from all other processors, it initiates the *Fission* process on its processor. A similar strategy was employed to coordinate each refinement of the *Fission* process and each *wave* of the *Fusion* process.

At the same time that the synchronization was occurring, there was also considerable message passing between the H_is at each of the three stages: *Initialization, Fission* and *Fusion.* All of the communications required between processors were handled by a multiplexer which established virtual channels between processors. As a result of the need for this amount of message passing, considerable programming time was spent in the debugging of the *diabolical deadlock* creeping into the design. However, this implementation was deadlock free for all of the test images that were

[2]The term *real* parallelism refers to the situation that two or more processes have the possibility of executing concurrently.

[3]The term *functional* parallelism refers to the situation when the processes are expressed concurrently for the purpose of ease in the writing of the code.

input to it. The main problem of the implementation related to the demand for memory at each processor. This arose largely from the inefficiencies in the coding of the *Synchronizer*. Most processors in the network had one megabyte of memory. This limit was reached with the third octave of SHM. At most only $7^3 = 343$ hexagons could be employed to represent any image. Although to date this problem has not been overcome, evidence suggests that a re-design of the *Synchronizer* will remedy the situation. It is anticipated that the number of desired octaves to be handled is many times greater than the present constraint.

4.7 Simulation Results

The results of the segmentation algorithm on the implementation as previously described is treated in this section. Data on four different images will be presented and discussed. Three of these images - a circle, annulus and trapezoid (see Figures 4.4, 4.8, 4.9), are synthetic and were selected to illustrate various points about the algorithm. The fourth image considered is that of the can of Pal[4] dog food which was used to illustrate the transformations discussed in Chapter 2. Each of the images will be considered at three resolutions.

4.7.1 Synthetic images

Each of the three synthetic images were created with 7^3 hexagons. At the lowest resolution, obtained by M_{10}, there are 49 *near copies* each of which contains 7 hexagons. At the middle resolution obtained by M_{100}, there are 7 *near copies* each of which contains 49 hexagons. The highest resolution is represented by the entire collection of all 343 hexagons.

[4]Pal is a trademark of the Mars Corp.

Circle The data generated from the first partitioning pass at the lowest resolution of the circle of Figure 4.4 is presented in Figure 4.5 in the form of patterns.

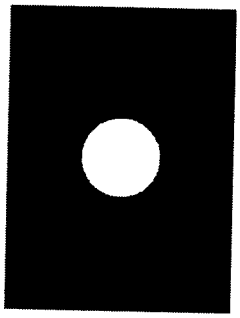


Figure 4.4: A synthetically generated circle.

The number of occurrences of each pattern is also indicated. It can be seen that of the 49 samples, 41 of them produced the same pattern, two conjugacy classes one of which contains the single centre hexagon of the cluster.

The image was created with only two grey values. Consequently, at the second pass for the next highest grey level, the process logically finished because each of the samples produced a pattern composed of 7 conjugacy classes.

The results from the 7 samples produced at the middle resolution are presented in Figure 4.6. Each of the 7 patterns is slightly different but each roughly approximates the shape of a circle. The sample centred at address 0 represents the closest possible approximation to a circle that could be represented with 7 hexagons.

The segmentation results of the full 343 hexagon representation of the circle is presented in Figure 4.7. Observe that the partitioning produced two conjugacy classes. As the shape of the class containing the darkened hexagons is as close to the shape of a circle as can be represented with a similar number of hexagons,

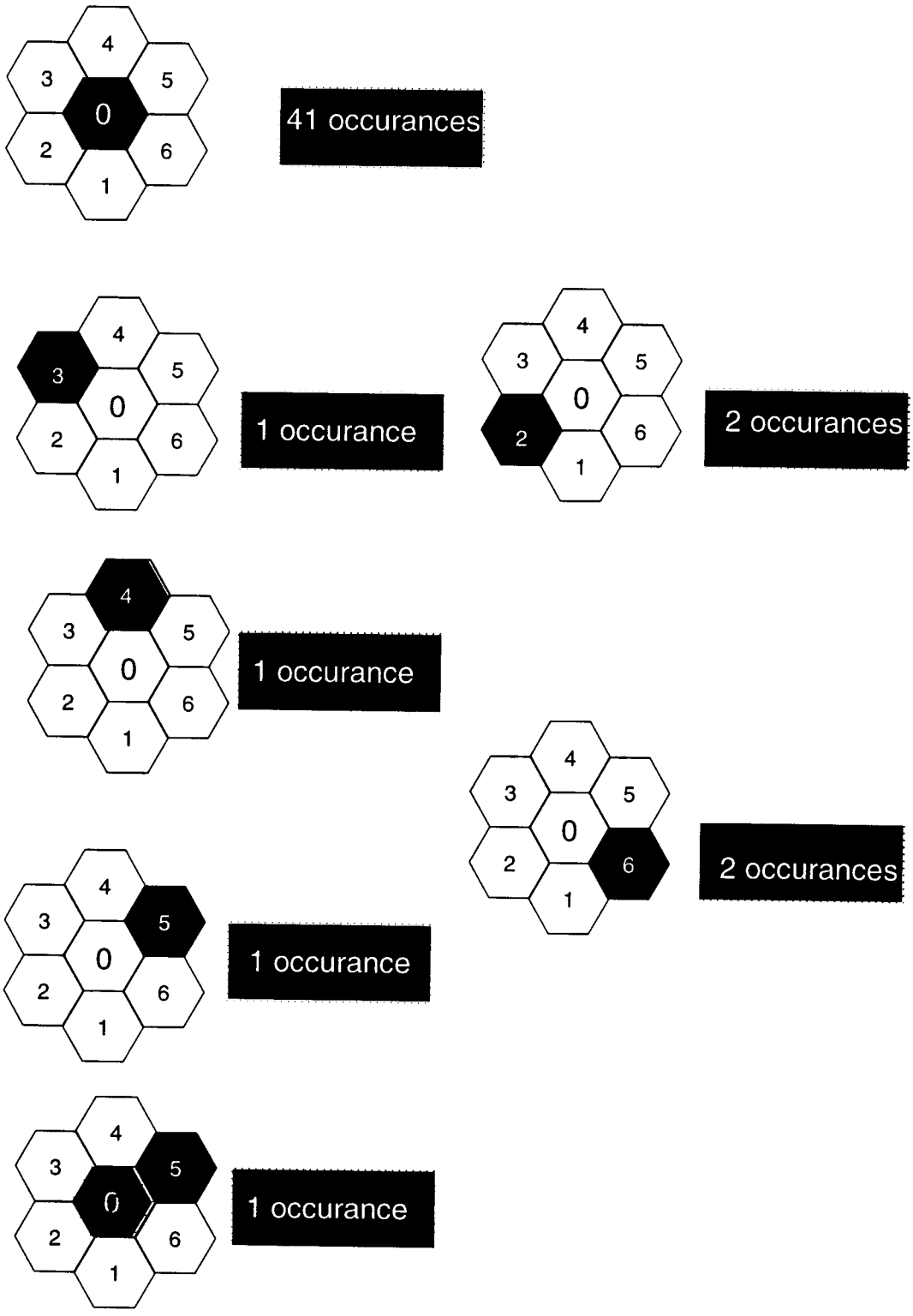


Figure 4.5: The 49 patterns generated by 2DSA for the representation of the circle (Figure 4.4) represented at the lowest resolution.

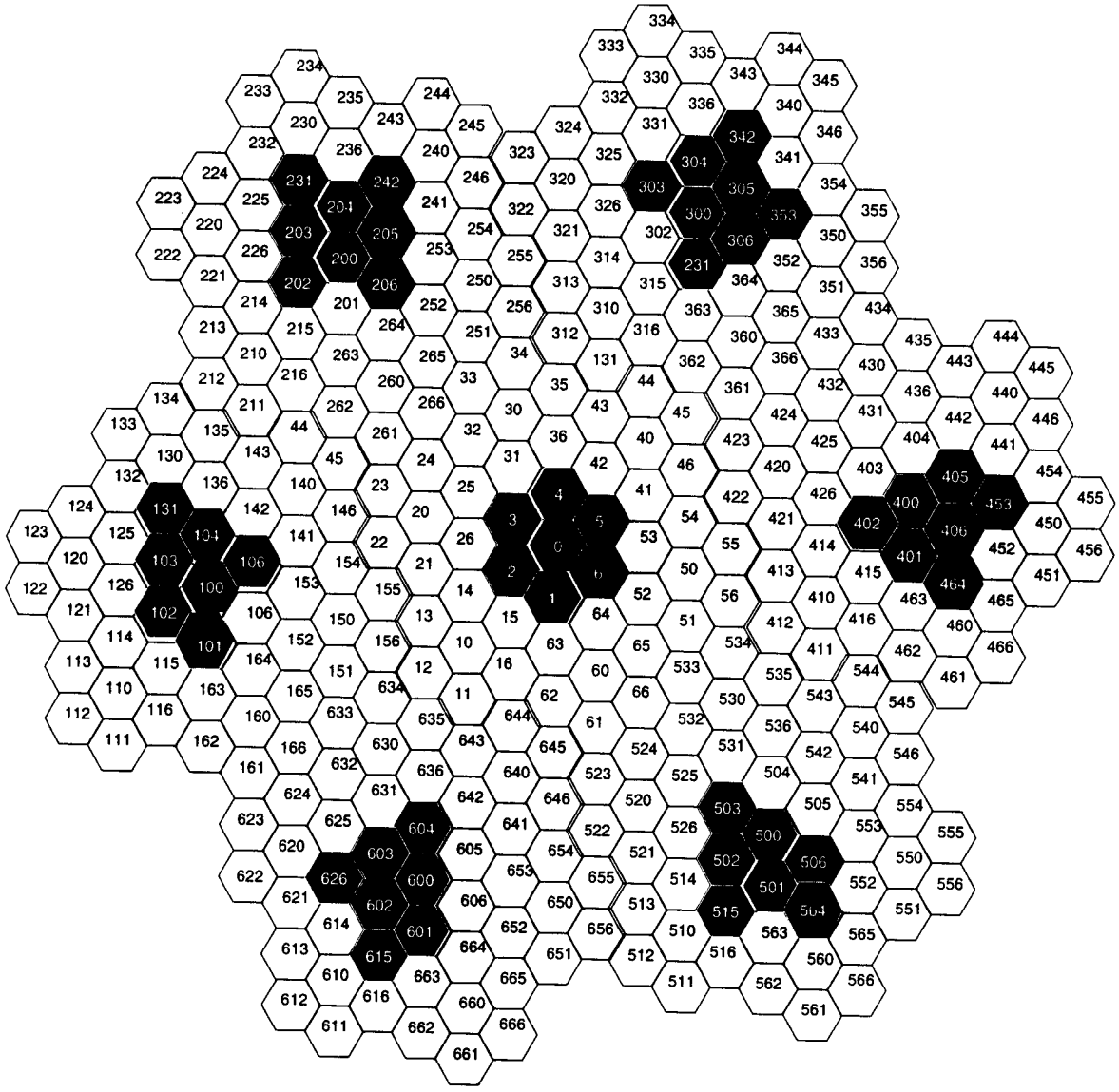


Figure 4.6: The 7 patterns generated by 2DSA for the representation of the circle (Figure 4.4) represented at the middle resolution.

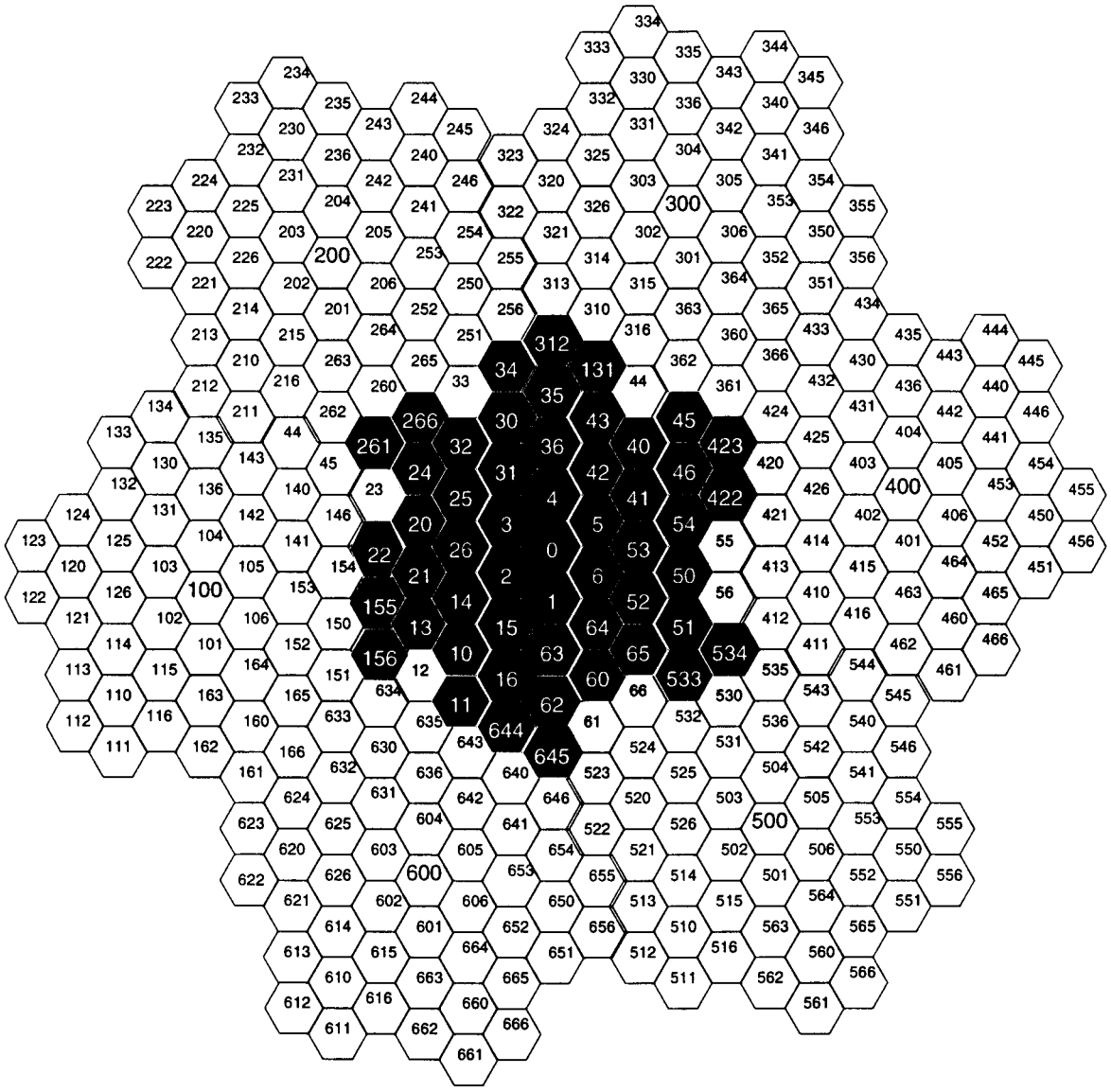


Figure 4.7: The single pattern generated by 2DSA for the representation of the circle (Figure 4.4 represented at the highest resolution.

the *best guess* at the shape represented by this pattern is a circle.

Annulus The segmentation results for the annulus of Figure 4.8 produced identical results to that of the circle for the first partitioning pass. This was not surprising

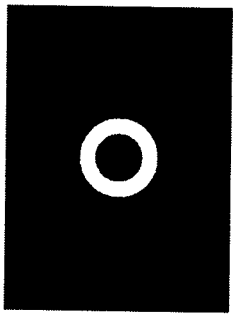


Figure 4.8: A synthetically generated annulus.

as the same equation was employed to generate the edge of the circle and the outer edge of the annulus. However, the inner circle of the annulus was generated with a different grey-value which produced a smaller δ value. Consequently, the inner circle was not picked up until the second pass. At the lowest resolution, the second pass did not produce any change in the variety of patterns. At the middle and higher resolutions, the inner circle emerged and generated three conjugacy classes roughly correlating to the three distinct forms observable in figure of the annulus.

Trapezoid The performance of 2DSA on the trapezoid of Figure 4.9 indicates how a non-circular shape emerges through the resolutions. At the lowest resolution, 25 of the 49 samples indicated a non-trivial pattern. All of these contained two conjugacy classes with a single hexagon in the smaller class. Eleven had the class which contained the zero address and twelve had the single element class at address five. Although no single shape is strongly discernible at this resolution,

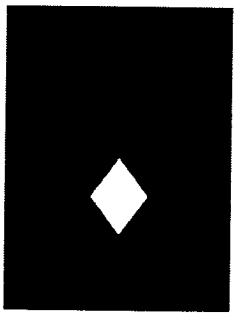


Figure 4.9: A synthetically generated trapezoid.

the analysis of the data would indicate the existence of a form located somewhat off centre from the optical axis towards address 50. This type of information would be used to manage foveal sensing in which case the sensor would be shifted in the direction of the address 50.

At the middle resolution of the trapezoid, a minor indication of shape is beginning to emerge. See Figure 4.10. The strongest indication is from the pattern centred at addresses 200 and 400 where both have four elements in their smallest conjugacy class and indicate the same pattern.

The data for the highest resolution is presented in Figure 4.11, where the trapezoidal shape is clearly visible. Given the resolution of the figure, 343 pixels, the trapezoidal shape has been captured with near perfect precision.

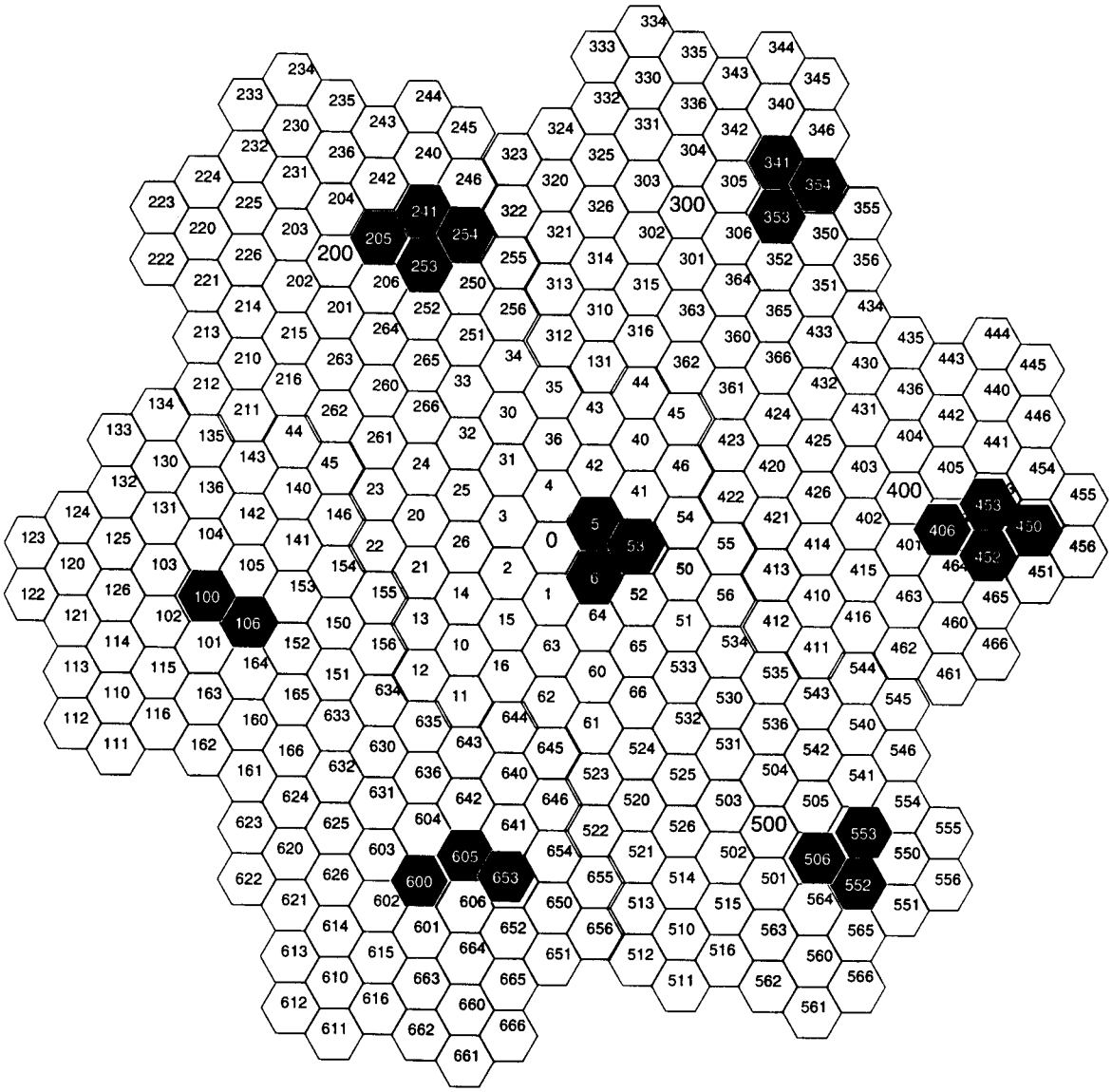


Figure 4.10: The 7 patterns generated by 2DSA for the representation of the trapezoid (Figure 4.9) represented at the middle resolution.

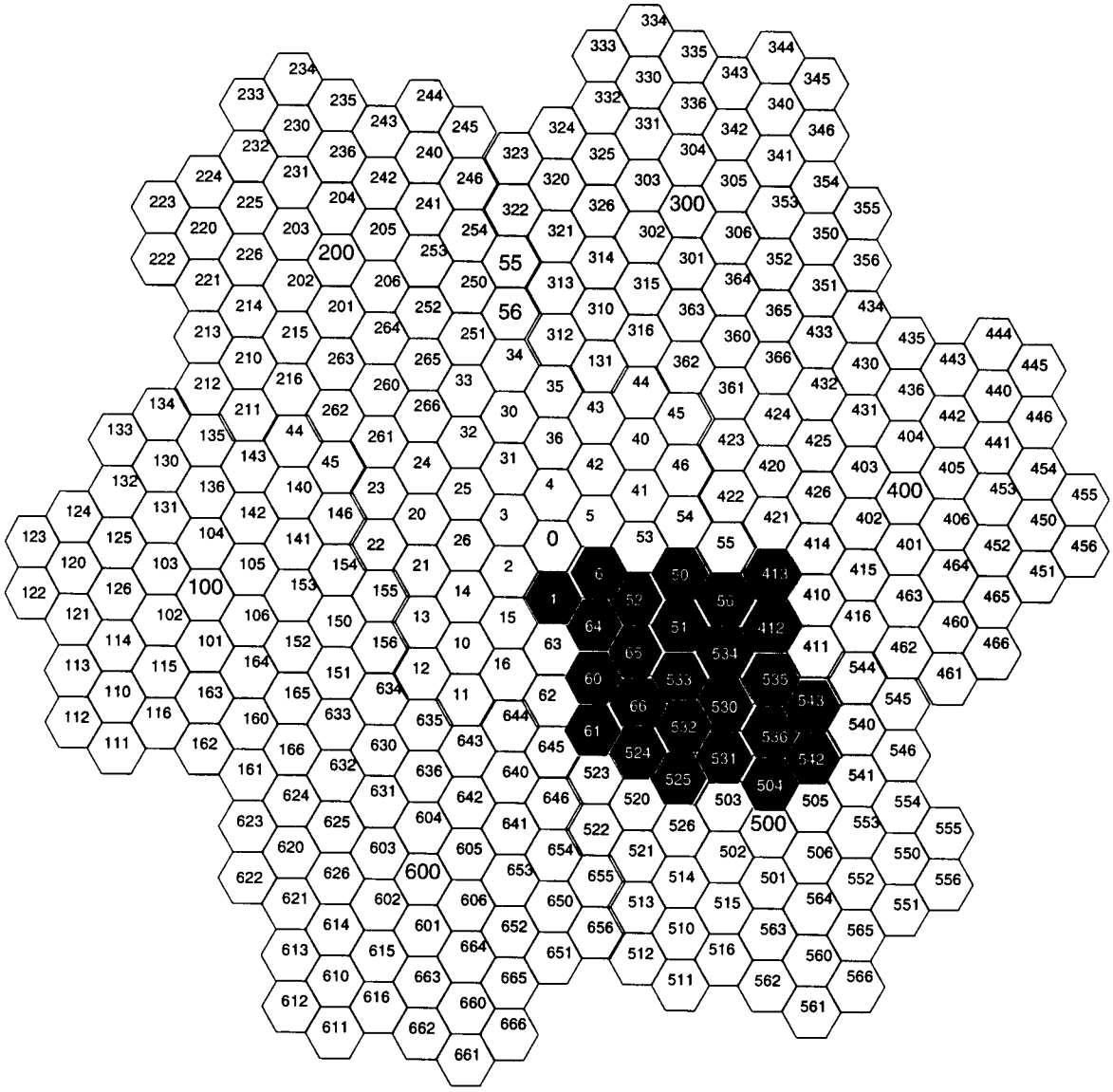


Figure 4.11: The single pattern generated by 2DSA for the representation of the trapezoid (Figure 4.9) represented at the highest resolution.

4.7.2 Natural image

The image of the can of Pal dog food, as presented in Chapter 2, is taken as an example of a natural image. Six transformations as generated from M_{10} are presented in Figure 4.12. Before a discussion is given of the results of applying 2DSA to this image, the salient features of these six transformations are recapped. The image in the upper left hand corner represents the original image at the highest resolution, 7^6 pixels. The image in (upper right) represents the image at a lower resolution as produced by applying the inverse of M_{10} to the image in (upper left). There are seven unique *near* copies of the image represented. Each of the copies is composed of a unique collection of 7^5 pixels from the original. Likewise, the image in (middle left) is obtained from applying the inverse of M_{10} to (upper right). It represents 49 unique *near* copies of the original. Each of these copies is composed of a unique collection of 7^4 pixels from the original. Subsequent applications of the inverse of M_{10} produces an increase by a factor of seven to the number of unique *near* copies and an accompanying decrease by the same factor in the number of pixels in each copy. Thus, after the 5th application of the inverse of M_{10} to the input image (lower right), there are 7^5 copies each of which is composed of 7 unique pixels from the original. Consequently, each of the six transformations represents the input image at a different resolution without *any* loss of light intensity information. This feature of 2DSA facilitates the commencement of the segmentation process at all resolutions of the input image without alteration to the light intensity data. This is a crucial point as the multi-resolution representations of the original image could have been achieved by an *averaging process* such as a Gaussian filter. However, with such a process, potentially important information would be lost at all resolutions bar the highest.

It was originally intended that 2DSA in its entirety be applied to the image at these

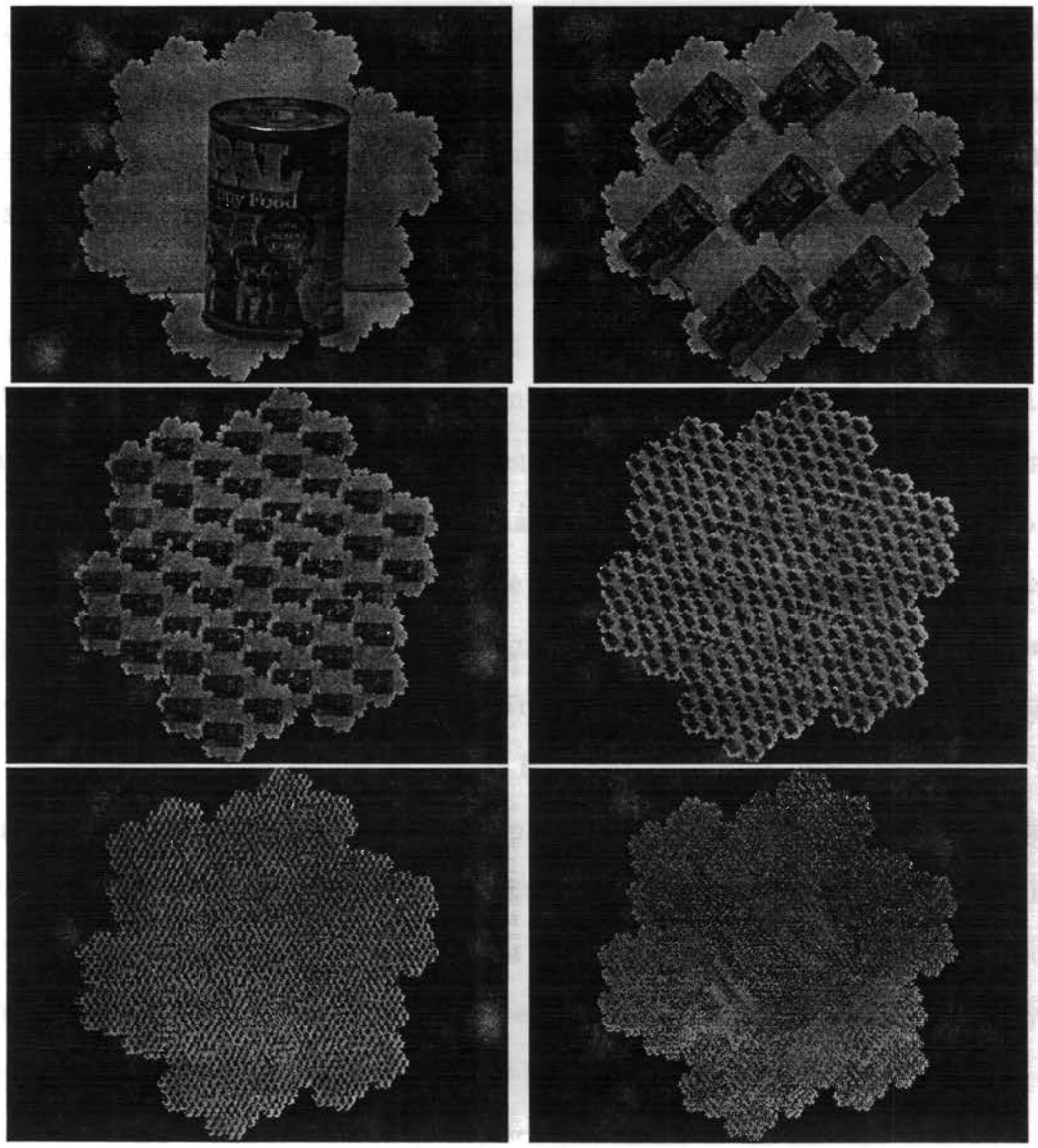


Figure 4.12: (upper left) original image with 7^6 pixels, (upper right) application of M_{100000} to the original image producing 7 unique copies each of which contains 7^5 pixels, (middle left) application of M_{10000} to the original image producing 49 unique copies each of which contains 7^4 pixels, (middle right) application of M_{1000} to the original image producing 7^3 unique copies each of which contains 7^3 pixels, (lower left) application of M_{100} to the original image producing 7^4 unique copies each of which contains 49 pixels, (lower right) application of M_{10} to the original image producing 7^5 unique copies each of which contains 7 pixels.

six resolutions. However, as the full implementation limited the representation to $7^3 = 343$ hexagons, a number of compromises were made in order to apply the segmentation process to the image represented at higher resolutions. The main compromise was to the parallelism inherent in 2DSA. A sequential version of the algorithm was designed. The amended 2DSA was implemented on the root transputer of the available network. This transputer had four megabytes of memory available and was able to handle images of size 7^5 pixels.

We can now proceed with the discussion of applying the amended version of 2DSA to the image in Figure 4.12. The full implementation of 2DSA would have segmented all *near* copies at all resolutions of the image concurrently. The amended version of 2DSA permits the segmentation of all of the copies at all resolutions to be performed in a sequential fashion. For the purpose of providing visual evidence of the effectiveness of the segmentation process, results for one copy from each of three distinct resolutions will be presented. The choice of copies for each resolution was arbitrary. First, consider the image as represented by the collection of 7^3 pixels taken from the centre-most *near* copy in (middle right) from Figure 4.12. At this resolution, very little of the detail contained in the original, highest resolution, image is present. Indeed, only the approximate shape of the object can be observed at such a resolution. Purely for the sake of visual display, the original 343 hexagonal pixels have been scaled up and Figure 4.13 is an attempt to portray the original image of the can of Pal dog food at this reduced resolution without sacrifice to size. It is important to stress the point that the segmentation was performed on the image as represented by the 343 pixels in the centre-most copy displayed in Figure 4.12 (middle right) and not on the image portrayed in Figure 4.13.

The results of segmenting the image at various δ grey-values are displayed in Figure 4.14. The main result of this segmentation is that the shape of a rectangle has

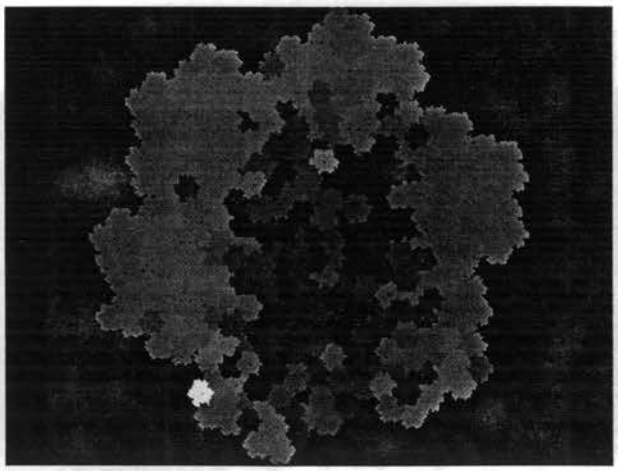


Figure 4.13: The image represents a scaling up of one of the copies from 4.12 (middle right).

emerged from the process. This shape results from the projection of a cylinder onto a 2-dimensional surface. The rectangular shape evolved as follows: Segmentation at the first (highest) δ grey-value produced two conjugacy classes. These two classes are represented in Figure 4.14 (upper left) by the collection of dark connected *blobs* and the collection of lighter shaded *blobs*. The perceived shape within the image results from the boundaries of the two classes. At the completion of this first iteration of segmentation, even though there is a definite shape present, the shape is not *recognisable*.

The results of the second pass of segmentation at the next highest δ grey-value is portrayed in Figure 4.14 (upper right). Here, a third class has emerged and is located near the middle of the lighter class and at the left hand side of the dark class. Observe that the human eye tends to join the new class with the darker class. This occurs for at least two reasons: firstly, the average grey-level of the new class is closer to that of the darker class than the lighter class. Secondly, such a joining results in the production of a longer straight edge (recognisable shape). The difference in average

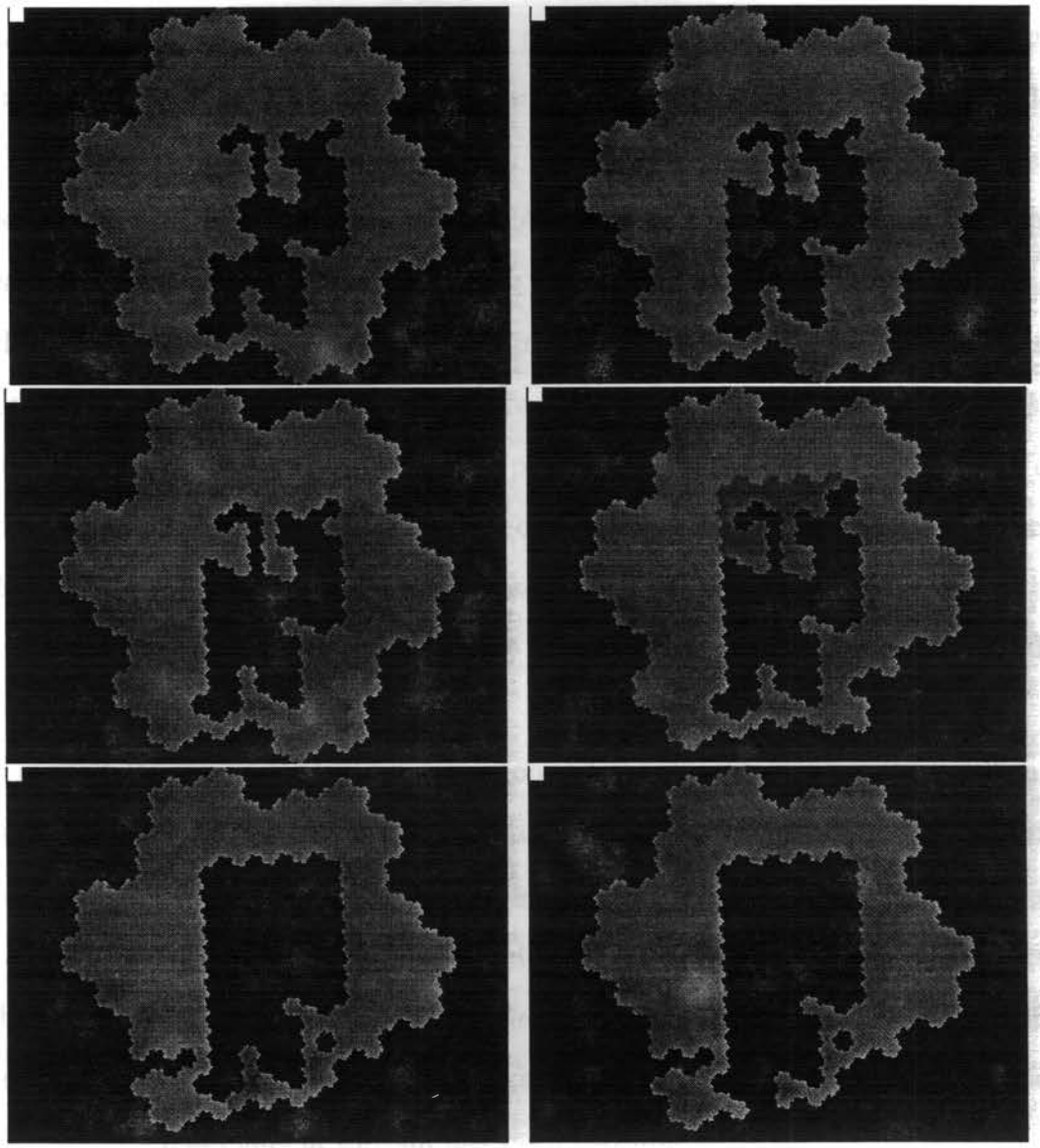


Figure 4.14: The collection of images represent six iterations of the segmentation algorithm as applied to the image from Figure 4.13; (upper left) first iteration at highest δ grey-value producing 2 conjugacy classes, (upper right) second iteration, (middle left) third iteration, (middle right) fourth iteration, (lower left) 8th iteration, (lower right) 12th iteration.

grey-levels can be quantified by the δ function, Equation 4.1. The effect of joining the new class to the darker class results from taking the complement of the lighter class. The third pass at the next highest δ grey-value resulted in the production of a fourth, rather small, class at the border of the field in the direction of 5 o'clock. See Figure 4.14 (middle left). The fourth pass isolated another class that seems to correspond to the top of the can and the PAL letters. The complement of the lightest class now produces a shape approximating three sides of a rectangle. See Figure 4.14 (middle right). Subsequent passes at progressively smaller δ grey-values continue to isolate other entities in the image. Figure 4.14 (lower left) represents the situation after the 8th pass. Observe the new class projecting in from the border near the edge of the field at the 7 o'clock location. This class corresponds to the line of intersection of the surface the can rests on and the background wall. Although a definite shape has emerged by the 12th pass, see Figure 4.14 (lower right), the lower right side of the can has remained undetected.

Figure 4.15 is a scaled up version of one of the 49 *near* copies displayed in the image of Figure 4.12 (middle left). At this slightly higher resolution using 7^4 pixels, in addition to the outline of the can appearing sharper, a somewhat blurred representation of the PAL is visible. The results of segmenting this image are displayed in the Figure 4.16. The shape of the can emerges in a similar fashion to that of the lower resolution. At this resolution though, the boundaries are sharper and more of the lower right hand side of the can has been captured.

The can is displayed at the second highest resolution in Figure 4.17. This figure is composed of 7^5 pixels and represents one of the seven *near* copies visible in the (upper right) image of Figure 4.12. At this resolution the PAL letters are clearly visible. The results of segmentation are displayed in Figure 4.18. The first iteration at the highest δ grey-value has included the pixels associated with the PAL letters in

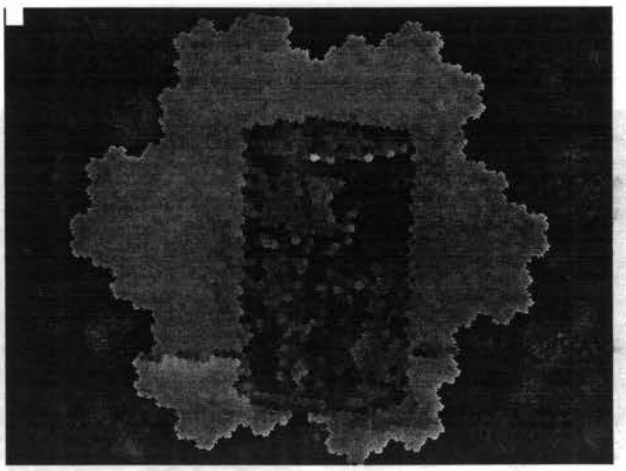


Figure 4.15: The image represents a scaling up of one of the 49 copies in Figure 4.12 (middle left).

the background class. See Figure 4.18 (upper left). At the next iteration, the PAL letters were isolated into a class by themselves. See Figure 4.18 (upper right). At subsequent iterations Figures 4.18 (lower left) and (lower right) displays the shape of the can with greater detail. In these two figures, the curvature of the top of the can is clearly visible.

It was stated at the outset of this chapter that a segmentation algorithm that is to be used as a component of a general purpose machine vision system should be capable of commencing the process without *a priori* information about the input image. A consequence of this paradigm is that all resolutions of the input image have equal priority at the commencement of the segmentation process. The results presented for the PAL image illustrates the significance of this requirement. Observe that different information was obtained at each resolution and that this information was obtained at differing computational costs. At the lowest resolution, an approximate rectangular shape emerged at the cost of considering only 343 pixels. At the highest resolution more detailed information, curvature of the top of the can and the PAL

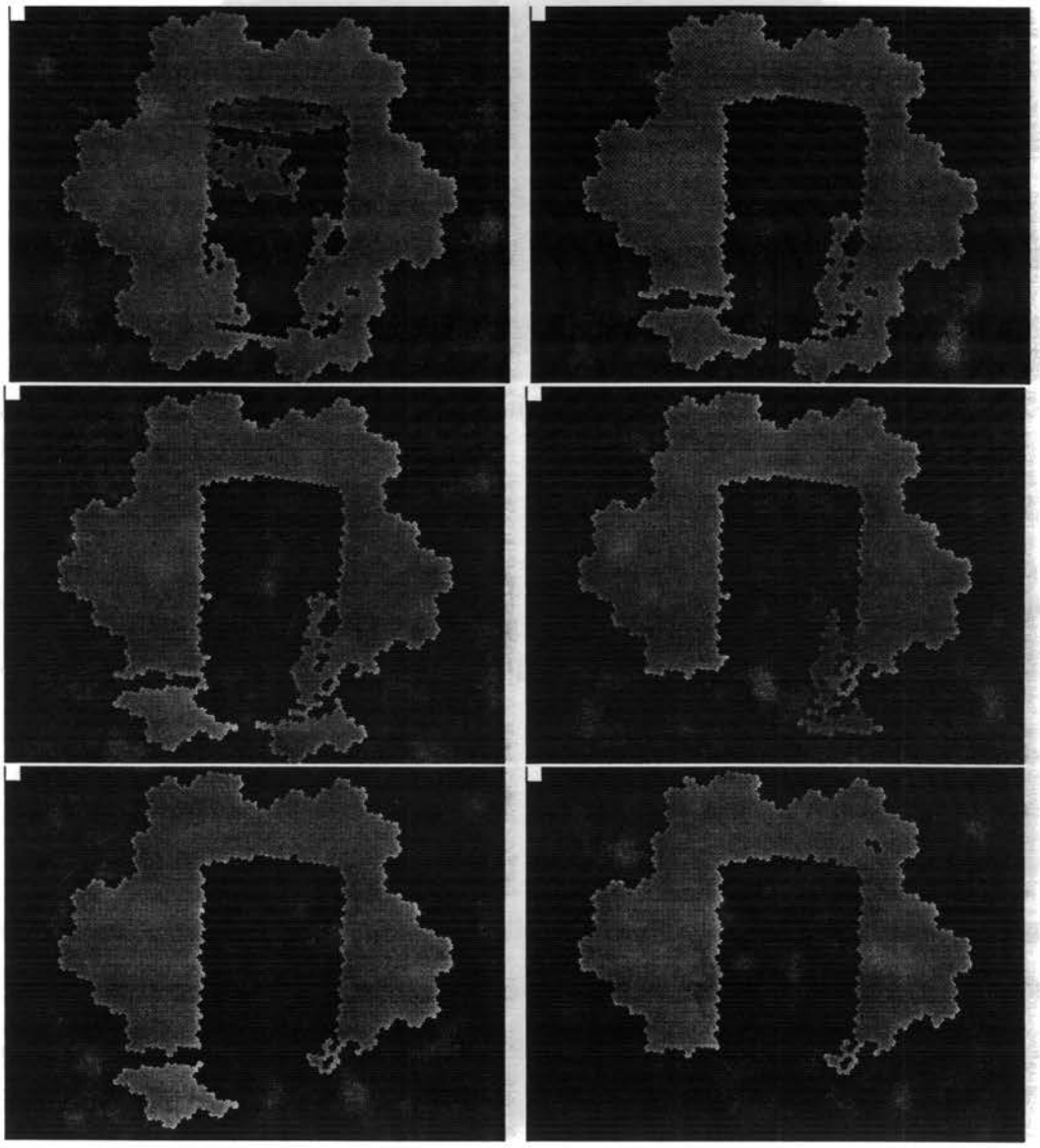


Figure 4.16: The collection of images represents six iterations of the segmentation algorithm applied to the image in Figure 4.15; (upper left) first iteration, (upper right) second iteration, (middle left) third iteration, (middle right) fourth iteration, (lower left) fifth iteration, (lower right) sixth iteration.

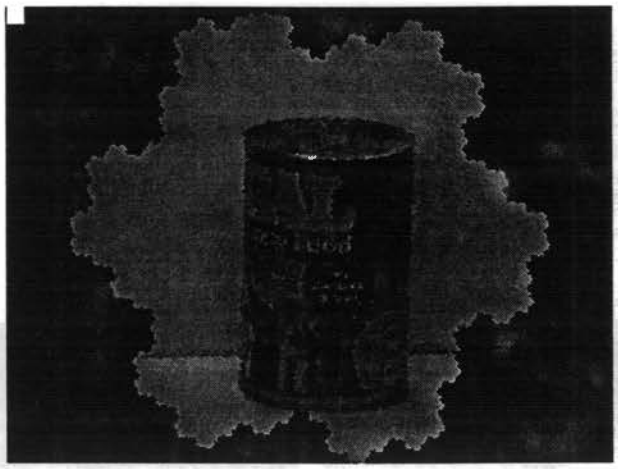


Figure 4.17: The image represents a scaling up of one of the 7 copies in Figure 4.12 (upper right).

letters, emerged at a higher computational cost. Thus, there is an inevitable trade-off between the density of detail and the computational cost of obtaining that detail.

4.7.3 Timing Results

The timing results of the segmentation of the PAL image are available for only the lowest three resolutions of the image. However, the results are of some interest. The first iteration at the lowest resolution mimicked the timing results for the trials presented for the transformations discussed in Chapter 3. The time increased linearly with the number of passes that were made at the lowest resolution. However, the time required for the two higher resolutions both increased by a factor of 10. The indication is that most of the time was spent in the actual passing of the messages between the machines. These results of the timing trials are only indications of how the algorithm will perform under arbitrary conditions at this stage. All of the problems of the implementation as discussed in Chapter 3 have remained in the 2DSA implementation. As such, these problems must be addressed before any store can be

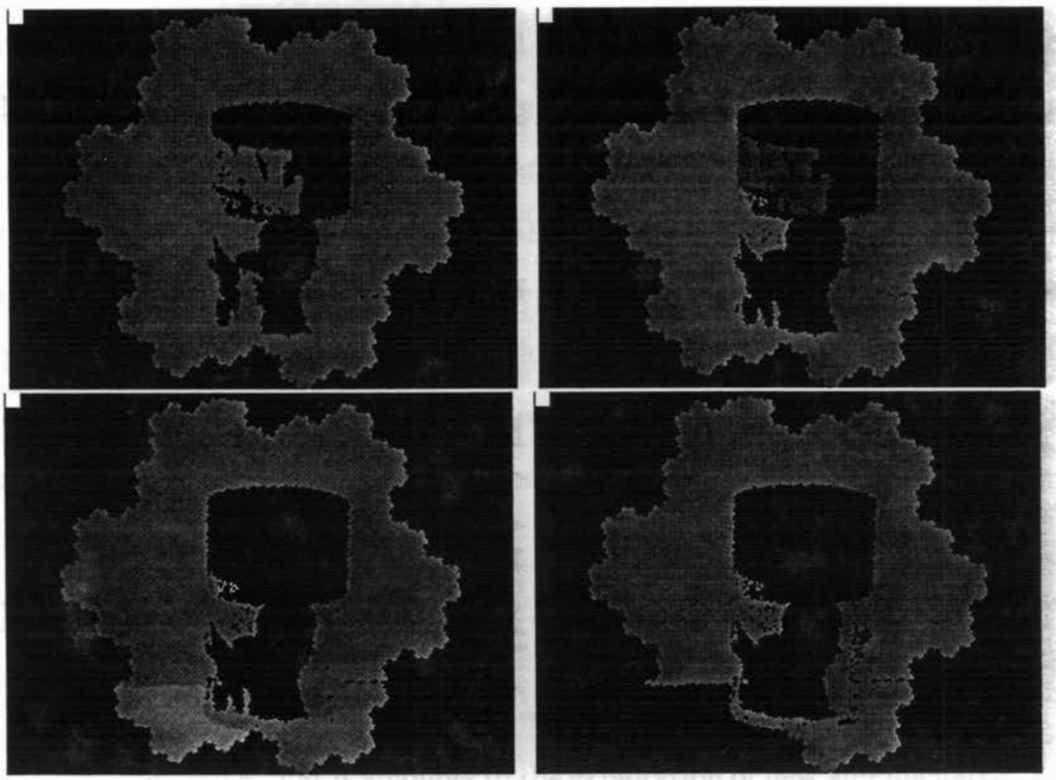


Figure 4.18: The collection of images represents four iterations of the segmentation algorithm as applied to the image of Figure 4.17; (upper left) first iteration, (upper right) second iteration, (lower left) third iteration, (lower right) fourth iteration.

placed on timing results for 2DSA.

4.8 Higher Level Segmentation Algorithms

2DSA is an application of the Gestalt law of *Similarity*[46]. It divides the raw image purely on the basis of light intensities. The output from this process results in the input image being represented at reduced computational complexity. This effect is similar to that achieved by the technique known as *Thresholding*, where the individual grey-level pixels are grouped after comparison with the threshold value. However, after this low level segmentation (either 2DSA or Thresholding) is complete, a higher level segmentation based on the shape of the conjugacy classes can serve to refine the process. This higher level segmentation might result from other *laws of visual perception.* Although none of the other Gestalt laws have been implemented to date, a few of these laws will be briefly discussed in Chapter 6 Section 6.1.3.

4.9 Discussion

One of the unavoidable problems associated with the creation of an autonomous segmentation algorithm is that it amounts to the exploration of new and uncharted areas. Although it is desirable to compare 2DSA with other segmentation algorithms, no benchmarks exist at present. Consequently, all analysis is purely subjective. However, four points regarding 2DSA can be stated. The first is that it operates autonomously, largely independent of the illumination conditions. Secondly, it indicates how the unique adjacency properties, see Figure 4.1, of clusters of hexagons provide computational power. Thirdly, it shows how SHLA can be employed in a parallel environment as a useful data structure for one of the low level image processing operations essential to machine vision. Fourthly, the choice of the δ function, Equation 4.1, appears to

produce adequate performance. It is probable that the function is not optimal and that further empirical investigations may lead to an even more effective δ function.

There is an implicit relation between 2DSA and the process of *edge detection.* A natural interpretation of the δ function is that it is a discrete derivative. That is, it measures the change in light intensity over a unit distance. Consideration of maximum δ grey-values then corresponds to the concept of higher order derivatives which are generally employed in conventional edge detecting techniques. The creation of each conjugacy class by 2DSA is associated with a contiguous sequence of hexagons at the border of the class. These hexagons are identified in the segmentation process when the fission process creates a disconnection between two adjacent hexagons. One might choose to call this collection of boundary hexagons (pixels) the *edge* of the conjugacy class. This edge detecting feature could be incorporated into 2DSA with little cost. Its inclusion would provide an enhancement to the reduction in complexity of the image's representation.

The ability to process multiple *near* copies of the image at the same resolution also has the potential to serve the higher level process of interpretation in at least two ways: Firstly, it can be used to centre an object within the image for subsequent foveal sensing. A second potentially powerful feature of processing multiple near copies at the same resolution, that has not been explored to date, is its effect on noise in an image. Natural images always contain noise. A conundrum in the interpretation process pertains to the removal of noise without *a priori* information. Pixels can only be classified as noise after the image has been interpreted. Before the process is complete any given pixel or small group of pixels may be of considerable significance. Consequently, the removal of arbitrary pixels from an image before the interpretation process is complete is fraught with danger. The transformations of SHLA that produce multiple near copies of the image all at the same resolution

provide an interesting approach to the problem. Each individual near copy represents a unique sample of the image and collectively they possess all the information of the input image. It is therefore likely that noise will be isolated to only a portion of the copies at lower resolutions. This property could serve the interpretation process by providing *noise reduced* copies without discarding information.

Although 2DSA does exploit parallelism at many levels - multiple resolution, multiple copies and that the independent construction of boundaries can merge into a single boundary without global information - the present implementation does have at least one global constraint. Before the splitting operations commence, global information is required to produce the maximal δ grey-value change. At the lower resolutions this is an insignificant search but as the resolution increases, the global search for the maximal δ will increase linearly with the amount of resolution. At this stage there is no indication as to the highest resolution that can be considered. However, there are a number of possibilities available to deal with this constraint. At the point where global high resolution becomes prohibitive, one approach would be to employ a windowing technique at high resolution on particular parts of interest in the scene.

Chapter 5

Perception of Time and Space

Kimo: "We can count and group patterns within a space, but, now we must try to count and group across space and time."

5.1 Introduction

The perception of motion in three-dimensional space is an essential feature of most biological vision systems. Without it, birds could not fly, flies could not avoid being swatted and monkeys could not swing in trees. It is no coincidence that the individual attributes of depth perception and motion perception are rarely found one without the other in biological systems. The attributes are highly interconnected which again demonstrates the totality of the vision process. The variety of biological strategies that have evolved to generate these attributes is immense. The one thing that all of the solutions have in common is that the analysis of two images of the scene is required because the photo receiving devices (eyes or cameras) capture the image from light projected onto a plane. The co-ordinates of a point projected onto a single plane do not correspond to a unique set of co-ordinates for the point represented in space.

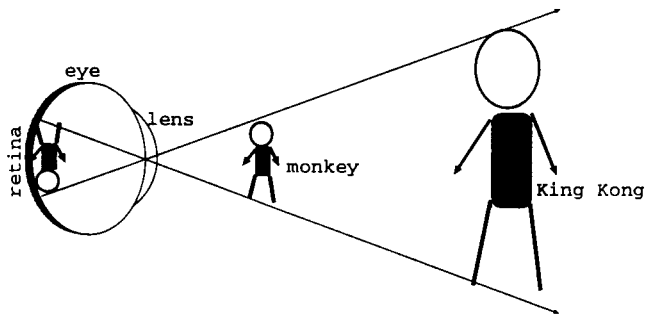


Figure 5.1: The projection of two images at different distances from the observer onto the retina.

To illustrate, consider the following two situations: viewing a monkey at close range and viewing King Kong at a greater distance. Assuming that the observer has no detailed information about primates, the probability that the observer will be able to distinguish the primates based solely on information derived from projections of

these images, using a single stationary eye or camera, is remote. See Figure 5.1.

The situation changes to one of easy distinction if any of the following variations on the single stationary camera approach is used.

i) Two distinct cameras capture simultaneous images;

ii) A single camera captures two images relating movement over time;

iii) A single camera captures two images from distinct points.

The first is known as binocular vision, the second is parallactic vision and the third is either binocular or parallactic depending on whether the two images were captured over space or time respectively.

The main results of this chapter will show how the SHLA, described in Chapter 2, and the segmentation algorithm, 2DSA, described in the previous chapter can be employed to gain part of the solution to the problem of real-time perception of motion in space. The two problems of perceiving depth and perceiving motion will be considered separately and then it will be shown how the two solutions integrate. In particular, this chapter is organized as follows: Section 2 provides an overview of some other related work. Section 3 presents an algorithm that produces a three-dimensional map from two images. Section 4 extends the results of the previous section to accommodate the dimension of time. A brief treatment of the way in which the algorithm from Section 3 could be implemented on a network of transputers is presented in Section 5. Section 6 considers two types of errors produced by the algorithm's method of measuring depth. The chapter concludes with a discussion of the results of the chapter.

5.2 Related Other Works

There have been a multitude of approaches that attempt to solve the problem of capturing three-dimensional information from two-dimensional images [49, 31, 1, 61, 48, 107, 99, 73, 71, 101, 64, 74] . Reviews of current techniques for three-dimensional image representation for model based systems can be found in [6, 75]. The authors treat both active systems, where the system emits an energy pulse and recaptures it for range analysis, and passive systems, which rely on the captured signal to be emitted from another source. Most of the passive systems reported seem to employ a noise reduction filter, such as a Gausian smoothing, to create the 3-D representation. The method of representation closest to the one presented in this chapter is the Octree structure [86]. However, the Octree is a hierarchical structure that employs smoothing to create the various levels of the structure. SHLA is a clustering structure.

A popular method for determining 3-D motion is that of the analysis of optical flow [78, 55]. A contour-based estimation of motion is presented in [25]. There is growing interest in the use of parallel algorithms for use in three-dimensional motion analysis. See [26, 82] as examples of this interest.

5.3 Perception of Depth

Most traditional methods of employing stereoscopic vision to gain the perception of depth generally amount to the resolution of triangles. Consider the triangle formed by a point P in space and two points E_l and E_r each at the centre of a distinct left and right eye (or camera). The problem is then to determine the distance, Z, from P to the base line joining E_l and E_r. See Figure 5.2.

Information about the focal distance of each eye and the distance separating the two eyes together with the angle of displacement of the projection of P from the

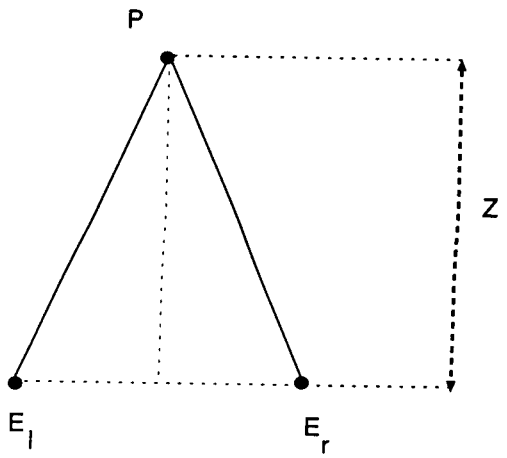


Figure 5.2: A point P at a distance Z from the base line joining points E_l and E_r.

optical axis are the inputs required to compute Z. The computation then results from the consideration of some elementary trigonometric relations. See Figure 5.3.

There is an assumption that the two eyes are aligned on the horizontal axis. The accuracy in the computation of Z is dependent on the precision of the measurements of their inputs. The time required to determine Z is mainly determined by the ability to obtain the two pairs of coordinates on each of the projections associated with P. This is the difficult part of the problem for a variety of reasons, not the least of which is that there is never a perfect correlation between points in the two projections. In fact, the richer the variation in depth across the scene, the poorer the correlation. There have been a number of approaches taken to solve the correlation problem. These range from sophisticated statistical techniques to feature detection methods.

The method of depth perception presented in this chapter is also a variation on the general technique of triangulation. It has been named 3DSA (Three Dimensional Segmentation Algorithm). It has a number of distinctive features:

- 3DSA executes concurrently with 2DSA. Output is generated from 2DSA on

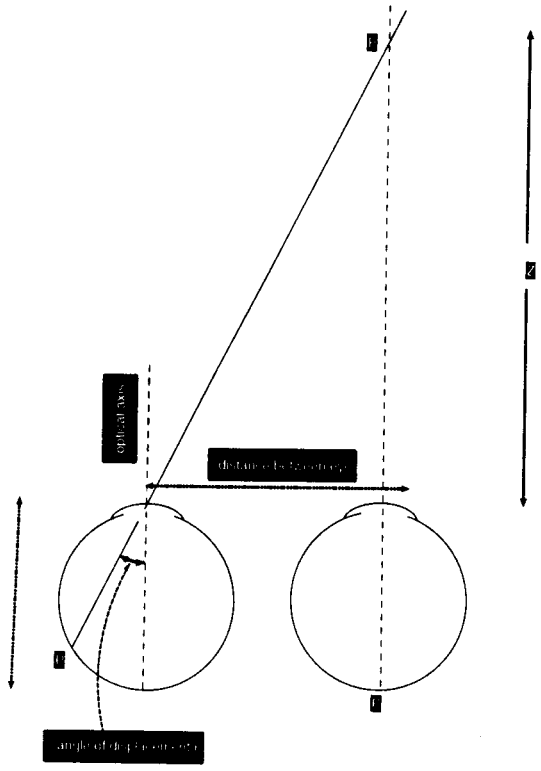


Figure 5.3: The geometry of stereo vision.

completion of its consideration of change in light intensities and then is input for a similar stage of 3DSA.

- The relative distance of various points on a surface to the observer is determined. That is, the depth information inferred will be that a given point on a surface in space is, relative to the observer, either closer to or further away than each of six neighbouring points. The determination of the absolute distance from any surface to the observer will be the jurisdiction of a higher level process.

- Although it is assumed that the two cameras are in horizontal alignment there is no requirement for the measurements of the focal length or the base line distance. Rather, it is assumed that the focal length and the base line distance are both constants for the duration of time it takes to capture the two images.

- The process generates depth representations concurrently for each of the resolutions of the image. The accuracy of the depth information is a function of the degree of resolution. The higher the resolution, the greater the accuracy.

- The process correlates patches in the two images creating a cyclopedean view of the visual world by attempting to match equivalence classes produced by the segmentation algorithm (2DSA). The non-matching equivalence classes also generate depth information.

5.3.1 A 3-D View of SHLA

The SHM introduced in Chapter 2 was inspired by the geometrical arrangement of the cones on the retina of primates. As a result, there may be a tacit assumption that the SHM is really just a two-dimensional structure but nothing could be further from reality. Just as the plane, as defined by the surface of the retina, is mapped into the multidimensional structure within the brain, the pixels as defined by the

camera, are mapped into SHM. Recall that the mathematical structure of SHM is a Euclidean ring (dimensionless). All spatial references to SHM have purely been matters of interpretation. It is just as easy to interpret SHM as a representation of a three-dimensional space.

To illustrate this, let SHM contain $7^2 = 49$ hexagons and let it represent the visual world in the following manner. Relative to and centred at an observation point, say E_l construct seven concentric spheres each of which is defined by a distinct radius. Let each of the seven clusters lie on the surface of a corresponding sphere such that the cluster with centre at address 0 is on the sphere of greatest radius, each of the clusters centred at 10, 20, 30, 40, 50, 60 lie on the surface of the next closer sphere, respectively. Then let the alignment of the clusters be such that the centre of each lies on the straight line segment joining E_l to address 0 (the optical axis). Let the area of each hexagon be proportional to its distance from E_l. Each cluster then represents a section of a distinct sphere and the greater the distance of the cluster from E_l the larger is the area of its component hexagons. See Figure 5.4. As a result of this geometry, each address has associated with it two numbers. One identifies the sphere and the other determines the direction from the centre of the cluster on the sphere. In the case of the present simple example where the addresses range from 0 to 66, the ten's component of a given address indicates the sphere it lies on and the one's component determines its relative position to the other hexagons on that sphere.

The mathematical dissection of the address is achieved as follows:

$$\begin{aligned} sphere &= (address\,/10) \\ direction &= (address\,mod\,10) \end{aligned}$$

Thus, SHM may be considered as a representation of a bounded visual world and an individual address may be interpreted as the centre of a hexahedron in three-dimensional space. See Figure 5.5.

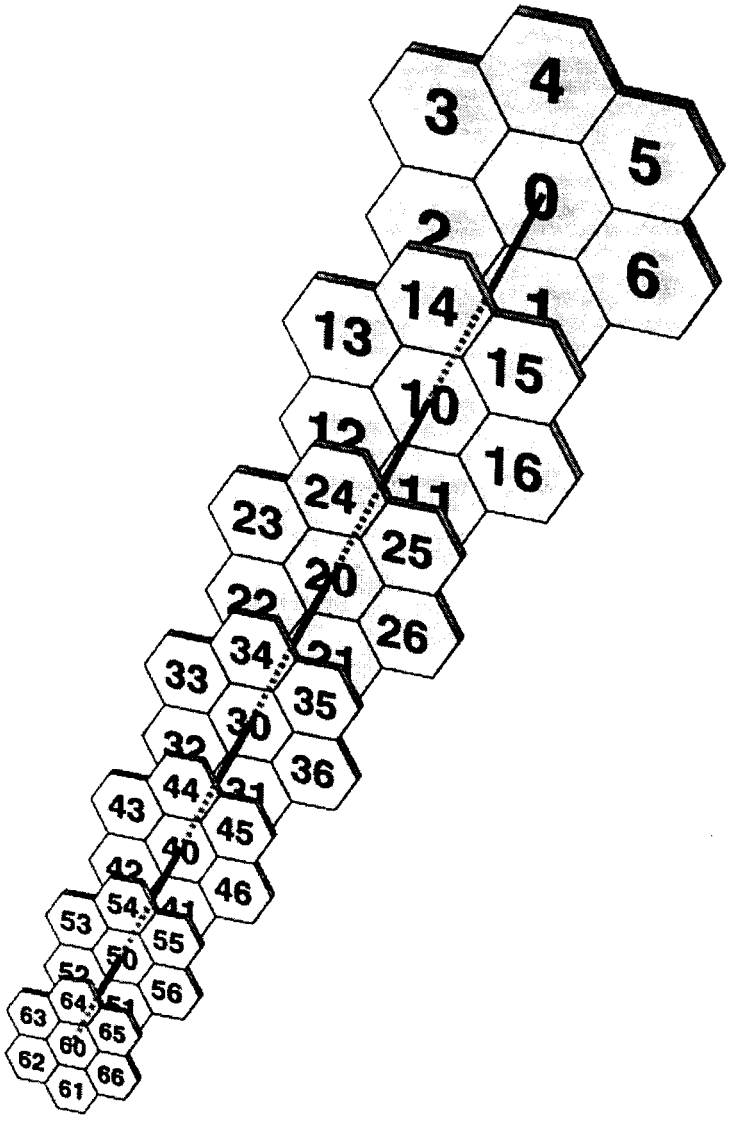


Figure 5.4: The figure indicates the relationship between the positions of seven clusters of the SHM to the seven units of 3-D space that they represent. The optical axis intersects each of the seven clusters.

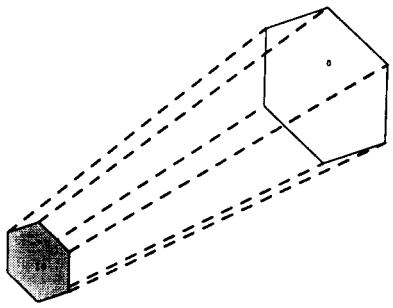


Figure 5.5: The optical axis intersects the centre of each hexagonal end of a hexahedron which is a component of the visual world.

One additional adjustment to the interpretation imposed on SHM in Chapter 2 is required to give SHM meaning as a representation of the visual world. This results from the fact that light does not emanate from all points in three-dimensional space. Rather, light emanates from the surfaces of objects within that space. The point is that not all coordinates in space are occupied by objects. The implications for SHM is that not all of its hexagons will have a light intensity associated with it. In other words, some of the hexagons will be transparent. This concept is, possibly, better explored by Gibson's [34] ecological terminology. He chooses to replace the geometrical constructs of *planes* and *spaces* with the ecologically equivalent notions of *surfaces* and *mediums*. Gibson's ecological terminology enhances the intuitive notion of the vision process. For example, his idea of *surfaces moving through a medium* gives a clear distinction between what it is that light passes through and what it is that light emanates from. The geometrical notion of space is a generalization of a plane and this tends to blur the distinction between planes and spaces in the visual context.

It is also necessary to re-interpret the physical significance of the elements of SHLA (transformations on SHM). Previously, T_x represented a transformation on a

two-dimensional space. Now, in order to achieve a transformation in three dimensions, the following enhancement of T_x is adopted:

Let $T_x^s = T_x$ applied exclusively to the sphere component of the addresses.

$T_x^d = T_x$ applied exclusively to the direction component of the addresses.

The short hand form of the combined components is:

$T_{x,y} = (T_x^s + T_y^d)$ where the first and second component of the (x, y) pair is the scalar value which is applied to the sphere and direction component of the address, respectively. A few examples will serve to illustrate the physical significance of the transformations:

$A_{0,x}$ produces the effect of translation within spheres, by an amount x. That is, none of the hexagons change spheres, due to the zero value of the first component.

$A_{x,0}$ produces the effect of translation across the spheres. That is, in depth with no change of horizontal and vertical position, due to the zero value of the second component.

$A_{x,y}$ produces the effect of an arbitrary translation through space.

$M_{1,x}$ produces the effect of rotation and scaling in the horizontal vertical plane.

$M_{x,1}$ produces the effect of rotation and scaling in depth.

$M_{x,y}$ produces the effect of an arbitrary rotation and scaling in space.

It is also worth noting that the term *scaling* has two different interpretations, depending on whether it takes place in the horizontal/vertical plane or in depth. The former is that which has already been observed in Chapter 2. The effect is one of viewing the image at different resolutions. The latter effect is one of *receding* or *advancing*. Note that translation in depth does not achieve this effect. For an image to

have the appearance of receding or advancing, it must have an accompanying decrease or increase in size. Translation does not produce a change in size, only position.

Before considering a few specific examples of the transformations, one additional observation should be made in order for the transformations to make sense. The transformation of a transparent hexagon need not be considered. This makes sense both visually and computationally. The former results from the observation that a transformation in space is principally concerned with what has moved into a component of space, not what has left it. The latter results from the simple fact that there is no point in trying to move *nothing* (a transparent hexagon).

With these conventions in mind, consider $A_{1,0}$ from two points of view. Firstly, from the viewpoint of the effect on the shift of the individual spheres:

Cluster in sphere 0 would move to sphere 1 with no directional shift:

$(0,1,2,3,4,5,6)A_{1,0} = (10,11,12,13,14,15,16)$.

Cluster in sphere 1 would move to sphere 3:

$(10,11,12,13,14,15,16)A_{1,0} = (30,31,32,33,34,35,36)$.

Cluster in sphere 2 would move to sphere 5:

$(20,21,22,23,24,25,26)A_{1,0} = (50,51,52,53,54,55,56)$.

Cluster in sphere 3 would move to sphere 2:

$(30,31,32,33,34,35,36)A_{1,0} = (20,21,22,23,24,25,26)$.

Cluster in sphere 4 would move to sphere 0:

$(40,41,42,43,44,45,46)A_{1,0} = (0,1,2,3,4,5,6)$.

Cluster in sphere 5 would move to sphere 6:

$(50,51,52,53,54,55,56)A_{1,0} = (60,61,62,63,64,65,66)$.

Cluster in sphere 6 would move to sphere 4:

$(60,61,62,63,64,65,66)A_{1,0} = (40,41,42,43,44,45,46)$.

Secondly, from the viewpoint of the effect on the image contained within the space, there are a number of scenarios to consider:

- If the image (non-transparent hexagons) was exclusively confined to spheres 0 and 5, the effect would be one of translating the image one sphere closer to E_l.
- If the image was contained in spheres 1 and 6, the shift would be to spheres 3 and 4 respectively. The visual effect is one of a relative shift of surfaces within the image.
- In general, the effect of the transformation is dependent on the image itself. This is an important point as it is related to the idea of an object being defined by the transformation which renders it invariant under the action of the transformation.

In a similar manner, the transformations of $M_{x,y}$ represent rotations and scalings in a three-dimensional space. In particular, $M_{1,x}$ for x = 2, 3, 4, 5, 6 represents rotations in the horizontal/vertical plane of 60, 120, 180, 240, 300 degrees respectively.

Rotations about the north/south axis of an image is achieved by $M_{x,1}$. The visual effect of this transformation has an interesting interpretation. Observe that the cluster in sphere 0 is not moved. Whereas the clusters in spheres 1 to 6 are cycled as indicated by the diagram in Figure 5.6.

If sphere 0 is considered to be at *visual infinity, the inference is that an image can't be rotated either in, from or out to infinity. This makes* visual sense.

5.3.2 Segmenting the 3-D SHM

In the previous chapter, considerable attention was devoted to the development of a two-dimensional segmentation algorithm, called 2DSA. Because vision is predominantly three-dimensional, direct applications of the algorithm to natural images are

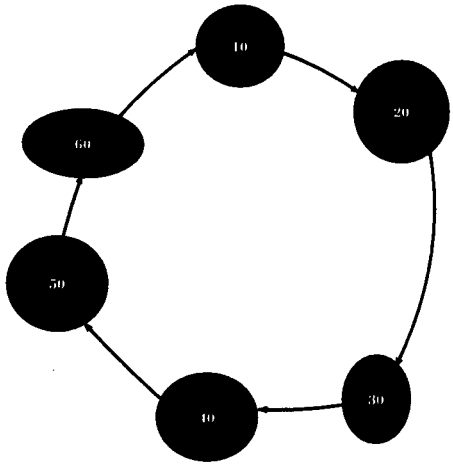


Figure 5.6: The figure indicates the relative movements of the seven clusters from Figure 5.4 when a transformation of rotation is applied to the SHM.

not abundant. The only real examples of two dimensional images in nature occur when the scene is far enough from the viewer that all points of the image appear to be at the same distance (visual infinity). For example, even though the night sky represents a richly three-dimensional space, it appears as though it is two-dimensional. This is due to the fact that the ratio of the distance from any extra-terrestrial object to an Earthly viewer and the distance between the viewer's two eyes is immense. In mathematical terms, this means that the value of Z of Figure 5.2 approaches infinity. Geometrically, the triangle defined by the points (P, E_l, E_r) collapses into a straight line segment. In spite of 2DSA's apparent limited applicability, it will be the lynch pin of the three-dimensional segmentation algorithm, (3DSA), described here.

The algorithm executes concurrently at each of the resolutions. The following three steps are repeated on inputs from each of the refinements of the 2DSA:

step 1 : The two images of the same scene captured from distinct points are segmented using 2DSA for a given change of light intensity.

step 2 : All new equivalence classes are initialized to sphere zero (furthest away from the observer).

step 3 : Individual equivalence classes from each image are translated forward (closer to the observer) so as to maximise matching of the similar equivalence classes.

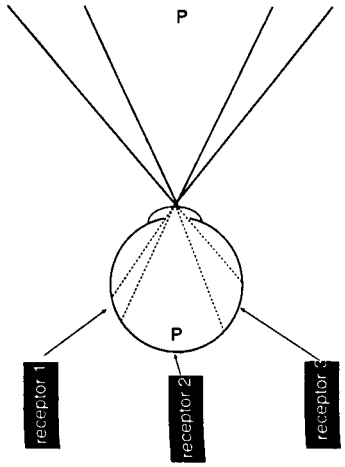


Figure 5.7: Horizontal cross section of a simplified left eye viewing a patch P in the visual world. The three photo receptors are labelled 1, 2 and 3.

The third step requires a few points of further explanation. There is an assumption of *non-accidental equivalence classes.* That is, two equivalence classes from the two images, that are of similar shape and lie on the same horizontal, tend to represent the same object from the scene. The process of correlating equivalence classes from distinct images is, in this context, called *matching.* Given this assumption, suppose further that two equivalence classes, one from each of the images that correspond in shape and vertical position, are found to *match.* A horizontal translation of the equivalence class from E_r sufficient to have its horizontal positions coincide with its counterpart in E_l provides the depth information. In order to appreciate the significance of the *horizontal translation*, consider Figure 5.7, 5.8 and 5.9. Figure 5.7

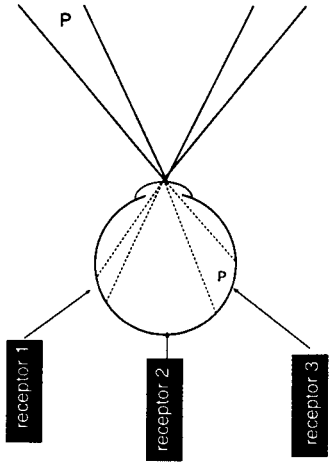


Figure 5.8: Cross section of the right eye's view of the same patch P displayed in Figure 5.7.

represents a horizontal cross section of the visual world, as captured by an absurdly simplified photo receiving device. The device is composed of a left and right component, each of which has three photo receiving elements aligned on the same horizontal plane. Suppose that a patch P is observed in the photo receiving element 2 of the left component. Note that this positioning only *locks* the patch into a corridor in the horizontal plane. The patch could be located anywhere in the corridor between the lens and infinity. Next, suppose that another patch is observed in photo receiving element 3 of the right component and that evidence suggests that the two observed patches are one and the same (they match). The patch is fixed in the intersection of the two corridors, as seen in Figure 5.9. Although the method for fixing a *patch* in space was described in the context of a simplified photo receiving device, the same argument applies for a photo receiving device with any number of elements. The only alteration being that a greater density of elements will generate more accurate depth information. The degree of error can, of course, be quantified in precise mathematical terms.

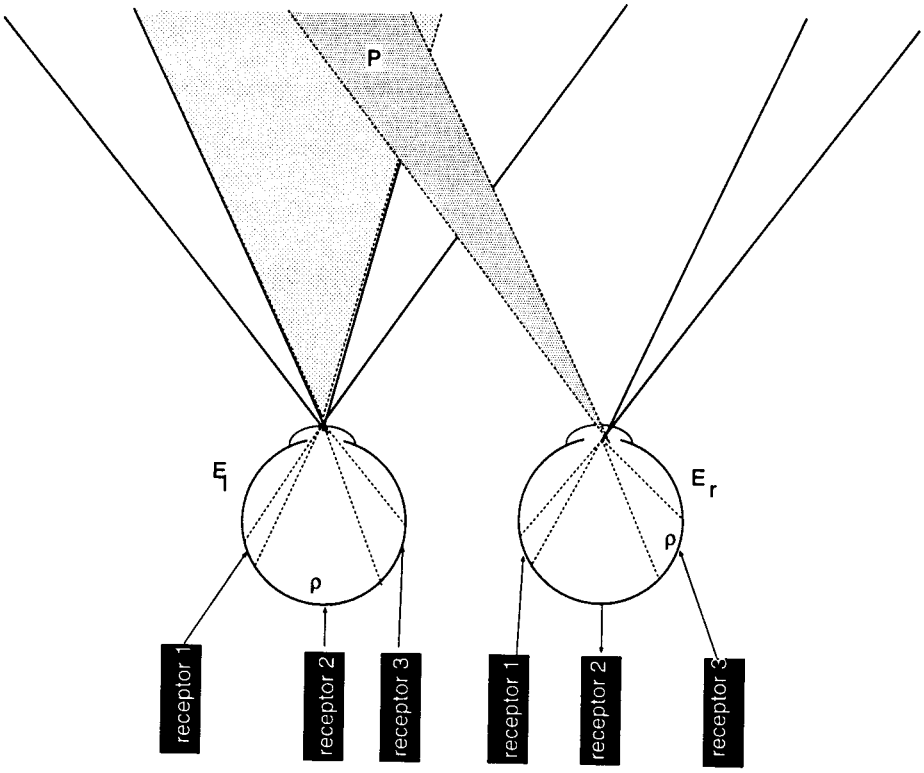


Figure 5.9: The combined effects of Figure 5.7 and 5.8 shows the patch locked in the intersection of two corridors.

The second issue to be resolved pertains to the generalization of the method of *connecting* and *disconnecting* hexagons in the cases of higher than two dimensions. The concept referred to was originally introduced in Chapter 4 with the definitions of section 3. In the higher dimensional SHM, the equivalence classes generated by 2DSA may exist in distinct spheres.

There is a problem to overcome. Light intensities, as a measure of the amount of light being reflected by an object in space, diminish exponentially as the object is moved away from the observer (the statement presupposes an assumption of constancy of other variables that affect light intensity). Alternatively, if light from a uniform source falls on an object with a surface of uniform, the observed light intensities from that object will diminish exponentially across the surface as the object is tilted away from the viewer. See Figure 5.10.

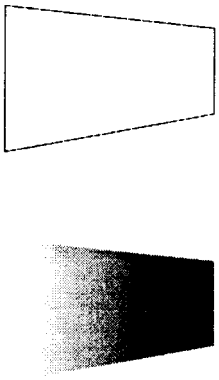


Figure 5.10: (upper) Trapezoid in a plane which is perpendicular to the optical axis of the observer. (lower) Same trapezoidal shape as in (upper) but with shading, giving the illusion of a rectangle in space tilted away from the perpendicular to the optical axis.

In order to adjust for the variation of light intensity over depth, the 2DSA is applied to a logarithmic transformation on the light intensities of hexagons in distinct spheres. This will eventuate in a possible adjustment of the labelling of the

equivalence classes.

There are, of course, other transformations on the equivalence classes that effect the perception of depth. Many of these have been described by the Gestaltists. One of these is the effect of texture on the perception of depth. This would appear to be a higher level function as it presupposes the recognition of a recurring pattern. Although it seems likely that this effect could be implemented as another concurrent module with 3DSA, it will not be treated here.

5.4 Perception of Motion

The perception of motion is dependent upon not only a representation of space but also of time. This section of the chapter will extend SHLA to incorporate time and will explore how SHLA can be employed to facilitate the perception of motion.

5.4.1 A Representation of Time

The concept of time is represented and imagined in varying ways by different cultures. In a typical western culture, time is represented on a linear scale with the present placed at the origin and extending in negative and positive directions. The negative direction represents a backward movement in time and the positive direction represents a forward movement in time. In Papua New Guinea culture,[1] time is perceived as spiralling into the future from the present. The present is placed at the centre and contains the past. For the purpose of this thesis, the Papua New Guinean representation of time shall be used.

[1]The spiral representation of time was explained to the author by a Papua New Guinean. The explanation was given in the context of providing insight as to why his son, an electrical engineering student, may be having difficulty with the Fourier Series. The son, shortly thereafter, mastered the Fourier Series. The author took much non-linear time to appreciate the explanation given him that day.

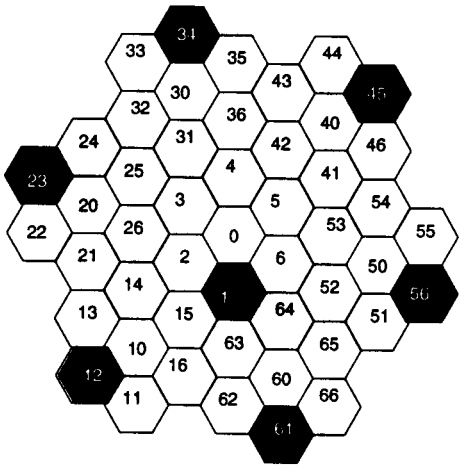
Spiral time, as it shall be called, is enumerated by Spiral Counting in a particular key. The key is dependent upon what it is that will be associated with the given moment in time. Consider a trivial example: if the units of time to be enumerated are to have nothing associated with them, the appropriate key is, one. A non trivial example is provided when each unit of time has associated with it, a unit of space. In the previous section, a simple example of 7^2 hexagons was used to represent a three dimensional space. In this case, Spiral Counting in the key of 100 would enumerate time and its associated units of space.

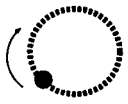
5.4.2 Transformations of Time and Space

The SHM can now be thought of as a four dimensional structure that has the potential to integrate time and 3-D space. It is the transformations of SHLA that realize the potential. Consideration of the following simple example will illustrate the point. Once again, let the space be represented by the smallest unit of visual organization: a cluster that will represent a surface in the horizontal/vertical plane. Let a darkened hexagon represent a patch against the white background of the other six hexagons. Thus, the patch in the context of a cluster has a position in space. Figure 5.11 represents such a space over seven units of time.

The position of the patch is indicated by its relative position to each of the addresses (0, 10, 20, 30, 40, 50, 60), which dually represents a moment in time and the centre of the visual field at that moment. Figure 5.11 can then be interpreted as a pattern of movement in space. The motion in each of the three figures is respectively, 5.11 along a path of a circle, 5.12, along a path of a straight vertical line, 5.13 a stationary object.

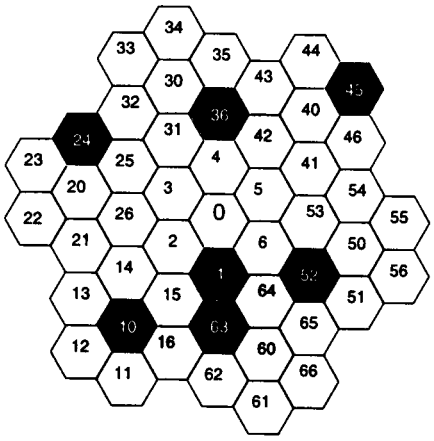
The qualitative description of the transformations of time and space, thus far given, can be rendered in mathematical terms with a slight extension to the notation.

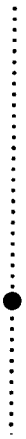




motion of object along the path of a circle

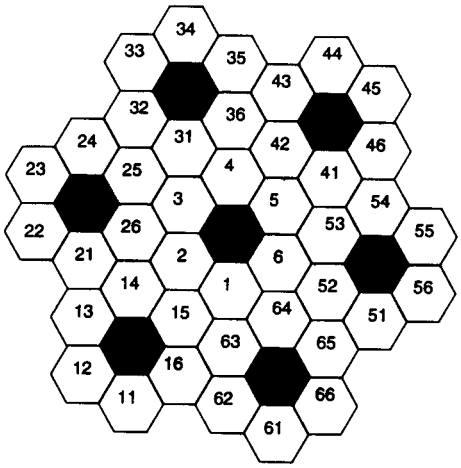
Figure 5.11: The figure indicates (right) the motion of an object along a circular path and (left) the representation of the position of the object in SHM at seven units of time.





object moving in a vertical path

Figure 5.12: The figure indicates (right) the motion of an object along the path of a vertical line and (left) the representation in SHM of the object's position at seven units of time.



stationary object

Figure 5.13: The figure indicates (right) a stationary object and (left) the representation in SHM of the stationary object over seven units of time.

Firstly, the general formulae for the extraction of the components of *direction*, *sphere* and *time* from the address are:

$$\begin{aligned} direction &= (address \bmod units - of - direction) \\ sphere &= ((address \bmod (units - of - space)) \,/\, (units - of - direction) \\ time &= ((address \,/\, (units - of - space)) \end{aligned}$$

Let T_x^t be the transformation T_x applied exclusively to the *time* component of the address. Thus, the previously defined short hand notation may be extended as follows: $T_{t,s,d}$ is the previously described $T_{s,d}$ with the insertion of the component for *time*. Thus, the transformation $T_{t,s,d} \forall t,\ s,\ d \in SHM$ represents the visual field in a subspace of SHM restricted by (s, d) at a particular moment, t, in time.

With this notation in hand, it is now possible to give a precise definition to the concept of *perception of motion* in SHM. Motion is determined by a sequence of transformations $T_{t_1,s_1,d_1} T_{t_2,s_2,d_2} \ldots T_{t_n,s_n,d_n}$ applied to SHM.

For example, the circular motion that was attributed to the representation of SHM as displayed in Figure 5.11 may be represented by a sequence of transformations. To illustrate this process, first consider an object in the centre of the visual field. This is represented most simply by a darkened hexagon, the object, in the centre of the fundamental unit of vision. See Figure 5.14. Secondly, apply the following sequence of transformations to this figure: $A_{1,0,0}M_{2,1,2}M_{2,1,2}M_{2,1,2}M_{2,1,2}M_{2,1,2}M_{2,1,2}$. This sequence is equivalent to the pair of transformations $A_{1,0,0}M_{6,1,6}$. In words, the object was translated from the centre of the visual field and then rotated in a circular path about the initial position over seven units of time.

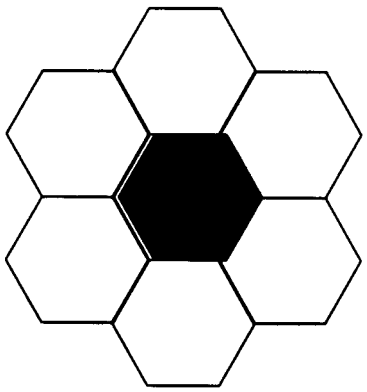


Figure 5.14: The figure shows an object (darkened hexagon) in the centre of the visual field.

5.4.3 Perception of Depth from Parallaxis

Parallaxis is the perception of depth derived from the analysis of relative positions over time. This information can be obtained from a single eye receiving images over time. The incorporation of time into the SHM facilitates discussion of parallactic vision. In practice, it turns out that the perception of depth from the parallactic effect is only a slight variation on the binocular system.

In binocular vision, the *cyclopedean* view of the visual world is produced by the merging of two distinct images from the same point in time. The point to note is that 3DSA merely merges distinct images. There was no requirement for the two images to be captured at the same moment. The only requirement from the processing point of view is that the time interval between successive images is short enough so that the *non-accidental assumption* is valid. So long as this assumption is valid, the 3DSA will be able to extract depth information. The relevant constraints on the process pertain to the *smoothness* of the movement. In particular, the greater the constancy in direction and velocity, the greater is the accuracy of the depth information. In general, it would be expected that the perception of depth from parallactic vision would be a simpler matter to implement as the binocular system requires additional synchronization. That is, the system must be able to keep track of the two streams of images.

5.5 Implementation of 3DSA

The implementation of 3DSA on a network of transputers is still underway. Consequently, no test results can be reported. However, a few words are in order to describe the main issues that are involved in the implementation. The most important point about the algorithm is that it is part of a larger module which concurrently executes with, and is similar to, the 2DSA. As might be anticipated, the problems associated with its implementation are similar to those of 2DSA. Indeed, the two main obstacles to be overcome relate to memory and synchronization.

The biggest problem encountered with the implementation of 2DSA, as described in the previous chapter, was the memory constraint imposed by each transputer. It was stated that, to date, the largest SHM to meet the memory constraints is $7^3 = 343$ pixels. This precluded the production of even the simplest non-trivial results possible.

The smallest non-trivial representation of three-space requires $7^3 = 7 \times 7^2$ hexagons in the SHM. The inclusion of the dimension of time to the three-dimensional space, would require a further power of seven hexagons to render the representation non-trivial. Clearly, a way around the memory constraint must be found in order to handle realistic images over the dimension of time.

The biggest problem that was overcome in the implementation of 2DSA related to the synchronization of the multitude of concurrent processes required for the algorithm. The 3DSA will have a similar synchronization problem to solve. This is in addition to a layer of synchronization between the dynamic interplay between 2DSA and 3DSA.

In view of the fact that the programming required for the implementation will take considerable time to achieve, an alternative approach might have been to implement the algorithm on a sequential machine. This would avoid both the problems of memory and synchronization. Such an approach might have produced demonstrable results from the algorithm. However, the approach was not taken for two reasons. Firstly, and most importantly, it is felt that the fundamental importance of such an algorithm is its scalability on a parallel architecture. This belief has been persistent throughout this work and stems from the conviction that the vision process is inherently a parallel one. The second reason for not pursuing a sequential implementation is of a more pragmatic nature. As it is intended that this work will ultimately result in the engineering of a machine vision system, (to be discussed in the next chapter), it is felt that any time spent on a sequential trial is a waste of effort.

5.6 Error and Illusions

Any vision system, be it biological or machine, has different types of error unavoidably engineered into it. In the case of the human vision system, one rather interesting

form of error is called *optical illusion.* This section considers two types of error in the perception of depth that would be generated from the output of 3DSA.

5.6.1 Error from Spatial Mismeasurement

The first example is drawn from the illusion experienced by a human binocularly viewing a tall, straight column. The illusion is one of *seeing* a curvature in the column[2]. If 3DSA was employed to handle input from a binocular system viewing the *tall, straight column,* a similar error in the perception of curvature would be predicted.

The explanation for the error in depth created by 3DSA is rather simple, if not expected. Error is incurred from *perfect matching* of equivalence classes. That is, the matching of equivalence classes that do not represent the same patch of a surface. For example, consider the equivalence classes generated by the binocular view of the cylinder, with uniform surface, depicted in Figure 5.15. The equivalence classes produced from the projection of the can onto the retina of the left and right eyes, have the shape of a rectangle (perfect match). However, each rectangle represents different patches of the cylinder's surface. Hence, the error creeps into the process by matching two equivalence classes that, in reality, match only in part. That is, the left vertical edge of the class from the left eye is occluded from the right eye and visa versa. This false positive match translates into error in depth representation from the algorithm by placing the two equivalence classes in the wrong corridor as discussed in section 5.3.2. More specifically, it is placed further from the viewer than it really is. The amount of error in depth is a function of the distance of the cylinder from the viewer. At visual infinity, there is no error; no part of the surface visible to one eye is occluded from the other eye. The closer the cylinder is to the viewer, the greater

[2]The ancient Greeks understood this phenomenon and compensated for this effect by placing a curvature in the opposite direction to make the column appear straight.

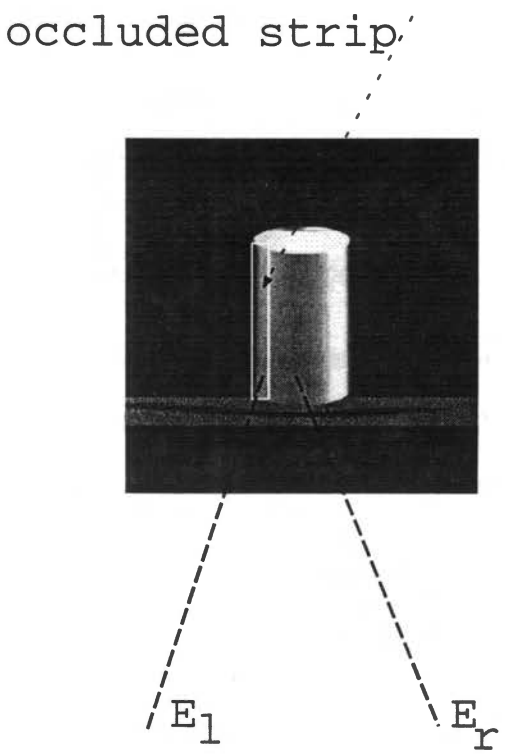


Figure 5.15: The curvature of the can produces an occluded strip from E_r which is visible to E_l.

is the degree of occlusion; accordingly the greater is the error in its positioning in depth.

Now in order to appreciate how the error in placing the depth of a cylinder is related to perceived curvature in a column, observe that the column is just a long cylinder. In addition, distinct vertical strips of the column are at different distances from the observer. The bottom end of the column is closer to the observer than is the top end. Consequently, there is greater error in the representation of depth at the bottom end of the cylinder than for the top end. This gradient of error in depth over the length of the column will translate into the perception of curvature.

5.6.2 Error from Time Mismeasurement

Another interesting example of *illusion of curvature* is provided by the well known phenomenon of perceiving the motion of a pendulum to be along the path of an ellipsoid when in reality the pendulum is swinging in a plane. This illusion of 3-D movement is experienced from binocular viewing with one eye receiving the light through a filter such as sunglasses while the other eye is receiving the light unimpeded. The viewing must be conducted with the head absolutely still and the background to the pendulum must be uniform. Otherwise, the vision processes will intervene and destroy the illusion. See Figure 5.16. There are a number of aspects to this illusion. One results from the fact that the speed at which the signal travels from the retina to the visual cortex is a function of the intensity of the light source that produced the signal. The effect of having each eye receiving different light intensities from the same scene, is that, by the time the images arrive at the visual cortex, they are out of synchronization.

Now, suppose that a binocular machine vision system is viewing the swinging of

a pendulum and that, for some reason, the two incoming images are out of synchronization. Suppose further that the incoming image from the right input arrives for processing by 3DSA, one unit of time later than the left input.

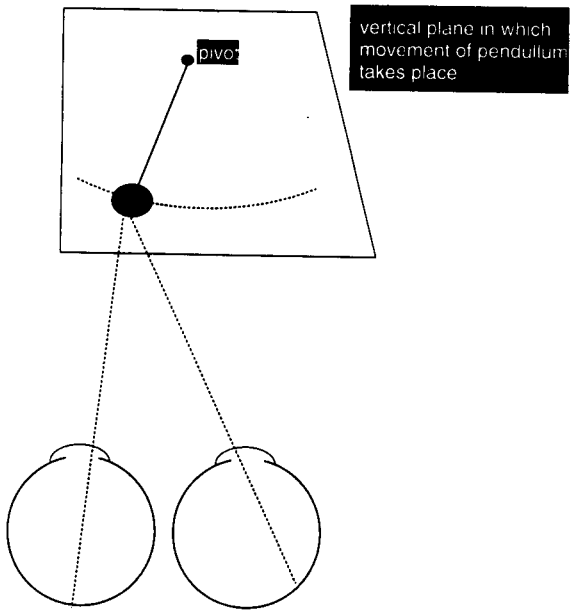


Figure 5.16: Binocular view of swinging pendulum with both eyes receiving unfiltered light.

Given this situation, the predicted error from 3DSA will produce an interesting effect. The prediction follows a similar argument to that of the one given for the illusion of curvature in the straight column, with the added component of time. As the pendulum is moving, it will be at two distinct positions at time t_{i-1} and t_i. However, a breakdown in the *non-accidental* assumption causes the two equivalence classes, which are of the same shape, to be matched. The position, in depth, is at that point of intersection of the corridors from the left and right cameras. The error incurred is proportional to the distance the pendulum has moved over the unit of time. See Figure 5.18.

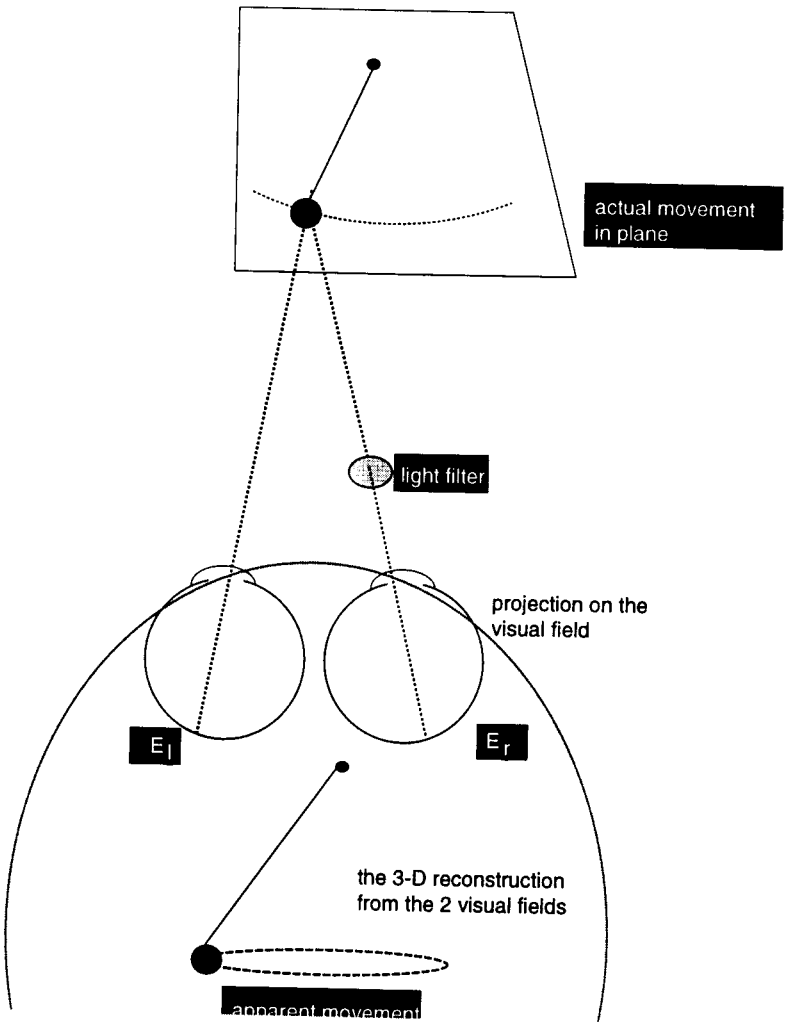


Figure 5.17: One eye, E_r, receives filtered light. The perceived motion of the pendulum is along the path of an ellipsoid.

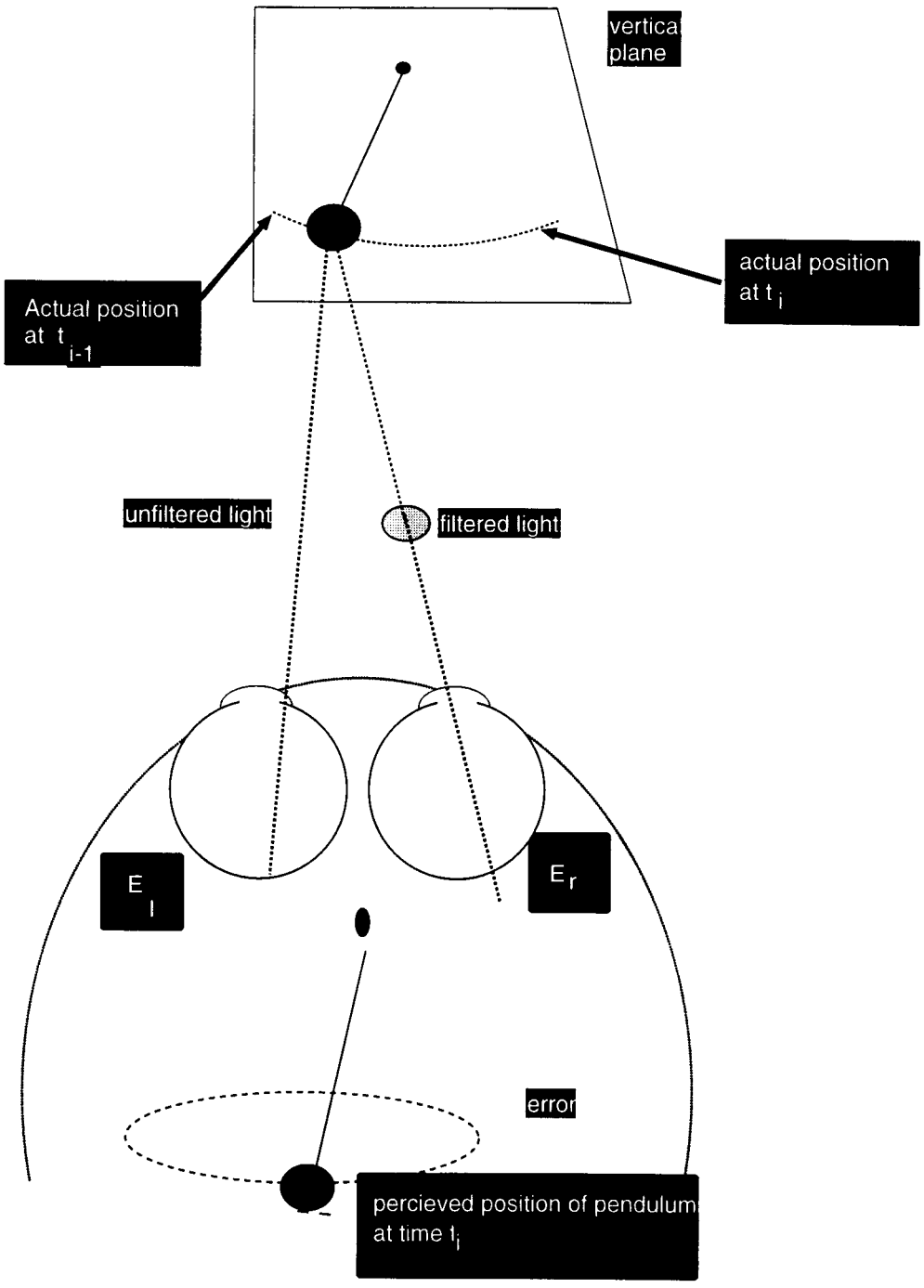


Figure 5.18: Showing the actual position of a swinging pendulum at two points in time and the perceived position of a single point in time.

If the pendulum was moving at a constant velocity, the error produced by 3DSA would also be constant. However, the speed of the pendulum is not constant. Rather, it has zero velocity at both points where the pendulum changes direction and has maximum velocity at midswing when it is in alignment with the gravitational field. The result is that, from the left there is no error, at the mid swing there is maximum error and the pendulum will appear to be moving towards the vision system. As the pendulum continues its movement to the right, the error decreases causing the pendulum to appear to recede from the vision system. When the pendulum changes direction, the sign of the error also changes and the error begins to increase again. The perceived change in depth will continue to be that the pendulum moves towards the vision system until mid swing when it will appear to recede once again. The path just described is one of counter clock-wise movement along the trace of an ellipsoid. If the synchronization of the inputs were reversed so that the left input arrived later than the right input, the perceived motion would be in a clock-wise direction.

5.7 Discussion

The SHM in this chapter has been extended in a natural way to include the dimensions of depth and time. There have been no fundamental changes to the method of computation of the transformations. The only real constraint to execution time is the degree to which the architecture can deliver scalibility. The implication for this extension to SHM pertains to its size and hence to the architecture's memory constraints. The size of SHM has increased by orders of magnitude. Consider the observation that the dimension of depth increases the memory requirements from two dimensions, by a quadratic factor and that time results in a linear increase to this. In particular, consider the implications for a one megabit camera operating at twenty five frames per second. Assuming that all information is kept, the memory

requirements would grow at the rate of 25×10^{12} units per second. At this rate, even the largest machines would reach capacity within the hour.

Obviously, with this capacity for growth of memory utilization, there must be a *graceful* way in which information can be compressed or just *forgotten.* Further discussion of the need for *graceful degradation*[3] is deferred until Chapter 6.

An interesting implication from 3DSA pertains to the choice of either binocular or parallaxis if one was to build a vision system requiring the perception of depth. The choice would appear to be a matter of application. In the animal world, birds and quadrupeds employ paralaxis and predators and primates use binocular systems. Birds and quadrupeds have a need to either spring or grasp at a stationary object from a stationary position. A predator requires precise information about the distance of the target if it is to leap without the ability to change its trajectory. A primate has a similar need if it is to jump from one limb of a tree to another. It can not change its trajectory in mid leap. Birds and quadrupeds have no such requirements. The implications for a machine system might be similar. The manipulation of a robotic arm from a stationary position may require a binocular system whereas a mobile robot navigating an obstacle course might find parallaxis superior.

[3]This term was coined by David Marr in [67]

Chapter 6

Future Directions

Kimo: "We have approached the eye of the spiral. It is time to turn around and work our way back out."

The philosophical basis from which this work emanates is that the vision process must be viewed as a *totality*. Crucial to realizing this paradigm is demonstrating how SHLA relates to the various components of a vision system. This will now be demonstrated from two perspectives:

i) A description of a general purpose machine vision system, named Akamai, will be presented

ii) Intriguing relationships between SHLA and biological vision systems will be discussed

6.1 Akamai

Akamai (pronounced ock-a-my) is the Hawaiian word for 'intelligent' and is the name chosen for the machine vision system currently being developed. One of the goals of Machine Vision yet to be attained is to create a general purpose machine vision system. Many special purpose machines have been built. These include: [?, 30, 43, 94, 83, 104, 79, 10, 92, 93]. The image understanding architecture at the University of Massachusetts, Amherst, consists of a 256K single-bit, Single Instruction Multiple Data (SIMD) processors at low level, 4K 16-bit Multiple Instruction Multiple Data (MIMD) processors in the middle and 64 MIMD processors at the top level [94]. The Hybrid Pyramidal Vision Machine at the Simon Fraser University has 512 SIMD processors at the low level and a 63 node transputer based MIMD processor at the top level [30]. The commercially available AIS 5000 consisting of 1K single-bit processors with mesh connected architecture [?] is a massive parallel processor.

These machines all exploit considerable parallelism and deliver enormous processing power but they fall well short of being general purpose machine vision systems. None of the machines developed so far can achieve the recognition of arbitrary objects

in real-time.

6.1.1 Overview of Akamai

Ultimately, it is intended that Akamai will be a general purpose machine vision system. At present, it is in the design stage; driven by the processing requirements and capacities of SHLA. There are four functional components of Akamai:

i) Image Acquisition Unit

ii) Image Analysis Unit

iii) Knowledge Generation Engine

iv) Locomotion Unit

Each of these four components has a biological counterpart and can be subdivided into smaller functional units. The Image Acquisition Unit corresponds to the eye. It consists of an image capture process (camera system) which corresponds to the front part of the eye, and a Pixel processor whose biological counterpart is the retina. The Image Analysis Unit may be thought of as the visual cortex of Akamai. It processes at three levels, referred to as low, intermediate and high[1]. The low level works on the raw image (light intensities) and produces a partitioning of the data. The intermediate level re-groups the low level partition into more meaningful classes. The high level involves various recognition processes. The Knowledge Engine corresponds to the cognitive processes of intelligence. The biological counterparts to Akamai's Locomotion Unit are the variety of anatomical structures which provide movement; for example, legs, neck and the socket of the eye.

[1]The terms low, intermediate and high level processes have currency in the vision literature which refers to broad classifications only.

Figure 6.1 displays the relationship of the functional units of Akamai. Each of these components will be discussed in the following sections.

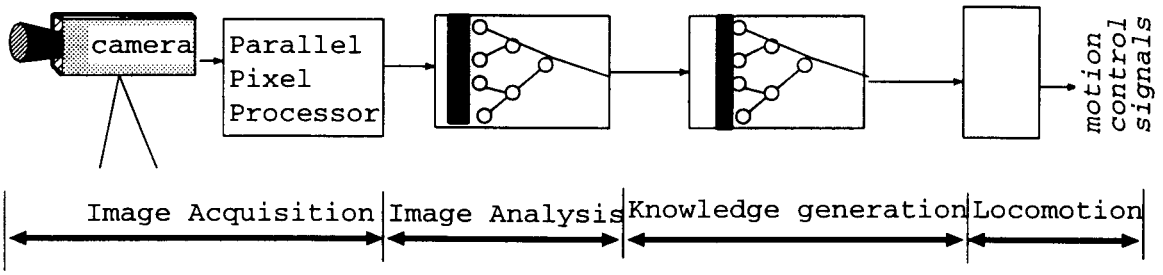


Figure 6.1: Architectural overview of Akamai.

6.1.2 Image Acquisition

The two functional components of the Image Acquisition Unit will now be described in detail.

Image capture

A dual camera system performs image capture. Each camera is a CCD high resolution area array with variable frame rate and sub-millisecond integration time. Each camera could be mounted either to act as a binocular system to gain high acuity depth perception from a stationary position, or as two monocular systems to gain the advantage of full spectrum *bird-like* viewing.

Pixel Processor

The necessity to have a *retina-like* process in an intelligent vision system can not be overstated. Recently, an innovative silicon retina has been developed by [69, 65] as an off-the-shelf component for a machine vision system. This apparatus emulates

certain aspects of the human retina. The Pixel Processor performs the interface from the camera to the Image Analysis Unit. It has four important features:

i) non-uniform sampling of the input image,

ii) transformation of the sampled image into multiple resolutions,

iii) parallel delivery of the multiple resolutions to the Image Analysis Unit, and

iv) dynamically change resolution delivery.

Non-Uniform Sampling

Biological vision systems provide a rich diversity of spatial sampling strategies. The retina in rabbits employs a uniform sampling distribution giving *snapshot like* images of their environment. It allows the input of optimal information without the necessity of eye or body movement. This is of great advantage to a non-combatant whose strategy is to avoid detection and then flee when the strategy fails.

Primates and predators, on the other hand, requires a more assertive form of vision; one which searches and homes in on particular targets. Their retinal sampling strategy is one of a non-uniform distribution. The densest sampling occurs at the centre of the retina. This is called foveal vision.

In the periphery of the retina, the spatial sampling decreases from the optical axis. See Figure 6.2. This strategy affords a certain computational advantage. The part of the scene requiring high resolution viewing can be centred on the fovea with a slight eye or body movement. An example of high resolution requirements is the precise knowledge of the location of small objects such as claws, branches or characters on a page. However, the requirement to detect a new object that has moved into the scene or of the general shape of a larger object can be realised by lower resolution sampling.

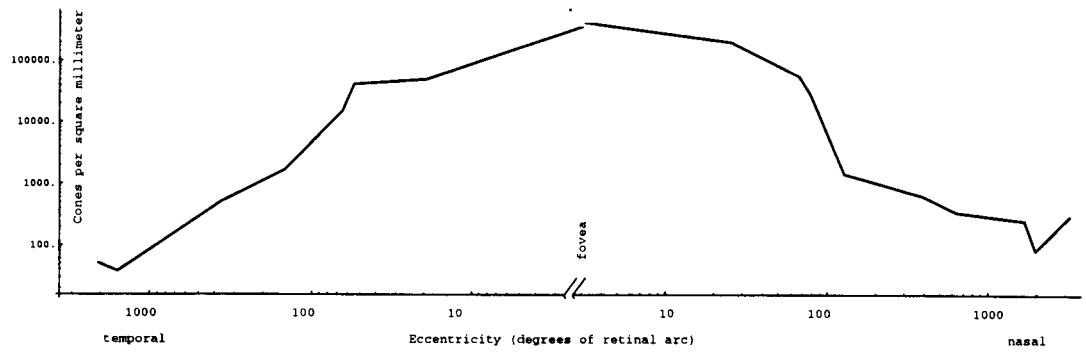


Figure 6.2: Cone distribution along the horizontal meridian in humans. Adapted from [23].

The strategy of non-uniform sampling has the potential for great computational saving so long as either the eye can move appropriately or the objects of interest in the scene move toward the optical axis. Examples of these two situations as applications for a machine vision system might be robotic navigation in the former case, and quality inspection of a product travelling a known path along a production line in the latter case.

The SHM lends itself to non-uniform sampling in a natural and computationally exploitable manner. By way of a simple illustration, consider the arrangement of hexagons in Figure 6.3. The addresses of the first seven hexagons (0 1, 2, 3, 4, 5, 6) are identified by Spiral Counting in the key of one. The addresses of the next six hexagons (10, 20, 30, 40, 50, 60) are identified by Spiral Counting in the key of 10. The last six (100, 200, 300, 400, 500, 600) by Spiral Counting in the key of 100. The density of hexagons over the three regions of this area yields:

region 1 7 hexagons per cluster = 7

region 2 1 hexagon per cluster = 1

region 3 1 hexagon per 7 clusters = (1/7)

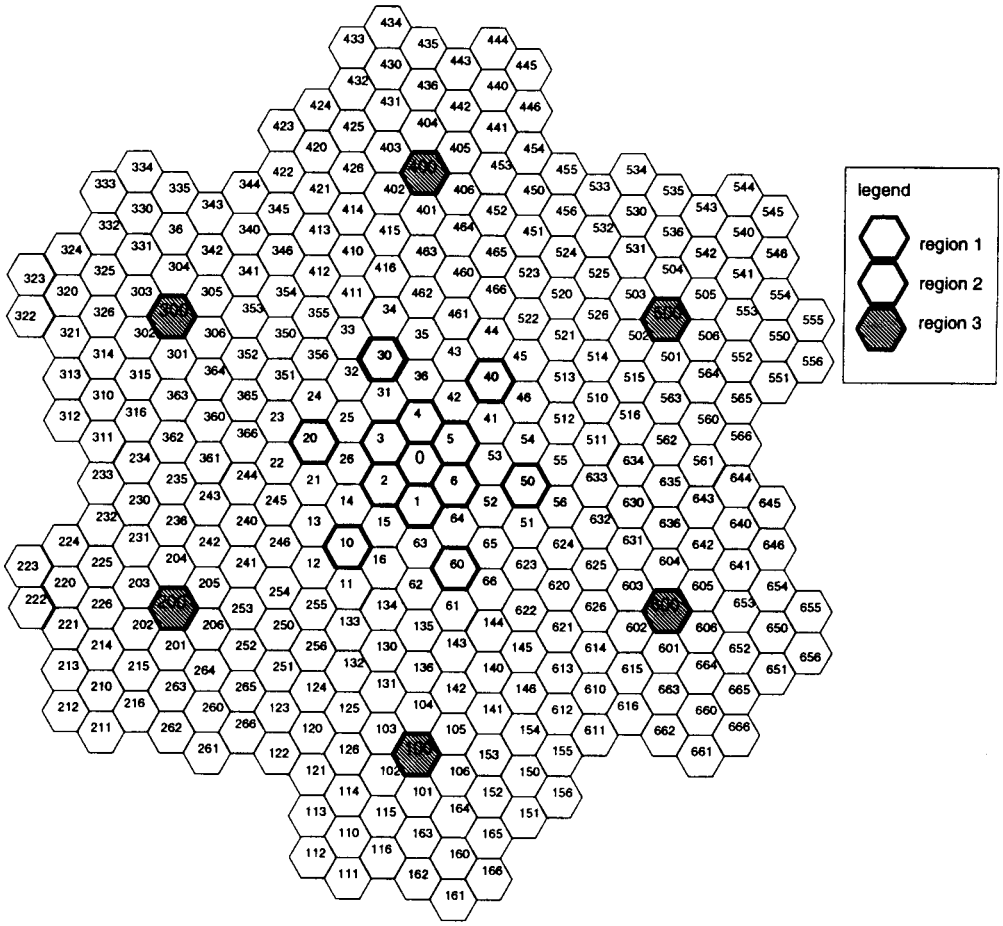


Figure 6.3: Distribution of a non-uniform sampling scheme. The shaded hexagons of the combined three fundamental units of vision in regions, 1, 2 and 3, indicates the distribution of one such scheme.

This constitutes an exponential decrease in density from the centre hexagon (line of optical axis) to the periphery.

In order to appreciate the potential decrease in computational complexity, compare the number of hexagons employed in the non-uniform versus uniform sampling strategies. The former has (7+6+6 = 19) hexagons and the latter has $7^3 = 343$ hexagons. The non-uniform strategy has $(19 \div 343) \times 100 \simeq 5$ % of the hexagons in the uniform case. Because the execution time of some of the low-level image processing, such as segmentation, grows exponentially with the number of hexagons to process, the non-uniform sampling strategy would then represent a potential for considerable reduction in computations.

Continuing the analysis of the hypothetical non-uniform sampling strategy, consider the problem of recognising a *small* object. If the objects in the environment were of a size compatible with being recognised with the foveal acuity, two situations may be considered. Firstly, if the eye were stationary or the objects moved in a straight line, it would be very unlikely that the objects would be detected by such a sampling strategy. This situation would constitute a *tunnel vision* scenario. On the other hand, if the eye was constantly on the move it is highly likely that the outer most peripheral hexagons would report the existence of an *unidentified intruder* in the environment. In response to this information, the eye could be moved for foveal viewing of the object. Thus, the combination of movement and non-uniform sampling forms a powerful partnership for an active vision system.

This hypothetical example also suggests that the successful employment of a non-uniform sampling scheme is dependent upon the applications involved. This motivates the first important feature of the Pixel Processor. That is, the size and density of the SHM's fovea and the rate of sampling decrease across the SHM to the periphery must be adaptable. This is achieved in one of two methods. In the case of *a priori*

information about the environment, these attributes could be determined at the outset. Alternatively, if intelligent vision is required, the attributes could be dynamically controlled with feedback from the Image Analysis Unit, as the process evolves.

Transformations to Multiple resolutions

The second important feature of the Pixel Processor is its ability to transform the input image into various resolutions. As was stated in Chapter 2, a general purpose machine vision system must be able to perform with minimal *a priori* information about the environment. One of SHLA's important properties that was stated in Chapter 2 and demonstrated in Chapter 3 is its ability to transform the SHM into all possible resolutions with equal computational complexity. The Pixel Processor will achieve this by employing the transformations $M_{10}, M_{100}, ...$

Parallel Delivery of Image

The third important feature of the Pixel Processor pertains to the manner in which the input image is delivered to the Image Analysis Unit. Once again, as it is not known *a priori* which resolution will contain the best *first pass* information about the environment, the strategy is to process all resolutions concurrently. Consequently, there are high speed serial links sending each resolution in parallel from the Pixel Processor to the Image Analysis Unit. It is also foreseen that there is a need to have dynamic re-configureability of these links to handle the following situation. In the event that the Image Analysis Unit detects *information overload*, it would send a message to the Pixel Processor to restrict the number of resolutions being sent and redistribute the load.

This facility may also have application in the multimedia environment. When a sequence of images is being transmitted either in parallel or sequentially, the receiving

component at times may not be able to cope with the rate of frame transfer. One strategy is to diminish the rate of transfer but possibly a more acceptable strategy would be to maintain the rate of image transfer and sacrifice resolution. The image would be transformed into an appropriate resolution and then that fraction of the original image would be transmitted. The effect would be to see the animation at the same speed but with less detail.

6.1.3 Image Analysis Unit

The Image Analysis Unit is the collection of processes that extract visual information from an image. These various processes will now be discussed.

Low level processing

A characteristics feature common to the *low level* processes is that their inputs are raw image (light intensity data), and their outputs are a collection of *organised equivalence classes*[2]. An example of the low level process, from this thesis, is the segmentation algorithm, 2DSA. In a sense, this algorithm is a quantification of the Gestalt law of visual perception known as *similarity.* It is anticipated that additional algorithms will need to be described for many of the other Gestalt laws governing visual perception. For example, *common fait* , which is the ability to segment a group of birds in flight as a single flock.

Intermediate Processes

The intermediate processes provide a further refinement of the low level outputs. An example of this level of processing is 3DSA, (described in Chapter 5) which creates a

[2]This concept of *organised equivalence classes* is probably closest to the idea of the *intrinsic image* as introduced by David Marr [67].

three-dimensional representation. One aspect of 3DSA, which is a general feature of the intermediate level, is that it receives inputs from both low and high level processes.

High Level Processes

A number of higher level processes still require development in order to create a realistic perceptual system. What follows is a description of one such higher level process. The common characteristic of the high level processes are that they involve some sort of *recognition* process. A simple example of this is provided by the Gestalt law of *good closure*. Specifically, it would be a process that selects equivalence classes that are *closed*. To illustrate, consider the equivalence classes represented in Figure 6.4. The collection of hexagons is partitioned into two equivalence classes. The unshaded

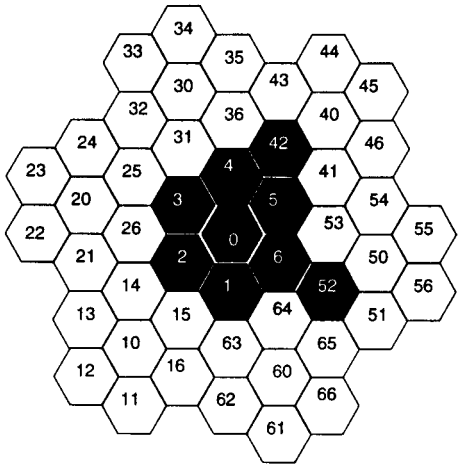


Figure 6.4: A collection of hexagons is partitioned into two equivalence classes. The unshaded hexagons constitute an open class and the collection of shaded hexagons is an example of a closed equivalence class.

class is open as it contains hexagons that are not completely surrounded by hexagons of another class. The class labelled by the shaded hexagons is closed so the meaning

of *closed* in this context is that the class is bounded by other classes within the collection. Observe that both classes in a) have sides exposed to the boundary of the cluster. Consequently, it is not known what other possible classes may be connected to either of them. In contrast, the class labelled with zero in b) can be *recognised*, with a first order guess, as having the shape of a circle. The *good closure* law, in this context, would be a simple matter to express as an algorithm and code.

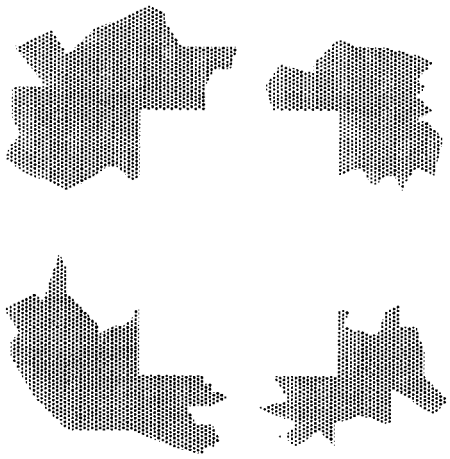


Figure 6.5: Viewing this figure induces the perception of a rectangle due to the presence of a corner etched in each of the four blobs.

Another more complex example of a high level process is provided by the process of creating *virtual equivalence classes*. This is an instance of the Gestalt law of *prognonce* [46]. Consider Figure 6.5. The low level segmentation 2DSA simply produces a partitioning consisting of five classes, four of which are closed and the fifth class (the background) is open. As soon as the human eye recognised the shape of the corner in each of the four closed classes and that they lined up in a certain manner, the visual perception of the sixth class is produced. This sixth class is closed and possesses the shape of a rectangle. It is a subset of the open class and is a virtual class because the light intensities within it and the light intensities in the open class are identical.

The key to the creation of this *virtual* class is the recognition of four corners which line up geometrically.

A precise description of an algorithm that creates *virtual* equivalence classes presupposes the existence of a higher order *feature extractor* that has as its inputs the outputs from 3DSA. With such *feature* information available, the creation of a *heuristic* to assemble the features should not be an insurmountable task.

Knowledge Generation Engine

It is impossible to separate *vision* from the process of *learning*. These two seemingly distinct processes are integral facets of one another in any biological vision system. Everyday examples abound. Consider the situation of a person who is beginning to collect puka shells[3]. These shells are, more or less, uniformly distributed over a beach. The would-be collector experiences a diminishing of the interval between successive finds. In other words, something *happens* that *empowers* the person to *see* the shells. This empowering involves the learning of the characteristic pattern and distinguishing between it and the surrounding similar patterns in the sand. The experience of this phenomenon by novice puka shells collectors is one which very few are able to explain. It is certainly not explained by simply memorising the pattern of the puka shell prior to the search and then commencing a *brute force* approach to the search. This process must be described in terms of the acquisition of a skill brought about by the interplay of vision and learning.

Therein lies the problem: *learning*, like *seeing*, is easily observable. However, it is deceptively difficult to understand. The field of *machine learning* is a new and exciting discipline of artificial intelligence where a number of techniques are attracting wide attention. These include a variety of symbolic algorithms, a renewed interest in neural

[3]Puka is a Hawaiian word meaning *hole*. Puka shells are thumb nail sized circular shells with a small hole in the centre which is used by the collector for threading into a necklace.

networks [52, 63] and genetic algorithms. All of these techniques will be investigated from the perspective of their compatibility with the computational features of SHLA.

Genetic algorithms could be of value in relation to the transformations produced by multiplication. Recall that one of the effects of the application of M_{10} to the input SHM was the production of multiple *near* copies of the image at a lower resolution. The segmentation algorithm, 2DSA, then worked on all the copies as independent images. The result of this processing is the production of a *population* of possible solutions to the recognition problem. A genetic algorithm could then work on this population of possible solutions to select the *best* one. Broadly speaking, it is anticipated that the learning module for Akamai will include a combination of these techniques. As the work on this aspect has recently begun, it is only possible to outline a few aspects of the cognitive unit.

The simplest level of this unit is called the *a priori* module. It is the mechanism by which the operator enters information about the environment applicable to a particular vision task. Examples of this includes any number of industrial applications where *a priori* information is available.

Another module of the cognitive unit is the *rejuvenator*; which is Akamai's data compression facility. After a vision task has been learned, the results of this experience must be stored in some accessible form. One of the data compression methods possible with SHLA is to store a sequence of related images (such as in a movie) as sequences of transformations that describe it. For example, if the movements of an intricate dance were to be remembered, it might be described in terms of the transformations on the dancer. The transformations in this case are the results of the Lie operators of SHLA.

6.1.4 Locomotion Unit

The discussion of Akamai's development has placed considerable emphasis on the *wholeness* of a vision system. The fact that movement is an integral part of that system means that it is essential to offer at least a brief treatment of this matter. Akamai is not a *disembodied eye*. It requires locomotion of two types, broadly classified as local and global movements. These two types of movements are analogous to the locomotion of the human vision system. The local movement is that of the cameras relative to the pixel processor. This motion must have the characteristics of being fine and quick. The global movement is that of the entire system relative to its environment. The requirements of this type of movement is highly dependent on the environment, just as is the case for biological systems. For example, the flight of birds and the pedestrian motion of primates are related to the requirements of their vision system.

6.1.5 Hardware for Akamai

All of the functional units interact with each other in varying degrees. While there are certain advantages in assembling a heterogeneous collection of computer components each targeted at specific tasks, [79], a homogeneous system provides for flexibility that can be tightened later in the implementation stage. The chosen computer hardware for the Image Analysis Unit and knowledge engine of Akamai is a network of transputers. Its architecture affords a number of desirable characteristics:

- massive parallelism,
- flexibility in the inclusion/deletion of nodes from the network,
- reconfigurability of its nodes both dynamically and statically,

- high speed serial communication links between nodes, and
- relatively low cost components.

To date only the T800 and T400 series of the transputer have been employed in the development of Akamai. As the T9000 series is now available it will play an important role in further developments. Its communication facilities are greatly enhanced.

6.1.6 Methodology

The previous sections of this chapter have outlined design considerations for the development of Akamai. In the process, the exciting prospect of producing a machine capable of the recognition of an arbitrary object in real-time has emerged. A clear direction is evident. The future development of Akamai is dependant on the assembling of an interdisciplinary team of researchers.

The methodology of concurrently working on design and implementation of various modules has demonstrated itself to be a fruitful approach. This approach is also consistent with the philosophical basis of the thesis; vision must be explored as a *totality*. The main theme of Akamai is that the SHM is related to all of its functional units. The principal tool for the manipulation of this space is the SHLA. SHLA's ability to work at numerous resolutions concurrently provides considerable scope for parallelism. It is conceivable, even likely, that recognition of an object at a high level process will be completed before the earliest 2DSA algorithm logically terminates. This would result from the coupled effects of the concurrency of the processes and the fact that the low resolution data from 2DSA should be available before the higher resolution data. This illustrates the need to consider the situation from, at the very least, the perspectives of computation and perception. Thus the combined skills of the computer scientist and the perceptual psychologist would be an advantage. This in turn cannot proceed without consideration of the engineering implications.

In addition, the insights provided by anatomical and physiological considerations of biological vision systems must be of value. The language that integrates and facilitates the interaction of all the aforementioned disciplines is mathematics. The importance of having a team that possesses all of these skills to bring Akamai to fruition can not be overstated.

6.2 Biological Considerations

A question on the minds of all who work in the field of machine vision systems is, *How is it that biological vision systems see?* This question promises to be with us for some time to come. Although it is not the intent of the thesis to attempt an answer to any part of the question, a number of issues have arisen from SHLA that pertain to it. This section will discuss some of those issues and in the process, *nibble* around the edges of the big question.

6.2.1 Understanding Biological Vision

The Gestalt psychologists at the beginning of this century made a pertinent observation with regards to the understanding of human visual perception. They suggested that this process could not be fully comprehended without, on the one hand, viewing it as a *whole* at its highest level, and on the other, coming to terms with the components that constrain it. In other words, there is a global nature to the process, a local nature and a link which connects these two natures. Since the early days of that observation, there have been worthy developments in each component of this trio. The most notable developments at the global level are those of Gibson [33, 34]; the concept of *visual constancy* and the *Figure Ground Theory.* At the local level, the pioneering work of the neurophysiologists, Hubel and Wiesel [50], on the function of

neurons in the visual striate secured credence to the hitherto speculations that the neurons possessed properties of a vector field (in the mathematical sense). A few years later in 1962, after the above mentioned contributions were established, the mathematician, Hoffman [46], employed the theory of Lie Algebra to establish the link connecting the global contributions of Gibson and their local counterparts from the work of Hubel and Wiesel. Hoffman's Lie Algebra approach forged the link by reconciling the following:

- the concerns of Gestalt psychologists with brain function as an organiser of perception
- modern knowledge of the neurophysiology of the visual system
- the need to proceed in a principled way from the local coding operations already mentioned above, to the global functional properties of visual perception
- the Gibsonian requirement that transformations and the invariance of pattern recognition under transformations be accounted for and in account of the evolution and consequent ecological validity of the system.

Dodwell gives an excellent account of this Lie Algebra approach in [27].

According to Dodwell, the full impact of Hoffman's significant contribution has not been accessible for two reasons. Firstly, its mathematical content requires a commitment of time and energy that most perceptual psychologists, amongst others, are either unable or unwilling to make in order to comprehend it. Secondly, the considered view of the computational community is that it is non-computational.

These criticisms aside, another weakness of Hoffman's theory is that it reflects few concerns imposed by anatomical constraints. In particular, the structure of the retina has not been accounted for. There is the tacit assumption of Hoffman's model that

the visual field, as represented by the retina, is a continuous space. The distribution of photo receivers on the retina is *discrete*. However, SHLA has exciting possibilities for reconciling the continuous structure of the visual world with the discrete structure of the visual field. The SHLA can be extended to a real-valued structure that possesses some similarities to Hoffman's model. In the next section this extension of SHLA will be described.

Apart from Hoffman's model of the human vision system, there have been many other attempts at producing a complete model. Although none of these have proven to be complete, many are worthy of note. A few examples are: [14, 85, 89].

6.2.2 The Analytic Component of SHM

The visual world, by virtue of the wave nature of light, is continuous and smooth. In mathematical terms, this is a manifold. On the other hand, the visual field as represented by the photo receiving device (retina or a camera) is discrete. If the photo receiving device is to provide a reasonable representation of the visual world, then it must follow that there exists a tight coupling of the visual field to the visual world. The SHLA extends in a natural way to a real-valued space to provide the desired coupling. The extension then forges an *analytic* tool that may be of use in both machine vision and the investigation of biological vision systems.

The first part of the problem is to obtain a continuous version of SHM. The construction principle employed to obtain the discrete space provides the mechanism.

In Chapter 2, it was shown that the SHM was in one-to-one correspondence with the integers base seven. The process begins with an SHM of arbitrary size. Consider the addresses of the hexagons in SHM as real numbers by simply placing a *.0* to the left of the one's digit. Declare an integer n = 1. Repeatedly perform the following steps:

i) Dilate SHM and enclose each address with six new hexagons giving each its appropriate note, *ray* to *te*.

ii) The address of each new note then is the address of $doh + (i \times 10^{-n} for\ i = 1 \cdots 6)$ corresponding to *ray* to *te*.

iii) Increment n by one.

At this point, the infinite collection of hexagons has real-valued addresses. In order to establish that this collection is actually a manifold, it must be shown that the areas of the hexagons converge to zero. To this end, consider two observations which result from the construction of SHM:

i) The metre of the primary octave in SHM is one

ii) The ratio of the metre of any octave to the metre of the next higher octave is $\sqrt{10 - (6\cos(\pi/3)}$. Call this ratio ρ.

It then follows that the metre of any lower octave, as determined by n, is $(1/\rho)^n$. Thus, the metre converges to zero, follows from the observation: Given an $\epsilon > 0$, $\exists$ an octave determined by n such that $(1/\rho)^n < \epsilon$. In the limiting case for n, the hexagons collapse to points.

In order to establish the required operations on this set of points, an adjustment to *Spiral Counting* is needed. This is established most naturally by extending the concept of counting in a *key* that includes the addresses of the lower octaves.

Let this collection of points, with their associated addresses be called Spiral Honeycomb Environment (SHE). SHE has a number of interesting properties:

i) For any two addresses in SHE there is an address possessing the property of being

(a) strictly greater than one of the addresses and

(b) strictly less than the other

ii) properties *(a) and (b)* induce a covering of the visual field, a bounded (2-dimensional plane).

iii) property ii) provides the tool by which real analysis can be performed on the image space.

The following theorem summarises the algebraic properties of the SHE.

Theorem 4

i) There is a bijective mapping from SHE to the positive reals base seven.

ii) SHE is a separable Hausdorf space.

iii) The extension of operations of SHLA to SHE is a Lie Algebra.

Two implications arise from Theorem 4. Both pertain to Hoffman's Liner Transformation Groups for Neurophysiology (LTG/NP) [46]. The similarities between the Lie operators of LTG/NP and SHLA are striking. How close is the relationship? In particular, the discovery of a *homomorphism* between the two structures would serve to motivate the investigation of SHLA as part of a model for the human vision system.

The second implication from Theorem 4, relates to its possible usefulness in finding the invariant transformations for particular images. More specifically, the discrete SHLA requires considerable computation work to be able to find and apply certain transformations in the contexts of *invariance*; whereas in mathematics it is often easier to work with analytic tools achieving results which would be either impossible or more difficult if one were working with purely computaional tools[4].

[4]Paul Halmos provides some interesting examples of cases when analytic techniques produce superior computational results to that obtained with computational techniques in [40].

6.2.3 The Retinal Cone Mosaic

The distribution of cones is densest in the fovea and decreases exponentially towards the periphery. This geometry has been exploited by many biological vision systems. There have also been attempts to model this geometry for machine vision [70, 4, 91]. Each of these models has been demonstrated to possess attributes possessed by the biological counterparts. However, one of the mysteries associated with this architecture is the prediction of a certain type of error, known as *aliasing.* In particular, a theorem from Information Theory, states that a non-uniform sampling of a signal will incur more aliasing error than that of a uniform sampling scheme given the same number of sample points [106].

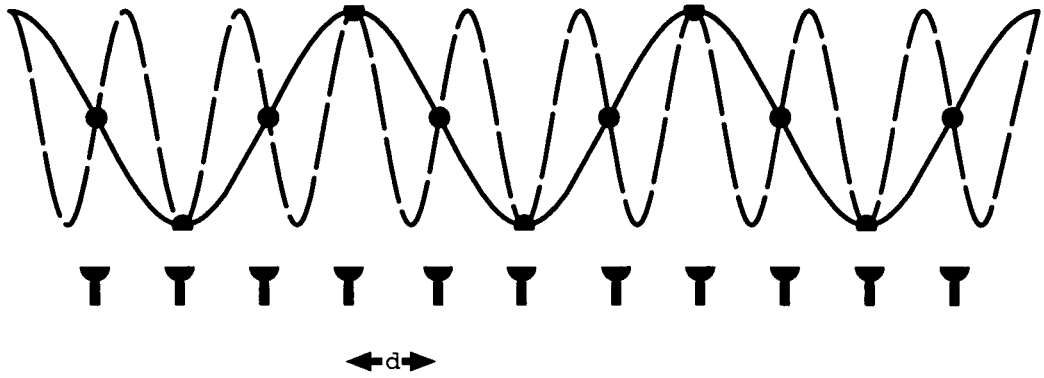


Figure 6.6: Two sinusoidal curves with the same amplitude and different frequencies appear as aliases when sampled by the array of photo receptors of the figure.

The aliasing problem is illustrated in Figure 6.6 with a one-dimensional array of sample points attempting to detect a sinusoidal curve. The *aliasing* effect is that the two curves will be classified as one and the same. The theory predicts that this situation is exacerbated when the spacing between the sample points is not regular. The non-uniform sampling of the retina of primates provides such a scenario. The question that emerges is, why would such a retina be allowed to evolve when

the predicted error has such great consequences? In other words, one might have speculated that the effect of not being able to precisely know the location of a potential threat might very well lead to its extinction. However, the reverse is the reality. The predators, for whom the effects of aliasing would constitute a disaster, have adopted this non-uniform distribution.

A possible explanation for this anomaly might be suggested by the sampling strategy adopted by Akamai, as described in section 6.1.3. More specifically, this strategy, which models the geometry of the primates to some degree, has in effect, equal spacing between sample points of a given resolution. This effect is produced by the resolution changing transformation M_x, where in each resolution of the image the spacing between the adjacent hexagons is of equal distance. This line of reasoning leads to the conjecture that the neurological pathways from the retina to the visual cortex may be employing a similar strategy. It might be of interest to examine the human vision system from this anatomical and physiological perspective.

6.2.4 Binocular Vision

The relationship between the architecture of a vision system and the optical illusion produced by the system has been the focus of attention of some interesting research [88]. The illusion of *seeing* a straight line as a curve under certain circumstances, produced by binocular vision systems, is well known if not well understood [46]. In Section 5.6, two examples of this phenomenon were discussed. The first related to the ancient Greek's practice of placing deliberate curves in columns in order to compensate for this error in perception experienced by binocular viewers. The second was the effect of perceiving elliptical motion in a swinging pendulum when the actual swinging motion was in a plane. The interesting point with regards 3DSA is that this same type of error is inherent to the internal representation of the three-dimensional

reconstruction of the scene. That is, the error was produced by the construction of the internal representation not in the recognition process at a higher level.

A few questions arise in consideration of the above mentioned observation. One such question is: How close is the predicted error in Akamai's 3DSA to the actual error experienced by humans? It would be of interest to make precise measurements of the human error and compare it to the 3DSA predicted error. If it turned out that 3DSA is a good predictor of the human error, it may then suggest a line of inquiry into further physiological understanding of binocular vision in humans.

Another question lies in considering the previous question in reverse. Can the study of optical illusion in humans provide results that could be used in the construction of algorithms for machine vision systems? In one sense, the measurable error of a system's performance indicates something about how that system works. For example, an object in motion appears to be smaller than when it is stationary. Does the study of this illusion have any potential for the development of algorithms governing the perception of motion?

6.2.5 Irregularities in the Cone Mosaic

Research into anatomical considerations of the cone mosaic on the retina of monkeys and humans has uncovered two perplexing irregularities. One is the variation in cone size and the other is the variation is cone placement on the retina.

Variations in cone size The diameter of cones increase with eccentricity from the fovea. This variation has implications to the vision process and is discussed in [106, 13]. However, explanations for this phenomenon are the foci of an interesting debate. One possible line of investigation which results from the SHM is a simple geometrical consideration as follows: The increase in cone size may result from a need for greater surface area in order to capture a critical

quantum of light. In the fovea, the surface of the cones are at right angles to the incident light rays. Towards the periphery, the rays of light strike the cones at decreasing angles from the perpendicular. This non-perpendicular angle results from the angle of defraction from the curvature at the side of the lens. It could be of interest to investigate the correlation of the angles of defraction to the surface area of the cones.

Variations in Cone Placement The non-uniform distribution of cones on the retina, as previously discussed, is irregular in the sense that apparently no existing model accurately predicts their locations. Once again this does not sit very well with the reconciliation of the process of *Natural Selection* and *Modern Information Theory*. There have been arguments put forward to bridge this apparent discrepancy [106]. However, there is no general agreement on this matter.

Considerations of SHLA might add to this debate. In particular, the transformations that produce multiple resolutions of the input image also produced multiple *near copies* within any given resolution. It was also suggested that this feature could also be of value to the knowledge generation engine of Akamai. The implication here is that the distribution of the hexagons within any particular *near copy* is highly regular. However, consideration of hexagons from distinct *near copies* would appear irregular. This suggests that it could be of interest to look at the distribution of cones on the retina in this context. It may be that the theorems from Information Theory apply to the human vision system to a higher degree than have previously been credited.

6.3 Summary

This chapter has explored the paradigm of *considering a vision system as a totality* from two perspectives. The first perspective described the development of a machine vision system named Akamai. The second perspective discussed some possible implications of SHLA to the understanding of biological vision systems. While some of the ideas presented in the section on *Biological Considerations* are controversial, they nevertheless have the potential to contribute to the investigation of understanding biological vision systems. The ideas are presented in the spirit of attempting to link the efforts of what otherwise may appear as divergent disciplines. It is certainly not the intention to suggest that the significance of SHLA is dependant on the validity of these implied links. Rather it is emphasised that SHLA was developed for a machine vision system implemented on a parallel computer architecture. Any light that may be cast into the domain of biological vision systems as a result of such speculations would be purely serendipidous.

The final word in this chapter must serve to relate the disciplines of machine vision and biological vision. The underlying assumption in both disciplines is that vision is somehow *computational.* Therefore, any computational principles that are discovered in either the realm of machine or biological vision systems must be considered by the other. Indeed, if vision is a *totality*, then surely the efforts of each discipline must serve the other.

Chapter 7

The Next Higher Octave

On a morning preceeding the judgement of the Challenge Pekanini approached Kimo with a lowered head and in a sombre mood.

Kimo: "Why so sad Pekanini?"

Pekanini: "We have not yet met the Challenge of The Third Servant."

Kimo: "Very few people ever meet such challenges. However, ultimately one is judged, not on the degree to which the Challenge is met, but rather how it is that the game was played."

With that pronouncement, Kimo turned his gaze to the horizon where a wave, like an idea approaching that moment of its expression, was forming out of the depths, growing and preparing for its expression. As it peaked, a lone surfer anticipating the moment, lifted high onto its crest, leaned into its face and accompanied by Kimo's conciousness, started the acceleration deep into the eye of the spiralling water.

Pekanini, realizing that the day's session was over, turned and with the warmth from the sand rising through her feet, her spirits lifted and she began her spiral dance.

This final chapter summarizes the results of the previous six chapters and provides a concluding statement to the work.

7.1 Summary

The material in this thesis has unfolded in six stages. The first stage involved articulating some of the perplexities of machine vision. The goal of demonstrating how a machine vision system could be endowed with the capability of emulating certain aspects of primate vision emerged from the observation that the arrangement of cones on the retina of primates possesses computational properties.

These computational properties were then formulated in mathematical terms which resulted in the theory of the Spiral Honeycomb Lie Algebra (SHLA). The significance of SHLA is that it describes a mechanism by which the continuous nature of light can be captured by a discrete machine so that effective processing can be performed on the associated image. It was also shown that additional investigation into the mathematical structure of SHLA has the potential to further enrich its theory and enhance its application.

It was shown how the computational potential of SHLA could be realised on the parallel architecture of a network of transputers. In particular, it was demonstrated that *real-time* performance of the primitive image transformations could be achieved on a network containing a modest number of transputers.

The practical applications of the transformations of SHLA were described. One application was in the form of a parallel autonomous segmentation algorithm, 2DSA. Although the target of *real-time* performance of the implementation of 2DSA was not achieved on the available network of transputers, the results did suggest that the algorithm has the potential to reach the target. Another application showed how

SHLA and 2DSA could be extended in a manner compatible with SHLA's implementation to handle higher dimensional vision data. In particular, a description of how the dimensions of time and depth could be naturally incorporated into the theory of SHLA was provided.

Two further avenues of exploration that result from SHLA's approach to machine vision were discussed. One was the partial description of the design of a general purpose machine vision system named Akamai. The essence of Akamai's design is that it embodies the principles of SHLA. Some other psycho-physical lines of exploration regarding biological vision systems were postulated.

The final stage ties the end to the beginning. This is done not in the spirit of ending the exploration but rather to provide the means to step to the next level of research, *one octave higher.*

7.2 Conclusion

This thesis has formulated a novel approach to machine vision that reconciles the local nature of machine computations with the totality of the vision process. This has been achieved by capturing fundamental computational properties inherent in the geometrical arrangement of cones on a primate's retina. It has been shown that the *fundamental unit of vision*, which was introduced in Chapter 1, uniquely affords the following attributes to a spatial data structure which is based on a hexagonal grid:

- All of the primitive image processing transformations (arbitrary translation, rotation, magnification and contraction of an image) can be achieved without incurring either rounding error or distortion of the image. The image will not degrade as a result of applying any sequence of transformations to it and that such a result could not be achieved if the image were represented and manipulated on a rectangular grid.

- Autonomous segmentation of a multi dimensional (space and time) image is achievable. In particular, the internal boundaries of an image can be determined without resorting to *a priori* information about the image. This type of segmentation could not be achieved on a rectangular grid without relying on *a priori* information or resorting to the vagaries of a heuristic.

- The primitive transformations may be combined to obtain the complex transformations required to describe the movement of objects in space and time. These complex transformations are as computationally simple as the primitives.

The fact that SHLA embodies all of the computational properties listed above and can be efficiently implemented on a computer possessing a parallel processing architecture confirms that SHLA stands as a powerful and new data structure on which the architecture for machine vision can now be based.

Appendix A

Kimo: "So, convince me that your counting is correct, Pekanini."

Appendix A Proofs of Theorems

This appendix contains the outlines of the mathematical proofs to the theorems of Chapter 2.

Theorem 1 *Suppose that the SHM is defined with the operations of Spiral Addition and Spiral Multiplication, as defined in Equations 2.1 and 2.2. Then,*

i) SHM is an abelian cyclic 7-group under the operation of Spiral Addition.

ii) SHM is a Euclidean ring.

Let the number of hexagons in SHM be

$K = 7^n$ *for* $n \geq 0$ *and let the arithmetical operations be modulo* K *as defined in Equations 2.3, 2.4 and 2.5.*

Then,

iii) The subset of SHM given by {0, 1, 2, 3, 4, 5, 6} is a finite field.

iv) The collection of addresses of SHM which are not a multiple of 10 form an abelian group under the operation of Spiral Multiplication and has order $6 \times 7^{n-1}$.

Proof: The proof is by induction on the metre of the addresses in SHM. The metre, see definition 2.3 in Chapter 2, associated with a given address is the length of the line segment joining the centre of the hexagon at that address to the centre of the hexagon at address zero. The metre of each of the addresses in the set (1, 2, 3, 4, 5, 6) is one. It is an easy matter to demonstrate the initial case of the induction by performing a hand calculation of the five properties of associativity, distributivity, existence of identities, existence of inverses and commutetivity all hold for the addresses of the set set (0, 1, 2, 3, 4, 5, 6).

Next, the general case of the induction will be treated.

Associative property : $a + (b + c) = (a + b) + c \forall\, a,\, b,\, c \in SHM$

Assume that the property does not hold for all a, b, c in SHM. Then as it does hold for the addresses whose metre is less than or equal to one, there must be a minimal counter example. That is, there must be an element, c say, for which the property does not hold but that for all elements whose metre is less than the metre of c, the property holds. There exists elements x and k such that

$$\begin{aligned} c &= x + k \text{ and } metre\, of\, x < metre\, of\, c \text{ and } k \in (1, 2, 3, 4, 5, 6). \\ a + (b + c) &= a + (b + (x + k)) \\ &= a + (b + x) + k, \text{ by choice of minimal counter example} \\ &= (a + b) + x + k, \text{ by choice of minimal counter example} \\ &= (a + b) + c. \end{aligned}$$

This last step shows that no such minimal counter example exists and so the assumption must be false. Therefore, the associative law for the operation of Spiral Addition is established. A similar argument provides the proof for the associative law for the operation of Spiral Multiplication.

Distributive and Commutative laws : The proofs follow near identical lines as was given for the associative law and are omitted.

Existence of unique additive and muliplicative identities : 0 is the additive identity and 1 is the multiplicative identity. This follows directly from their respective definitions.

Existence of unique additive and multiplicative inverse : The proofs of this property for both operations are constructive. Given an element a $\in$ SHM which is of the nth octave then a is expressible as:

$a = a_n, a_{n-1}, ...a_1 \forall\, a_i \in (0, 1, 2, 3, 4, 5, 6)\, and\, 1 \leq i \leq n.$

The additive inverses of the elements in the set (0, 1, 2, 3, 4, 5, 6) are (0, 4, 5, 6, 1, 2, 3). Then, the additive inverse of a denoted $-a$ is given by, $-a = -a_n, -a_{n-1}, \ldots - a_1$

It then follows that $a + (-a) = 0$.

The multiplicative inverse of the element, a, is constructed by considering the situation in two cases as follows:

Case 1: The element is not a multiple of 10.

Let the inverse of a be denoted by a^{-1} and let $a^{-1} = b = b_n, b_{n-1}, \ldots b_1$ for $b_i \in (0, 1, 2, 3, 4, 5, 6)$.

First observe that the multiplicative inverses of the elements in the set (1, 2, 3, 4, 5, 6) are (1, 6, 5, 4, 3, 2), respectively. The construction now proceeds inductively on the subscript of b.

$$\begin{aligned} b_1 &= a_1^{-1} \\ b_2 &= -(a_1 b_1) \times b_1 \\ b_3 &= -(a_3 b_1 + a_2 b_2) \times b_1 \\ b_4 &= -(a_4 b_1 + a_3 b_2 + a_2 b_3) \times b_1 \\ &\cdot \\ b_n &= -(a_n b_1 + a_{n-1} b_2 + \ldots + a_2 b_{n-1}) \times b_1 \\ &= (-(\sum_{i=0}^{n-2} a_{n-i} \times b_{i+1})) \times b_1. \end{aligned}$$

Case 2: The element is a multiple of 10.

First find the value $m > 0$ minimal to the constraint that the product of the element a and 10^m is greater than 10^n and the least significant digit of the modular value of the product is not zero. Then, let k denote the modular value of this product. Next, find k^{-1} by employing the algorithm from **Case 1**.

Lastly, a^{-1} is then equal to the product of k^{-1} and 10^m.

The above algorithm for generating a^{-1} can be use in an inductive proof to show that $a \times a^{-1} = 1$.

The establishment of the five properties for each of the operations of Spiral Addition and Spiral Multiplication is intact. We can now conclude that SHM is an abelian group under the operation of Spiral Addition and the non-zero elements of SHM have the properties of an abelian group bar the Existence of the inverse for each element. The Euclidean property follows immediately from the definition when d(a) is defined as the absolute value of a. These properties together with the distributive law describes SHM as a Euclidean ring.

The fact that the set (1, 2, 3, 4, 5, 6) forms a finite field follows from the observation that each of its elements has a multiplicative inverse within the set.

The modular form of the multiplicative operation provides the existence of the multiplicative inverse for all non-zero elements.

Theorem 2 With regards the transformations on SHM:

i) Each of the sets $\mathcal{A}$ and $\mathcal{M}$ forms an algebra under the operations of addition, multiplication and scalar multiplication (equations 2.8 through 2.10).

ii) The external direct sum of $\mathcal{A} \oplus \mathcal{M}$ forms an algebra under the pairwise operations of addition, multiplication and scalar multiplication. Let this algebra be denoted by T.

Proof: These results follow from the establishment that SHM is a field. The details of the proof of the properties of an algebra proceed along standard lines that may be found in most good books on introductory abstract algebra [42], page 217.

Theorem 3 .

i) SHLA is a Lie Algebra.

ii) SHLA is a subspace of the dual of SHM

Proof: As SHLA has been established as an algebra, Theorem 2, the result that it is a Lie Algebra follows directly from the definition of the Lie operator [38] page 209.

The result that SHLA is a subspace of the dual of SHM largely follows as a consequence of the definition of the dual of a vector space ([42] page 146).

Glossary

Pekanini: "Can you please remind me of a few things, Kimo?"

Glossary

Computational: The term computational has different meanings to different people and is used in a variety of contexts. Reference to the term *computational model* in the context of this thesis will generally relate to the speed and accuracy with which that model can execute on a machine. In terms of speed, it will mean real-time execution. In terms of accuracy, it will generally mean that the model is discrete, in the mathematical sense, or can be treated as a discrete model without loss of effect.

Foveal sensing: The fovea is the centre part of the retina. It has the highest density of cones and takes up only 2 or 3 degrees of arc on the retina's surface. The term *foveal sensing* refers to the process of employing the fovea to achieve acute vision.

Invariance: This term shall generally be used in the context of transformations on images, such as rotations or translations. It will refer to the geometrical property of some aspect of the image being unchanged under the action of the transformation. For example, the moving of one's head with the eyes open generates a transformation of the scene being viewed. However, the objects that make up the scene are largely unchanged with regard to their shape and relative position.

Real-time: This term refers to the performance of a computing system. If the desired outputs of the system can be delivered at a rate commencerate with handling subsequent inputs, the system is said to be operating in real-time. By necessity, a biological vision system must be a real-time system.

Visual field: is the projection of the visual world onto the retina of the eye. This is a two dimensional space.

Visual world: The time-space continuum that the eye or camera is pointing at.

Other Publications by the Author

Akamai - algorithms and performance of an image processing system using parallel procesing. In *Applied Machine Vision Conference*, Minneapolis, Minnesota, June 1994. Society of Manufacturing Engineers. (with T. Hintz)

A General Purpose Parallel Pixel Processor for Real Time Machine Vision Applications. IEEE, 1993. (with A. Nachimuthu and T. Hintz)

Transputer implementation of primitive image processing operations. In *Parallel Computing and Transputer Conference*, Brisbane, Australia, November 1993. IOS Press. (with T. Hintz.)

A spiral lie algebra approach to active machine vision. Technical report, 1994. (with T. Hintz)

Design details for a parallel proscsssing product inspection vision system. Technical report, University of Technology, Sydney, 1993. (with T. Hintz and N. Anantharaj)

Spiral architecture in machine vision. In T. Bossamier, editor, *Australian Occam and Transputer Conference.* IOS Press, Amsterdam, 1991. (with T. Hintz and W. Moore)

The spiral resolution pyramid - optimization in a message passing parallel processing image system. In S. Atkins and A. Wagner, editors, *Transputer Research and Applications 6.* IOS Press, 1993. (with T. Hintz and A. Nachimuthu)

Bibliography

[1] ADJOUADI, M., AND CANDOCIA, F. A stereo matching paradigm based on the walsh transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 12 (December 1994), 1212–18.

[2] AHUJA, N. On approaches to polygonal decomposition for hiearchical image representation. In *Computer Vision, Graphics, and Image Processing* (1983), vol. 24, Academic Press, pp. 200–214.

[3] AHUJA, N., AND TUCERYAN, M. Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt. *Computer Vision, Graphics and Image Processing 48* (1989), 304–56.

[4] ALOIMONOS, I., WEISS, I., AND BANDYOPADHYAY, A. *Active Vision.* Int. J. Comp. Vis, 1988, pp. 1:333–356.

[5] ARBIB, M., AND LIAW, J. Sensorimotor transformations in the worlds of frogs and robots. *Artificial Intelligence 72* (1995), 53–79.

[6] ARMAN, F., AND AGGARWAL, J. Model-based object recognition in dense-range images - a review. *ACM Computing Surveys 25*, 1 (March 1993), 5–43.

[7] ATHANAS, P., AND ABBOTT, A. L. Real-time image processing on a custom computing platform. *Computer* (February 1995), 16–24.

[8] BARROW, H., AND TENENBAUM, J. Recovering intrinsic scene charactreristics from images. In *Computer Vision Systems* (New York, 1978), A. Hanson and E. Riseman, Eds., Acadimic Press.

[9] BATCHLER, B. *Practical Approach to Pattern Classification.* Plenon Press, 1974.

[10] BERNARD, T., NGUYEN, P., DEVOS, F., AND ZAVIDOVIQUE, B. A programmable vlsi retina for rough vision. *Machine Vision and Applications 7* (1993), 4–11.

[11] BIGUN, J. A structure feature for some image processing applications based on spiral functions. *Computer Vision, Graphics, and Image Processing 51* (1990), 166–194.

[12] BLACKBURN, N. Automorphisms of finite p-groups. *Journal of Algebra 3* (1966), 28–29.

[13] BOSISOMAIER, T., SYNDER, T., AND HUGHES, A. *Vision Res 25* (1985), 145–147.

[14] BRACCINI, C., GAMBARDELLA, G., SANDINI, G., AND TAGLIASCO, V. A model of the early stages of the human visual system: Functional and topological transformations performed in the peripheral visual field. *Biological Cybernetics 44* (1982), 47–58.

[15] BURT, P. Tree and pyramidal structures for coding hexagonally sampled binary images. *Computer Graphics and Image Processing 14*, 3 (November 1980), 249–270.

[16] BURT, P. J. *Machine Vision.* Academic Press, 1988, pp. 1–30.

[17] CARLSON, C., KLOPFENSTEIN, R., AND ANDERSON, C. Spatially inhomogeneous scaled transforms for vision and pattern recognition. *Optics Letters 6*, 8 (1981), 386–88.

[18] CHARNIAK, E. *Introduction to Artificial Intelligence.* Addison-Wesley, 1984, ch. Vision, p. 94.

[19] CHOUDHARY, A., AND RANKA, S. Parallel processing for computer vision and image understanding. *Computer* (February 1992), 7–9.

[20] COK, R., GERSTENBERGER, J., LAWRENCE, C., AND NEAMTU, H. Applications of a parallel image processor. In *Transputing '91* (1991), P. Welch, Ed., IOS Press, pp. 15–47.

[21] COK, R. S. Prentice-Hall, 1991.

[22] CROWLEY, J. A local feature based human face recognition system. *IEEE* (1994), 32–36.

[23] CURCIO, C., SLOAN, K., PACKER, O., HENDRICKSON, A. E., AND KALINA, R. Distribution of cones in human and monkey retina: Individual variability and radial asymmetry. *Science 236* (May 1987), 579–82.

[24] CYPHER, R., AND SANZ, J. Simd architectures and algorithms for image processing and computer vision. *IEEE Transactions on Acoustics, Speech and Signal Processing 37*, 12 (December 1989), 2158–74.

[25] DAVIS, L., WU, Z., AND SUN, H. Contour-based motion estimation. *Computer Vision, Graphics and Image Processing 23* (1983), 313–26.

[26] D'HAEYER, J., AND BRUYLAND, I. Parallel computation of image curve velocity fields. *Computer Vision, Graphics and Image Processing 43* (1988), 239–55.

[27] DODWELL, P. *Transformations after Gibson: Biological and Ecological Constraints on Vision.* 1991.

[28] DUNCAN, G., AND LEUNG. Processing of offset vertical seismic profiling data. In *1991 WTUG Conference* (1991), pp. 355–366.

[29] EBRAHIMI, T., AND KUNT, M. Image compression by gabor expansion. *Optical Engineering 30*, 7 (1991), 873–79.

[30] ENS, J., AND ET.AL. A hybrid pyramidal vision machine for real time object recognition. In *Transputer Research and Applications 5* (IOS Press, 1992), A. Veronis and Y. Parker, Eds., pp. 90–103.

[31] FLYNN, P. 3-d object recognition with symmetric models: Symmetry extraction and encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 8 (August 1994), 814–18.

[32] GARDNER. *Spirals, In the Unexpected Hanging.* Simon and Schuster, 1969.

[33] GIBSON, J. *The Perception of The Visual World.* Houghton-Miffin, Boston, 1950.

[34] GIBSON, J. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

[35] GIBSON, L., AND LUCAS, D. Vectorization of raster images using hierarchical methods. *Computer Graphics and Image Processing 20* (1982), 82–89.

[36] GIBSON, L., AND LUCAS, D. Vectorization of raster images using hierartical methods. *Computer Graphics and Image Processing 20*, 1 (September 1982), 82–89.

[37] GONZALEZ, R., AND WOODS, R. *Digital Image Processing.* Addison-Wesley, Massachusetts, 1992.

[38] GORENSTEIN, D. *Finite Groups.* Harper's Series in Modern Mathematics. Harper and Row, New York, 1968.

[39] GU, Y. L. An exploration of orientation representation by lie algebra for robotic applications. *IEEE Transactions on Systems, Man, and Cybernetics 20*, 1 (February 1990), 243–48.

[40] HALMOS, P. *I want to be a Mathematician.* Springer-Verlag, 1985.

[41] HARGITTAI, I., AND PICKOVER, C., Eds. *Spiral Symmetry.* World Scientific, 1992.

[42] HERSTEIN, I. N. *Topics in Algebra.* Blaisdell, Waltham, Massachusetts, 1964.

[43] HIRSCHAND, E., PAILLOU, H., MUELLER, C., AND GENGENBACH, V. A versatile parallel computer architecture for machine vision. In *Transputing '91* (1991), P. Welch, Ed., IOS Press, pp. 828–58.

[44] HOARE, C., Ed. *Inmos Limited occam 2 Reference Manual.* Prentice Hall, 1988.

[45] HOCHBERG, J. Machines should not see as people do, but must know how people see. *Computer Vision, Graphics and Image Processing 37* (1987), 221–37.

[46] HOFFMAN, W. The lie algebra of visual perception. *Journal of Mathematical Psychology 3* (1966), 65–98.

[47] HOLSCHNEIDER, M. Wavelet analysis over abelian groups. *Applied and Computational Harmonic Analysis 2* (1995), 52–60.

[48] HORAUD, R. New methods for matching 3-d objects with single perspective views. *IEEE Transactions on Pattern Analysis and Machine Intelligence Pami-9* (May 1987), 401–12.

[49] HORN, B. Understanding image intensities, 1977.

[50] HUBEL, D. *Eye, Brain and Vision.* Scientific American Library, 1988.

[51] HUCK, F., FALES, C., JOBSON, D., PARK, S., AND SAMMS, R. Image-plane processing of visual information. *Applied Optics 23*, 18 (September 1984), 3160–67.

[52] HUETER, G. Neural networks automate inspections. *Quality* (January 1993), 41–4.

[53] HUGHES, A. *Handbook of Sensory Physiology.* Springer - Verlag, 1977.

[54] JEACOCKE, M., AND LOVELL, B. A multi-resolution algorithm for cytological image segmentation. *IEEE* (1994), 322–26.

[55] KANATANI, K. Structure and motion from optical flow under orthographic projection. *Computer Vision, Graphics and Image Processing 35* (1986), 181–99.

[56] KASPRZAK, W., AND WETZEL, D. Consecutive tree search for dynamic road scene analysis. *IEEE* (1994), 312–16.

[57] KOHLER, W. Gestalt psychology, 1929.

[58] KORHONEN, T. Evaluation of speed-up and efficiency in real-time parallel processing- on-line pattern recognition in the venus experiment. *Transputer Communications 2*, 2 (June 1994), 101–19.

[59] LAWTON, W., AND RESNIKOFF, H. *Fractal tiling for multiple mirror optical devices.* United States of America, 1990.

[60] LINDEBERG, T. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[61] LIU, Y., AND HUANG, T. Estimation of rigid body motion using straight line correspondences. *Computer Vision, Graphics and Image Processing 43* (1988), 37–52.

[62] LOWE, D. Organization of smooth image curves at multiple scales. *IEEE* (1988), 558–67.

[63] LYNCH, M., AND DAGLI, C. Stereoscopic neuro-vision for three-dimensional object recognition. *Mathematical and Computer Modelling 21*, 1/2 (1995), 185–215.

[64] MACHTOVOI, I. Artificial-vision stereo system as a source of visual information for preventing the collision of vehicles. *Journal of Optical Technology 61*, 10 (1994), 741–43.

[65] MAHOWALD, M., AND MEAD, C. The silicon retina. *Scientific American* (May 1991), 40–6.

[66] MARESCA, M., LAVIN, M., AND LI, H. Parallel architectures for vision. *Proceedings of the IEEE 76*, 8 (August 1988), 970–81.

[67] MARR, D., AND POGGIO, T. A computational and theory of human stereo vision. *Procedings of the Royal Society B*, 204 (1979), 301–328.

[68] MAY, D. Towards general purpose parallel computers. In *1991 WTUG Conference:Invited Paper* (1991), pp. 2–42.

[69] MEAD, C. Neural hardware for vision. *Engineering and Science* (June 1987), 2–7.

[70] MESSNER, R., AND SZU, H. An image processing architecture for real time generation of scale and rotation invariant patterns. *Computer Vision, Graphics, and Image Processing 31* (1985), 50–66.

[71] MINGYUNE, D., AND WAHL, F. Using space continuity and orientation constraints for range data acquisition. *Pattern Recognition 27*, 8 (1994), 987–1004.

[72] MUELLER, P., BLACKMAN, D., AND FURMAN, R. *An Introduction to Neural and Electronic Networks.* Academic Press, 1990, ch. 7, pp. 131–53.

[73] MURRAY, D., AND BUXTON, B. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence Pami-9*, 2 (March 1987), 220–28.

[74] OBAIDAT, M., AND ABU-SAYMEH, D. A microcomputer-based video-pattern generator for binocular vision test. *IEEE Transactions on Instrumentation and Measurement 43*, 1 (1994), 89–93.

[75] OZANIAN, T. Approaches for stereo matching, a review. *Modeling, Identification and Control 16*, 2 (1995), 65–94.

[76] PAHLAVAN, K., AND EKLUNDH, J. Mechatronics of active vision. *Mechatronics 4*, 2 (1994), 113–23.

[77] PALMER, S. *Human and Machine Vision.* Academic Press, 1983, pp. 269–95.

[78] PRAZDNY, K. On the information in optical flows. *Computer Vision, Graphics and Image Processing 22* (1983), 239–59.

[79] RAUTIOLA, K., PEHKONEN, K., STAHLE, L., AND JOKITALO, P. Design of tms320c40 signal processors and programmable logic based prototyping environment of real-time machine vision architectures. *Microprocessing and Microprogramming 38* (1993), 663–68.

[80] ROJER, A., AND SCHWARTZ, E. *Design Cosideration for a Space- Variant Sensor with Complex-Lograrithmic Geometry.* IEEE, 1990.

[81] ROSENFELD, A. Image analysis and computer vision. *Computer Vision and Image Understanding 62*, 1 (1995), 90–131.

[82] RYGOL, M. A parallel 3d vision system. In *Image Processings and Transputers* (1992), H. Webber, Ed., IOS Press, pp. 131–152.

[83] RYGOL, M., MCLAUCHLAN, P., COURTNEY, P., POLLARD, S., PORRILL, J., BROWN, C., AND MAYHEW, J. Parallel 3d vision for vehicle navigation and control. In *Transputing '91* (1991), P. Welch, Ed., IOS Press, pp. 48–67.

[84] SAITO, Y., KOMATSU, S., AND OHZU, H. Scale and rotation invariant real time optical correlator using computer generated hologram. *Optics Communications 47*, 1 (1983), 8–11.

[85] SAKITT, B., AND BARLOW, H. A model for the economical encoding of the visual image in crerebral cortex. *Biological Cybernetics 43* (1982), 97–1083.

[86] SAMET, H. *A Design and Analysis of Spacial Data Structures.* Addison Wesley, 1990.

[87] SANDON, P. A pyramid implementation using a reconfigurable array of processors. *IEEE* (1985), 112–18.

[88] SCHWARTZ, E. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research 20*, 8 (1980), 645–69.

[89] SCHWARTZ, E. A quantitative model of the functional architecture of human striate cortex with application to visual illusion and cortical texture analysis. *Biological Cybernetics 37* (1980), 63–76.

[90] SCHWENK, T. *Sensitive Chaos.* Rudolf Steiner Press, 1965.

[91] SHERIDAN, P., AND HINTZ, T. Spiral pyramid vision. In *Australian Transputer and Parallel Applications Conference.* IOS Press, Melbourne, Australia, November 1992.

[92] SHERIDAN, P., HINTZ, T., AND ANANTHARAJ, N. Design details for a parallel proscsssing product inspection vision system. Tech. rep., University of Technology, Sydney, 1993.

[93] SHERIDAN, P., HINTZ, T., AND NACHIMUTHU, A. The spiral resolution pyramid - optimization in a message passing parallel processing image system. In *Transputer Research and Applications 6* (1993), S. Atkins and A. Wagner, Eds., IOS Press.

[94] SUNWOO, M., AND AGGARWAL, J. *VisTA-An Image-understanding Architectures and Its Programming Environment.* Academic Press, Inc, 1991.

[95] TAN, C., PANG, C., AND MARTIN, W. Transputer implementation of a distributed multiresolution dynamic scene analysis system. Tech. rep., National University of Singapore, June 1992.

[96] TAR, M., AND BLACK, M. A computational and evolutionary perspective on the role of representaion in vision. *CVGIP: Image Understanding 60*, 1 (1994), 65–73.

[97] TAR, M., AND BLACK, M. Purposive reconstruction: A reply to 'a computational and evolutionary perspective on the role of representation in vision'. *CVGIP: Image Understanding 60*, 1 (July 1994), 103–08.

[98] TISTARELLI, M., AND SANDINI, G. Dynamic aspects in active vision. *CVGIP: Image Understanding 56*, 1 (1992), 108–129.

[99] TSAI, R., AND HUANG, T. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence Pami-6*, 1 (January 1984), 13–27.

[100] TSOTSOS, J. Analyzing vision at the complexity level. *Behavioral and Brain Sciences 13* (1990), 423–69.

[101] VLEESCHAUWER, D. D. On the smoothness constraint in the intensity-based estimation of the parallax field. *Multidimensional Systems and Signal Processing 6* (1995), 113–135.

[102] WATSON, A. *Algotecture of visual cortex.* Cambridge: Cambridge University Press, 1990.

[103] WEBBER, H., Ed. *Image Processing and Transputers.* IOS Press, 1992, ch. KITTEN - A Foveal Image Tracker, pp. 97–111.

[104] WELCH, P., AND WOOD, D. Kitten - a foveal image tracker. In *Image Processing and Transputers* (1992), H. Webber, Ed., IOS Press, pp. 97 –111.

[105] WETZEL, D., NIEMANN, H., AND RICHTER, S. An image sequence analysis system for driver assistance. *IEEE* (1994), 150–54.

[106] WILLIAMS, D. Through the photoreceptor mosaic. In *Trends in Neuro Science* (May 1986), vol. 9, pp. 193–8.

[107] WU, C., WANG, D., AND BAJCSY, R. Acquiring 3-d spatial data of a real object. *Computer Vision, Graphics and Image Processing 28* (1984), 126–33.

[108] YASREBI, M. A review of various philosophies on the design and analysis of parallel computer architectures for vision. *The Australian Computer Journal 24*, 3 (November 1992), 139–51.

[109] YATAGAI, T., CHOJI, K., AND SAITO, H. Pattern classification using optical mellin transform and circular photodiode array. *Optics Communications 38*, 3 (August 1981), 162–65.

[110] ZEKI, S. The visual image in mind and brain. *Scientific American* (September 1992), 43–50.