Rupert Young — Ph.D. Dissertation

This cover page added to Rupert's dissertation in order to highlight his introduction to and explanation of Perceptual Control Theory.

Post to CSGnet

[From Dag Forssell (2017.05.31 16.30 PST)]

<snip>

I drove some 1,200 miles round trip in May of 2007 to see Phil Runkel a month before he passed on, and again in September to attended the celebration of his life. I came away with three cartons containing what to me appeared to be the PCT section of his library. It included the 2000 doctoral dissertation of a certain Rupert Young.

Just now I made a point of digging it out and looking at it more closely. What a marvelous work. Beautifully printed and bound, and significantly with a splendid introduction to PCT.

Chapter 2 ***Perception and Behaviour*** brings Perceptual Control Theory into the realm of Artificial Intelligence with notes on pages 15-16 and 18.

Chapter 3 ***Perceptual Control Theory*** (pp 23-37) holds not only a good introduction of PCT, HPCT, and terminology, plus good examples, but also a superior expose of 3.2 ***The Conventional Error***, with 6 progressive diagrams and a thorough discussion.

Chapter 4 ***Basic Perceptual Control Systems*** (pp 39-52) provides a rather detailed overview of the math, slowing factors, transport lag and such; all with numerous color charts.

Chapter 11 ***Conclusions and Future Work*** (pp 141-147) begins with a discussion of perspectives.

Runkel underlined part of the paragraph that reads:

> *All* of the patterns and transitions which can be observed in the Game of Life arise from these three simple rules. None of the behaviours are specifically implemented but *emerge* from the above rules. Here, then, lies the danger of taking the observer's perspective.

Red pen underline typical of Runkel. Always meticulous, he added the apostrophe ☺.

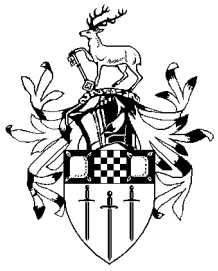
Runkel also typed one of his typical notes on a 2x2 inch paper and taped it next to Fig 11.1: ***The Game of Life***

> "Showing that many complex patterns can be produced from 3 simple rules. How easy it is, in observing events, to look for lots of rules—to mistake apparent complexity for underlying complexity."

Visual Control in Natural and Artificial Systems

Rupert Young

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey



Centre for Vision, Speech and Signal Processing
School of Electronic Engineering, Information technology and Mathematics
University of Surrey
Guildford, Surrey GU2 5XH, U.K.

January 2000

Summary

The desire to produce artificial vision systems which behave in an intelligent, human-like way or which can autonomously and automatically perform tasks currently only performed by humans has been a goal of Artificial Intelligence research for many decades.

Until recently much of the research concentrated on extracting visual representations of objects from single, static scenes. The last decade has seen an increase in interest concerning mobile robotics for navigation, planning and autonomous control as well as for the interpretation of events in real, dynamic scenes.

Presented in this thesis is research on artificial vision systems from two different, but both necessary, standpoints. One concerns low-level vision-based behaviour of object tracking based upon a naturalistic theory of perception and behaviour within living systems. The other takes a more application and engineering based approach and its goal is to address high-level scene interpretation and control of processing resources.

Numerous experiments are presented to demonstrate the various issues. The two main experiments, corresponding to the two research streams, are a system which is able to fixate complex multi-coloured objects and a fully integrated vision system for predicting and following, with a mobile sensor, events in a dynamic scene.

Key words: Perception, Behaviour, Control Theory, Fovea, Active Vision, Visual Integration, Calibration, Scene Evolution, Grammatical Modelling.

Email: R.Young@surrey.ac.uk

WWW: http://www.ee.surrey.ac.uk/cgi-bin/R.Young/index.html

Acknowledgements

The research presented in this thesis would not have been possible without the supper of many people and the provision of many excellent facilities. I would, therefore, like to thank:

- My supervisors Prof. John Illingworth and Prof. Josef Kittler for their constant support and guidance over the last few years.
- George Matas for his patience during the many detailed, technical discussions about the work as well as for his significant intellectual input into the research.
- Charles Galambos for being a major source of education for the subtleties and nuances of programming in C and C++.
- Prof. Josef Kittler, again, for building an excellent research centre with the latest in software and hardware technology without which much of the development would have been impossible.
- Bill Powers and the Control System Group for the endless discussions on the nature of Perception and Behaviour within living systems.
- Radek Marik, George Matas, Charles Galambos and the many others who developed and contributed to the AMMA and GAMA software libraries which were an invaluable resource.
- Dekun Yang and Alan Soh for, respectively, supplying the modules for Cylindrical Object Recognition and Calibration Chart Detection.
- EPSRC and CVSSP for generously funding me for the duration of this research work.
- Sanjay Pandit for his friendship and calmness at all times and Kenneth Jonsson for the beer drinking partnership at those times when it was necessary to take our minds off work.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Setting the scene

1.1 Motivation

What makes us tick ? This question has engrossed mankind for thousands of years. The mechanics of solely (supposedly) human concepts such as intelligence, autonomy, thought, consciousness, perception and behaviour have been endlessly discussed and dissected. Behind such curiosity has been not only the desire more deeply to understand ourselves, but also to create artificial systems which replicate Man. However, not until this century has technology reached such a level that the attempt to build systems with Artificial Intelligence (AI) can be taken seriously.

In this thesis we report our research that takes advantage of the enormous advances in computer technology over recent decades to further the investigation into the nature of living systems and one possible architecture for their artificial counterparts. There are myriad starting points for such research of which we have chosen visual perception and behaviour. Vision is the dominant sense in many animals, and certainly in humans. To our mind vision is the most interesting sense as there is a wealth of applications in which artificial vision systems could take over, enhance or even improve upon the tasks currently performed by humans. In terms of artificial systems, incorporating vision is sensible and desirable not only due to the multitude of tasks which it would allow, but also that a vision-less system would barely resemble major human-like abilities, given that it is the major way by which we learn about, navigate around and interact with the world.

Interaction is a key concept when dealing with any intelligent system, whether it be natural or artificial. No living system has a passive one-way relationship with the world, it *behaves*, whether that be holding a conversation, swimming through the

mud or setting down roots. Senses without action are pointless, which bring us to the other side of the perception coin which is the concern of this research, *behaviour*, specifically behaviour associated with visual perception and visual systems.

1.2 Objectives

Two quite distinct areas of research are presented in this thesis. Although both are concerned with vision and action, the philosophical foundations, guiding principles, objectives and methodologies are quite different. One is *Perceptual Control Theory* (PCT) [87, 88, 91], a radical theory of the nature of perception and behaviour in living systems. The other is a more conventional computer vision approach, hereafter referred to as the *Vision As Process* (VAP) [27] project.

The aim of the PCT research presented in this thesis is two-fold. In general terms the goal is to present the concept of *control of input* and how it applies to a real system and, more specifically, to develop a visual system which is able to control its fixation (viewpoint) relative to complex objects.

Distinguishing an object in a real scene is no trivial matter. Different parts of a scene may have similar, and therefore distracting, attributes as that of the target. For example, an object may have the same colours as its background, the question then becomes where does the object finish and the background start, or different objects may have the same colour. In figure 1.1 the coloured faces have many colours in common both with each other and the background. The problem can be alleviated, to some extent, if different perceptual dimensions are considered. If it is possible to detect shape or motion characteristics, say, specific to the target it will be more easily distinguished. The majority of previous work on fixation and tracking in dynamic views has mainly concentrated on single perceptual dimensions where the target is easily distinguished [50, 81, 107] in complex views or simple views where the target has no distractions.

One way to move this research forward could be to add extra perceptual dimensions to the system. However, we favour a different approach involving higher-level characteristics of a single dimension. Neurophysiological evidence [53, 29, 113] suggests that the brain is structured in just such a way, with higher areas representing increasingly abstract aspects of the environment. Part II describes some preliminary work drawing on the principles and methodology of PCT.

The objective of the VAP project is to build artificial vision systems which are able to interpret what is happening in a dynamic, real-world scene and act accordingly. The

Figure 1.1: Multi-coloured objects

way this is envisaged to be accomplished is to integrate diverse perceptual modules within a mobile robotic framework working on the principles of attentional control of processing and active parameters with the purpose of achieving computational efficacy.

The majority of Computer Vision research has concentrated on single visual modules or techniques in isolation. The VAP project represents one of the first attempts to bring together and integrate the diverse elements necessary for a complete vision system.

The recent research effort in computer vision, under the acronym VAP (vision as process) [27], clearly demonstrated the benefit of enhancing the commonly advocated active vision paradigm [1, 2, 5, 41] by the concept of continuous processing which facilitates the exploitation of temporal context to make the scene interpretation problem manageable. Accordingly, the vision system is not only able to control the camera to focus on regions of interest or to adopt a new view point to simplify a scene interpretation task, but most importantly, it is processing the input visual data on a continuous basis. The latter has the advantage that the degree of knowledge about the imaged scene (identity, location and pose of objects in the scene) is continuously maintained and this in turn simplifies the complexity of future visual tasks. This idea mimics the ability of the human vision system to build a model of the surrounding environment and use this model to generate visual expectations (even with closed eyes). In terms of machine perception, the capability to exploit temporal context

translates into the requirement to build a symbolic scene model which is utilised in solving instantaneous visual tasks and is continuously updated.

We have developed a vision system which adheres to the VAP philosophy [57]. The scene is regularly sampled by visual sensing and the outcome of processing directed towards a particular visual task is entered into a scene model database. The database contains symbolic information about the types of objects detected, their position in the 3D world coordinate system defined for the environment in which the sensor operates, and their pose. The system has been shown to exploit multiple cues to generate object hypotheses and to verify the content of the scene model. It also has the capability to interpret dynamic scenes. In particular, it has been demonstrated how the combined use of temporal context and a grammatical scene evolution model enhance the processing efficiency of the vision system [70].

In the above studies the camera of the vision system acted as a static observer. We have recently extended the system capability by placing the camera on a robot arm with a view to performing scene model acquisition and maintenance experiments with an active observer. As the ego-motion of the active observer is known to sufficient accuracy, it should be possible to verify the presence of objects in the scene model database from any view point. However, a mobile camera raises the issue of accuracy and stability of calibration. If calibration is inaccurate, the prediction of the appearance of objects in the scene model will be rendered useless for efficient comparison with the observed data.

In this thesis we investigate the influence of calibration errors on scene interpretation and scene model maintenance using an active observer. We show that a single view calibration does not yield calibration parameters which are sufficiently accurate. This leads to inaccurate estimates of object positions. Moreover, the positional estimates cannot be improved by viewing an object from several viewpoints, as the location estimates are biased towards the initial calibration viewpoint. We show that the problem can be effectively overcome by means of a multi-view calibration process. This avoids over-fitting the camera/grabber chain calibration parameters and facilitates reliable and computationally efficient scene model maintenance.

We then turn to the main goals of this research, to control the processing and view point of the camera in relation to a dynamic real-life scene. The problems being that the exhaustive search of a model database is impractical in terms of efficiency and random changes in viewpoint will not adequately track the action. The solution involves incorporating a representation of the evolution of scene events in terms of *grammatical* models. We investigate the effects of using such models, to hypothesise

Figure 1.2: Breakfast table scenario

objects and viewpoints, on the computational processing resources of the system.

The work described in part III successfully integrates modules for the detection of regions of interest, object recognition, camera calibration, mobile robot/camera control and scene evolution models into an active vision system capable of controlling the nature of processing and the choice of viewpoint with respect to dynamic, real world scenes. The specific type of scenario used is that of a breakfast table scene (see figure 1.2) where objects are placed in real-time and the action is followed and interpreted.

1.3 Contribution

The original contribution of this thesis falls into the following areas,

- The common tendency in Computer Vision has been to encode objects in terms of their geometric properties. We describe a generic encoding scheme for complex and multi-featured objects which relies upon the *number* of features present, not their position. The histogram of features represents the target from a particular viewpoint.

- A fixation measure, a comparison of the histogram of the current view with that of the target, was developed which gives an indication of the validity of the current view.

- The encoding scheme was extended further to include higher levels of abstraction. The higher level within this hierarchy encodes features which are highly specific to the target allowing successful segmentation from distracting elements in a cluttered scene.

- Segmentation of the target, based upon the encoding scheme, provided an even more useful fixation measure which results in values for the direction and distance to the target.

- A visual fixation and tracking system based upon Perceptual Control Theory is described. As opposed to conventional approaches the system controls its *input* (the fixation offset) by varying the direction and *velocity* of its tracking output.

- The use of the compact foveal representation of the scene is combined with the tracking control system and the fixation measure to produce a fast, real-time tracker.

- We demonstrate experimentally the unreliability of using camera calibration data derived from a single view for different camera viewpoints. We also present a resolution with a procedure for calibrating a mobile camera. The method uses images taken at a finite set of positions covering the working environment. The resulting calibration data is valid anywhere within that area.

- Thresholds for target object match values were determined experimentally. These thresholds allowed the database search and match facility to confirm the identity of an object without searching the entire database.

- Temporal modelling is used to guide the processing of a vision system and the movement of its mobile camera module. This is achieved by modelling the expected evolution of scene events as grammatical rules and facts within a production system.

- The vision system described towards the end of this thesis signifies a major step forward in the paradigm of Computer Vision systems representing as it does one of the first systems of its by integrating diverse visual modules to form a coherent whole as well as implementation as an actual working prototype.

1.4 Outline

In the remainder of part I of this thesis we go back to the roots of the study of intelligent systems by a brief look at the history of modern AI, as well as discussing the different methodologies of computer vision, to put into context the current approaches to research into perception and behaviour. Parts II and III separate the two diverse areas of research reported. Part II concerns *Perceptual Control Theory* (PCT). We start with an introduction to the theory and describe how it provides a possible explanation for all types and levels of the behaviour of living systems. Chapter 4 describes some simple experiments which show the behaviour of a single basic control system. We digress slightly in chapter 5 to explain the structure and use of the foveal representation of the visual scene which is used later in the fixation experiments. The last chapter in Part II brings together the control systems of PCT, the techniques of foveal fixation and simple colour segmentation to realise the functionality of automatic visual fixation to multi-coloured objects.

The research in Part III takes a more conventional approach to artificial systems. Chapter 7 outlines the aims of the recent *Vision as Process* (VAP) project along with a description of the corresponding architectural design for an integrated vision system adhered to by this research. One of the major obstacles to building a successful mobile vision system is determining the spatial relationship between the camera, which will be at many different positions, and the external world. This process, camera calibration, and the issues involved is discussed in chapter 8, along with our technique for overcoming the multi-view problem. The results of our camera calibration method are made use of in chapter 9 to show how the location of an object, predicted by a moving camera, is maintained accurately with respect to its actual position. The last chapter of Part III combines the preceding concepts and techniques and extends the visual system to include control of the camera viewpoint and the computational processing resources by means of scene evolution models of dynamic scenes.

We conclude in Part IV with a general discussion of research perspectives and a summary of the two threads of research as well as laying out some recommendations for future work.

Chapter 2

Perception and Behaviour

It is wise, and courteous to the reader, to define terms prior to a discussion. A problem arises, however, with respect to *Artificial Intelligence* given that *intelligence* itself is a somewhat vague and high-level concept and the knowledge of the foundations and the processes which give rise to intelligence are largely unclear. Therefore we will leave to the philosophers the actual meanings of the concepts and instead describe the types of approaches which have been taken in the field, with particular reference to perception and behaviour.

We start with a brief look at the history of AI and describe the different ways people have thought about what constitutes intelligent agents and how those views have changed over the years. In the second part of the chapter we apply a similar discussion to the specifics of vision, concentrating on the advantages (and necessity) conferred on visual systems by being able to *interact* with their environment.

2.1 A Brief History of Artificial Intelligence

Although a number of different disciplines, such as psychology, philosophy, biology and neuroscience, have been around for many years with the aim of studying intelligence, there are two main factors which not only gave birth to modern AI but also shaped its research goals and methods over the next few decades from its inception in the middle part of this century. Those two factors were the advent of the modern computer and the work of the mathematician, Alan Turing, who was widely recognised as the father of AI.

During the second world war Turing contributed significantly to the successful attempt by the Allies to crack the German "Enigma" code. The problem was overcome

by applying the necessary mathematical principles to a mechanical device which was able to carry out the vast amount of computations involved in a fraction of the time it would have taken a human. It was perhaps inevitable, given a seemingly "intelligent" operation, that great interest arose, establishing a proper link between mathematics, computers and intelligence. Spurned on by Turing's later ideas [115] the field of AI was born, distinct from the conventional life sciences, as a discipline concerned with tackling what were, essentially, mathematical problems with some relation to human intelligence. The types of problems tackled were those which, it was thought, required high-level cognitive skills: those involving reasoning, problem solving, planning and logic, problems such as chess, the travelling salesman, the Towers of Hanoi, expert system design and natural language processing [63, 96]. It is no coincidence, given the mathematical background of the main participants, that the problems tackled were, on the whole, well-defined mathematically and somewhat remote from the real-life behaviour of living systems.

These early approaches followed a particular philosophical attitude towards human cognitive processes. This viewpoint was known as the *Physical Symbol Systems Hypothesis* which stated that the

> necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system [78].

The implication was that any system exhibiting general intelligence was a physical symbol system (a symbol manipulator), and that any physical symbol system (e.g. digital computers) could (given the right configuration) exhibit general intelligence.

One of the main techniques to come out of AI research was that of *Search*, which involved ways of finding the optimal solution from a large number of possibilities. For example, with chess it involved looking many moves ahead in order to determine which current move would provide the best route to winning the game. Similar computational approaches were taken with early, and subsequent, research into artificial vision. Although these programs have become quite sophisticated in recent years (chess programs have reached the level of Grand Masters) they gave no insight into the nature of human intelligence. The methods and processes used by the programs to solve the problems were not the same as those employed by humans.

Computer Vision did not take off until the early 1980's with the advances in computer processing power and the work of people such as David Marr. In his seminal work, *Vision* [67], he outlined a computational approach and architecture for vision

systems, though concentrating on how information, such as texture and shape, can be derived from single, static images.

Artificial Intelligence never fulfilled its early promise, and so, due to the resounding failure of the traditional approaches, in the late 1980's and early 1990's a number of researchers moved away from looking at high-level human cognition to more humble (though certainly not more simple) problems faced by more simple agents when navigating their environment. This branch of AI, known as *Artificial Life*, took a bottom-up approach and began to look at such things as visuomotor control in insects [36], simple navigational behaviours [10, 14, 31, 68] and ways of *evolving* simple artificial creatures [25].

In contrast to the detached symbol manipulating systems of traditional AI these new branches were inspired by biological living systems and were primarily concerned with the interaction between the system and the environment. Harnad [46] had suggested that mere symbols were meaningless without a grounding to the entities they were supposed to represent. These new lines of research provided that grounding by furnishing the systems with sensory abilities in order to perceive the world. Even with symbol grounding the Physical Symbol System Hypothesis was no longer sufficient, or adequate, to describe or explain the interactions or dynamics of the new approaches. The new philosophical viewpoint to arise came to be known as the *Dynamical Systems Hypothesis* (DSH) [103, 117, 118, 119] which concerns the continuous change (of variables) within the environment and interacting agents.

Meanwhile, throughout the decades from the 50's to the present day a quite different theory of perception and behaviour within living systems was being developed independent of mainstream Cognitive Science, a theory which did not fit easily with the conventional stances of the life sciences or AI. PCT [87] explains the functional architecture and basic mechanism of the nervous system as *control of input*. What this means is that living systems are constantly *acting* in order to bring *inputs* (perceptions) in line with desired values (goals). In other words, *behaviour* controls *perceptions*, or *output* controls *input*. For example, the diameter of the iris in the eye changes in order to achieve the *desired* amount of light falling on the retina, or we act (stealing or not) in such a way as to achieve (or maintain) our perception of honesty. This concept is in stark contrast to the usual view of a cause-effect relationship from inputs to outputs (stimulus-response) or the control of behaviour.

The control systems described by PCT are undoubtedly dynamical in nature, however it would be an injustice to describe PCT as *merely* a Dynamical Systems Theory. Invoking the DSH as a system model is only *descriptive* whereas PCT and its as-

sociated language are *explanatory*. For example, both weather systems and living systems are dynamical in nature. Dynamical Systems Theory may well *describe*, from the viewpoint of the observer, what is going on in both systems but does not *explain* the underlying nature and mechanisms from which that observed behaviour arises. The difference between the two is that living systems are actively controlling variables to keep them in certain states instead of being at the whim of natural forces.

2.2 Vision

The predominant theme running through the Machine Vision arm of AI has been that of *reconstruction*. The idea was to analyse single, static images with the purpose of constructing three dimensional models of whatever objects were present in the scene. Marr, one of the most influential proponents in this area, regarded vision as

> the process of discovering from images what is present in the world and where it is

and so is,

> first and foremost an information processing task. [67](p. 3).

A task that creates internal representations of the world from retinal images. Marr followed the policy of least commitment which states that it is necessary fully to process all visual stimuli in case the resulting information might be required.

There seems little reason to believe that such a reconstructive process is the basis on which animal or human vision is founded [5, 122], especially considering the arguments suggesting the intractability of the approach [108, 123]. Of course, there may be applications where 3-D reconstruction is desired or suitable, but the lack of such a strategy in a visual system should not be, a priori, any indication of failure.

The non-traditional approach to vision advocates the policy of most commitment [5, 75] and emphasises the advantages of active or animate vision. The policy of most commitment specifies that it is only necessary to process what is required for the problem at hand. Most of the time it is not necessary to build complex representations, as objects can be discriminated by a relatively few distinguishing features [17]. If a particular viewpoint is not optimal for discrimination, attentional action based upon selected target characteristics enables an animate vision system

to move to a more suitable position. Such action basically carries out what would be computationally expensive model transformations with the passive, reconstructive method [108]. Action also allows a system to gather information that is actually a reality instead of trying, perhaps incorrectly, to fit an internal model to a distorted view. In effect, an animate vision system can use the world as its own model [9].

Nelson [76] suggests that vision is more suited to *recognition*, identifying relevant situations, than *reconstruction*, the general transformation of information between different forms. The reconstructivist approach processes information without any regard for its relevance for the animal, whereas recognition is only concerned with specific problems of interest.

In the natural world there are many different types of visual system, from simple *pit* eyes of a few receptors to human eyes with a cornea and lens and thousands of receptors [60]. Eyes may only be of use to distinguish light and dark areas, if that is all the animal needs to know. In fact there is often a close correspondence between the physiology of the eye and the ecological niche of the animal. For example, the higher density ganglion cells of animals in flat, open environments are formed in "visual streaks" corresponding to the horizon, whereas the density in arboreal species is radially symmetrical [60]. This would indicate that the function of visual systems arose out of a reaction to *relevant* aspects of the environment and not as a method of identifying *all* aspects.

Animals need to be active in order to survive [39, 76]. They must be able to crawl, swim, run or fly around in order to find food and avoid predators. Also, they must be able to discriminate between safe and dangerous aspects of the environment, which is what the senses, vision included, are used for.

The purpose of visual systems is governed by the environment and is not simply a means of recording it. In other words,

> biological vision evolved to permit animals to act within, and react to their environment [75](p. 5)

and,

> The satisfaction of getting things right is not much compensation for being eaten because you took too long to decide what was rushing towards you [102](p. 21).

Furthermore, to avoid redundant and time consuming (and hence potentially fatal to the animal) processing,

> The economical way to organise a nervous system is for perceptual mechanisms to detect whatever information is needed to guide the animal's activities, and no more [17](p. 250).

The consequence is

> that vision is more readily understood in the context of visual *behaviors* that the system is engaged in, and that these behaviours may not require elaborate categorical representations of the 3-D world [5](p. 1635).

The way that areas of interest are found and investigated is by actively controlling the parameters of visual sensors of a system to enable it to interact in real-time with a complex, dynamic world; *Active Vision* [8, 108].

2.3 Summary

To clarify the terminology, and to put into context the research topics of this thesis, here are the broad definitions of the categories of the machine and natural vision found in the literature.

Active sensing The reconstruction of shapes and surfaces by *actively* projecting an infra-red signal onto an object. The signal, usually in a grid, is extracted from images and, by analysing the pattern of distortion, the surfaces are reconstructed. Active sensing is not the topic of this thesis but is noted here as it is sometimes called Active Vision.

Reconstructive Vision Constructing 3-dimensional geometric models of objects from single or multiple images by extracting primitive visual elements such as edges and outlines.

Active Vision Machine Vision involving the control of parameters such as, position, orientation, focus, zoom and aperture of a mobile camera system.

Animate Vision Concerns visual behaviours that a system is engaged in while it is interacting with its environment.

Perceptual Control Although an all-encompassing theory of living systems, in terms of vision it concerns how actions control desired visual perceptions, or goals.

These categories are by no means mutually exclusive and most research involves overlaps between different areas. Part II of this thesis falls into the last category, though it does incorporate some more traditional feature detection techniques. The significant aspects of this theme of research is that it is concerned with low-level behaviours of living and environmental systems which are dynamic in nature and often unpredictable, precluding the more traditional approaches of predictive modelling or symbol manipulation. Concerned as it is with low-level visual control this area of research could be seen as a potential module for the system which is the subject of Part III of this thesis. The roots of Part III are to be found in the more traditional Active Machine Vision paradigm with some leanings towards reconstructive vision. The goal is to take a high-level view of the problem and to integrate diverse modules displaying different visual behaviours in order to control parameters of both the internal processing and position of the system.

The overriding principle of both areas of research is that perception performed in isolation of action results in systems of very limited usefulness, or even, in terms of natural systems, a grossly inaccurate portrayal of the intimately entwined nature of perception and behaviour.

Part II

Perceptual Control Systems

Chapter 3

Perceptual Control Theory

3.1 Introduction

Perceptual Control Theory (PCT) [65, 87, 88, 91] offers a radically different way of viewing the operation and functionality of living systems in contrast to the view conventionally held in Cognitive Science. The traditional way of looking at how living systems, including humans, work is that the output of the system is controlled [11, 15, 22, 33]. That is, the organism computes, or determines, *specific* actions necessary to carry out a task from its perception of the current state of the world. In other words, perceptions control actions, inputs control outputs or stimuli control responses. To put it another way, there is a direct relationship from the inputs of the system to its outputs.

PCT, on the other hand, claims that what living systems are *actually* doing is *varying* their outputs to control their inputs. *Specific* actions are not computed, but rather action is varied in order to bring about a desired input (perception) or goal [24, 26].

At first glance the differences between these two philosophical stances may seem merely semantic. However, as we shall see, the adoption of one viewpoint or the other leads to completely different (and perhaps incompatible) notions, methodologies and goals for both our understanding of natural living systems and the design of artificial systems.

In this chapter we explain the ideas and concepts behind PCT contrasting with those of conventional Control Theory and Cognitive Science as an introduction to the next few chapters which describe some preliminary research into vision systems based upon PCT.

3.2 Some Examples

For the benefit of the reader not familiar with PCT we first describe a few everyday examples of PCT in action.

A graphic example of PCT is manifest in the everyday activity of driving a car. The most important aspect of driving is to keep the car in the correct lane in order to avoid accidents. In PCT terms this is achieved simply by adjusting the steering wheel (output) according to a perceived position (input) of the car. The amount the wheel is turned does not need to be specifically calculated, the wheel is simply turned until the car is perceived in the appropriate position. This method will effortlessly overcome any disturbances to the system whether they be due to internal changes or external influences, such as a wind acting upon the car. Contrast this with a non-control system, which could be called a *measure and model* approach. As with PCT it is necessary to represent in some way how far away the car is from a desired position, but it is now also necessary to relate that error to a specific amount that the steering wheel needs to be turned in order to bring the car back on course. Obviously, this requires precise calibration of the relationship between not only the steering and driving wheels, but also between the driver and the steering wheel. A wind force would influence the car's position so this would also need to be measured and its effects modelled and compensated. Many other factors can also affect how the car behaves such as tyre pressure, road surface and engine condition, all of which would need to be measured and modelled in an open-loop approach that requires the computation of a specific output. PCT greatly simplifies the situation by *controlling* the *input* and dispensing with the need for measuring and modelling the environment.

The operation of the iris in the eye is traditionally seen as a stimulus-response system. The change in lighting conditions (stimulus) in the environment induces a response, a corresponding change in the size of the iris. The PCT view would say that there is a certain amount of light falling on the retina which is desired by the brain. The size of the iris is then varied in order to maintain that desired amount of light. Although the external lighting conditions may be changing, the perceived amount of light (the input) remains constant due to the changing output (iris size). At no point is there a correspondence between the lighting conditions and the size of the retina. For example, imagine the case where the external lighting conditions remain constant but the desired input changes. In this situation the iris size (response) will change even though there is no "stimulus". Alternatively, there could be a situation where both the external lighting conditions and the desired

amount of light are changing in the same direction and are balancing each other out. In which case the size of the iris would remain constant. In other words there is a stimulus, but no "response"!

As a final example, consider the simple task of picking up a cup of coffee. The traditional approach to this task (as reflected in the robotics research [45, 38]) is to compute the specific muscle tensions required to place the hand in the right position to pick up the cup. The next step is to compute the changes required in those tensions to move the cup through the correct trajectory to the lips. Computationally, this is a horrendous problem involving specific variable values for thousands of muscle fibres. If not exactly right the cup could shoot off in any direction. Further complicating matters is the fact that each time you take a sip the weight of the cup is *different* requiring a *different* set of values for all the variables. To get this right it would be necessary constantly to measure the weight of the cup of coffee!

From the PCT viewpoint it is not necessary either to know the weight of the cup or to compute specific values of the muscle fibres. The tensions in the muscles are simply varied so that the cup, or hand, is in the desired position. If the cup were going in the wrong direction the tensions would be changed (not by specific amounts) until the right direction was achieved. The same applies with the changing but unknown weight of the cup. The outputs are *varied* in order to maintain the desired feel or view (perceptions) of the cup's position and trajectory.

3.3 PCT Terminology

The basic architecture of a Perceptual Control System is shown in figure 3.1. There are four main signals involved. The perceptual signal, p, is the input to the system and is the variable which is controlled. It arises as a result of some input function which converts external variables to an internal signal. In the case of the eye example it is the amount of light *detected* on the retina. The input function comprises the operation of retinal cells which convert the light on the retina to internal signals. The reference is the *desired* value for the perceptual signal. The error signal is the difference between the reference and perceptual signals and provides the motivation for a control action of the system. The disturbance signal (the only one outside the system) represents elements in the environment which affect the perceptual signal, i.e. the changing light conditions in the eye example or the wind in the driving example.

There are two other key functions which are characteristic of a Perceptual Control

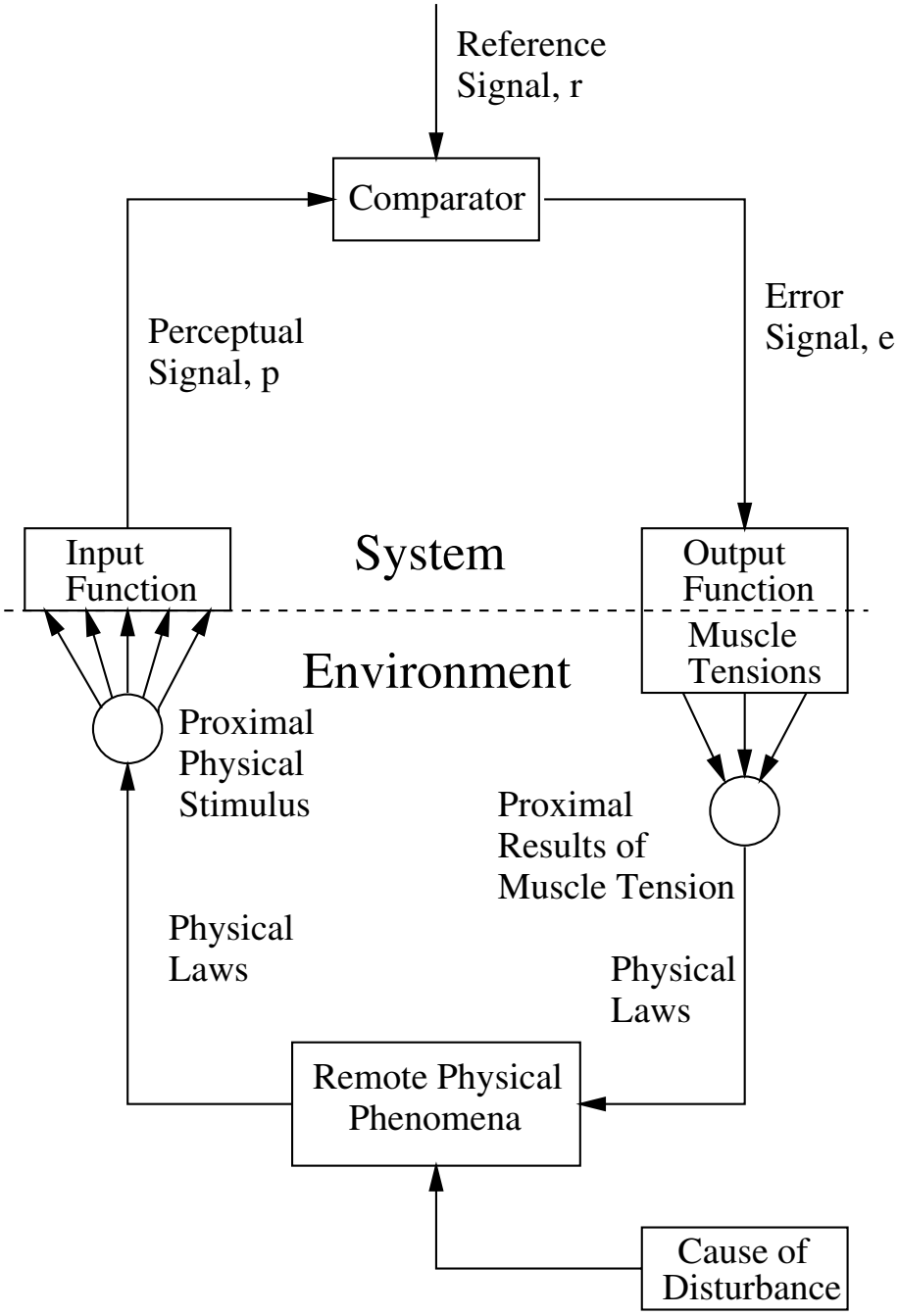


Figure 3.1: PCT Model

[General model of a feedback control system and its local environment. (Reproduced with permission from Powers [87], p. 61)]

System apart from the input function already mentioned. They are the comparator and the output function. The comparator basically subtracts the perceptual signal from the reference signal to give the error signal, which is some representation of how far we are from the goal. The output function transforms the error signal into some actions or outputs of the system. In the eye example, the output would be to change the opening of the iris to increase or decrease its size depending upon the sign of the error signal. However, it is not necessary to compute some specific amount for which to change the size of the iris, but as the perceptual control system is a continuous feedback system the output would change the size of the iris until the error signal becomes zero. The essence of a perceptual control system is that it is a continuous negative feedback system counteracting any disturbance to the input signal.

3.4 The Conventional Error

Feedback control systems which regulate a variable in the face of unpredictable disturbances are nothing new. Two hundred years ago James Watt invented the centrifugal governor, the initial application being to control the speed of a steam engine. Since then control systems have been applied in countless situations and can be studied under the discipline of *control engineering* [3, 37, 111]. There is

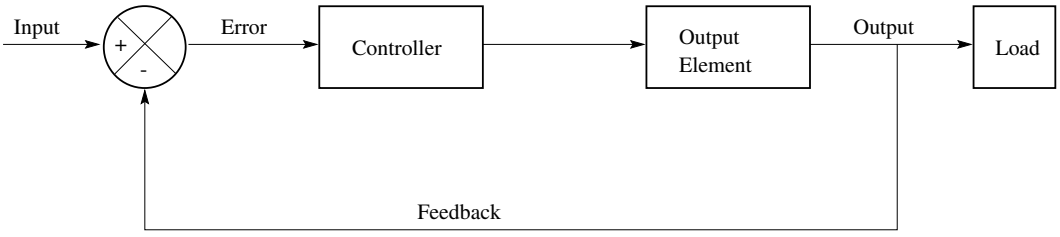


Figure 3.2: Standard Modern Control Theory model

also nothing new about the application of control theory to the behaviour of living systems. What *is* new, however, is *how* control theory is interpreted in the realm of living systems. As we shall see a gross error is made in the conventional life sciences when applying control theory to living systems resulting in a functional and architectural view of living systems which is both unnecessarily complex and incompatible with the actual function of living control systems.

Let us first go back to the basic control system as shown in figure 3.2. The purpose

of this system is to maintain the output variable at a particular value corresponding to the input or goal.

A more concrete example is shown in figure 3.3, with a system for controlling the speed of an electric motor. A sensor measures the actual speed of the motor which is compared with the desired speed, in the error detector (comparator). The controller either increases or decreases the voltage applied to the motor according to the sign of the error (difference in speeds) and by an amount proportional to the size of the error. For example, if the actual speed is less than the desired speed the voltage is

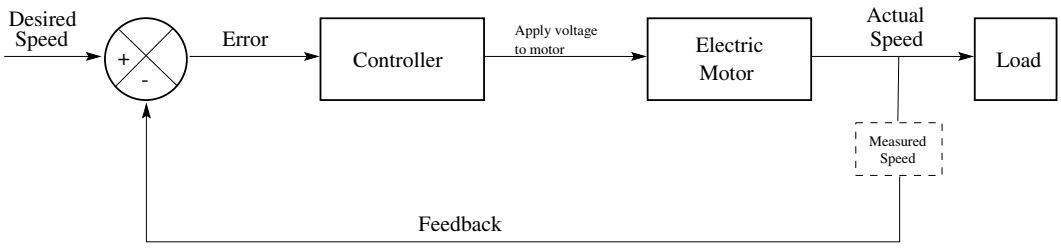


Figure 3.3: Modern Control Theory model of electric motor

increased until the error is zero and the desired speed is reached. These processes are performed continuously and simultaneously until the desired speed is reached. So, in the terminology used in the diagram the control system is controlling the *output* of the motor.

Given the unpredictable nature of the world it would seem natural and logical that living systems utilise the principles of control systems. Now let us try to redraw the control diagram in terms of living systems (figure 3.4). As we all "know" the inputs and outputs of living systems are, respectively, their perceptions (stimuli) and motor actions (responses). It naturally follows that if we apply control theory

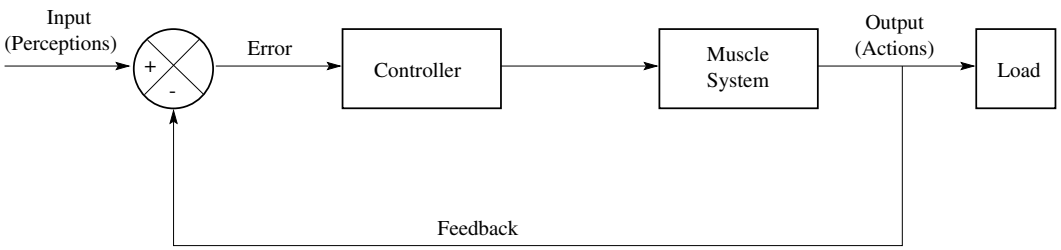


Figure 3.4: Modern Control Theory model for living systems

to living systems then what living control systems do is to control their actions, or outputs. However, let us examine the diagram more closely. At the comparator we are trying to compare two signals, perceptions and actions, which are of an entirely *different* nature. How, for example, do you compare the visual information you are receiving about the position of you hand relative to a tea cup with the values of muscle fibre tensions in your hand ? Supposedly, we could introduce a transducer to convert from one signal to another, as can be seen in the iris control system of figure 3.5.

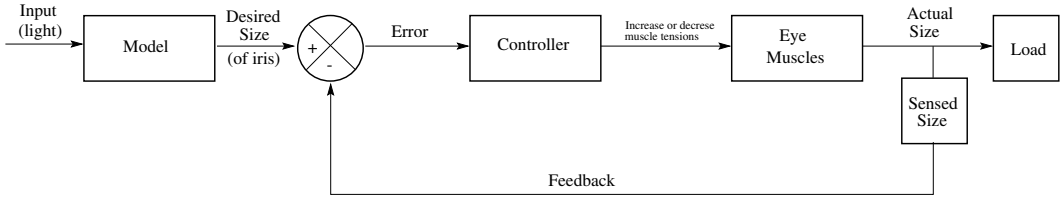


Figure 3.5: Modern Control Theory model applied to iris control system

In this case we have a model which relates the incoming light to an appropriate desired size of iris which can then be appropriately compared with the actual size and then suitably controlled. At first glance this may seem reasonable, however there are a number of serious flaws with this setup. First, a complex model is required in order to derive the correct size of iris from the incoming light. Second, there must be some way of sensing the current state of the output, the iris size. Finally, and by no means the least, this system is not directly controlling the actual goal which is the amount of light falling on the retina. If there are any imperfections in the model then the amount of light falling on the retina will be incorrect even though the size of the iris is right.

However, these problems are merely symptoms of a much more basic and mundane flaw: the terms used in control theory and conventional life sciences *do not* refer to the same variables [88]. In other words, the "input" and "output" of control theory are not compatible with the terms "input" and "output" as they are used and understood in the life sciences. If we look again at the first control diagram (figure 3.2) we see that the input is the *goal* value of the output and the value which is fed back from the output is the *sensed* value of the output. The actions are actually between the controller and the output element (see the application of voltage in the motor example). So let us relabel the diagram with less ambiguous terminology, as shown in figure 3.6 and then using this terminology, re-apply control

theory to living systems in the context of the eye example, as detailed in figure 3.7.

Figure 3.6 shows a control system with the goal variable being compared to the actual or sensed value. A physical element is changed, according to the error,

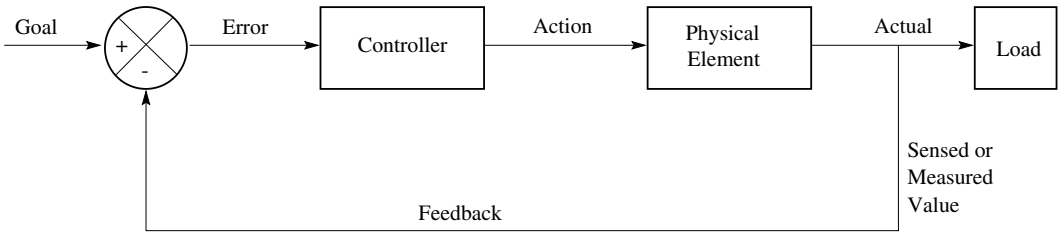


Figure 3.6: Modern Control Theory model with general annotations

which affects the variable being measured. Figure 3.7 shows the correct way of applying the elements of goal value, sensed value and action to a perceptual system. What is now being controlled is not the size of the iris, as in figure 3.5, but the sensed variable, the light on the retina. Now it can be seen that there is no need for elaborate models relating inputs to outputs, no need to sense the output and the goal is being directly controlled.

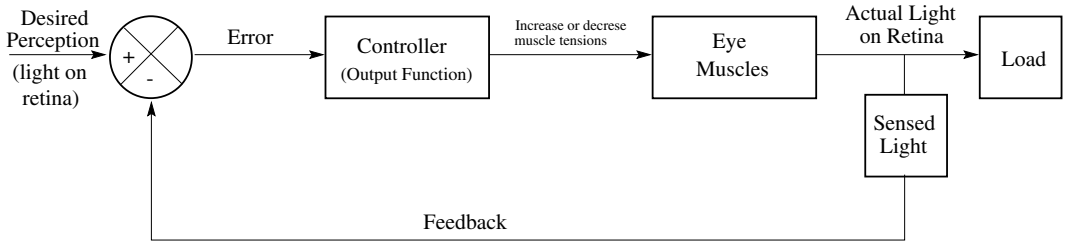


Figure 3.7: Perceptual Control Theory model of iris control system

This error, of misapplying the terminology of control theory to living systems has added support to the enduring view of living systems as stimulus-response systems. PCT offers an alternative view which is both clearer in conception and simpler in operation.

3.5 Hierarchical Perceptual Control Theory

The value of PCT comes not only from the recognition that in the application of control theory to living systems, such systems control their inputs, but also from the acknowledgement that Perceptual Control applies to all levels of the behaviour of living systems. Hierarchical Perceptual Control Theory (HPCT) extends the basic ideas and principles of PCT to add the different levels of behaviour within living systems.

All living systems are made up of a hierarchy of interdependent levels of perceptual control systems. At the lowest level the inputs come from the environment. Higher level systems take their inputs from the inputs of sets of lower level systems and so represent increasing levels of abstraction from the sensed raw environmental variables. The outputs of the higher levels combine to form the reference signals of the lower levels. Using the eye example again, there may be different amounts of light desired on the retina depending upon the task at hand. If you are just sitting around doing nothing in particular, the amount of light falling on your retina may be relatively unimportant and have one particular value. If, however, you are reading a book there is a particular level of light which is most suitable. So, at some higher level there will be a desire to read the book which will output signals for the light which is suitable for the task which sets the reference for the iris control system.

Below we briefly describe each of the levels currently identified within HPCT for humans. It should be noted however, that these levels are not cast in stone and only *roughly* describe the different types of behaviour associated with human beings, which can be associated with distinct levels of control.

Intensity The lowest levels of perception (controlled variables), are those which represent the only level which has the direct experience or interface with the environment, *intensities*. Here we a talking merely about the neural signals that arise when the environment impinges upon our bodies. Those intensities may be the signals when your skin is touched, when light falls on your retina or when sound vibrations reach your ears. The lowest level, then, is the intensity of stimulation of sensory nerve endings.

Sensations Our experience, however, is not simply a matter of the intensities which arise in sensory nerve endings. What we normally refer to as the senses taste, touch, smell, sight and sound are not direct experiences of the world but are collections of intensities of nerve impulses. Colours are perceptions which are

combinations of many different intensities that arise within the retinal nerve cells. Sounds are perceptions arising from collections of many nerve cells within the ear. Sensations depend upon a multitude of signals from the intensity level.

Configurations In turn the next level of perceptual experience, configurations, depend upon collections of sensations. To experience the perception of an object depends upon many different sensations of colours, shadings and textures, for example.

Transitions Our perception of the world does not consist only of static perceptions, but also of changes within, and of, configurations. Motion, for example, depends upon a series of different configurations. The perception of a musical melody depends upon a changing set of configurations of musical notes.

Events The next level of perceptions is events, which are units of perceptual experience which have a beginning, a middle and an end. A spoken word, for example, could be seen as a unit of perceptual experience which consists of a set of transitions between different morphemes of spoken language.

Relationships Events themselves are independent of each other, but how those events relate to each other gives rise to different experiences at the relationship level. A vase *on* a table is a different perceptual experience than a vase *under* a table. The experience of watching a TV depends upon the relative position of the TV and your line of sight. As we go up the levels, perceptions become more and more abstract, in that it gets increasingly difficult to put your finger on something in the world and to say that here is that perception. You can not look at the world and point to the existence of a relationship perception involving yourself and the TV.

Categories Perceptions are things which exist only within ourselves and are not in the external environment. Categories take this a stage further in that they refer not to specific objects within the world but to collections of things that have similar attributes. The category of a chair, for example, does not refer to a specific instance of a chair in the "real world", but to things which have the attribute of being able to sit upon them.

Sequences The order in which we perceive things is also important and gives rise to different experiences. The perception of a dog chasing a cat is not the same as the other way around. The ordering of words in a sentence is vital to our perception and understanding of language.

Programs Whereas a sequence is a straight ordering of perceptions and events, a program involves choice points. Until the program is in action it can not be determined what the result of these choice points will be. They will be dependent upon the actual perceptions experienced. For example, when driving from A to B your driving behaviour will depend upon the state of the traffic lights at which point you have to make a choice, if red "stop", if green "go".

Principles We choose particular programs of behaviour in order to maintain certain concepts of principles, such as honesty or trust. To maintain the perception of ourselves as being honest we behave in such a way, such as not stealing, so as to maintain that perception depending, of course, upon our own definition of honesty.

System Concepts At the highest level sets of principles are brought together to maintain system concepts, such as democracy or law. If our democracy is threatened we protest or go to war.

Some important implications of HPCT are that perceptions depend upon other perceptions and that all of experience is perception. Although there may well be an objective reality "out there" our only direct experience of the "reality" is at the level of intensities. Everything else are internal phenomena of our mind.

3.6 Another example

It may not be immediately apparent how the different levels interact and fit together, therefore in this section we present another example, of the common task of writing a letter, which involves, and requires, the control of variables at many different levels.

The standard view of such a task is that at some point in the brain a command is issued to "write a letter", which is then converted into motor outputs and "hey presto", a letter appears. Hopefully, it is clear to the reader that there are some serious problems with this approach. First, that some very complex computations would be necessary to perform the task. Second, the process could not cope with any unpredictable events which interfered with the task. Finally, if it were the case that we control our outputs we should then be able to perform the task blindfold.

The PCT approach, on the other hand, would say that we start off with a reference perception which represents the state of your perception when there is zero error,

when the goal has been achieved [90]. In this case the reference is more of the form "a letter has been written".

To start with we may be at the program level when we want to define the *type* of letter. So we will have a program saying something like "if business letter then sequence A, if personal letter then sequence B", where the sequences define the different parts of the letter like address positions, date, signature etc. The output would be a sequence that we want to *perceive* at the next (lower) level.

So, to actually write the letter I am sitting at my desk with pen in hand and paper oriented correctly in front of me. I am writing a business letter so the reference set at the sequence level is sequence A (my address top right, date, their address top left, opening, body of letter, signature).

But first I have to position my pen, so I control a relationship between the tip of my pen and a point on the paper a couple of inches in from the right at the top. The perception being the difference, visually, between pen and point. The output is movement of the hand. When they coincide no further action is taken.

The element "my address top right" is also a sequence of distinct events starting with my name. My name is yet another sequence, of letters. Controlling a sequence involves setting the references of lower levels in a certain order. So to write my first name I control the sequence of letters R-u-p-e-r-t, first setting a lower reference of "R". The other letters will not be set as references until the preceding one is complete.

The letter "R" is a reference at the configuration level. To write the letter "R" requires controlling the position of the pen (relationship), the speed of movement of the pen (transition), the feel of how tightly the pen is held (sensation) and the sensed forces of the fingers (intensity). I write the letter "R" in one go starting bottom left. I hold the pen firmly, but softly, applying enough (but not too much) pressure on the paper such that it is easy to move the pen across the paper and that the shade of ink is to my liking. I move the pen across the paper in a swift movement that allows the tip constantly to change position resulting in the letter "R" appearing. The processes continue on in this manner until the entire letter is written.

It is difficult to convey the true state of affairs in the written medium and so it is useful to bear in mind that we are talking about a massively parallel set of massively connected neural control systems such that many different variables at many different levels are being controlled *simultaneously*. So while I am controlling the variable

related to the pressure of the pen on the paper I am also controlling a variable related to the sequence that makes up my address as well as the orientation of my head related to the paper, the dryness/wetness of my throat, the light falling on my retinas, the letter writing program, and probably millions of other variables of which we are totally unaware.

3.7 Learning and Re-organisation

Control systems are not born with the ability to be able to control variables, at all levels. Like anything else the abilities need to be acquired. Initially, according to the nature of the organisation of the nervous systems, the actions you are able to take may be ineffective against the variables you are attempting to control. Furthermore, you may not have the perceptual apparatus in place to be able to perceive things which are affecting you.

Powers [87, 91] postulates that at a fundamental level human control systems continually perform to keep certain *intrinsic* physiological and biochemical variables at particular values. Such variables include body temperature, blood glucose levels and carbon dioxide levels. When these variables are not at their correct values then *intrinsic error* is experienced. Whenever intrinsic error persists re-organisation [87, 85] within the nervous system takes place. That is, connections between nerve cells are altered. If this re-organisation has no positive effect on reducing the intrinsic error it continues. If, however, the effect of the behaviour of the new structure of the nervous system does reduce the error, then any re-organisation is stopped or delayed. In this way the system learns to control.

3.8 Significance of PCT

PCT offers a quite different view of and approach to the study of living systems [92] than that commonly held within the conventional life sciences. From a philosophical perspective there are a number significant advantages to think the PCT way.

Although the universe *appears* extremely complex Science has shown that the underlying principles are simple. In terms of living systems, evolution by its very incremental nature produces beings built upon simple underlying principles, although their behaviour may appear to be very mysterious and complex. In contrast to conventional approaches PCT provides a simple process, the control of input, which is

able, theoretically, to account for behaviour at all levels of experience, no matter how complex it *appears*. The control processes involve only simple "computations", such as subtraction and integration, which are certainly plausible in terms of neural function (see chapter 3 in [87]).

Of great concern to the Cognitive Science community is how the world is *represented* [13, 12, 19, 23, 21, 116] in the brain. GOFAI (Good Old-Fashioned Artificial Intelligence) such as the Physical Symbol Hypothesis [78] favours strong representation where symbols have direct and discrete neural representations in the brain. The problem with this view of representation is how it can occur. PCT is only concerned with control and controlled neural signals. Whether or not they could be said to *represent* anything in the world is incidental, what is important is the ability to control perceptions. In this sense the variables in the brain have no, or weak, representation. For similar reasons PCT does not require geometric or predictive models to be explicitly represented.

One of the most profound and far-reaching implications which arises from the PCT view of living systems is that behaviour is a side effect of perceptual control. Psychological research has concentrated on studying this behaviour with an aim to discovering more about how people work. From the PCT standpoint the standard approach is misguided and the majority of previous psychological research requires, at best, re-evaluation. Marken [66] draws the distinction between the study of perceptual control and the study of observed behaviour by the analogy of the Dancer and the Dance. The dance is the observed side-effect of the dancer's efforts to control many perceptual variables. To study the dance as opposed to the dancer and his controlled variables is to miss the point. Even worse, behaviour may have little to do with the control system but actually reflect an external disturbance applied to the system. For example, in the case of the driver counteracting the effects of wind on the car's position, any steering behaviour is actually reflecting what the wind is doing. Measuring and studying this behaviour will tell you about the disturbance but not about the control system.

3.9 Conclusions

Apart from being a candidate for an all-encompassing theory of Perception and Behaviour within living systems PCT provides a practical way of coping with the unpredictable nature of the world in terms of low-level interaction with the environment. This fits in well with the general aims of the Active Vision paradigm with

which this thesis is concerned. Low-level visual modules based upon PCT could provide an efficient and robust way of maintaining successful interaction between the world and a high-level vision system.

Chapter 4

Basic Perceptual Control Systems

4.1 Introduction

Given the dynamic nature of control systems, attempting to communicate their operation and function through the static medium of the written word is not the ideal method. The reader, therefore, is strongly encouraged to try out for themselves any of the real-time, interactive demonstrations available. A set of PC simulations can be down-loaded from ftp://burkep.libarts.wsu.edu/csg/billdemos/. More readily accessible are some online Java demos at http://home.earthlink.net/~rmarken/demos.html. In both cases the simulation of the tracking task is recommended. Also highly recommended is the control demonstration of the walking behaviour of a six-legged bug, http://www.sys.uea.ac.uk/~jrk/PCT/Archy/Archy.html.

The tracking task neatly demonstrates one of the profound implications of Perceptual Control Systems, that there is no correlation between the inputs and outputs of the systems. In fact the output correlates with the disturbance applied to the input. As already mentioned in section 3.8 the output is more a reflection of the disturbance than of internal cognitive processing.

The tracking task has been discussed in detail elsewhere [48, 88]. In this chapter we present some simulations of basic, single control systems. The purpose is to investigate how the behaviour is affected both by disturbances as well as different values of the control parameters involved.

4.2 The Math

The mathematics which describes perceptual control systems is very simple. As discussed in chapter 3 and presented in figure 3.1 the error, e, is the difference between the actual and desired input perception,

$$e = r - p,$$

where r is the reference (desired) signal and p is the actual perception.

The output function which relates the error to action can be of a number of different varieties. The most common are the proportional function,

$$o = ge$$

where g is the gain, or amplification factor, and the integrating function,

$$o+ = (ge - o)/s$$

where s is a slowing factor. The function of the slowing factor is to ensure that the output changes gradually as in real physical systems, as opposed to instantaneously. For simulation purposes the input function is,

$$i = o + d$$

where d is the disturbance, and simply represents the concept that the input is affected by the actions of the system and any disturbances.

Although the control signals are theoretically continuous variables, in these simulations, due to the nature of computers, we must deal with discrete variables which change from one iteration to the next. As the values can change during an iteration we must be careful where we quote the value. For our present purposes the values quoted are all at the beginning of each iteration.

One important point to remember in these simulations is that we are assuming that all the signals have the same units or, at least, that the signals are unit-less. So we may say that the magnitude of the output is the same as that of the disturbance. In a real system this may not be the case. For example, in the iris control system any disturbance is only relevant in terms of its effects on the amount of light entering the eye, whereas the output is the size of iris aperture. It would be meaningless to say that the output is the same as the disturbance. What we really mean is that the *effects* of the output counteract the *effects* of the disturbance.

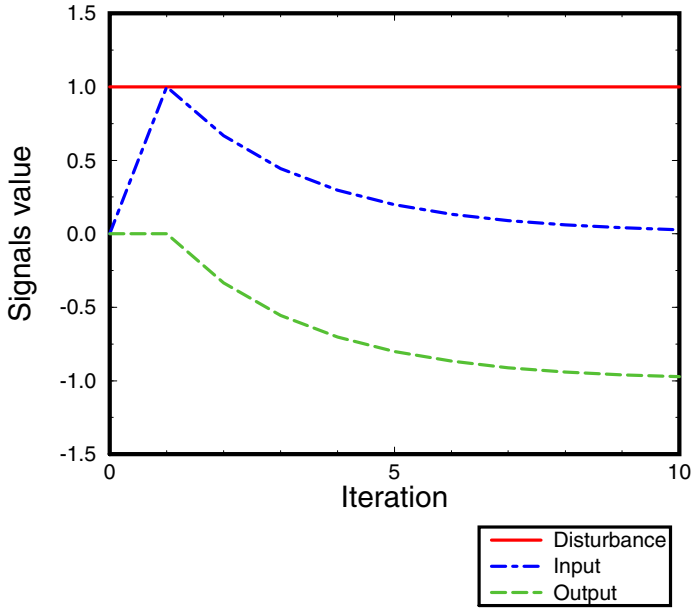


Figure 4.1: Basic control with constant disturbance, $s = 1500$

Iter.	Ref.	Input	Error	Output	Dist.
0	0.0	0.000	0.000	0.000	1.000
1	0.0	1.000	-1.000	-0.333	1.000
2	0.0	0.667	-0.667	-0.555	1.000
3	0.0	0.445	-0.445	-0.703	1.000
4	0.0	0.297	-0.297	-0.802	1.000
5	0.0	0.198	-0.198	-0.867	1.000
6	0.0	0.133	-0.133	-0.911	1.000
7	0.0	0.089	-0.089	-0.940	1.000
8	0.0	0.060	-0.060	-0.959	1.000
9	0.0	0.019	-0.019	-0.987	1.000
10	0.0	0.013	-0.013	-0.990	1.000

Table 4.1: Signal values for 10 iterations of control to constant disturbance, $s = 1500$

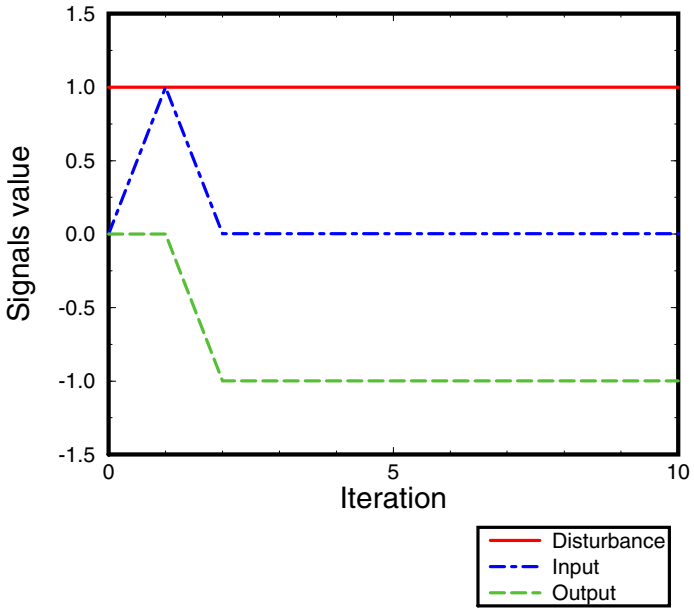


Figure 4.2: Basic control with constant disturbance

4.3 Basic Control

In this section we present some graphs and tables of the results of some simple simulations of basic control in order to illustrate the main points concerning the behaviour and functionality of perceptual control systems. In all cases the value of the gain is 500 and the reference is zero.

4.3.1 Constant disturbance

Figure 4.1 and table 4.1 show control with a constant disturbance of 1.0. The error signal also starts at 1.0 as there is zero output. Gradually the output increases (in a negative direction) to -1.0 and the error is zero at which point the input equals the reference.

Table 4.1 shows the signal values for the first 10 iterations. Notice that the *change* in output becomes smaller and smaller with each iteration. This is because the error is decreasing and the change in output is a function of the error.

Figure 4.2 and table 4.2 show a similar situation except that the slowing factor is less, 501, almost equal to the gain. Because the g/s ratio is ≈ 1, the output change is virtually the same as the error. In other words, the input comes to match (almost)

Iter.	Ref.	Input	Error	Output	Dist.
0	0.0	0.000	0.000	0.000	1.000
1	0.0	1.000	-1.000	-0.998	1.000
2	0.0	0.002	-0.002	-0.998	1.000
3	0.0	0.002	-0.002	-0.998	1.000
4	0.0	0.002	-0.002	-0.998	1.000
5	0.0	0.002	-0.002	-0.998	1.000

Table 4.2: Signal values for 10 iterations of control to constant disturbance.

the reference in one iteration. So far the slowing factor has been greater than the gain value. This ensures that the change in output is always less than the amount required to bring the input back to the reference value, thus avoiding overshoots and oscillations. In other words the g/s ratio is less than 1. Let's see what happens if $s < g$. Figure 4.3 shows control with s=270 and so the g/s ratio is 1.85. At each iteration the change in output is greater than the error, resulting in an overshoot. With each iteration the overshoots decrease and eventually settle down. The reason the oscillations die out, despite overshoots, is because the ratio of 1.85 means that the *magnitude* of the error at the next iteration is determined by half this value, ie. 0.925. As this is less than 1 the error will always decrease.

Figure 4.4 shows the case where the oscillations do not die out but increase. The g/s ratio is now greater than 2 (taking the one iteration lag into account). Again the *magnitude* of the error at the next iteration is determined by half this value, ie. > 1. Therefore the error will keep increasing.

The stability, and sensitivity therefore, of a control system depends upon the g/s ratio. As long as $s > g$ feedback will be negative and control successful. Figure 4.5 shows control with a variety of disturbances ($g = 500, s = 501$). In each case the output counteracts the effects of the disturbance with the input remaining close to the reference. The first, particularly, illustrates that the output correlates with the disturbance and *not* the input.

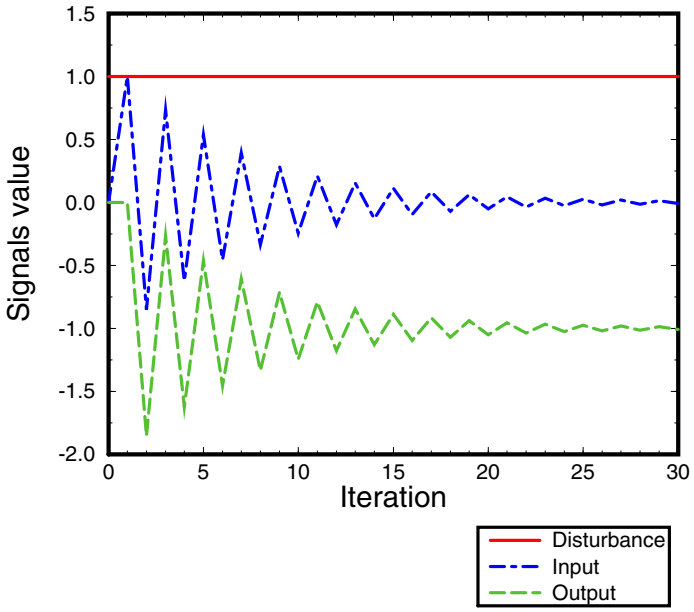


Figure 4.3: Basic control with constant disturbance, $s = 270$

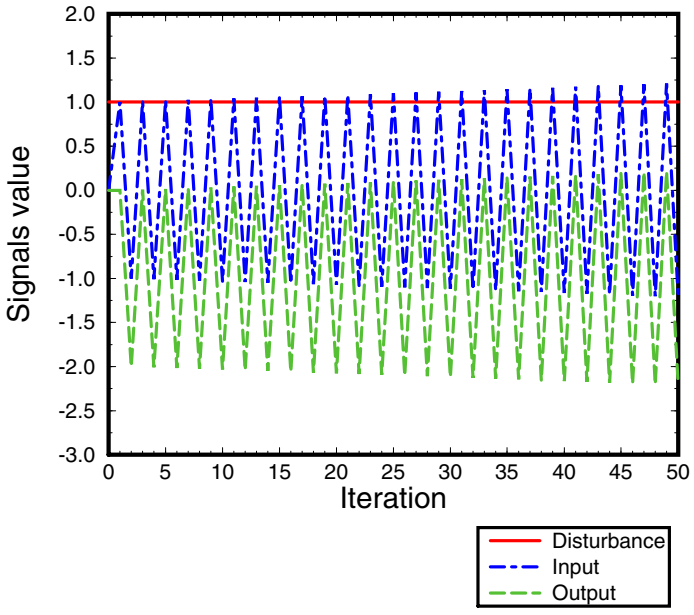


Figure 4.4: Basic control with constant disturbance, $s = 250$

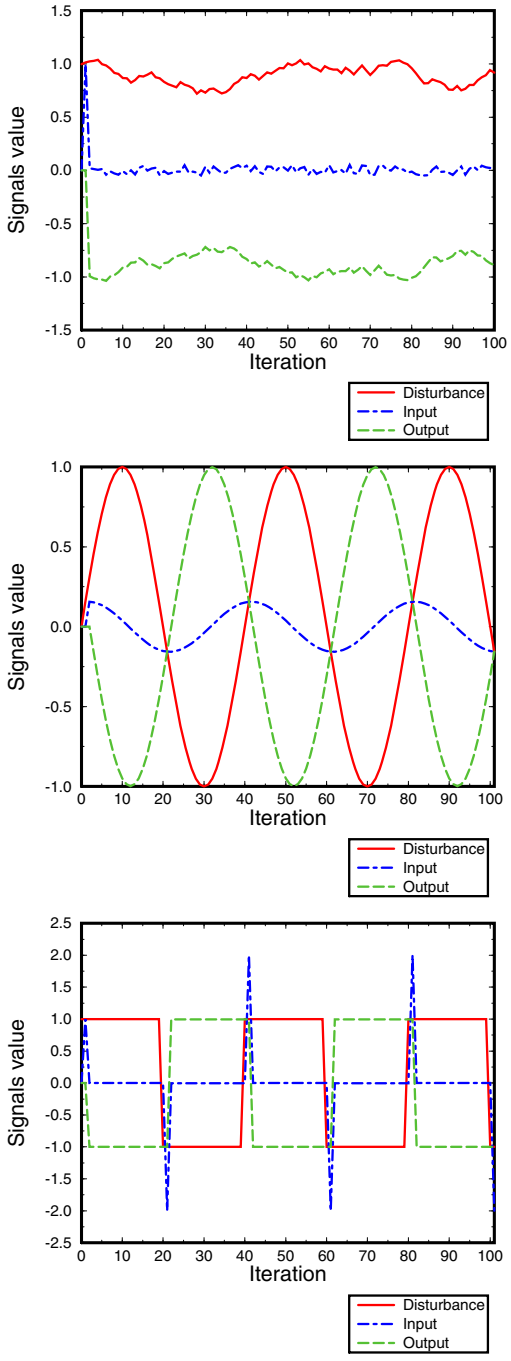


Figure 4.5: Basic control with random, sine and square disturbances

4.4 Basic Control with a transport lag

In real physical nervous systems signals are not instantaneously available, but take time to propagate around the system. Here we show how the problems introduced by transport lags need not be catastrophic.

4.4.1 Constant disturbance

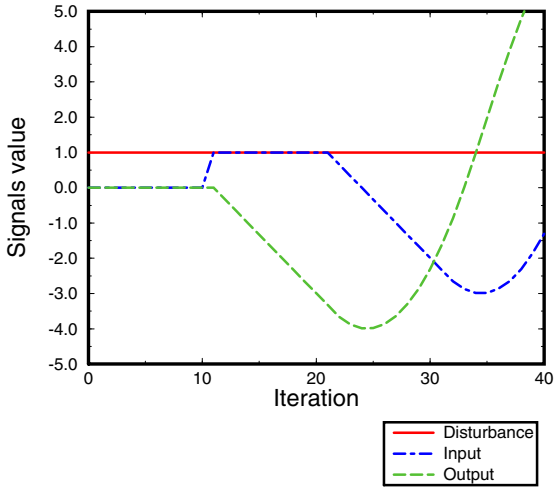


Figure 4.6: Basic control with transport lag and a constant disturbance, s = 1500

In Figure 4.6 a 10 iteration input lag is shown and the input signal displayed is that at the comparator. In the first 10 iterations the input remains at zero even though the disturbance is 1. Therefore, the error is zero and so no output. In iterations 11 - 20 the input is 1 as the signal from the sensors arrives. From iteration 11, as the error is now non-zero, the output grows(-ve). Although this affects the outer input immediately the effects will not reach the inner input for another 10 iterations. As the inner input has not been affected the error is still large and unaffected resulting in ever increasing output. From iteration 21 the effects of the output start to show and the input starts to move towards the reference. However, it continues past the reference due to the previous effects of the output which are only now showing through. In iterations 21 - 30 the output changes direction as the error has changed sign. The error keeps increasing as the input moves away from the reference resulting again in increasing output but in the other direction. The system continues in this fashion and oscillates out of control as the changes in output are too large for stable control.

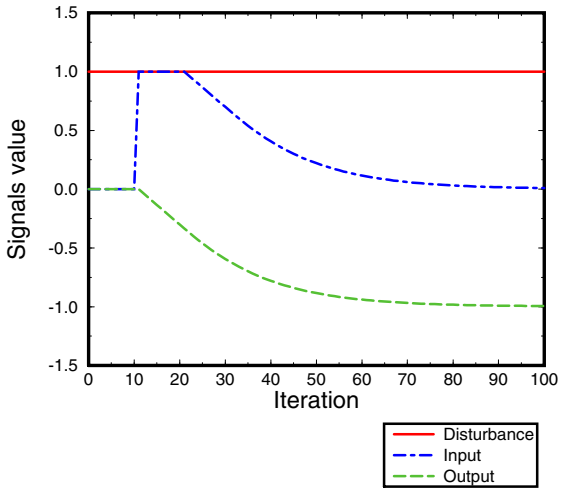


Figure 4.7: Basic control with transport lag and a constant disturbance, s = 15000

Figure 4.7 shows the same situation but with a greater slowing factor. Once the lagged input reaches the comparator the output changes much more slowly than the previous system, due to the greater slowing factor. From iteration 20, although the inner input is lagging behind the effects of the output the error is decreasing, such that the changes in output are small enough not to cause an overshoot. Control involving a transport lag is possible as long as the response of the system is slow enough. The longer the lag the slower the response must be.

4.5 Adaptive Control

In this section we look at a control system with an adaptive output function [89]. With the basic control system we provided the function that transferred the error to the output which was basically the amplification factor, the gain. Furthermore, it only took into account the operation of the system at one point in time (the present). Here the transfer function is learned by the system itself and extends into the past by looking at previous error signals resulting in an output pattern generator which would be able to maintain control even if the input signal is intermittently sampled.

4.5.1 More Math and Terminology

The output signal is a function of not just the current value of the error signal but also past values. We can look into the past as far as we like and the extent is denoted

by n, which represents the number of past program loops from the current position. A weight is associated with each error signal value. Therefore, we have the same number of weights as error values and the weighted sum of the error values gives a value related to the *change* in output. The weights denote the amount of influence each past value will have on the output.

The weights (which are initially zero) are adjusted by a small amount, on each iteration of the program execution, defined by past error values, the current error value and a learning rate. So, as the current error decreases (as control improves) then so does the amount of adjustment.

The new value of the weight is,

$$tau_{j_i} = tau_{j_{i-1}} + le_0e_j$$

where tau refers to the weight, l is the learning rate, e is the error value (e_0 being the current error), j is the weight number and i is the number of the current program iteration. The result of this process is that the weights *adapt* to the pattern of the input, generating appropriate output which keeps the error at a minimum.

The weighted sum w then is,

$$w = \sum_{j=1,n} tau_j e_j$$

and the output is,

$$o_i = o_{i-1} + gw$$

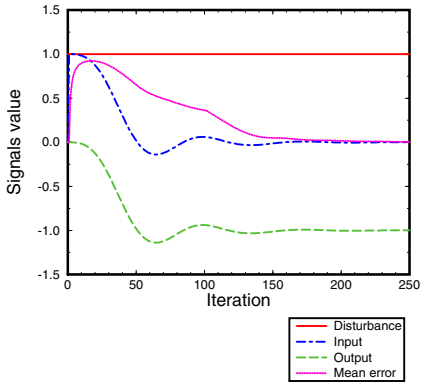
where g is the gain, in this case acting as a scaling factor determining the amount of allowable change in output.

The weights are also allowed to decay, which is necessary so that adaptation can take place to a new pattern of input. If the decay rate is d then,

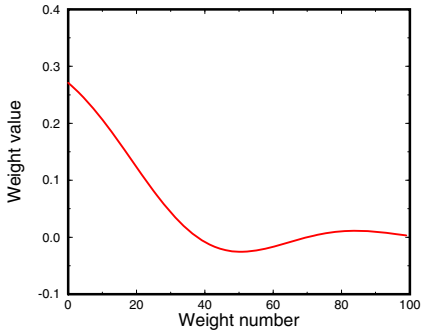
$$tau_{new} = tau_{old} - tau_{old}d$$

signifying that the weight is decreased by a proportion of itself.

The following variables values are used in the simulations, unless stated otherwise, gain $g = 0.01$, learning rate $l = 0.01$, decay rate $d = 0.0$ and error history n = 100. The *mean* error displayed in the graphs is the average of the error history, ie. the average of the last n error values.



(a) Control signals



(b) Transfer function after 250 iterations

Figure 4.8: Adaptive control with a constant disturbance

4.5.2 Constant disturbance

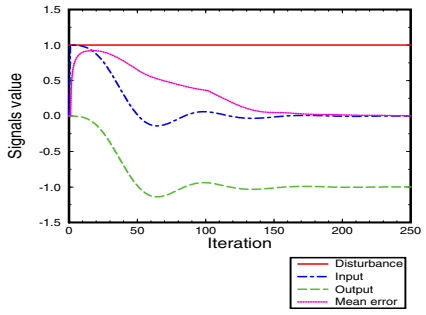
Figure 4.8a shows the control signals for 250 iterations of adaptive control to a constant disturbance with the values of the transfer function (the weights) plotted in 4.8b. The weights with a low weight number refer to the most recent error values. So, from 4.8b the weight of the current error signal is about 0.27 and the oldest is about 0.0. What this means is that the most recent errors contribute more to the change in output than the oldest.

The graphs in figure 4.9 shows similar simulations but with slight variations of the parameters to see what their effects are. For easy comparison figure 4.8 is repeated in 4.9 a and b.

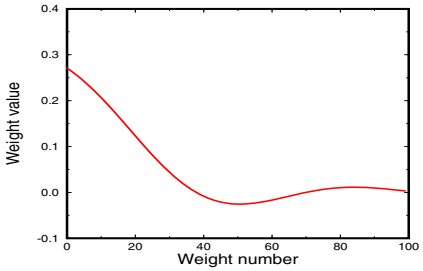
In figures 4.9c and d the gain has increased to 0.1. The result of this is that the change in output at each iteration is greater compared to 4.9a and b, and so control is achieved more rapidly. The transfer function, 4.9d, which reflects the past error signal, has the same shape as 4.9b but is less stretched out as the error signal (which, in these simulations is the inverse of the input signal) is more frisky in this case. Moreover, as the reference is reached sooner than in the previous simulation the weight values have had less time to build up, which is why they are lower.

Figures 4.9e and f show control with a lower learning rate, of 0.001. In this case adaptation takes longer and smaller changes are made to the weights.

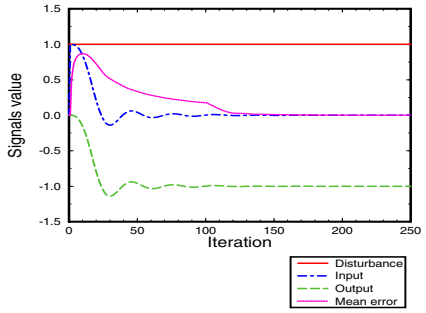
In figures 4.9g and h there is a non-negative decay rate which means that when the reference is reached, and the error is zero, the weights will decay to zero, as shown. This is because the weight changes due to the decay are a function of the



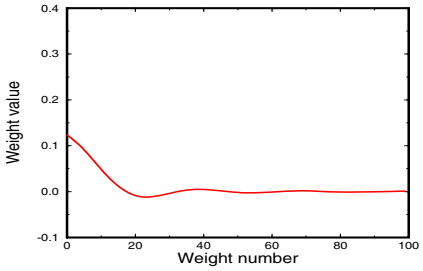
(a) g = 0.01, l = 0.01, d = 0.0



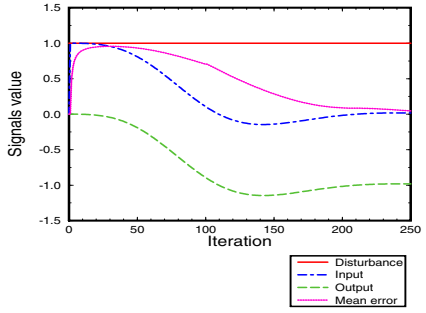
(b)



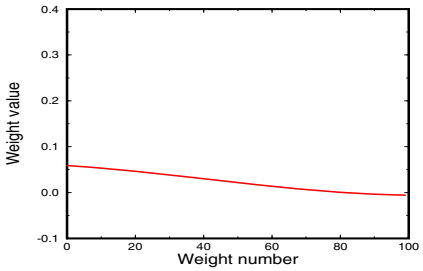
(c) g = 0.1, l = 0.01, d = 0.0



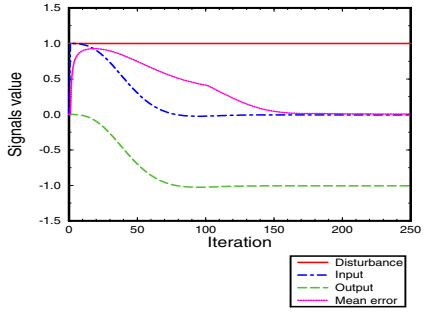
(d)



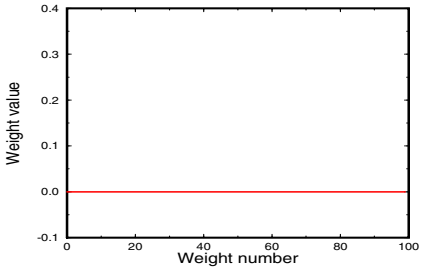
(e) g = 0.01, l = 0.001, d = 0.0



(f)



(g) g = 0.01, l = 0.01, d = 0.0005



(h)

Figure 4.9: Adaptive control with a random disturbance

weight *value* whereas, the weight changes due to learning are a function of the error. Depending upon the values of the decay and learning rates there will be the case where one cancels out the other resulting in the reference never being reached.

As with basic control there will be particular values of the adaptation parameters which achieve optimal response as well as control with and without oscillations.

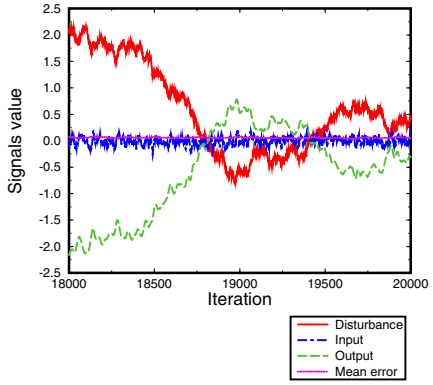
Figures 4.10, 4.11 and 4.12 show adaptive control to random, sine and square disturbances, respectively. In each case control in the second case is worse due to the decay factor reducing the magnitude of the weights. The transfer function for the random disturbance case shows a recent peak with the remainder of the weights at zero. This would indicate that, as would be expected, with *random* values, the only useful error signal is the current value.

For the sine and square disturbance cases the transfer function reflects the pattern of the historical error signal. If the input were removed, after adaptation, the output would continue unaffected for a short time.

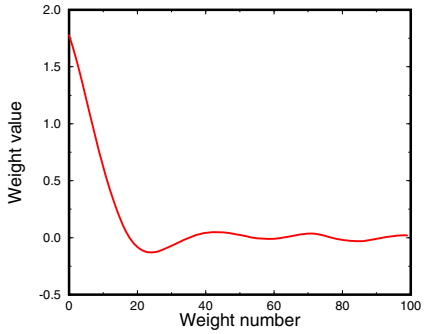
4.6 Summary

In this chapter we have presented simulations which demonstrate the main function of control systems, that the output of a properly designed control system will counteract any disturbance, keeping the input close to the reference. How close will depend upon the frequency of the disturbance. Transport lags are not a fatal problem but can be overcome with the appropriate system parameters.

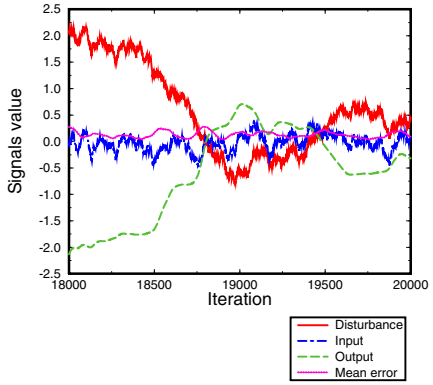
The perceived need to anticipate future events are often put forward [16, 42, 73] as a reason for the necessity for predictive modelling as opposed to the simple control systems described. We have also presented some preliminary demonstrations of how perceptual control systems can anticipate, in a simple way, based upon output pattern generation derived from adaptation to past input signals.



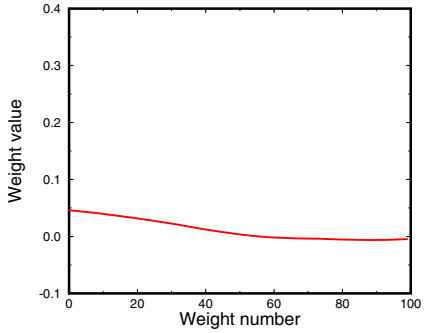
(a) Control signals, d = 0



(b) Transfer function after 20000 iterations

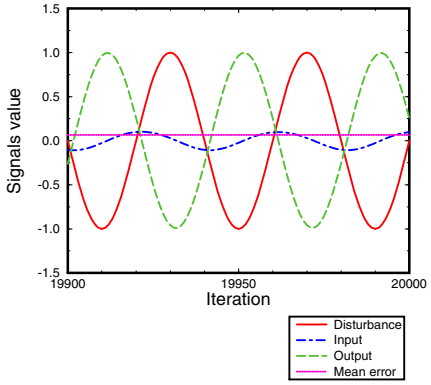


(c) Control signals, d = 0.005

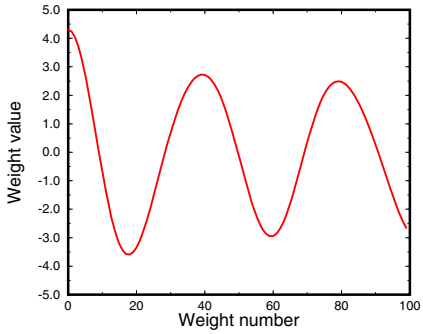


(d) Transfer function after 20000 iterations

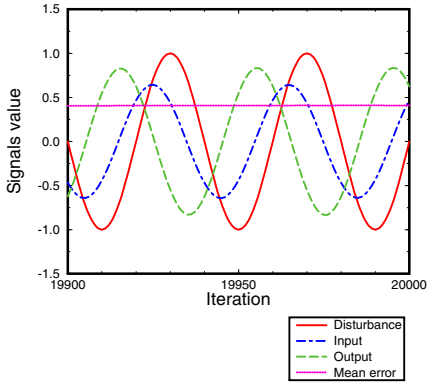
Figure 4.10: Adaptive control with a random disturbance



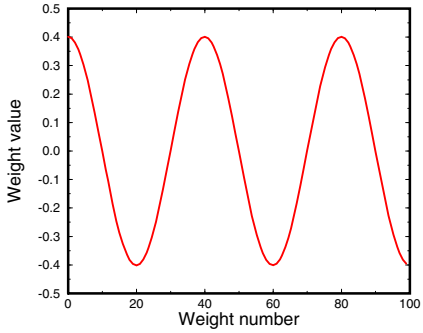
(a) Control signals



(b) Transfer function after 20000 iterations

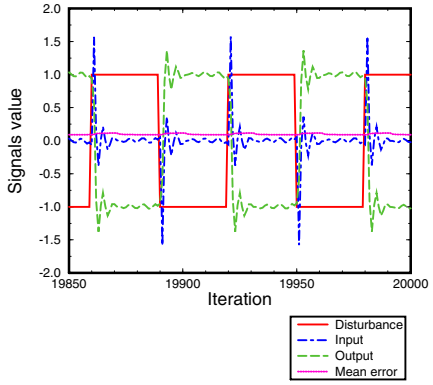


(c) Control signals, d = 0.005

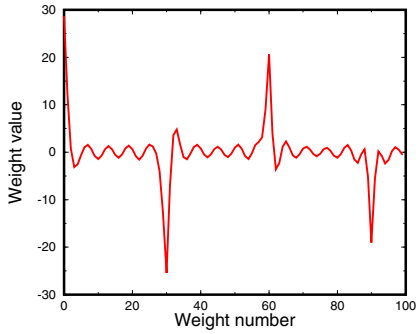


(d) Transfer function after 20000 iterations

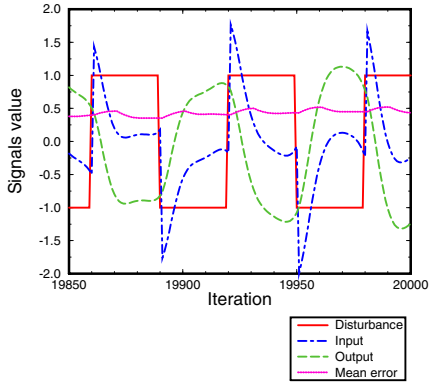
Figure 4.11: Adaptive control with a sine disturbance



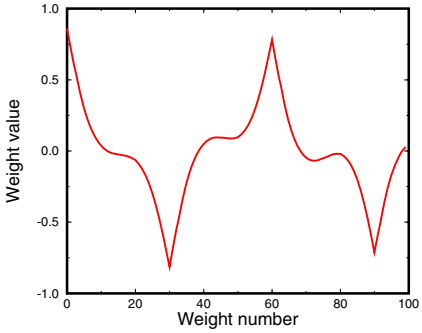
(a) Control signals



(b) Transfer function after 20000 iterations



(c) Control signals, d = 0.005



(d) Transfer function after 20000 iterations

Figure 4.12: Adaptive control with a square disturbance

Chapter 5

Foveal Fixation

5.1 Introduction

The great majority of animals have some type of visual system. One of the most readily apparent observations of the visual behaviours of animals, including humans, is that eyes are constantly moving, to attend to significant elements of the visual environment. How such fixation could take place is the subject of this chapter and the next. In this chapter we discuss some aspects of animal vision with particular reference to the foveal distribution in the retina. We outline some of the advantages of using the foveal representation [62, 128] of the scene over the more standard uniform representation, the main issue being that the foveal representation provides an intrinsic way of driving fixation. The fixation *measure* associated with the fovea had been proposed for simple binary shapes [120]. In this chapter we describe how it has been included in a framework which provides the capability for fixation to more complex and multi-coloured objects [130], for use in the fixation control system in chapter 6.

5.2 Animal vision

Evolution has derived a vast variety of visual systems [51, 59, 82, 83, 80, 125]. The motivation behind the development of vision as with other sensory systems is to differentiate between properties of the environment. With vision it is by detecting electro-magnetic radiation. Even the lowly single–celled amoeba responds to light, which acts directly on the cell causing it to move through water [104]. In more complex animals only specialised cells react to light and are organised in various

structures. Some progressive examples of single-chambered eyes range from the pit eye, and pinhole eye without a lens and with up to a few thousand light reacting cells (receptors) to human eyes with millions of receptors and a focusing arrangement of a lens and cornea [79].

The structure of the eye is a good indication of both the animal's ecological niche and types of behaviour it can exhibit [60]. Animals that inhabit flat open environments have retinal cells organised in "visual streaks", whereas those in arboreal environments such as forests have cells which are radially symmetrical. Further examples can be seen by looking at specific species. Beavers have a thickened cornea allowing them to see underwater. Bats and moles have minute eyes, which are all they need. Worms just have eye-spots that tell them the difference between light and darkness. Insects with compound eyes have arrangements of cells which are excellent at detecting motion, essential for small irritating creatures.

Birds of prey have a small pit called the *fovea* which has a high concentration of visual receptors. The pit has the effect of magnifying the image in this area to give the high resolution needed to spot small animals from afar (typically 10cm objects at 1500m) [129].

For behaviours that require shape recognition any organism that requires an adequate degree of resolution needs an image focusing system such as a lens and cornea and a large number of receptors. The optimum distribution of the receptors is graded from a peak at the centre (fovea) to the periphery. Figure 5.1 shows a variety of receptor distributions one might envisage, beginning with the type found in humans, which exhibits the distribution law $1/\theta$, where θ is the visual angle.

If the distribution was averaged out (figure 5.1b) the resolution would be too low, whereas if it was uniformly as high as in the fovea (figure 5.1c) there just wouldn't be enough space to fit them all in as the number would need to be increased by a factor of *10,000*. The graded distribution also has an advantage over two uniform areas of low and high distribution (figure 5.1d) as it is then easier to bring a target into the centre [18].

A practical limitation of a foveal system is that the visual scene is processed serially. Wouldn't it be an advantage to process in parallel and attend to the whole scene at once ? Well, the outcome in such a system might be that two areas of equal importance in the scene are detected, resulting in the urge to go in two different directions at once !

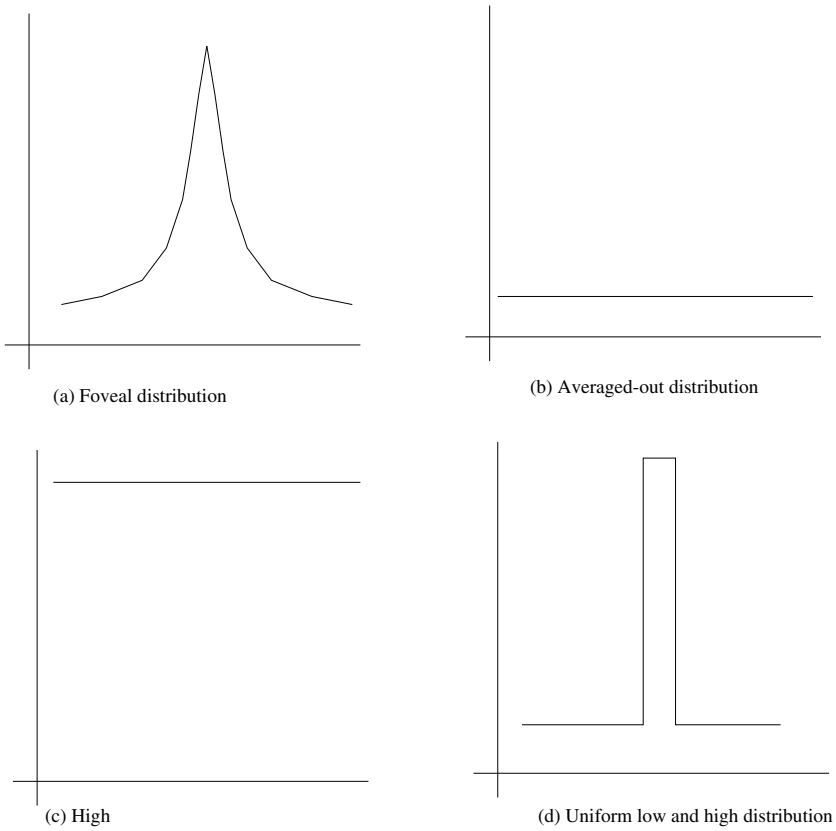


Figure 5.1: Foveal distribution

5.3 Foveal Representation

5.3.1 Uniform v. Non-uniform Representations

The predominant retinal representation in the animal kingdom is graded from a central point, the fovea, with a high resolution, to the periphery with a low resolution. In principle, a non-uniform distribution provides a more efficient active vision sensing environment than the standard uniform representation.

Consider a simple example of a visual scene of, say, ten objects, five which are of interest and five not. Also, consider two ways of viewing the scene. First, a method whereby the entire field of view is covered by high resolution. Second, a method which consists of a low-resolution representation of the field of view along with a smaller high-resolution window which can be moved around the same area. In order to evaluate the measure of interest of each object, the uniform method attributes equal priority of processing to all areas, even to those which turn out not to be of interest. The latter, non-uniform approach is able to evaluate (perhaps incorrectly)

each area at a low resolution before further processing at a high resolution in order of priority according to the evaluation. Ideally the five objects of interest would be processed first. Even if the low-resolution evaluation is only correct 10% of the time, the amount of processing with the non-uniform method will still be less than that of the uniform method which has no method of evaluation. The reason why the uniform system has to process everything at a high resolution is because it does not have the same predictive abilities. It would be more expedient to process only the areas of interest within a scene. However this is not possible without *some* pre-processing. The non-uniform method, with its low-resolution periphery, represents a compromise between only processing areas of interest and processing all areas at a high resolution. It should be noted, however, that this is only possible where there is meaningful information to be extracted at a low resolution, such as colour blobs. If such information is not available then it *would* be necessary to process all areas at a high resolution to find the areas of interest. The retinal representation, described next, is a special case of a non-uniform approach with some interesting properties particularly suited to overt fixation.

5.3.2 The Foveal Transform

One of the main practical advantages of the retinal distribution is data reduction [6, 18, 97, 98, 109], in that only a small portion of the space is sampled at a high resolution. Other advantages include the provision of a natural interest operator (the centre of view), and less complex algorithms for tracking [114, 120], as well as the easy detection of size and rotation changes by vertical and horizontal translations (provided the object is foveated [7, 97, 109, 101]). The use of the foveal representation does, of course, require appropriate motor behaviours to put the high resolution fovea over the areas of interest [6, 7, 18, 97, 98, 99, 114, 120, 109].

The foveal software used in our experiments converts a uniformly sampled image into a log-polar representation. The image is separated into receptive fields radially from the centre (see figure 5.2a). The fields nearest the centre are a single pixel and grow exponentially larger out towards the periphery.

The grey-level (or colour) values at each receptive field are averaged and stored in a rectangular array called the *cortical* projection (due to the similarity with the structure of the visual cortex)(see figure 5.2b). Therefore, the axes of the cortical projection correspond to the angular position, θ, the angle measured at the centre from a common origin, and the radial distance, r, of the receptive field from the centre. Figure 5.3b shows the foveal representation of a face (5.3a). The eyes can be

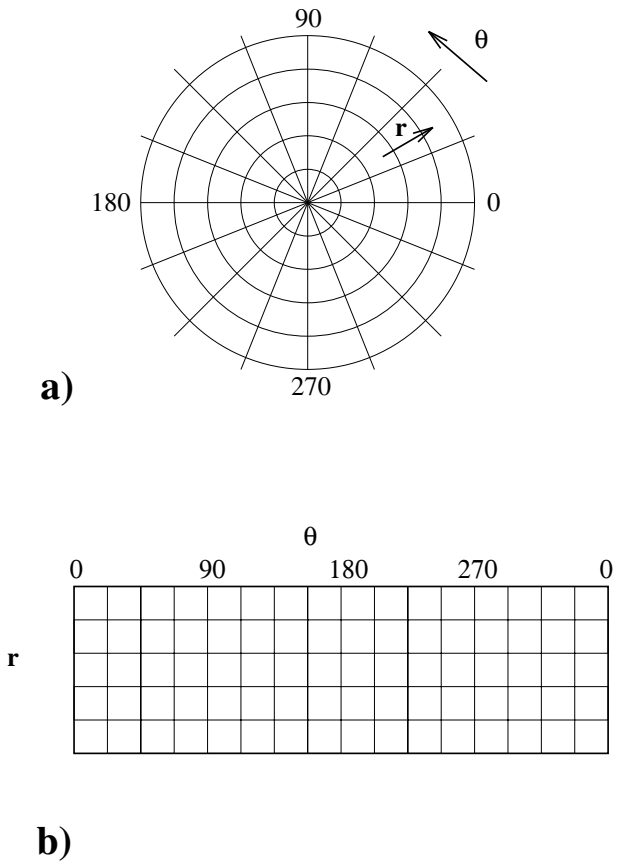


Figure 5.2: Each segment in the uniform representation (a) maps to a pixel in the cortical projection (b) indexed by r (the distance from the centre) and θ (the angle measured anti-clockwise from the horizon).

seen in the left-half of the transformed image and the nose and mouth on the right. Each row of the foveal representation represents a ring of receptive fields with the top line derived from the centre of the input image, the fovea.

Certain parameters can be manipulated with the software, such as the size of the input image, the position in the image to use as the fovea and the scale, in order to get a zooming effect on the foveal representation. The software is from a retina-like simulation from the Department of Communication, Computer and Systems Science at the University of Genoa. The original code has been modified to provide a command-line interface and a library of routines for transforming both grey-level and colour images into the foveal representation. The modifications enable the cortical projection to be transformed back to the uniform representation to aid visualisation of the retinal view, as well as procedures that construct an array of transformation details that produce the cortical projection with different zoom values (scale). The

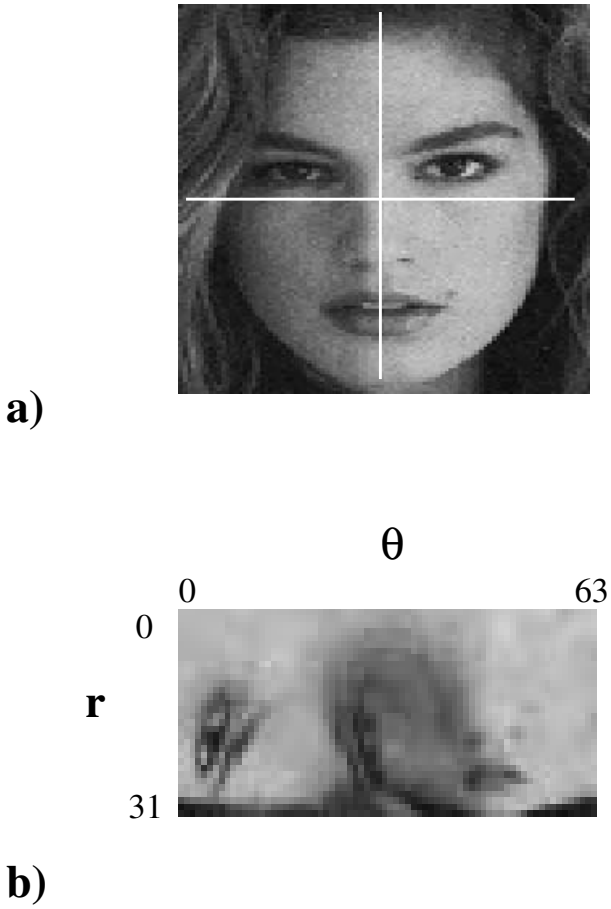


Figure 5.3: a) A face (256 x 256 pixels) and b) its foveal representation (64 x 32 pixels).

result of being able to change the scale of the transformation is that a particular area or object can, in effect, be segmented from interfering areas by allowing it to take up the entire field of view for intensive analysis. This is the covert equivalent of moving closer to an object.

The top half of figure 5.4 shows the foveal representation transformed back to the uniform representation. Each square area represents one colour input signal and is taken as the most prominent colour which falls on that area. There are 32 rings of square regions each with 64 elements. These can easily be mapped into a rectangular array which is more suited to processing within a computer program. The array of the same scene is shown in the bottom half of figure 5.4, where each row represents one ring. The rings from the fovea to the periphery map to rows from top to bottom, respectively. This foveal representation of 2,000 pixels signifies a substantial reduction in the amount of the information that needs to be processed, compared

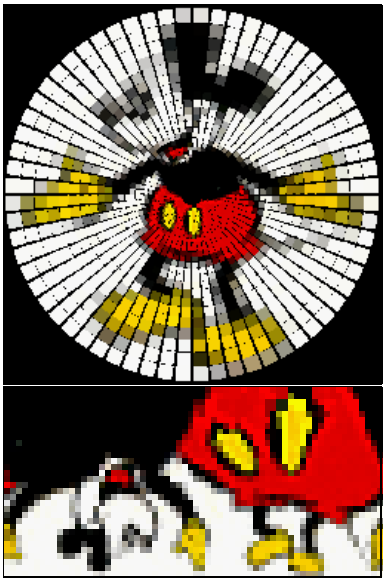
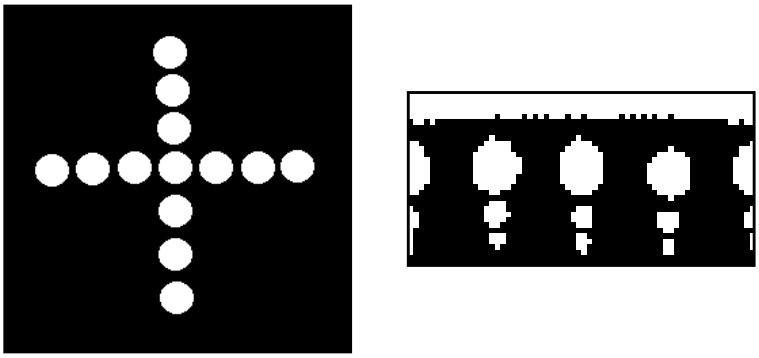


Figure 5.4: The foveal representation of a well-known cartoon character

with the standard uniform image of 60,000 pixels covering the same field of view.

5.4 Foveal Fixation Measure

The basis for the motivation of behaviour and the measure of fixation is derived from a very simple property of the foveal representation, referred to as *pixel count* [120]. Figure 5.5 demonstrates this property on a simple circular shape. Figure



a) Uniform image

b) Foveal image (cortical projection)

Figure 5.5: Demonstration of the property of maximum pixel count at the fovea.

5.5a shows a standard, uniform image with a white circle in a number of different positions. Note that wherever the circle is in the image its size is the same, i.e. the

pixel count is constant. Figure 5.5b is the foveal representation of the same scene with the foveal point centred on the uniformly sampled image. Now notice what happens to the circle the further away it is from fovea. At the fovea the circle has a maximum pixel count (the band of white at the top of the foveal image) which gets smaller and smaller the further from the centre. Given this property it is possible to determine when the circle is fixated, with a controlling mechanism that adjusts position and maximises the pixel count. Incidentally, this method equates to finding the centroid for an arbitrarily shaped figure.

The scheme can be extended slightly to operate with a specific view of an object, which equates to a specific pixel count. So the current view would be correct when the current pixel count is the same as that of the model.

$$Error = P_{REF} - P_{PERC}$$

This equation simply states that the error signal is the reference pixel count, P_{REF}, minus the current, or perceptual, pixel count, P_{PERC}. This difference can be used to drive the fixation to the correct point. The real world example in section 5.5.2 uses this principle, but instead of just counting one binary feature the input function counts multiple colour features corresponding to a particular object.

5.4.1 Representation

The representation of objects and scene used is simply that of a count, or histogram, of features relevant to a target. In the current system colour features are used mainly because they can be processed relatively quickly compared to other types of features. It is intended, however, that the scheme can be extended to any kind of feature as the measure is dependent upon the feature *count* and not the feature *type*. The main rationale for a histogram of features is indicated in the later discussion on *pixel count*, however this type of simple computation (addition) and representation (activations) is consistent with what is possible with neurons or groups of neurons.

5.4.2 Input function

In order to encode (compute the pixel histogram of) a target object a training image of the object is converted into the foveal representation by applying a log-polar transformation [98]. To avoid the washing-out of colours in an area of pixels, the method used for deriving a single colour from the uniform image is to take the modal value of the set of colours as opposed to the average of the area. The RGB

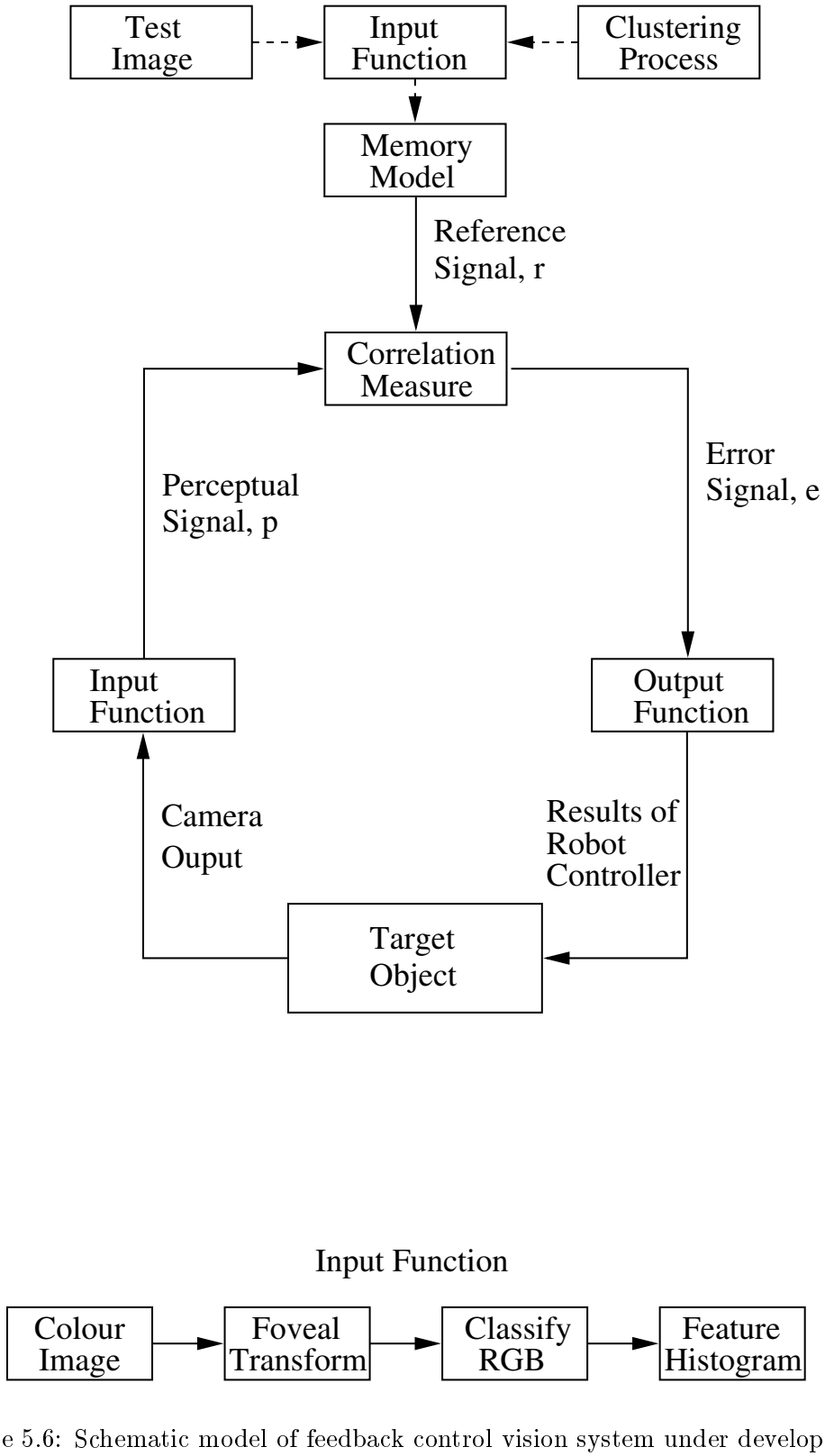


Figure 5.6: Schematic model of feedback control vision system under development.

feature vectors at each pixel in the log-polar image are then adjusted for intensity by,

$$C(r,g,b) = \frac{c^{1/G}}{r^{1/G} + g^{1/G} + b^{1/G}}$$

where G is the gamma value of the camera and c is the red, green or blue values.

Within the learning process, performed on the training image, the feature vectors are clustered according to a k-means clustering process resulting in a reduced set of vectors (the cluster means) that represent the features specific to the model object. The histograms for the model and the scene are produced by counting how many features from the foveal RGB image fall within each cluster, for the set of clusters representing the target object. For example, at each pixel the RGB vector is adjusted for intensity and then compared with each of the ten, say, cluster means. If it is within three standard deviations of a particular distribution it is said to belong to that cluster and the histogram is incremented appropriately.

5.4.3 Comparator

The philosophy of this system is not to *identify* objects in a scene but to model the general behaviour by which an animate system can position itself relative to a specified object. Such behaviour could, however, be used as a mechanism for providing an identification system with the best available information.

The behaviour is dependent upon a measure of how close the current view is to that desired. The measure is derived from a comparison between the model and scene histograms. Such correlation methods include the sum of squared distances and the Bhattacharya distance. Results presented in section 5.5.2 use the former method,

$$\omega = \sum_{i=1,n} (x_{1_i} - x_{2_i})^2$$

where x_1 and x_2 are the corresponding bin values from each of the two histograms.

5.4.4 Controller

The output function, or controller, relates the error signal (fixation measure) to the *direction* of movement of the artificial animate system in three dimensions. Within this control-based scheme it is not necessary to compute the specific values of the position parameters but to change them in such a way that the error signal is minimised and a specific *input* is realised. One possibility is that the automatic control

of the sensor can be achieved by a simple gradient descent technique. However, as we shall see in the next chapter a more direct method of moving to the target is available.

5.5 Fixation Measure Experiments

5.5.1 Expected behaviour of fixation measure

In this section an analysis is made of the idealised behaviour of the error signal in response to movements of a sensor, with six degrees of freedom, relative to a known object, again a simple circle. The movements to be analysed are as follows,

Foveal fixation The change of the point of fixation in a grid of points covering the object.

Distance to object Movement along the line of sight with a constant, central fixation point.

Object rotation Movement of the sensor around the object retaining a constant fixation point and distance. Equivalent to rotating the object.

The above categories cover all the movements required for an animate system to position itself in a learned viewpoint relative to a known object. This type of behaviour pervades the animal kingdom irrespective of the level of intelligence, as a means of providing the animal with the best available information as a basis for further behaviour or recognition.

Figure 5.7 shows the behaviour of the error signal as a result of the three types of movement. Figure 5.7a shows the error landscape (minimum shown as a peak) for the change in fixation point, 5.7c the plot for translation along the line of sight and 5.7e the rotation of the object. In the next section this expected behaviour, for a simple synthetic object will be compared with the behaviour of more complex, real world objects in a laboratory environment.

5.5.2 Experimental behaviour of fixation measure

The experimental setup consists of a JVC colour CCD camera with an Ernitec controllable lens, attached to the end-effector of a PUMA762 robot arm (see figure

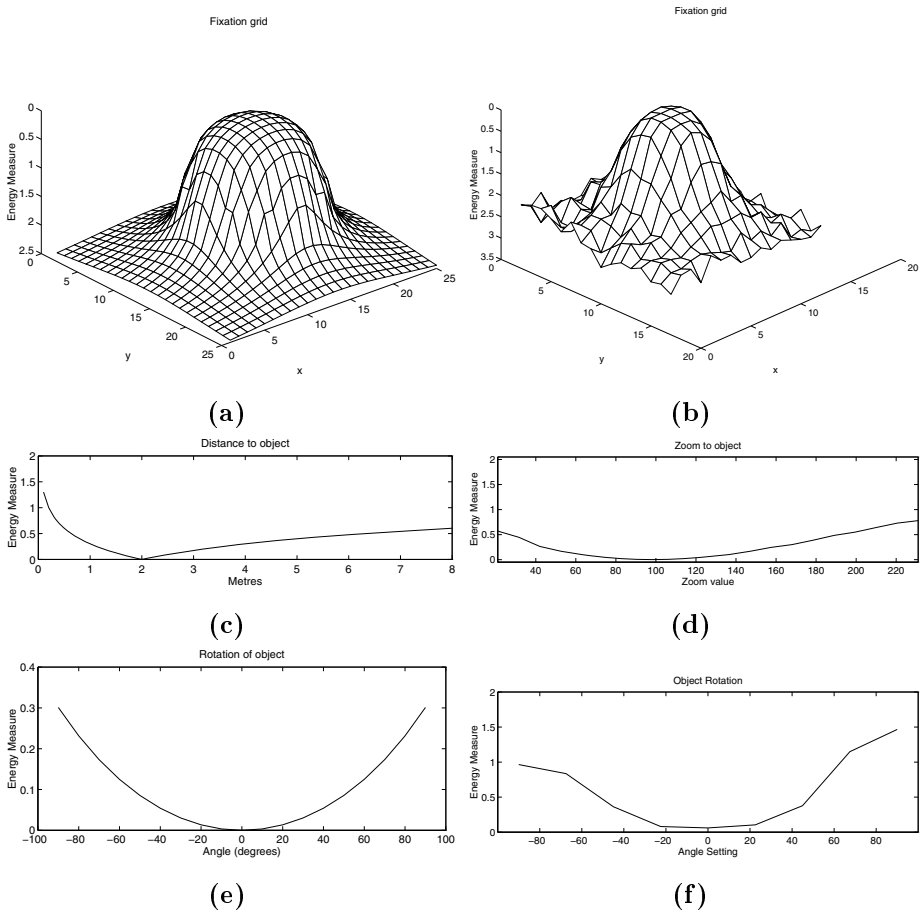


Figure 5.7: a), c) and e) Idealised behaviour of fixation error signal. b), d) and f) Experimental behaviour of viewpoint error signal.

5.8a). The camera can be controlled as if it were independent of the arm by rotating it about and translating it along any of its three axes.

The objects used as targets were the Halloween masks as shown in figure 5.8b. These masks were chosen for a number of reasons. Their relative complexity, arbitrary shape and 3-dimensional nature makes them more realistic objects for an unconstrained environment than, say, a blocksworld scenario. Of course, their distinctive colours make them more suitable for the current system which is only, at present, extracting colour information. For each of the movement categories described in section 5.5.1 the robot was moved accordingly, relative to a target Halloween mask. At each fixation point the correlation measure from section 5.4.3 (error signal) was recorded. The fixation grid experiment was equivalent to fixating an even grid of points over the scene in figure 5.8c. The results are shown in figure 5.7. Due to practical difficulties of determining a long line of sight along which the camera could

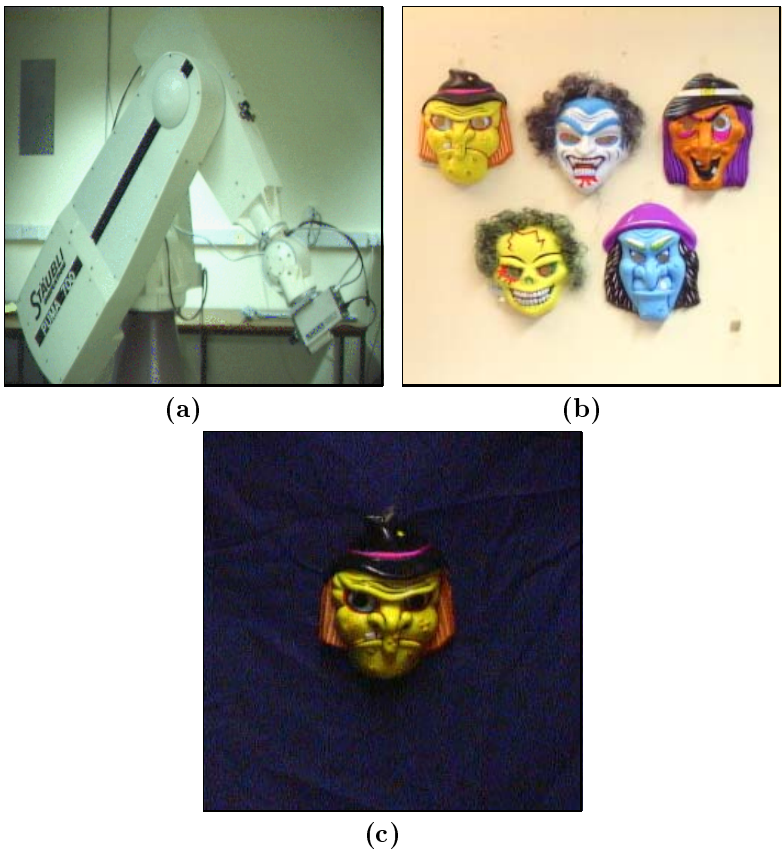


Figure 5.8: a) Robot/camera system. b) Target objects. c) Experimental scene.

move without violating safety protection, the camera was kept stationary and the camera zoom varied to emulate changing 'distance to object' (see figure 5.7d). The experimental results compare favourably with the predicted behaviour shown on the left hand side of figure 5.7, though the fixation grid does show unevenness at the extremities of the scene, the critical issue being that in each plot there is a gradual progression to a minimum point which a controller could use to position the camera appropriately.

5.5.3 Discussion and Conclusion

In this chapter a fixation measure has been presented for use in controlling the gaze of an artificial, active vision system with respect to multi-featured, complex, non-synthetic objects. At present the scheme only uses low-level features, analogous to those in the primary visual cortex, and hence will be corrupted by non-target objects in the field of view. The next chapter addresses this issue and presents work (continuing the analogy with the brain) which includes increasingly higher-levels

of feature complexity that, with a similar feature count measure, will give added discrimination to the system.

There are many other research projects that endeavour to investigate and model the type of attentional, recognition behaviour discussed in this chapter. Nordlund and Uhlin [81] and Tunley and Young [114], are concerned with optic flow and control to motion features and Weiman and Juday [120] with control to the foveal pixel count of binary shapes, all for tracking purposes. Hoad and Illingworth [49] and Pahlavan and Eklundh [84] describe methods for automatic control of stereo head-camera parameters, the former with open-loop fixation to single colour regions of interest in static scenes. Spratling and Cipolla [107] describe a robotic, visual servoing system that has some goals in common with ours, but one which operates by computing transformations between the error signal and specific motor actions, on outline contours in static scenes.

Static, open-loop fixation also features in many attention-based systems. Prime examples of such systems are Westelius [121] (symmetry and edge points), Milanese [72] (colour and local curvature), Culhane and Tsotsos [28] (intensity and edges), Rao [94] (image patches) and Grimson et al [40] (colour, edges and depth). The properties and benefits of the foveal representation have been recognised and discussed by, among others, Sandini and Dario [98], Tunley and Young [114], Weiman and Juday [120] and Westelius [121]. Although motion information is not explicitly incorporated into our model the importance of control to motion, in animate systems, for responding to things of potential significance is recognised. Extending the current system to include control to a maximum pixel count of motion features would be relatively simple.

In contrast to open-loop fixation systems, the work presented in this chapter represents a move towards a more realistic system which is able to perform in real, dynamic environments. The system incorporates some of the features and ideas of the systems cited above, but extends and brings together the goals to develop an active, dynamic, versatile vision system that is able to respond to complex, real-world objects. The main feature described in this chapter has been the proposed foveal fixation measure. It provides a means by which the system is able to move to fixate a known object without having to pre-process the entire scene.

Chapter 6

Visual Fixation Control

6.1 Introduction

The previous three chapters laid the foundations for the visual fixation control system presented in this chapter. Described and discussed were Perceptual Control Theory, foveal scene representation, multi-coloured object encoding and a visual fixation measure. These are extended and combined to produce a system which is able to fixate and track complex objects [134]. The fixation measure is extended to include higher levels of abstraction which gives information more specific to the target allowing segmentation from surrounding, distracting regions.

Described first are two essential parts of a control system, the input, or perceptual, signal and the output which results in the input being controlled. The first experiments show the behaviour of the control system when fixating in real-time to a real object with detailed reference to the input and output signals. The model acquisition procedure for multi-level control is also described followed by some relevant experiments to complex multi-coloured objects.

6.2 Fixation input signal

In succeeding sections we describe how particular regions of interest in a scene are segmented from the background. As described in the previous chapter, each row and column of the pixels, in the foveal distribution, that make up the region of interest represent the direction and magnitude from the centre of the field of view. From this information we are able to derive a fixation signal which can be used

as the input to a standard PCT control system. Sparks [106] reports that animal visual fixation works in a similar manner. Populations of activated cells in a neural map in the *superior colliculus* define the direction and magnitude of eye movements [101, 100]. So, instead of using the measure from the previous chapter, which only gave a correlation value but no information about where to move, we can control a variable which more directly represents how to get to the fixation position. This

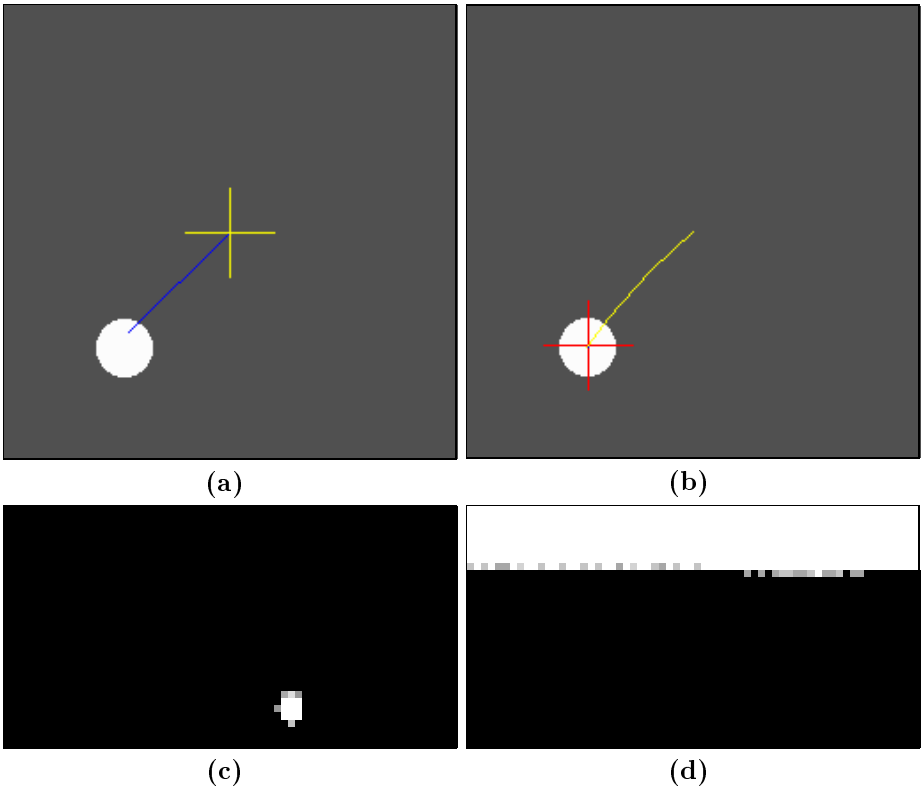
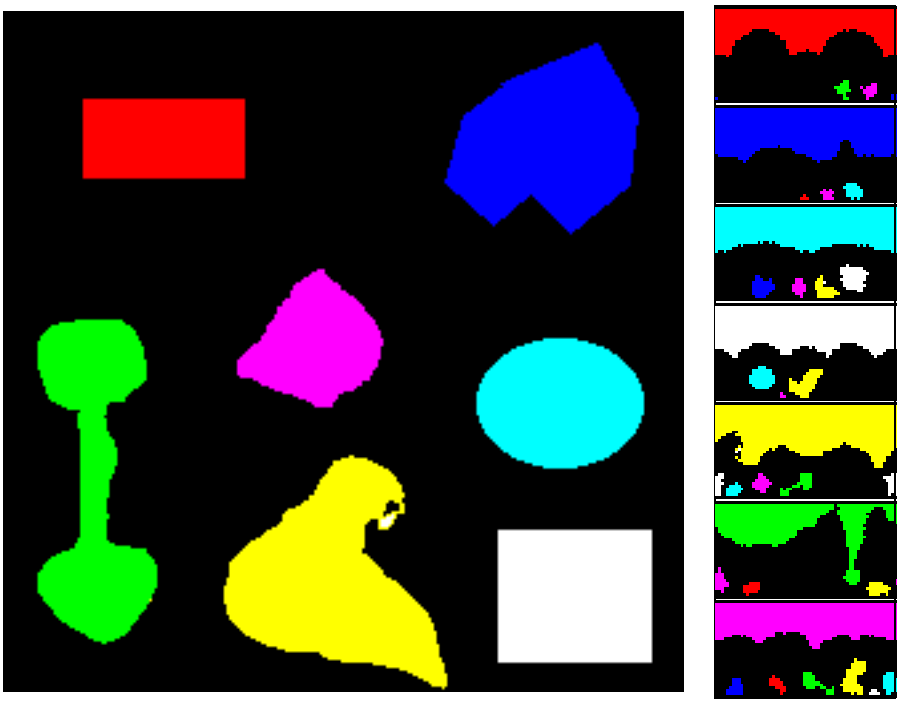


Figure 6.1: A simple, single-level fixation control simulation. Images (c) and (d) are the foveal representations of the initial (a) and final (b) uniform scenes, respectively.

fixation signal is derived by simply taking the mean of all the position vectors within the region of interest. Figure 6.1c shows the foveal representation of figure 6.1a where the small white blob corresponds to the white circle in 6.1a. The dark line in 6.1a from the central cross hair is a visual representation of the fixation input signal derived from the mean of the position vectors of the blob. Figure 6.1b and d shows the end result of control of the fixation signal. The cross hair tracker is now centred on the target circle. Notice in 6.1d that the circle now corresponds to a white band in the foveal view. What has happened is that the tracker has moved until all the position vectors are in equilibrium (their average is zero) resulting in the fixation on the centroid of the region.

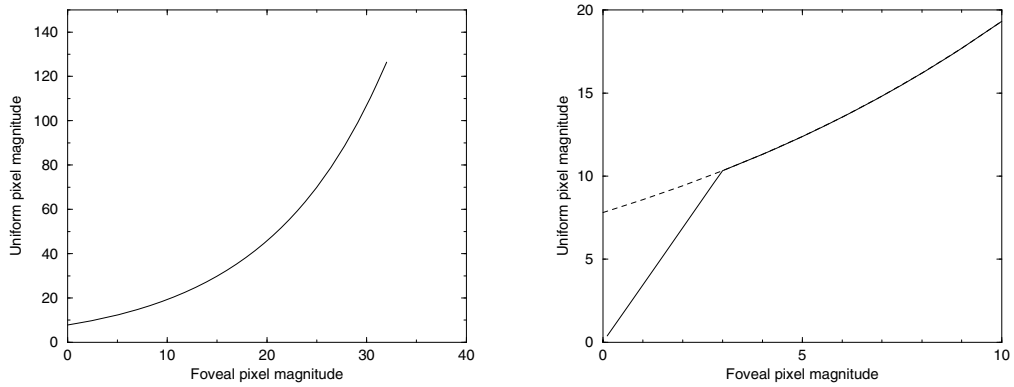
The above effect of fixation on the centroid occurs not only in regular geometric figures but also for irregular shapes as shown in figure 6.2. The image in figure 6.2a contains a number of irregular coloured shapes. The foveal view when fixated on the centroid of each object is shown in figure 6.2b. One slight problem with the foveal



(a) (b)

Figure 6.2: Simple colour fixation. a) The image of simple coloured figures, b) The foveal view when fixated on the figures, clockwise from top left.

representation used is that there is a blind spot in the centre of the image. This is shown in figure 6.3 where the distribution plot of the number of pixels from the centre of the uniformly sampled image against the representation meets the y-axis at approximately 8 pixels. Although this is a small proportion of the scene as a whole the practical result is that there will be some small oscillations around the centre as the input signal jumps directly from 0 to 8 pixels. Figure 6.3b shows this area expanded and the proposed resolution which is to re-represent this area as a gradual decline from 10 pixels to zero.



(a) **(b)**

Figure 6.3: Distribution of the foveal representation showing the blind spot.

6.3 Image and Robot output

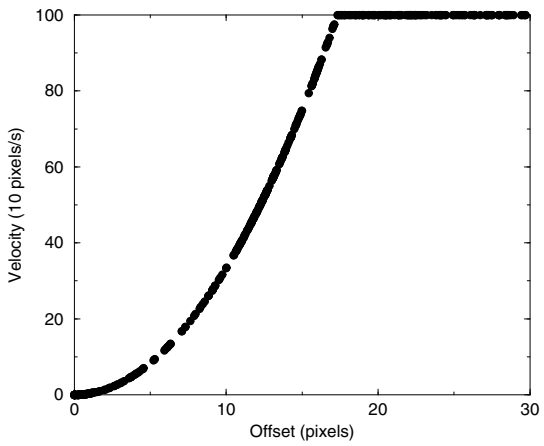
We have developed software which implements tracking control off-line in synthetic and real images, on-line in real, live scenes with a fixed viewpoint as well as with the mobile PUMA700 robot arm and camera system. In the live, fixed view experiments the movement of the robot is represented by a moving cross-hair. The input which is controlled is the size of the offset from the centre view to the region of interest, with the reference signal being zero. The region of interest is the segmented area of features which belong to the target. As the reference signal is zero, the error signal is the same as the input signal,

$$\vec{p} = \sum_{i=1,n} \vec{x_i}/n$$

where $\vec{x}$ are the position vectors of the n segmented features with respect to the centre of the current field of view, in Cartesian coordinates.

The output signal is the direction and *velocity* of the movement towards the target, the velocity being a function of the error signal (see figure 6.4). The direction is defined by the unit vector of the error signal (same as input) and the velocity for, image based output, is,

$$v = g\bar{p}^2$$



(a)

Figure 6.4: The output velocity as a function of the pixel offset. A limitation is introduced to emulate a real physical system.

where g is the gain and $\bar{p}$ is the modulus of the error signal in pixels. The output is limited to a maximum of 1000 pixels per second (see figure 6.4). For the output of the robot arm the velocity is,

$$v = g\theta$$

where θ is the angle of rotation through which the camera must be turned to fixate the target.

Therefor, as the sensor centre gets closer to the target the velocity decreases until fixation, when the error will be zero and so, therefore, the velocity. A practical result of relating the error signal to the *velocity*, in this way, avoids oscillations and jerky movements as fixation is reached.

With the real-time robot controller it is possible to execute commands defining the direction and velocity of movement required. The image processing is performed in parallel with the robot movements and so it is not necessary to wait for a movement to cease before updating the error signal. Also commands can be sent to the controller while the robot is in motion which override all previous commands. In this way we are able to continually monitor and control the fixation signal. Control within fixed views is handled in the same way with the exception that the movement (of the robot simulating cursor) in each iteration is computed discretely from the current desired velocity and the length of time of each iteration.

6.4 Single-level Control

Single-level control is sufficient for tracking simple lights or areas in grey-level or colour scenes. Areas within an image of a particular grey-level range (such as the brightest) can easily be segmented, from which the fixation signal of a blob can be derived. Similarly for colour regions, particular objects can be delimited by defining the upper and lower thresholds for the red, green and blue values. Figures 6.1 and 6.2 show examples of tracking simple regions in simulated images. Experiments in tracking to simple lights and single-coloured objects have been performed successfully in real-time with the robot.

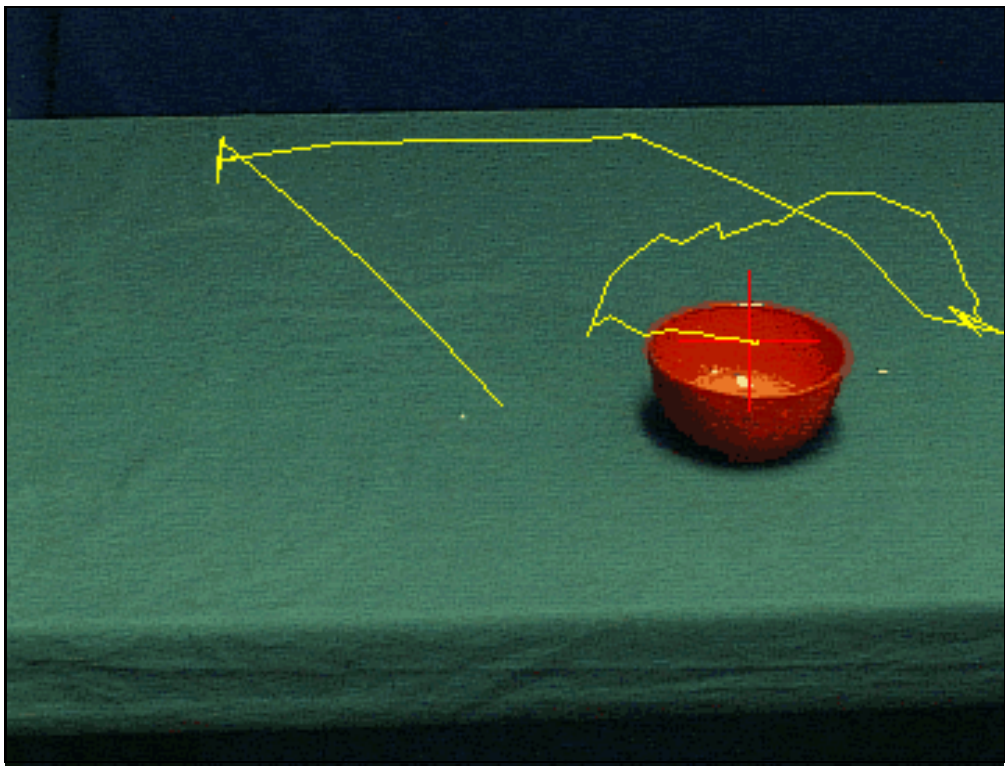


Figure 6.5: The yellow line shows the trajectory of the red bowl as it is tracked by the control system.

Figure 6.5 shows a real scene of a red bowl on a blue background. The yellow line shows the trajectory of the bowl starting from the centre of the image. The bowl was moved rapidly three times (up to the left, to the right and down to the right) with a pause between each movement and then slowly to the left. The signals for the pixel distance to the target (input signal) and the velocity of movement (in units of 10 pixels per second) are shown in figure 6.6a, for the duration of the tracking. The three larger peaks correspond to the rapid movements and the smaller

to the slow movement. The second large peak is shown in more detail in figure 6.6b. The circles show the points on each iteration of the processing loop indicating a sampling rate of about 20 frames per second. It can be seen that the system rapidly detects the movement of the target and the output immediately goes to its maximum of 1000 pixels per second. The position of the tracking cursor, as indicated by the input signal, also moves rapidly to fixate the target, the whole process taking approximately 250 milliseconds. If the robot arm were actually moved the behaviour would be similar except that the time it took to reach its output velocity would depend upon the physical system itself. The smaller peak is shown in more detail in figure 6.7. Although there is always some residual error it is kept at a minimum by varying the velocity output to maintain tracking of the moving object.

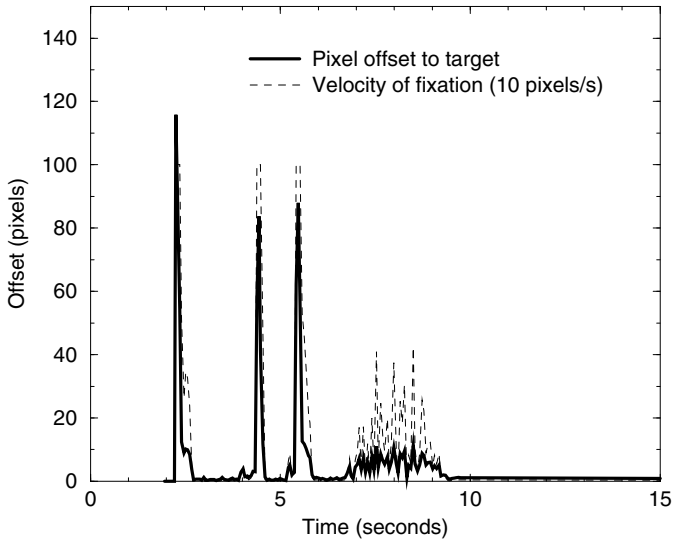
6.5 Object model representation and acquisition

A couple of problems arise when extending tracking control to multi-coloured objects:

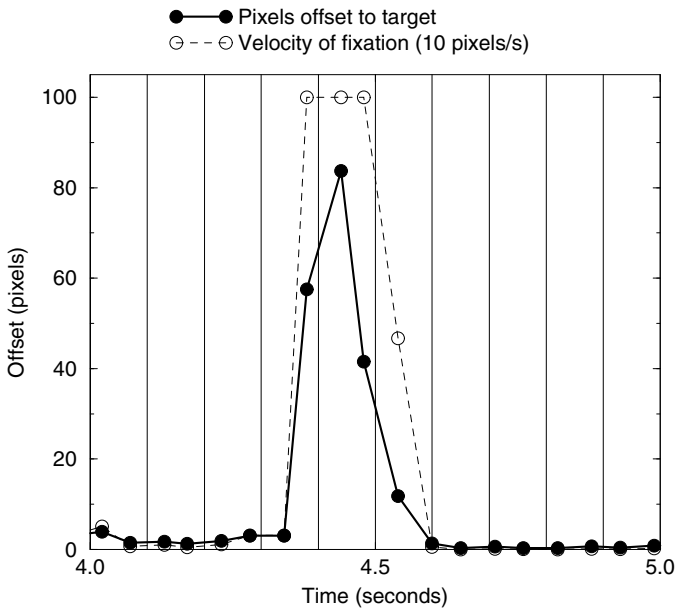
- Determining the RGB values of the different colours which belong to a target object and,
- distinguishing between areas of the same colour which belong to different objects (or the background)

The first problem is partly addressed by the method of model acquisition employed. The target object is isolated from its surroundings and the RGB vectors at each pixel are recorded and clustered (for the purposes computational efficiency) into a small number of *ideal* vectors (10-20) which are said to represent the input vector weights for the object when it is assumed to be under perfect control.

Input vectors at higher, additional levels are derived by examining a 3x3 area of the preceding level. Within this area the feature *types* are counted giving an input vector to the next level which represents the *number* of each of the features present (see figure 6.8). This process is repeated for each subsequent level. Adding these higher levels partly solves the second problem as the input vectors will be more specific to the target object than to others.

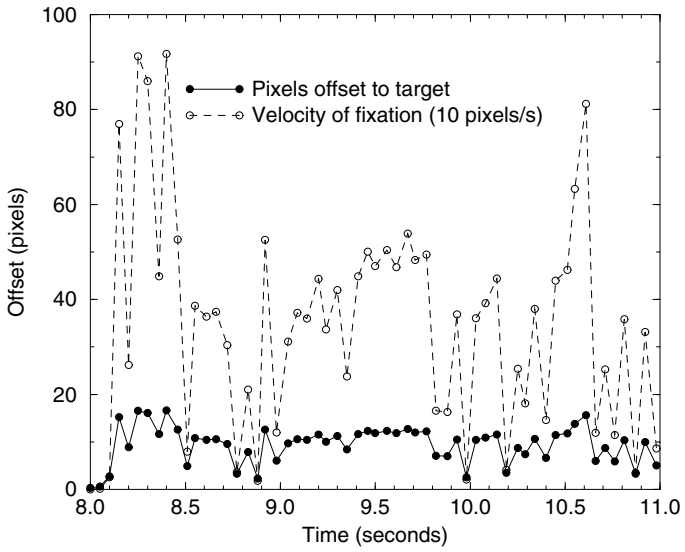


(a)



(b)

Figure 6.6: The pixel offset and output velocity for a typical tracking experiment.



(a)

Figure 6.7: Expanded area from figure 6.6.

6.6 Multi-level Control

Figure 6.9 shows a block diagram of a multi-level control system. Level 0 processes the basic RGB vectors and higher-levels (only one is shown) the vectors from the 3x3 pixel area. The input which is controlled at the highest level is the perception of the direction and magnitude of movement to the target. The intermediate, colour processing levels, it should be noted, are not actually controlling variables, but are assumed to deal with uncontrolled perceptions with previously organised input functions. In the current case of fixation control we are more concerned with the *location* of perceptions than with their values. In the *perception types* images (see figure 6.9) the non-white pixels show the areas of interest which have become activated for the particular target. The pixels are colour-coded according to the feature type, their magnitude indicated in the signals arrays. In this particular example the target is the blue face on the bottom-right of the main image The first of the perception types images shows many features activated spread over the scene. This indicates that, at this level, there are many features in common between the target and non-target areas. The second perception types image shows only a small cluster of activated features indicating that at this higher-level the target does not have any features in common with other areas. Segmentation of target dependent features allows us to selectively fixate the target by computing the fixation vector from the position

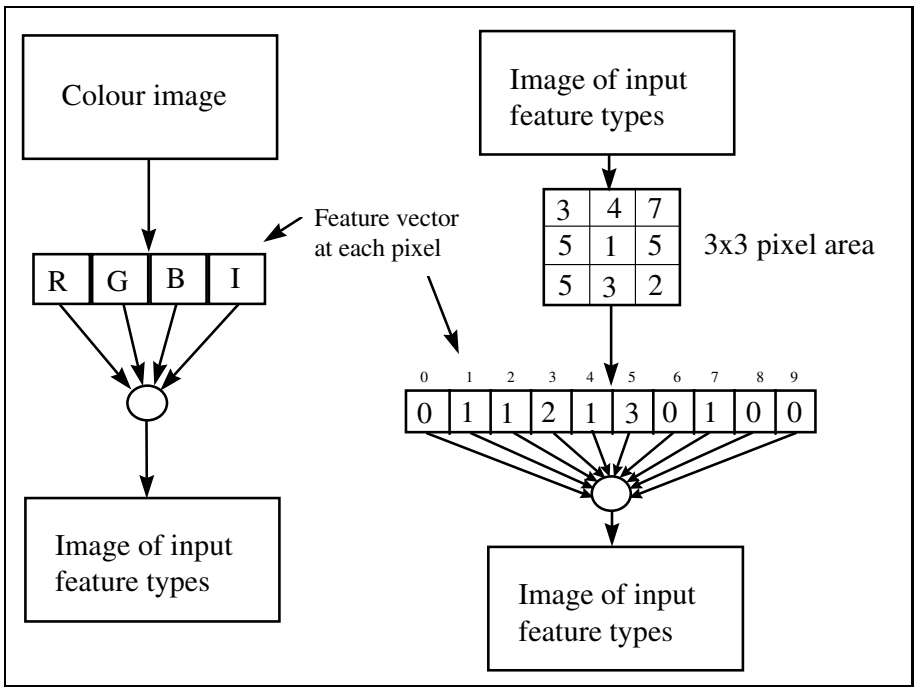


Figure 6.8: On the left is the input function for the lowest level, the RGB image. On the right the higher levels derived from a 3x3 pixel region of its preceding level.

vectors of the features.

Some preliminary results of the multi-level control system are shown in figure 6.10. Each row of images show the results of fixation for each of the Halloween mask targets, clockwise from top left. The columns, from left to right, show the results using levels 0, 1 and 2. In each case the starting position is the centre of the image and the cross-hair indicates the end position (which should be the nose of each face) with the dark line showing the course of fixation.

From the left column it can be seen that control, solely with level 0, is poor. Although fixation is made towards the correct targets, interference from background and extraneous signals adversely affect the fixation signal. Control which includes level 1 (centre column) is greatly improved, with fixation terminating, correctly, at the centre of the target face each time. Including another level (level 2, right column) does not seem to improve control further and in fact seems slightly worse. However, this is probably more to do with the fact that much of the signal is lost at this level, rather than with higher levels not being of benefit. Given the instability of the input signal at higher levels we limit the hierarchy to the lower two feature processing levels.

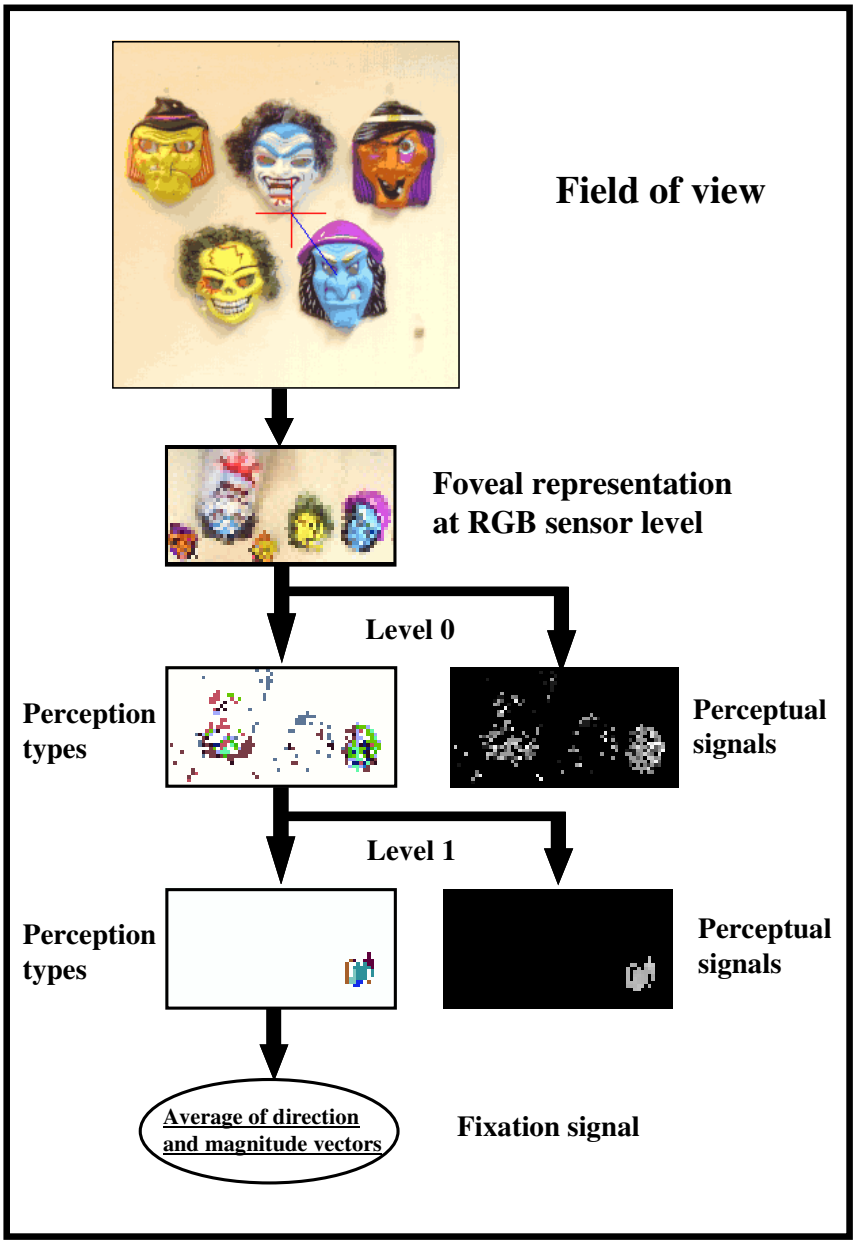


Figure 6.9: The two level colour processing control system used in our experiments. The outputs from these levels, of the magnitude and direction to the target, define the input to the highest level (fixation) control system.

Figure 6.10: Multi-level control. Each row shows the results of fixation control to each of the faces (clockwise from top left) at levels 0, 1 and 2.

6.7 Conclusions

The fixation system presented in this chapter performs well in real-time on simple lights and single coloured figures in synthetic and real scenes. Results have also been presented of some preliminary work concerning fixation to more complex, multi-coloured objects. Control improves with added levels in a hierarchy. Each level embodies signals which are more specific to the target object enabling the target to be more easily distinguished from its surroundings. The main problem is deriving the input functions and their weights. In the present scheme the signals at the higher levels are rather impoverished with much of the lower level inputs being lost sometimes resulting in erratic control. Future work would benefit from further investigation into the reorganisation and development of the input functions.

We have presented some preliminary results in off-line images which show that good fixation control, to complex objects, can be achieved with signals based only upon colour. Control may be improved further by including feature dimensions such as edges, motion and texture to add even greater discrimination.

Part III

Machine Vision

Chapter 7

An Integrated Vision System

7.1 Introduction

In this part of the thesis we describe the methodology and goals for building generic active vision systems with the general aim of interpreting dynamic scenes and actively responding to events. This requires the integration of diverse visual modules. The procedure and architecture we use follows the principles, structure and aims of the recent VAP project [27]. In this chapter we describe the VAP approach along with the particular architecture of our experimental integrated active vision system, and outline the corresponding visual modules used.

7.2 VAP Objectives

Research in computer vision has focused on developing procedures for solving discrete problems for extracting information about visual scenes [32, 54, 55]. The reasons for this tendency is partly due to people's proclivity to break a complex problems into smaller chunks, and partly due to the highly influential representational approach developed by David Marr [67]. The basic theory consisted of applying (discrete) visual algorithms on single images which extracted different types of information such as, edges, shape and texture. The results were placed on what Marr referred to as the 2 1/2 D sketch which was to be used as the basis for the construction of 3D representation of the scene and its constituents.

More recent research [2, 4, 8, 108] has argued that the complexity of visual problems can be significantly reduced by controlling both the sensor and the processing

resources, the hypothesis being that the spatio-temporal context plays a crucial role in predicting both *what* is going to happen next and *where* it is going to happen, enabling processing and sensor position to be first applied according expectations [71]. Arising from this paradigm shift towards visual integration and control are the following set of problems which need to be overcome, and which, broadly, form the basis of the objectives of the VAP project:

1. Control and scheduling of discrete knowledge sources.
2. Optimisation of knowledge source parameters.
3. Sensor control.
4. Control of processing resources.
5. Scene model maintenance.
6. Evaluation of expectations in dynamic scenes.

Some preliminary work has been done elsewhere for the control and scheduling of knowledge sources [30, 44, 61] and for the optimisation of parameters [95] and will not be addressed here. We do, however, address the remaining four topics.

The control of the sensor and processing gives rise to new conceptual problems in addition to the practical problems. As the sensor is mobile it becomes necessary not only to maintain a scene model in terms of the contents of the world of interest outside the current field of view but also in terms of the accurate location of those contents with respect to the sensor. The control of the sensor position depends upon the relationship between current and future events. The autonomous control of sensor and processing requires some in-built knowledge of such relationships. We propose to model such spatio-temporal events linguistically, by constructing grammars of the possible sequences of temporal scene events.

7.3 VAP Architecture

Figure 7.1 depicts the basic modules of the VAP vision system architecture. Previously defined models of objects are stored in the **object database** in terms of the relevant knowledge sources (colour, texture, geometric properties). The **scene description database** contains those objects from the database of possible objects

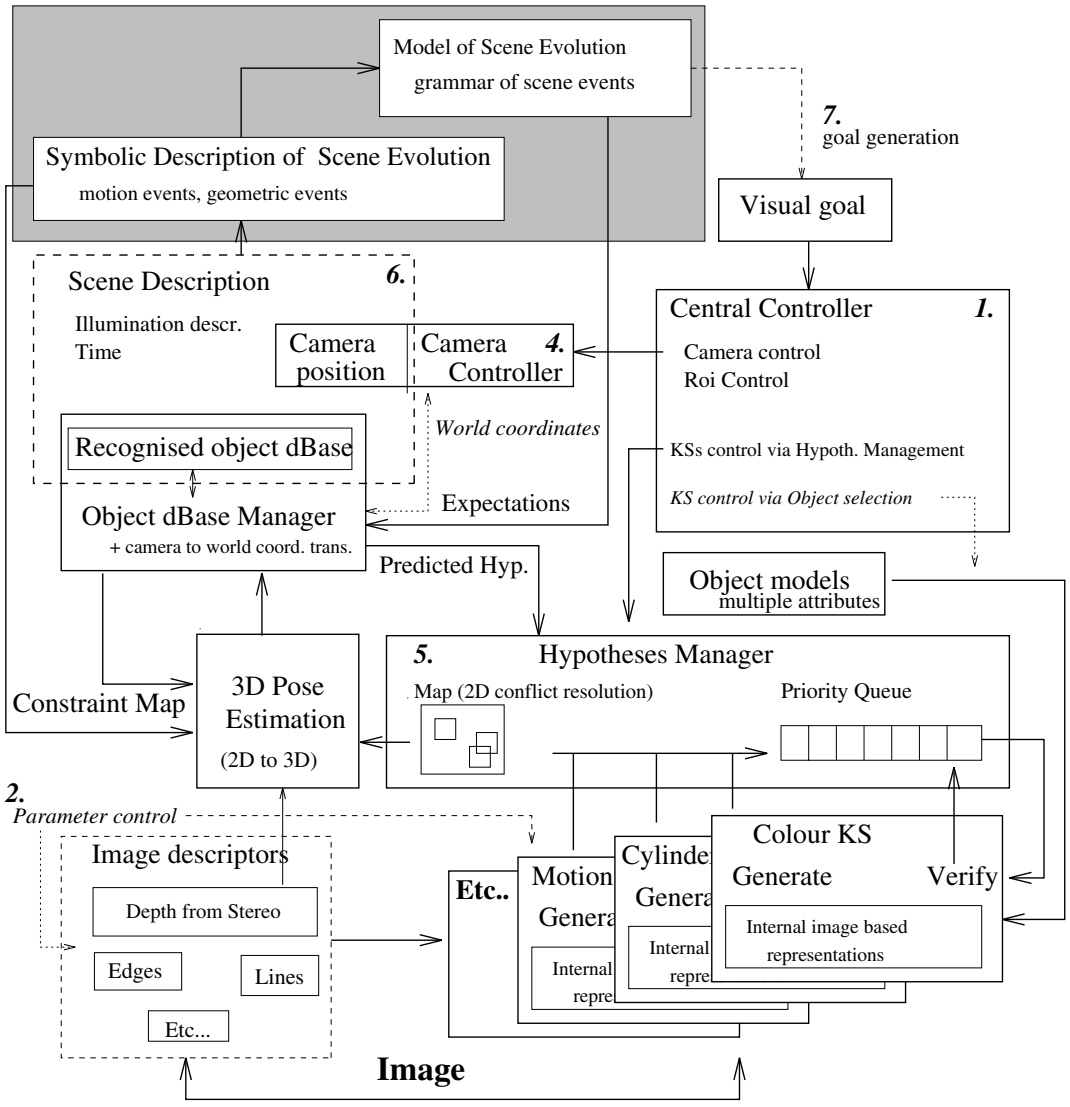


Figure 7.1: VAP Architecture

which are detected in the current scene. The information includes their location in a camera-independent coordinate frame, ie. their world 3D position.

The scene description reflects our knowledge about the environment at a given time and is continually updated according to the dynamic nature of the scene. The central controller governs the overall operation of the system by taking expectations from the **model of scene evolution** and issuing instructions to the **hypotheses manager**, **knowledge sources** and **camera controller**. The hypotheses manager combines the bottom-up information derived from the current scene, via the knowledge sources, with the top-down spatio-temporal expectations to assign a priority to the possibilities for both the camera look position and the current scene models.

7.4 Experimental System Architecture

The structure of the experimental vision system which is the subject of this part of the thesis is shown in figure 7.2. Incoming images are analysed for regions of interest defined by changes which have taken place in the environment, denoting objects which have been placed or removed. The outlines of all objects are extracted from the regions of interest and, after adjustment for camera pose, are matched against the object database. The scene description is then either updated or confirmed. According to the context, as defined by the grammatical model of scene evolution, the sensor is re-positioned and the sequence of matching through the object database is prioritised according to the probabilities assigned to the expected objects. Each of these modules are described in detail in the following chapters with the exception of two, which we briefly describe now.

7.5 Regions of Interest

Regions of interest (see figure 7.3b) are determined by comparison of the current image with a background image of a static tabletop scene. Any areas which show a significant chromatic difference [69] are likely to represent new objects or events and are, therefore, deemed interesting. The chromatic differencing results in a binary image indicating those pixels which have changed from the base image, according to the chosen threshold parameters. By connected components analysis the pixels are grouped together into separate, contiguous areas. Rectangular regions are then derived by finding the boundaries of the contiguous areas. Any regions which are

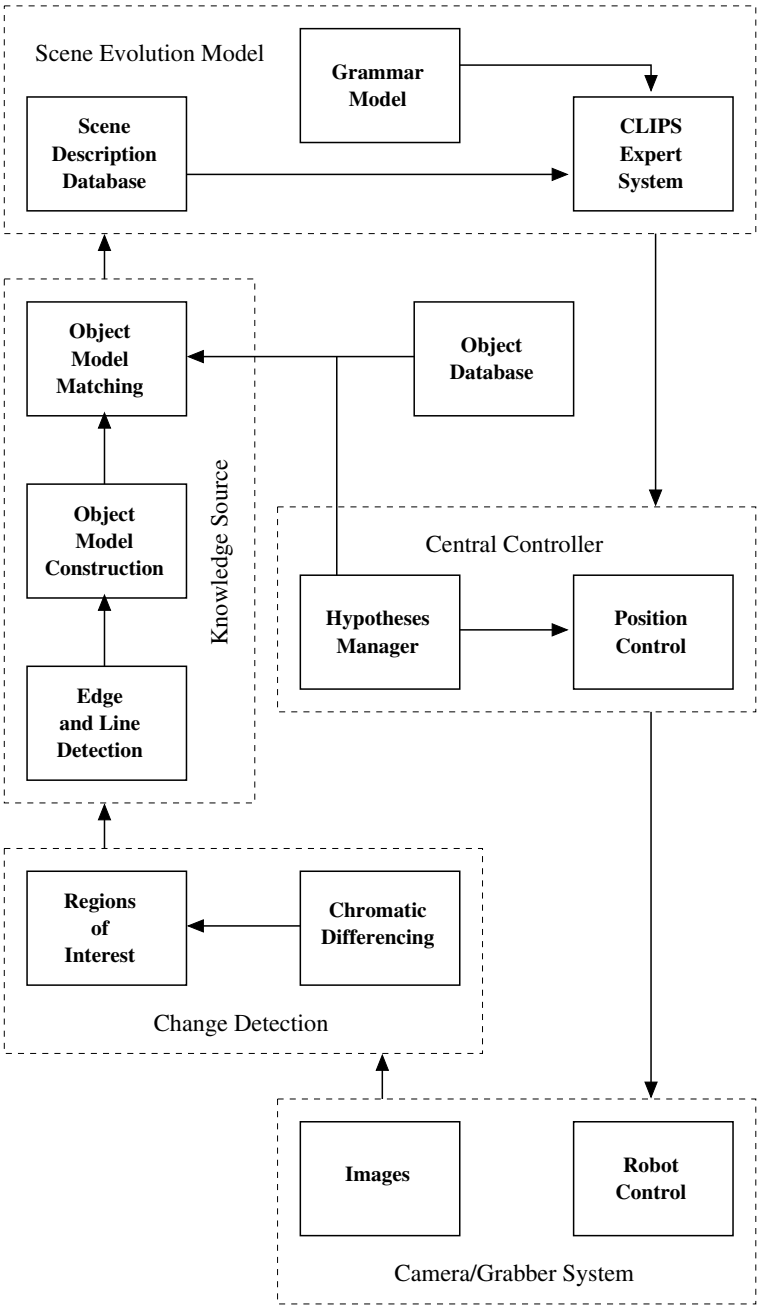


Figure 7.2: VAP inspired experimental architecture

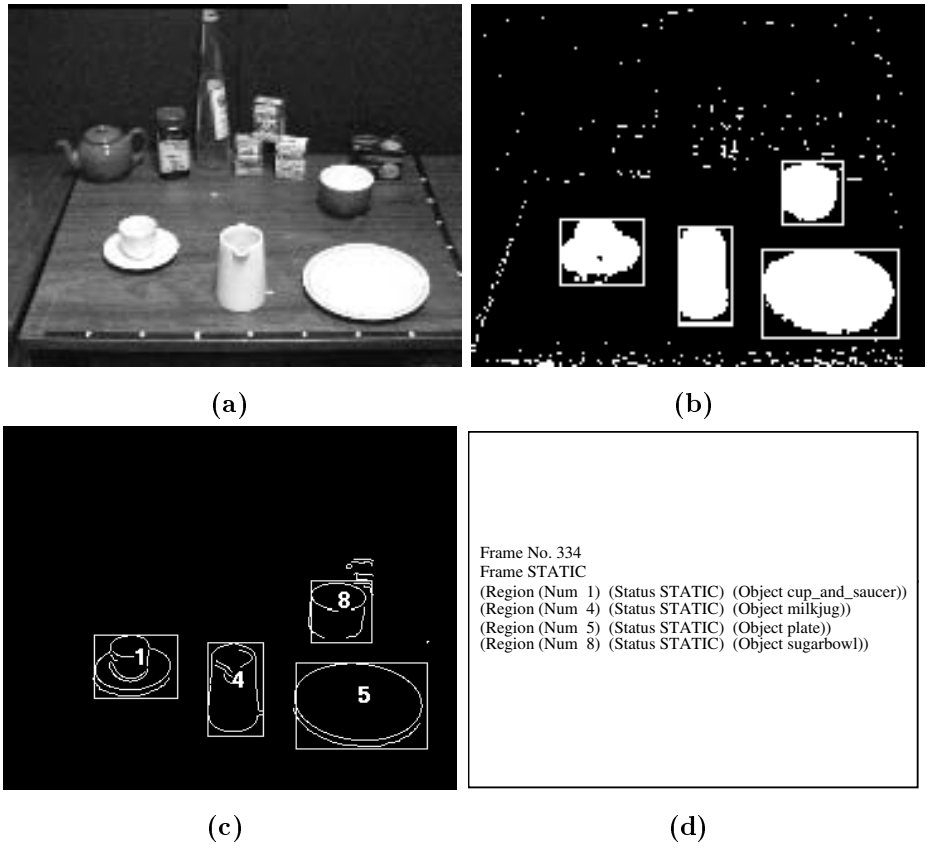


Figure 7.3: Processing steps a) One image in sequence b) Colour difference regions c) Edges within regions of interest d) Symbolic interpretation of regions

less than 100 pixels square are rejected as being too small to be interesting, most likely due to noise.

7.6 Cylindrical Object Recognition

The object recognition approach used in the present system has been reported in detail elsewhere [127]. It uses a dedicated recognition engine for each type of object that can be found in a breakfast scenario. In particular we can cope with plates, saucers, sugar bowls, cups and milk jugs. The recognition scheme assumes some prior knowledge and constraints. All objects must be placed on a common, flat ground plane. The transformation between the camera coordinate system and the ground plane coordinate system must be known (established through calibration). The recognition procedure adheres to the processing steps shown in figure 7.3.

Within each rectangular region which is output from the change detection stage the edges are extracted, by a standard Canny detector, and linked to form lines within the regions which represent the outline of any objects present. For each stored model in the database of known objects the model is transformed according to the pose estimation of the camera resulting in a list of pixels, in the image coordinate system, that would (exist) if the object were actually in the region in question. We then have two sets of pixels, one derived from the stored model. A *match* value is determined by summing the Euclidean image distances from each model pixel to the closest image pixel. The lower the value the better the match. The model which produces the lowest value which is below an experimentally derived threshold (see chapter 9) is accepted as the identification of the object. If the stored database is large, this process represents a major processing bottleneck, as it may be necessary to search through, and attempt to match, all the models before the correct one is found. In a later chapter we will see how this is overcome by prioritising the search according to predictions which rely upon a model of scene evolution.

7.7 Experimental set-up

All our experiments were carried out with a COSMICAR/PENTAX 25mm lens on a JVC TK1070E camera attached to a PUMA700 robot arm (figure 7.4). All processing was performed on a Silicon Graphics Power Challenge machine.

The images were captured with a Sirius image grabber. All software was written in

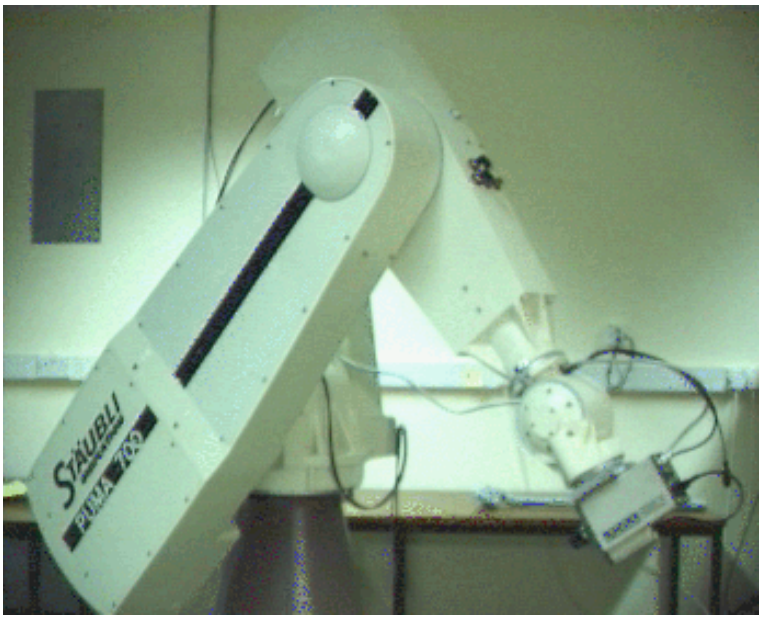


Figure 7.4: Robot/camera system

C++ making extensive use of the AMMA library classes developed at the Centre for Vision, Speech and Signal Processing.

7.8 Summary

In this chapter we have described the overall architecture for an integrated Active Vision System which is the goal of this research. Also described, briefly, were some of the modules, such as change detection and object recognition, we use which were the subject of research by others. We are not committed to using these particular techniques but envisage substituting different modules for the purposes of evaluation and comparison.

The remaining chapters of this part of the thesis fill in the missing parts from the whole resulting in a working, interactive vision system. The next chapter provides a solution to the enduring problem of camera calibration, necessary for 3D object recognition. Demonstrated in chapter 9 is the benefit of accurate calibration data for maintaining the position of an object with a moving camera. Chapter 10 describes the full system with particular reference to the modelling of the temporal evolution of a scene.

Chapter 8

Camera Calibration

8.1 Introduction

One of the major goals of Computer Vision is akin to a surveying task of measuring the 3D position of objects by optical means. The way this is done can generally be described as the task of equating the position of objects in acquired images with their 3D world coordinates. If a stereo vision setup is used one solution is to find the points in each image pair which correspond to the object and compute the object position, by triangulation, from the known orientation of the two cameras. However, in the situation which we are examining here, of a single camera, the triangulation method is not available. If only a single camera is to be used the accuracy of the recovered 3D values depends upon the degree to which the parameters of the camera and its relationship to the world are known. It is necessary to determine, therefore, to a high degree of accuracy the *projective transformation* between the image and the world, which requires the intrinsic and extrinsic parameters of the camera. The intrinsic parameters we are concerned with are the focal length, the image centre and the radial lens distortion factor. The extrinsic parameters are the rotation and translation values which define the relationship of the camera coordinate system with respect to that of the world. The process by which these parameters are derived is known as camera calibration.

A particular goal of ours is to be able to maintain the known world position of an object even though the camera pose changes, dramatically and arbitrarily within a working area. This requires calibration data which is reliable enough to predict, from a new view, the object image position to within a few pixels. Any more than that and we run the risk of attempting to match our models against erroneous or

background object edges.

Calibration has been performed reliably from static viewpoints allowing accurate estimation of 3D world coordinates [43, 47, 56, 58, 112]. However, problems arise when the camera pose is changed. Due to the large number of free parameters compared to the number of constraints, the calibration parameters are biased to fit the data for the view from which they were determined. In our experience the projection errors based on camera parameters established from a single view increase rapidly as a function of the distance of the viewpoint from that at which the calibration was performed. Re-calibration at different views gives unstable estimates for both intrinsic and extrinsic parameters. In fact the estimates are so unstable it makes little sense to consider these parameters 'intrinsic'. The instabilities have often been attributed to changing environmental conditions (e.g. temperature) or mechanical non-rigidity of the camera system. We present a new method for camera calibration which overcomes the over-fitting problem and yields very stable estimates of the camera parameters.

The standard way to increase the precision of calibration from a single view is, either to use a chart which covers the whole field of view, or to use cleverly designed 3D objects. With our procedure only a simple planar chart is required and we are able to determine parameters which can be used at a later date or perform the calibration experiments in conjunction with 3D recognition experiments such as the maintenance of scene models. We extend Tsai's [112] calibration method to optimise the parameters over many views resulting in values which rapidly converge during the procedure and remain consistent over time. The multi-view method requires and exploits the knowledge of the motion of the camera system.

Our basic strategy can be summarised as follows,

1. A robot/camera system is moved to a large number of positions with its line of sight oriented roughly towards a calibration chart.
2. At each pose the image positions of feature points on the chart are detected.
3. The re-projected positions of the points are computed from estimated values of the camera parameters and compared with the detected centres.
4. The total error over all views is minimised to obtain the optimal values of the intrinsic parameters.

In the reported experiments the intrinsic parameters converge to stable and consistent values. The focal lengths, for example, determined from many executions of the

procedure agree to 0.28 mm. The resulting calibration data is tested successfully in poses not used in the initial procedure.

We present a series of experiments which examine the behaviour of the derived values of the camera parameters in a variety of situations ranging from a single set of parameters at a static view to multiple sets of parameters at multiple camera poses. We conclude that the best way of determining the effective parameters is to optimise a single set of the intrinsic and extrinsic parameters over multiple views of the scene. In this chapter we first describe the camera model assumed sufficient for our purposes, some of the issues in camera calibration and optimisation and the chart detection technique. Section 8.5 outlines the experiments performed. The results are also presented in section 8.5 and the conclusions discussed in section 8.6.

8.2 Camera Model

The camera model we employ is the basic *pinhole* model with radial lens distortion as used by Tsai [112]. The goal of camera calibration is to recover the *projective transformation*, such that 2D image points can be converted to their 3D world counterparts, and vice versa. Four coordinate systems need to be determined in order to compute the transformation; in terms of the world, the camera, the camera sensor plane and the image. Three sets of data are required, the extrinsic and intrinsic parameters and details concerning the camera sensor.

The extrinsic parameters define the position and orientation of the camera with respect to the world, and comprise the rotation matrix (R) and the translation vector (T), such that a point in world coordinates (x_w, y_w, z_w) can be defined in terms of its corresponding camera point (x_c, y_c, z_c) as,

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$

The 3 x 3 rotation matrix can also be expressed as three parameters of rotation, the roll, pitch and yaw angles around the z, y, and x axes, respectively [38]. With the three elements of the translation vector T_x, T_y and T_z gives six extrinsic parameters in the calibration implementation used in our procedure [124]. Transformation with the extrinsic parameters give the coordinates of a world point in terms of the camera coordinate system with the origin at the optical centre and the z axis along the camera's optical axis.

There are many possible intrinsic parameters, including different types of lens distortion, which could be used in the model. For compatibility with other research and available software [124] we constrain ourselves to four intrinsic parameters, the focal length, f, the image centre C_x and C_y and a radial lens distortion factor, κ, which are required for the remaining stages. The next step is to convert from the camera coordinate system to the plane of the CCD sensor.

$$\begin{aligned} X_u &= f\frac{x_c}{z_c} \quad \text{and} \\ Y_u &= f\frac{y_c}{z_c}, \end{aligned}$$

where X_u and Y_u are the coordinates on the undistorted (ideal) sensor plane.

Due to geometric lens distortion the sensor coordinates require adjustment with the lens distortion factor giving the true sensor position of the point,

$$\begin{aligned} X_d &= \frac{X_u}{(1+\kappa\rho^2)} \quad \text{and} \\ Y_d &= \frac{Y_u}{(1+\kappa\rho^2)}, \end{aligned}$$

where $\rho = \sqrt{X_u^2 + Y_u^2}$.

Finally, the image point is derived by,

$$\begin{aligned} X_i &= d_x^{-1} X_d s_x + C_x \quad \text{and} \\ Y_i &= d_y^{-1} Y_d + C_y, \end{aligned}$$

where d_x and d_y are the distances between the centres of the sensor elements and s_x is a scaling factor compensating for any uncertainty in the timing of the start of the image acquisition. These three camera sensor parameters are assumed to be constant, the values of d_x and d_y are given by the camera manufacturer and, for our present purposes, s_x is taken as 1.0.

8.3 Calibration and Optimisation

Camera calibration involves finding the optimal values for the parameters which correspond to the minimum error between points in an image observed from the world and points re-projected into the image plane from the projective transformation and

the known world coordinates of those points. The calibrated camera parameters are unlikely to be the same as those specified by the manufacturer of the camera. The parameters we derive are the *effective* parameters of the pin-hole camera model as opposed to the *true* parameters of a thick-lens camera model. As images are 2-dimensional any point within an image translates to a line in the world coordinate system. Therefore, there is not a one-to-one correspondence between image and world points. To overcome this problem we constrain all world points and objects to lay on, or to be related to, a known common plane. The intersection of the line with this plane then gives us a point which corresponds to the image point.

The most basic calibration situation comprises a single, static view of a set of points lying on a common plane (coplanar). However, there could be many variations of the calibration parameters which satisfy a specific view giving an acceptable re-projected error. If the world points are coplanar there can, particularly, be considerable ambiguity in the estimated f, T_z values. This can be seen in table 8.1, section 8.5.2, where the values of the f and T_z parameters vary quite considerably depending upon the circumstance. Note, however, that the *ratio* of the values remains constant indicating that a parameter minimum has been found for that particular view, but which is unlikely to generalise to different views.

To some degree the f, T_z ambiguity can be alleviated by using non-coplanar calibration points [112] (which requires some mechanism for moving the calibration points in space or for correlating points on different planes) and the intrinsic parameter bias can be alleviated by calibration from multiple poses [93]. We draw on these two concepts and derive a procedure which involves taking a series of views of a fixed calibration chart. The views are defined by known camera movements. All views are used jointly to determine the camera intrinsic parameters and the initial extrinsic parameters. By moving the camera in different planes (instead of the calibration points) and calibrating at a large number of positions which cover the desired working environment of the robot/camera system, we obtain very stable estimates of the calibration parameters.

The approach we use, however, is not the simple averaging of the values of the intrinsic parameters computed at the different views. Our goal is to derive values which result in a small error between the predicted and actual positions of objects when projected into the image. Averaging may or may not achieve this. The value of our approach is that we directly minimise the quantity in which we are interested, the re-projected error.

Along with Tsai's [112] camera model we also employ his calibration algorithm using

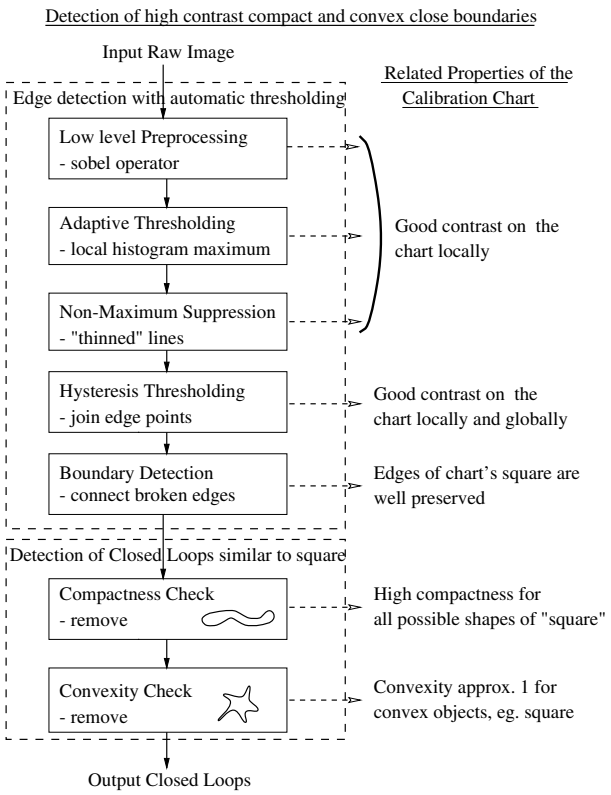


Figure 8.1: Chart detection algorithm

Willson's [124] implementation. There are five stages of parameter optimisation. The purpose of the first four are to derive parameters which are close to a solution in preparation for fine-tuning by the fifth and final stage which is the Levenberg-Marquardt optimisation algorithm. We use the full five stages only for the initial pose in our experiments. Subsequently, for pre-optimisation estimates, we use the same intrinsic parameters and derive the extrinsic parameter estimates by chaining the initial values with the known movements of the camera. As these estimates are derived from the initial optimised values they are sufficiently close to the solution that we then dispense with the first four optimisation stages and only use the fifth.

8.4 Chart Detection

Successful camera calibration relies heavily on the accurate and consistent detection, in images, of precisely known world points. Points extracted from the image of conventional objects are notoriously unreliable. The object edges extracted from images at different poses may actually refer to different world points. There is

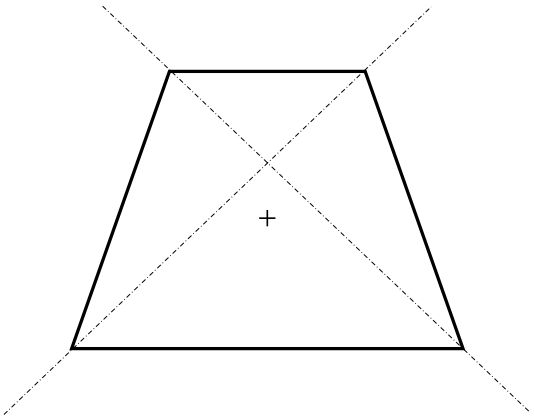


Figure 8.2: Comparison of centre of gravity with true centre of square. The trapezoid is the outline of a square seen in perspective. The true centre is shown by the intersection of lines (dashed) through the corners of the square, whereas the centre of gravity is represented by the cross.

also the added problem of finding the correspondence between points in different images. To overcome these ambiguities we use as world registration marker points on a calibration chart. In addition we apply a fast and robust technique for the accurate detection of the centres of the squares of the calibration chart. The chart detection technique is founded on the premise that the image gradient for pixels on the chart will be highest locally and close to the global maximum. The resulting edge strings are closed, compact and convex. The main stages of the method involve the processing of local image gradients, the feature extraction of the centre of gravity of each square and the linking of possible nodes to segment each individual chart. Figure 8.1 shows a block diagram of the entire process and more details can be found in Soh et al [105].

The types of charts used here consist of either square or circular elements. Both the number and size of the elements can vary. One possible source of error with the method described above of detection the centre of gravity of each chart element is that the centre of gravity may not actually be the centre of the chart element, depending upon the angle of incidence. Figure 8.2 shows the outline of one of the square chart elements seen in perspective. The centre of gravity is the point which is equidistant from the four corners. Whereas, the *actual* centre of the square is the intersection of the diagonals. For this reason we also investigate the use of the detection of the *corners* of the chart elements. The corners are defined by the intersections of lines which are *fitted* to the edges of the squares.

Figure 8.3: Example view of calibration chart

8.5 Calibration Experiments and Results

The reliability of the parameters of the projective transformation can be determined by examining the *re-projected error*. Two sets of pixel values for the chart elements can be derived. One set is constituted by the values resulting from the chart detection process. From these values, along with the known dimensions of the chart, the camera parameters are computed. The other set can be obtained by re-computing the (predicted) image positions from the camera parameters and the known dimensions of the chart. The re-projected error is the mean of the Euclidean distances between the pixel values for the corresponding points in each set.

8.5.1 Experimental procedure

In the multi-view experiments, the robot/camera system was placed, every 200 mm, in a 800x800x600 virtual 3D grid, less the poses that were unreachable or violated the camera protection protocols. The direction of sight of the camera was always oriented towards a fixed 5x5 calibration chart (figure 8.3). It was ensured, however, that the chart was off-centre in the image so that the lens distortion factor would have an effect within the camera model equations, otherwise it may not be optimised. The camera was placed in 50 different poses for the multi-view optimisation procedure. The resulting camera parameters were then tested in 60 other poses, half of which were outside the initial 3D grid and half inside.

In order to test the temporal stability of the derived intrinsic parameters the experiments were repeated several times a day over the period of two weeks. In case the temperature of the hardware had any effect on the results the experiments were

Corners	**Focal length (mm)**		**Tz (mm)**		**f/Tz Ratio**		**Error**	
Chart Type	Mean	σ	Mean	σ	Mean	σ	Mean	σ
4x4 20	14.54	4.38	1782.0	480.16	0.0081	0.000227	0.267	0.011
4x4 40	22.46	0.81	2653.9	86.33	0.0085	0.000033	0.295	0.007
5x5 40	20.91	0.71	2515.5	86.00	0.0083	0.000023	0.276	0.007
8x8 20	21.87	0.82	2610.1	96.17	0.0084	0.000023	0.206	0.009
8x8 40	20.48	0.39	2564.5	49.20	0.0080	0.000047	0.248	0.007

Circles	**Focal length (mm)**		**Tz (mm)**		**f/Tz Ratio**		**Error**	
Chart Type	Mean	σ	Mean	σ	Mean	σ	Mean	σ
4x4 20	53.28	25.47	6030.3	2813.92	0.0088	0.000103	0.099	0.010
4x4 40	20.29	0.66	2426.9	73.16	0.0084	0.000023	0.085	0.008
5x5 40	20.58	0.30	2475.0	30.75	0.0083	0.000017	0.090	0.004
8x8 20	20.19	0.74	2438.1	78.31	0.0083	0.000055	0.129	0.004
8x8 40	21.17	0.11	2648.5	13.12	0.0080	0.000007	0.151	0.003

Table 8.1: Results of f and T_z for 100 calibrations from the same view.

carried out first thing in the morning and after the equipment had been left powered up all day.

8.5.2 Single-view calibration

Chart Type

An important relationship between calibration parameters is that between the focal length, f and the third element of the translation vector, T_z. As mentioned in section 8.4 there are different types and sizes of charts that can be used for calibration purposes. In our first experiment we fixed the camera in one position with the line of sight of the camera at an angle of approximately 45^o to the plane of the table and performed the calibration in order to determine the effect of the different charts on the derived values for f, T_z and the f/T_z relationship. The calibration was repeated 100 times for each chart and the values of the derived focal length and T_z recorded.

Table 8.1 shows the results for some statistics of the focal length and T_z as derived from Tsai's [112] five stage calibration process. The two tables show the results for the extraction of corners from square charts and for the centres of gravity from circular charts. The chart type indicates the number of elements in a chart and

the dimension between elements, in millimetres. For example, the chart type **8x8 20** means a chart consisting of an 8x8 grid of elements (64 in total), the centres of which are 20 mm apart.

There are a number of things to notice,

- the statistics, σ, improves with the number and size of the chart elements
- there is a large difference in the mean values of the focal length for the different charts indicating significant instability in determining the values from a single view
- the f/T_z ratio is constant over all situations. The reason for this is that from the single pose there is a wide range of values of f, along with its appropriately scaled dependent variable T_z, which can satisfy the minimisation process. In other words, there are many local minima for the values of the camera parameters which will produce an equally low error.
- the mean re-projected error is fairly constant across chart types, approximately 0.25 pixels for corners and 0.11 for circles.

To summarise, in our experience the results obtained from single-view calibration are neither stable nor repeatable. Considering that the re-projected error *is* stable, we feel there is strong evidence for the conjecture that the camera parameters are biased to fit the extracted coordinate data for that particular view. If the view was slightly different there may be a large effect on the parameters.

Scale of chart and intrinsic parameters

It has been reported elsewhere that the stability and precision of derived parameters depends upon the size of the chart within the image. The usual conclusion is that the reliability of the parameters improves as the size of the the chart increases. To investigate this claim we carried out an experiment where we translated the camera along its line of sight every 100 mm in a range of 1800 mm. The size of the chart, therefore, decreased with increasing distance from the chart. At each pose calibration was performed 50 times and the mean and standard deviation of the derived focal length were recorded. Figure 8.4 shows the mean and standard deviation plotted as a function of distance from the starting point. Experiments were performed for different size charts as well as for the corners of square elements

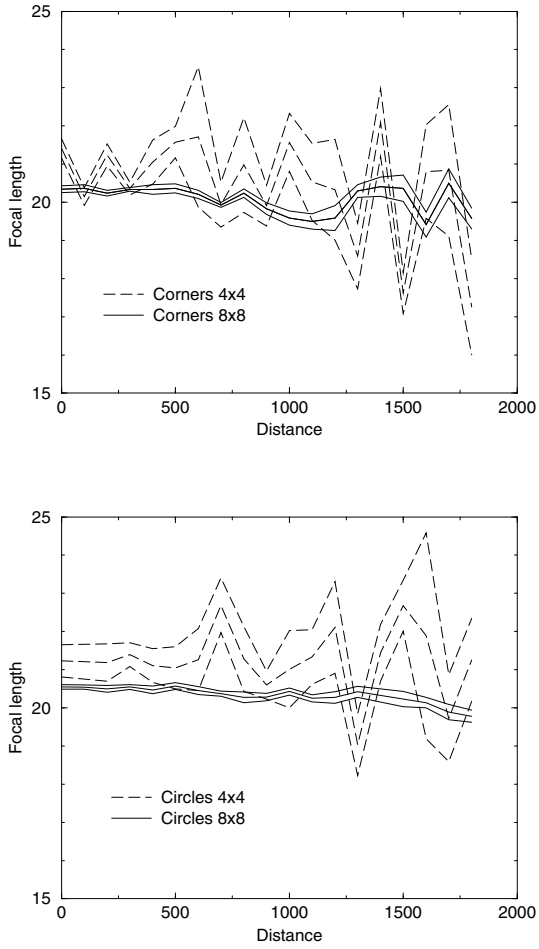


Figure 8.4: Focal length, f, and standard deviation, σ, plotted against scale (distance from chart). There are two sets of three lines in each graph. Each set represents $f+\sigma$,f and $f-\sigma$, from top to bottom. At each pose (scale) 50 images were taken and the calibration parameters derived from each image. The mean and standard deviation of f was computed for the 50 measurements. This process was carried out for two different sizes of chart elements, as represented by the two sets. The upper graph shows the results for detected corners of square chart elements and the lower for circular chart elements.

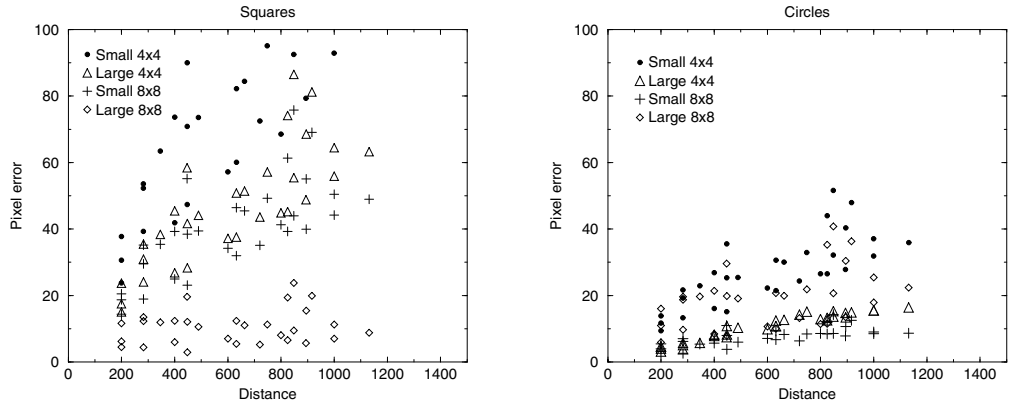
and the centre of gravity of circles. Two points are clear from the graphs. One, that the instability of the mean focal length value increases as a function of scale. The second, that the variance of derived values also increases. These results confirm the earlier claim and suggest that the best results can be obtained when the chart covers the entire field of view. However, it is often both inconvenient and impractical for the chart to fill the scene for one, let alone multiple views and we wish to move towards the situation where the chart holds a less conspicuous position in the scene. The remaining experiments address this issue in more detail.

Single-view generalisation

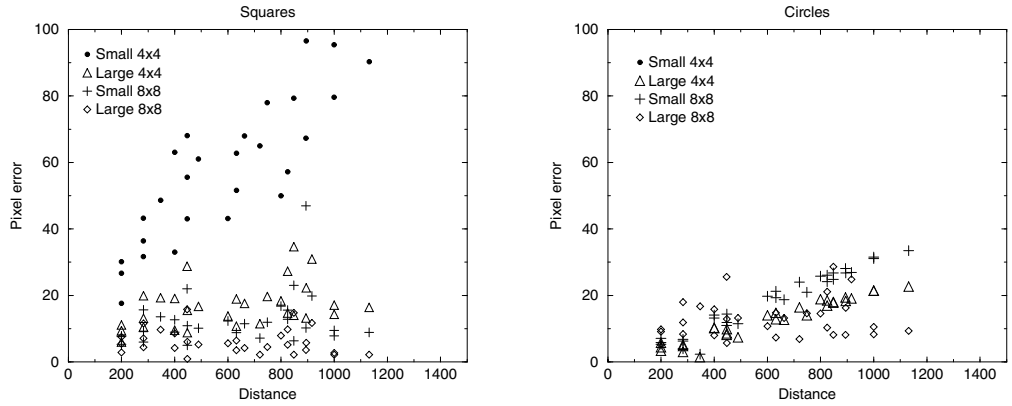
We have seen how the calibration parameters are affected by the chart size and type and that reliability improves with the size and number of chart elements. This knowledge, so far only applies to static, single views of the camera. We would like to be able to obtain calibration information that would be appropriate for the same scene but from different poses. This set of experiments investigates how the calibration data, taken from a single view generalises to other views.

For the grid of poses described in section 8.5.1 the calibration data was obtained for the initial pose only. The robot arm was, subsequently, moved to 30 of the other poses. At each pose the extrinsic parameters were computed from those of the initial pose and the known relative movements of the robot. Along with the intrinsic parameters from the initial pose the re-projected error was calculated at the subsequent poses in order to determine how well the single view calibration data generalises to *different* poses.

As can be seen from figure 8.5 different types and sizes of charts were tested and the re-projected error was plotted against the distance from the initial pose. To test the repeatability of the results the same experiment was performed twice, in figure 8.5a and b. Although, a general trend of size of error to distance can be seen the more pertinent result is the *size* of the errors. Although the bigger charts produce smaller errors they are still unacceptably large and could be anything up to 40 pixels. Furthermore, the results of the two executions of the experiment were *not* consistent, showing very different values. For example, in the first experiment (8.5a) the small 8x8 circles chart has smaller errors than the large 8x8 chart, a situation which is reversed in the second experiment (8.5b) and in the second experiment the errors from the small 4x4 circles are so large as to be off the graph. Our goal is to develop a technique which not only produces relatively small errors but errors that



(a)



(b)

Figure 8.5: Re-projected error of single view calibration applied to new views. Corners of square chart elements and centres of gravity of circle chart elements are used as features.

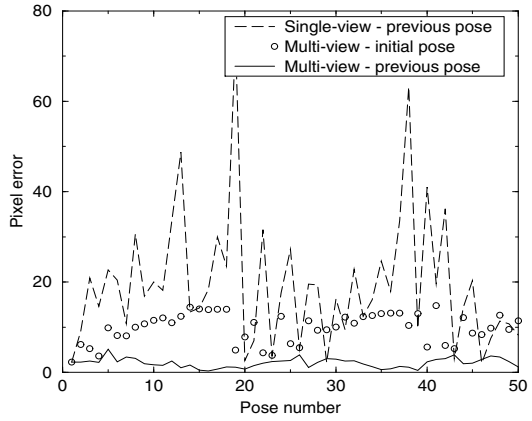
have little variation over the volume of space to be calibrated and that can be shown to be repeatable.

8.5.3 Multi-view calibration

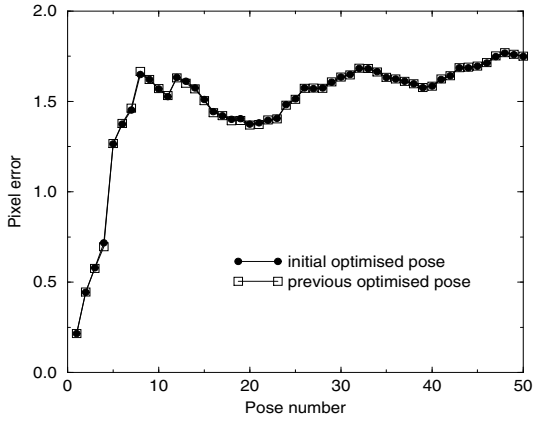
The purpose of the multi-view calibration experiments is to derive a set of intrinsic parameters which can be used at any pose of the camera and which produce a consistently and acceptably low re-projected error. In these experiments we optimise the camera parameters by minimising the error not just for one single view or for many independent views, but over many views *simultaneously*. The rationale being that we are directly minimising the goal quantity, the error over multiple views, and that the computed parameters will converge to general values which will produce a similar minimum error value from *any* view.

The main elements of the procedure for the multi-view calibration experiments are as follows;

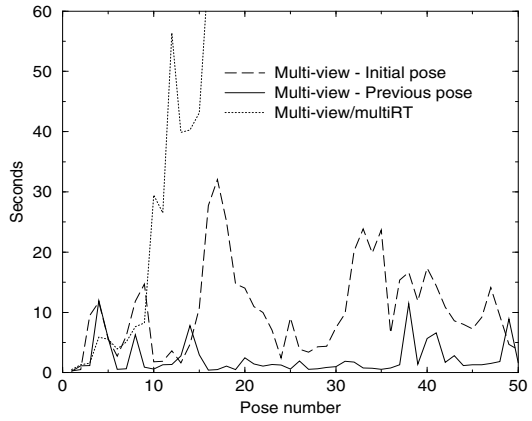
- At each pose an image of the chart scene (figure 8.3) was taken
- Within the image the corners of the chart elements were detected using Soh et al's [105] robust chart detection system, and recorded.
- At the first position, initial values for the extrinsic and intrinsic parameters were derived from Willson's [124] implementation of Tsai's [112] calibration algorithm.
- At each subsequent position the parameters were optimised to reduce to a minimum the error over all views.
- The current extrinsic parameters for each subsequent pose were computed from the initial extrinsic parameters chained with the known relative movements of the camera.
- The predicted positions of the chart in the image were determined by projecting the known world coordinates of the chart into the image with the current extrinsic parameters and the overall intrinsic parameters.
- The re-projected error quoted is the mean of the distances between the re-projected image points and those from the chart detection method. However, the actual values used in the optimisation process are those projected onto the sensor plane.



(a) Average pixel error at current pose re-projected from estimated values of camera parameters

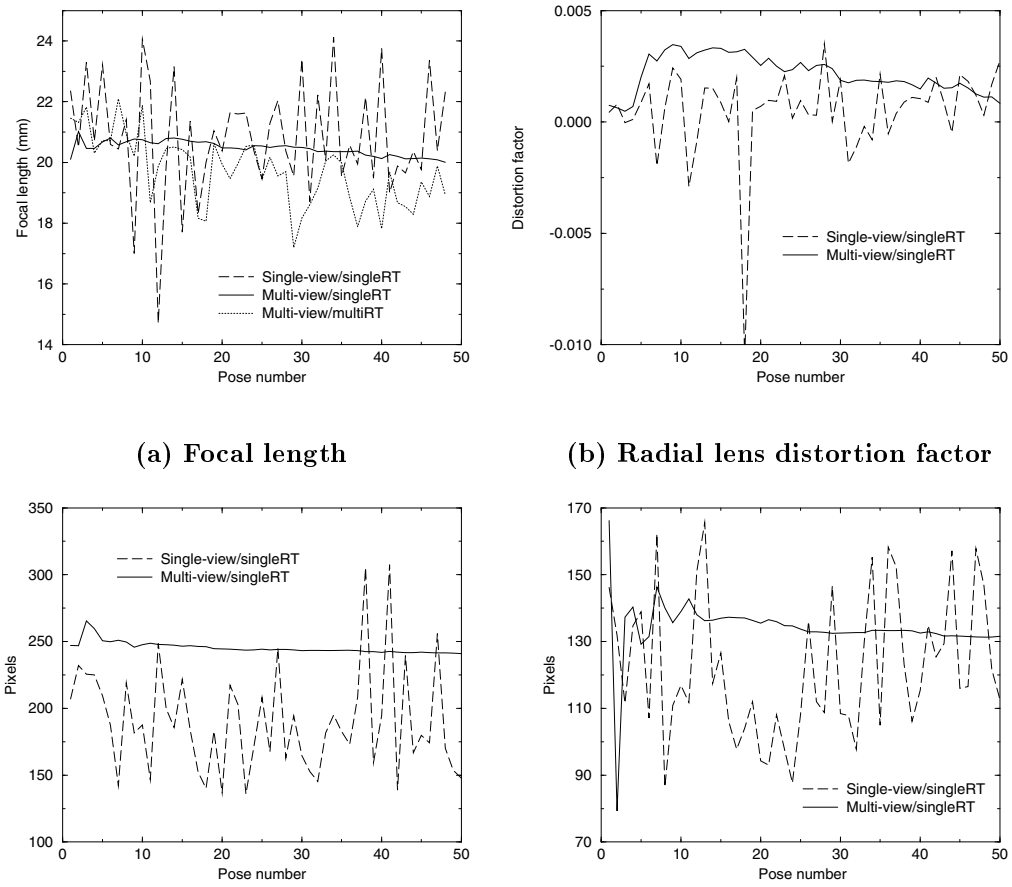


(b) Average pixel error over all poses re-projected from optimised values of camera parameters



(c) Processing times for the three optimisation procedures

Figure 8.6: Re-projected errors from estimated and optimised values of camera parameters.



(a) Focal length

(b) Radial lens distortion factor

(c) Image centre X

(d) Image centre Y

Figure 8.7: Typical behaviour of intrinsic parameters during a multi-view calibration experiment.

To see the relationship between different calibration processes various re-projected errors for pre-optimised estimates of the camera parameters (rotation and translation, **RT**) were computed and compared. There were three single RT estimates and one multiple RT estimate, as follows:

Single-view/singleRT The preceding pose single-view data is chained with the last camera movement. The aim is to establish how the data from one view generalises to the next.

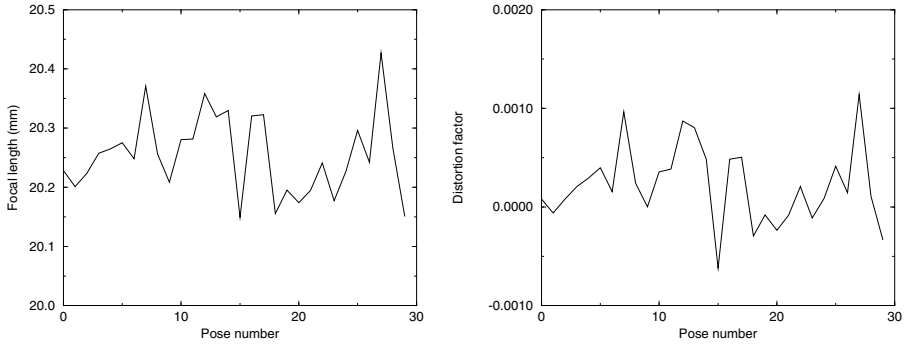
Multi-view/singleRT (Initial pose) The RT of the initial pose is obtained prior to any optimisation. For each subsequent pose the starting point of optimisation for the camera parameters are be the same.

Multi-view/singleRT The RT of the initial pose is determined *after* optimisation at the preceding pose. Essentially the optimisation performed from this estimate is over the same variable space as the previous estimate but should be closer to the optimum, and hence take less time to reach the solution.

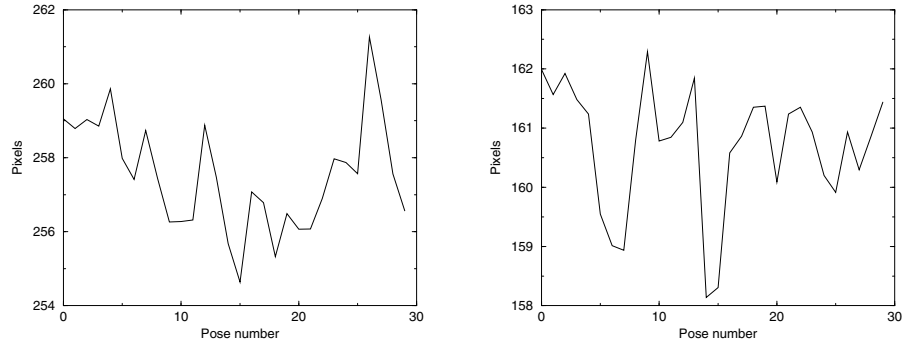
All the experiments discussed so far have involved optimising ten parameters, the four intrinsic and six extrinsic parameters. However, these all rely upon knowing the camera movements. As an additional experiment we remove the relative constraints between the poses and optimise the RT parameters at *every* pose, along with one set of intrinsic parameters. In other words, $4 + 6n$ parameters, where n is the number of poses.

Multi-view/multiRT The estimates for the RT at each pose are taken from the first pose chained with the relative movements.

From the above estimates the optimisation process was carried out at each of the 50 poses. We now look at the results. The results of errors (in pixels), re-projected just into the current view, from the three **singleRT** estimates of the camera parameters are shown in figure 8.6a. The circles indicate the errors when the estimate is taken from the calibration of the preceding pose. The dashed line is the estimate taken from the initial pose. Both show large and erratic errors in contrast to the third (solid) line which represents the estimate from parameters optimised over all previous views. This confirms our expectations that sets of parameters derived from separately calibrated single views are inconsistent and supports the view that the parameters are biased towards the data.



(a) Focal length **(b) Radial lens distortion factor**



(c) Image centre X **(d) Image centre Y**

Figure 8.8: Distribution of intrinsic parameters over all multi-view/singleRT experiments

Figure 8.6b shows the errors after optimisation for the **multi-view/singleRT** estimates from the initial pose calibration and that from the optimised values from the previous pose. Both are virtually identical indicating a similar minimum was found, however the processing time for the latter was generally shorter than for the former, as shown in figure 8.6c, indicating that the minimum was closer to the solution. The significance of this result is that the accumulated optimised values of the camera parameters give a much better estimate when extrapolated into a new pose, indicating that the parameters are getting closer and closer to their optimal values with each additional pose.

Also shown in figure 8.6c are the processing times for the optimisation process for the **multi-view/multiRT** estimate, which rapidly increases after about ten poses. This type of estimate and optimisation may be useful in a situation where the camera movements are not reliably known, such as with a hand-held camera. The processing time increases rapidly because the number of parameters being optimised increases making a minimum value much harder to find. It would be impractical to use this method for on-line calibration, however depending upon the accuracy required it may be sufficient to optimise over a small number of views, keeping the processing times to a manageable level.

The results for the optimised values of the focal length for the **multi-view/multiRT** in a single experiment over 50 poses are shown in figure 8.7a. Compare this with the other values shown. Although more stable than the erratic single-view calibration data the best performance is delivered by the multi-view optimisation which is constrained by known robot movements. The other graphs in figure 8.7 show similar results for the other intrinsic parameters. In all cases there is little consistency across the values derived from single-view calibration from different poses, whereas the multi-view values quickly converge and remain stable.

Figure 8.8 shows the plots of values for the intrinsic parameters for all the 30 experiments with some statistics shown in table 8.2. It is clear that the values obtained from the multi-view procedure are consistent, repeatable and show a low variation over the two week period of the experiments. Compare the mean values and variation of the focal length in figure 8.8a and table 8.2 with those in figure 8.7a (for the single-view) and table 8.1. The latter are unstable, non-repeatable and erratic in comparison.

Figure 8.9 shows an example of the re-projected errors at the test poses, those not used in the calibration process. There is little difference between the errors for the inner and outer poses indicating that the intrinsic parameters found are suitable

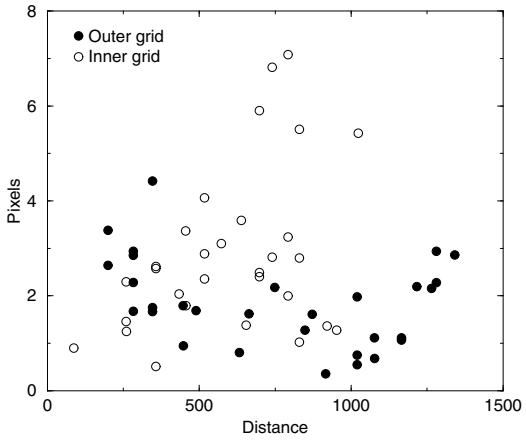


Figure 8.9: Re-projected error for test poses.

Intrinsic parameters		
Parameter	Mean	σ
Focal Length	20.26	0.07
Image Centre X	257.53	1.46
Image Centre Y	160.71	1.04
κ	0.0002	0.0003

Table 8.2: Results of all intrinsic parameters

for any pose. By comparing the magnitude of the errors with those computed from single-view calibration data for figure 8.6a generalised to new poses, it can be seen that the errors obtained from the multi-view calibration are much lower. Furthermore, these results are repeatable and of an acceptable size allowing us to achieve our goal of model maintenance by a moving robot/camera system. We found that we got almost identical results for subsequent experiments of the test poses with each pose giving an error of similar magnitude each time.

The repeatability of the errors determined in the test poses can be seen from figure 8.10 which shows the average re-projected error for the two test grids for each of the 30 experiments. Both grids show a low mean values though with the inner grid about half a pixel greater than the outer grid. These results indicate that the camera parameter values which have been determined by our multi-view calibration procedure are close to the optimal values for the camera/grabber configuration used. This is further supported by the fact that a low error is obtained on the poses which

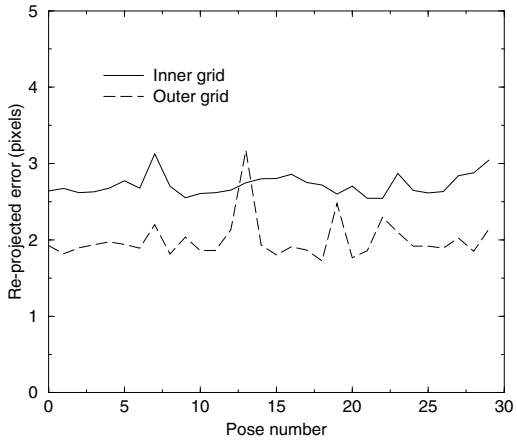


Figure 8.10: Re-projected error of multi-view calibration data applied to the two test grids over all experiments.

were outside the original volume of experimental poses.

8.6 Summary

We have presented a series of experiments with the purpose of investigating, in-depth, the characteristics and usefulness of the values of camera parameters determined in a number of scenarios involving calibration from single and multiple views of the camera system.

There are two main conclusions concerning single-view calibration. First, although a small error can be achieved for the same view the parameter values obtained do not generalise to different views resulting in large errors, of scores or even hundreds of pixels, depending upon the type of chart used. Second, the parameter values are not repeatable over time. Subsequent experiments do not show consistent values even to the extent that each time the grabber is switched on the mean values for the intrinsic and extrinsic parameters are different.

Others have addressed the problem of calibration for a mobile camera or with multiple images [34, 64, 93, 126]. However, the number of positions or images is generally small and there has been little attempt to identify the intrinsic parameters that would be applicable to any general pose of the camera. Puget and Skordas [93] presented five different methods of calibration at eight different poses of a robot arm.

The poses were over a relatively small area and the experiments did not include investigation of the intrinsic parameters over time or testing in poses not used in the original optimisation.

We have presented an experimental procedure which reliably recovers consistent and stable values for the intrinsic parameters of a camera. These values have been tested on additional poses within a proposed working environment and achieved an image error of approximately 2 pixels. The information derived from this procedure would be ideal for use with a robot where the movements are known, to predict the image position of 3D world objects from any pose.

Subsequent research by Fedor et al [35] has investigated the performance of the chart detection process. One of the very useful aspects of their work was that they used synthetic images for which the ground truth (the position of the chart elements) was available, allowing them to determine the chart *detection* error as well as the re-projected error. The experiments involved evaluating the performance of the chart detector while controlling the distance and rotation angle of the chart with respect to the camera as well as the focus of the lens system. The results of the experiments allowed them to make significant improvements to the chart detection system as well as concluding that the best results can be achieved by using the centres of gravity of squares instead of the corners. We expect our multi-view calibration procedure will perform even better after taking these factors into account.

Chapter 9

Model Maintenance

9.1 Introduction

Two issues are covered in this chapter, as preliminaries to the working active vision system, which is the topic of the final chapter in this part of the thesis. One issue concerns how the presence of a predicted object can be confirmed without matching every model in the object database. The solution is to determine *match threshold* values for the object in question [131]. The other issue is the maintenance of the predicted position of objects [133] with respect to its actual position, in succeeding frames taken from a moving camera. A couple of experiments are described which show this process in action and demonstrate the crucial importance of the projective transformation parameters obtained from our multi-view calibration procedure.

9.2 Match Thresholds

The technique we used for object recognition requires matching an outline of an object with *every* model in the database, in order to find the *lowest* value which we take as confirmation of the correct object. Two potential problems arise with this procedure. First, the object in the scene may not actually be represented in the database, but will still result in a lowest match value resulting in misidentification. Second, in order to find the correct model it is *always* necessary to search the entire database. Fortunately, we are able to take advantage of the fact that the match has significance in itself and not just with respect to the complete database. Under ideal circumstances the correct match value will be zero. However, due to noise and extraneous image detail the actual value will never be zero, but will consistently be in

a range close to zero. Therefore, instead of checking our object hypothesis against the whole database, we need only check the target model. If the match value is within the threshold the hypothesis is confirmed. This procedure also overcomes the second problem of falsely accepting an unknown object.

9.3 Experiments and Results

9.3.1 Determination of Match Thresholds

The match threshold experiment was performed with a database of five objects, the saucer, cup_and_saucer, milkjug, plate and sugar bowl. Each object was placed in different positions in the field of view covering the whole of the tabletop scene. At each point the values of the match between the observed object and every model in the database were measured and recorded. The results are shown in figure 9.1,given as the number of positions in which the goodness of fit (match value) falls within a particular interval, for each of the five models. The solid line represents the distribution of the match values for the correct model, whereas the dotted line is the mixture histogram of match values with all the other models. From these results, match thresholds for the saucer, cup_and_saucer, milk jug, plate and sugar bowl were derived as 4.10, 2.35, 3.50, 4.60 and 4.00, respectively.

9.3.2 Position Prediction

Our hypothesis concerning the maintenance of models and calibration data is that the data derived from the multi-view procedure will produce a small variation in the predicted object positions from different poses, whereas the data from single-view calibration will not generalise well to new views, resulting in a large variation of predicted position. Having determined two sets of calibration data, from single and multiple camera poses (chapter 8 and [132]), our next step was to determine the effect of the parameters on the predicted position of a recognised object.

The robot/camera system was successively positioned at 3D grid points 200mm apart, covering a 650x650x450 virtual 3D volume (barring the poses that were unreachable or violated the camera protection protocols). The direction of sight of the camera was always oriented towards the same point on the tabletop. In the initial position the object, a milk-jug, was recognised and its 3D position computed. At each subsequent position, using the known camera movement from the robot control system, the new image position of the object model was computed.

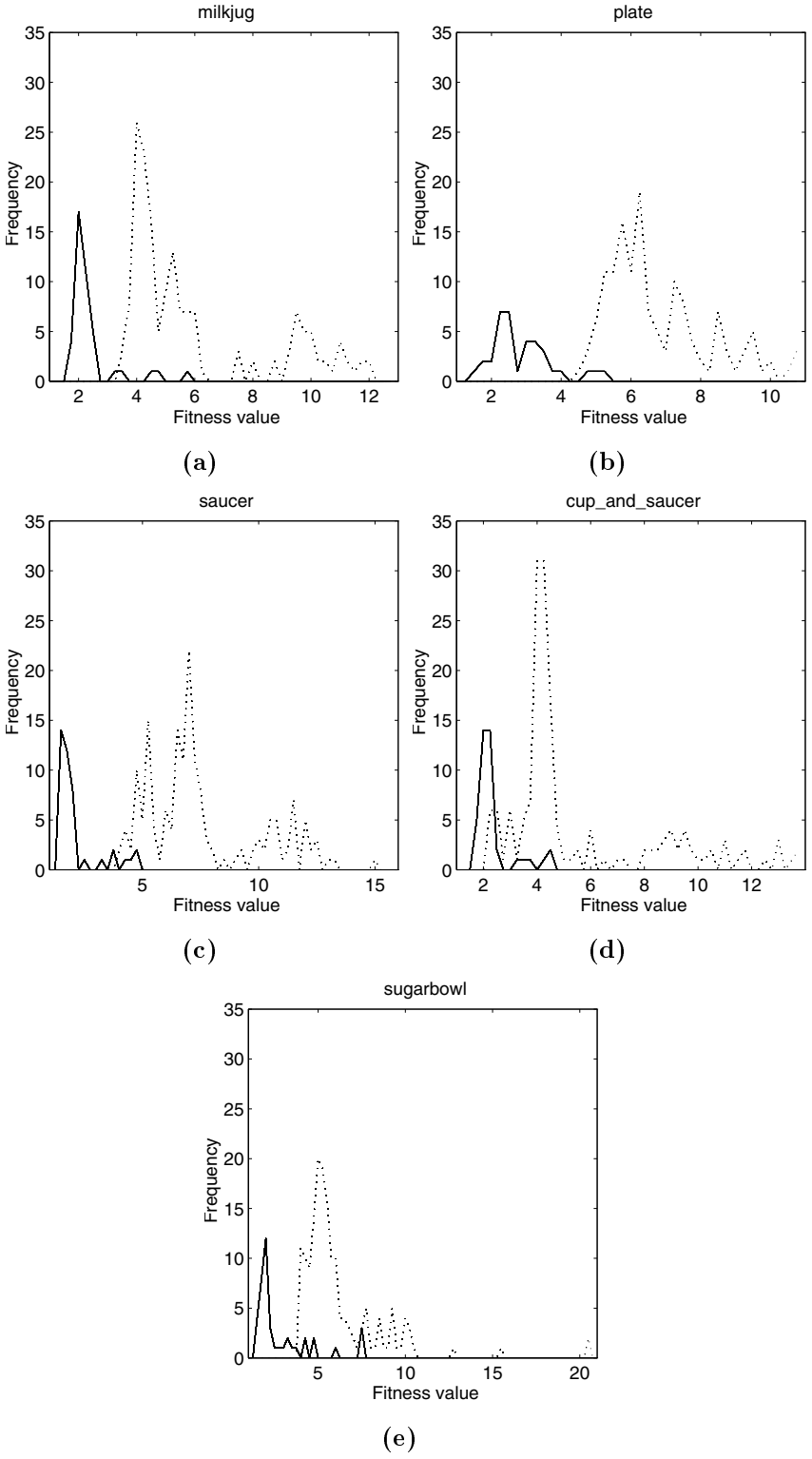


Figure 9.1: Determination of match thresholds

Generally there would be some error between the new model position and the new edges. This discrepancy was resolved by optimising the x and y world position of the object such that the error between the re-projected model and the edges in the image was minimised. A typical example is shown in figure 9.2. The first image indicates the initial re-projected error between the edges and the model (thick grey lines). The subsequent images show the minimisation process in action with the final image showing the ultimate match. With this process, a new world position of the object is determined at each new pose of the camera. Ideally, if the projective transformation is accurate, the computed position should be the same in each case.

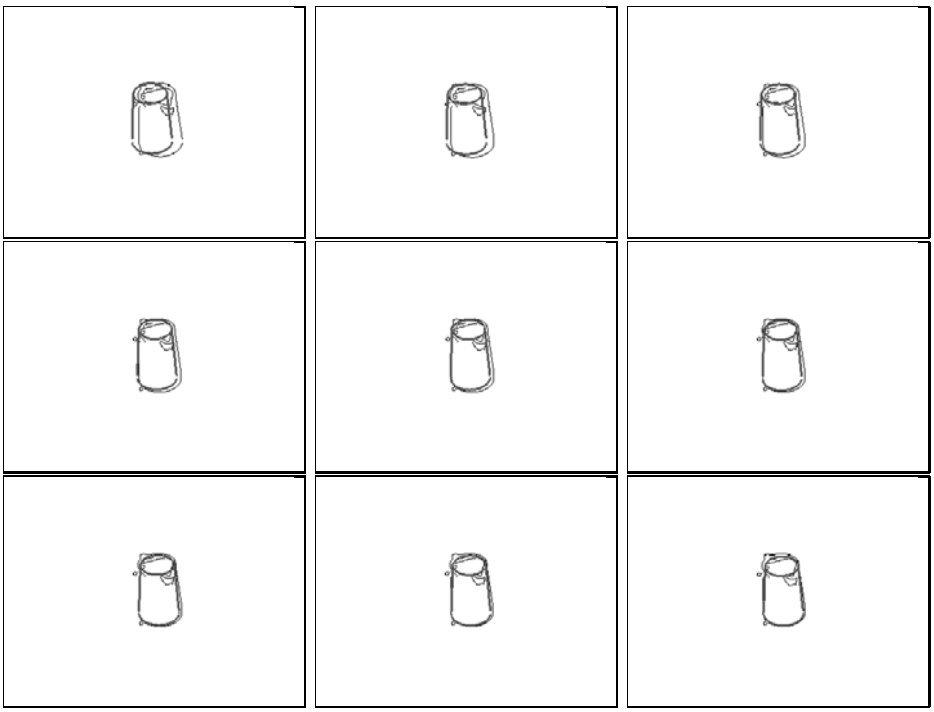
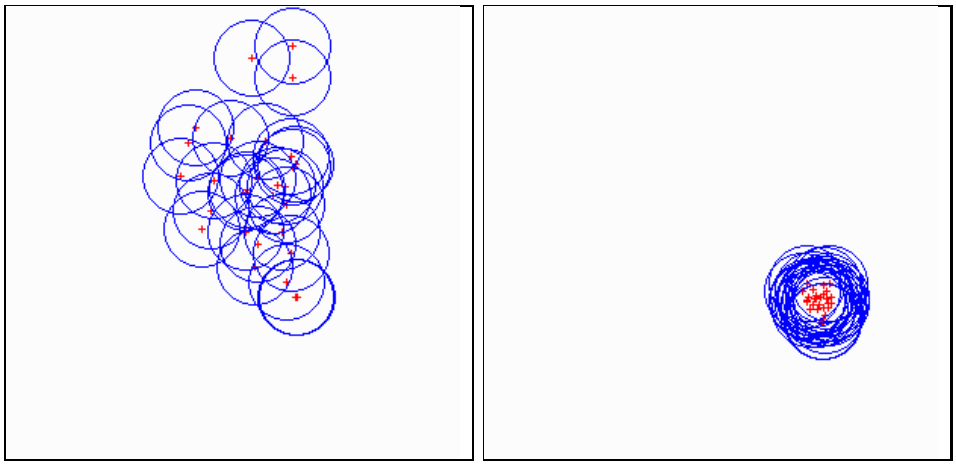


Figure 9.2: The process of optimisation of the world object position as seen in the image position of the re-projected model.

To test the above hypothesis we performed the experiment as described with both sets of calibration data with the expectation that the more reliable the data the more precise the predicted position. As the results indicate, our expectations were confirmed.

The plot of the predicted object world positions at each pose of the camera is shown in figure 9.3. The circles represent the diameter of the base of the object on the tabletop seen in plan view with the cross indicating the centre. Figure 9.3a shows the predicted positions derived from the experiment using single view calibration data

and 9.3b that for multiple view data. It is clear that the predicted positions derived from the multiple view calibration show a marked improvement over the single view, with a close clustering of points. Table 9.1 shows the mean and standard deviations of the distance of each point from the mean position of all the points, for both the single and multi-view calibration.



(a) (b)

Figure 9.3: Plan view of predicted positions from single view calibration (a) and from multiple view calibration. Each circle represents the base of the object, with the centre shown by a cross. The area of each image covers 600x600mm and the diameter of each circle is 100mm.

The predicted image positions derived from the single view calibration were, in fact, so poor that the regions allocated for edge detection were so far from the actual edges required that the position optimisation could not converge to the correct solution. This was resolved by manually indicating (with the cursor) an initial image location at which to start the position optimisation procedure.

A surprising outcome of this experiment is the finding that with the single view calibration the estimated camera parameters give rise not only to a larger variance in object position estimation but also to a bias. Thus the object pose cannot be refined merely by viewing it from several viewpoints and averaging the position estimates. However, the converse is true, provided the camera is calibrated using the multiple view calibration procedure. Thus with appropriately calibrated camera, an estimate of the position of an object can be refined by taking multiple views of the same object and subsequently averaging the positional measurements before inserting the object identity and its position into the scene model database.

Predicted world position data		
Parameter	Single view calibration	Multiple view calibration
x	331.105	442.728
y	240.409	391.170
Mean distance	82.2	15.4
σ	48.5	7.0

Table 9.1: Results of the predicted world position experiments. The mean x and y coordinates (millimetres) of predicted world positions are shown first, followed by the mean distance between the position predicted from each experiment to the mean position along with the corresponding standard deviation.

9.3.3 Occlusion Experiment

A further experiment demonstrates the application of the improved knowledge of the projective transformation by recovering from a situation where the detected object is occluded. In figure 9.4 the left hand column shows three successive scenes of the tabletop. The second shows objects occluding the target object from the first image. The robot is then moved to a new pose where the target is not occluded and an attempt is made to recover detection by utilising the predicted position from ground plane knowledge and the robot movement.

The result of this experiment is also shown in figure 9.4. Each row shows a different scene with the columns showing, from left to right, the grey-level image, the extracted edges and the edges with the region boundary and the superimposed model, if an object is detected.

In the first scene the milk-jug is detected (and its 3D world position recorded) as indicated by the thick grey outline in the right hand column, which matches the edges seen in the centre column. In the second scene the target object is occluded and no object is detected. In the final scene the camera is moved to a pose where the milk-jug is no longer occluded. From the world position derived from the first scene, the known robot movement and the precise knowledge of the projective transformation the image position of the object in the new scene is accurately determined, leading to successful matching of the re-projected model and the observed edges.

At present we only show how detection recovery is possible given a suitable new non-occluded pose and not how the choice of that pose is made. We leave to the next chapter determination of the camera viewpoint.

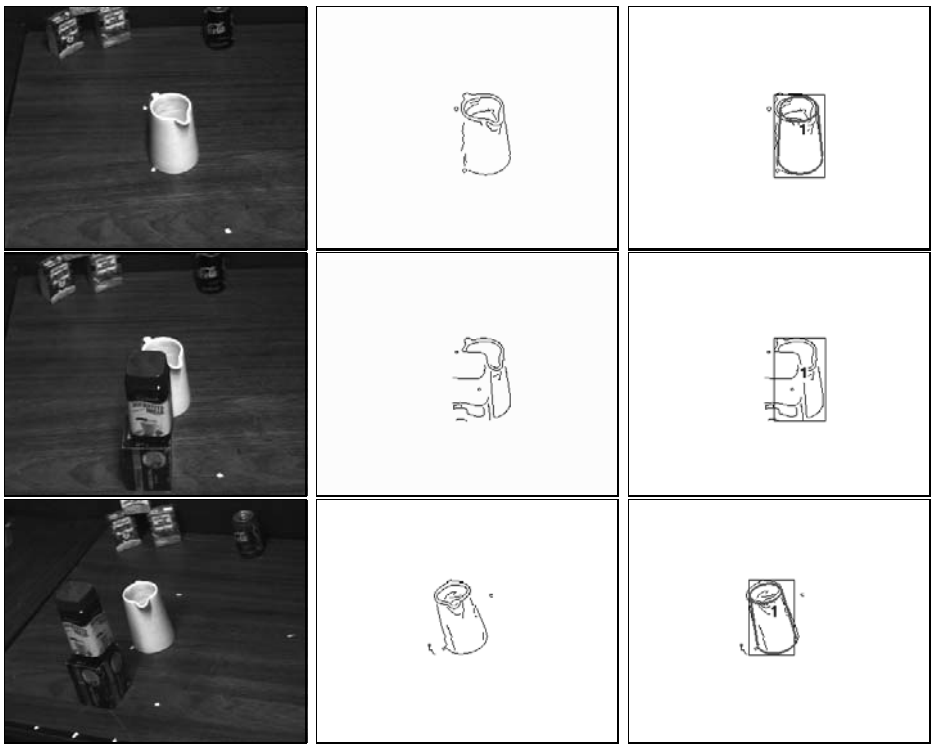


Figure 9.4: Occlusion experiment. Each row shows a different scene from the robot. In the first scene an object, a milk-jug, is recognised. In the second the milk-jug is occluded and recognition fails. The robot is then moved to a new pose and the object is recovered.

9.4 Summary

Two topics have been addressed in this chapter which which lay vital groundwork in preparation for the active vision system presented in chapter 10. We have shown how object match thresholds can be determined experimentally, allowing confirmation of object hypotheses avoiding costly search through the database.

The other main topic addressed here is the influence of the camera calibration accuracy and stability on object recognition and its 3D pose determination in the context of *active* vision. Previous work in the area of scene interpretation and 3D reconstruction has concentrated mainly on static poses with calibration determined from a single view image of the scene. We have shown that precise maintenance of 3D position cannot be achieved in the absence of reliable projective transformation parameters and that the required accuracy cannot be delivered by the single view calibration approach. The technique of computing calibration data from multiple views [132] was shown significantly to improve the predictions of object positions

from different view points assumed by a mobile camera.

Also presented was an experiment showing the application of the ability to predict object 3D position to the standard problem of occlusion. The system was able to verify the initial interpretation after a target object was occluded.

Chapter 10

Visual System Control

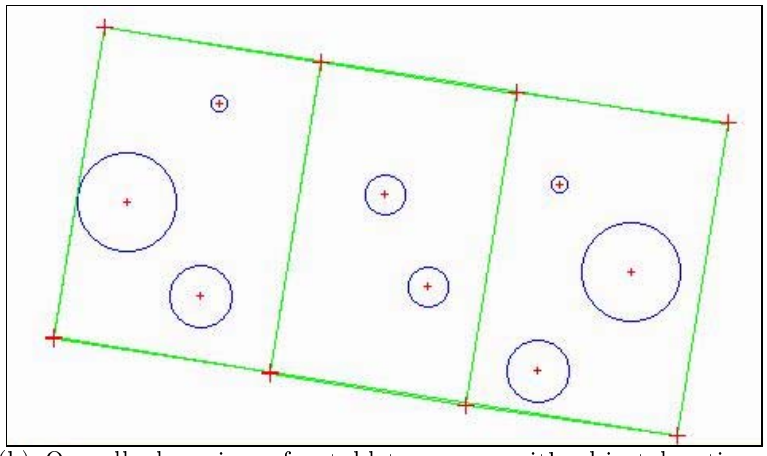
10.1 Introduction

The previous few chapters described the goals and context of the VAP vision project and how the current work fits into that framework. We have seen that in order to utilise a mobile sensor for object position maintenance it is necessary to accurately determine the camera parameters.

In this chapter we see the full benefits of robust multi-view calibration by introducing a fully integrated vision system which is able to move between different views of a large scene. In addition to integrating the diverse elements previously described we introduce a further level of processing by modelling the *temporal* evolution of the scene and its events.

The main focus of computer vision research has been concerned with the spatial structure of the world. Evident examples include spatial image analysis for the extraction of lines, textures and numerous other features. What has received little attention [52, 74, 77, 110] has been the structuring of the world in the time domain. The evolution of events at different times is highly consistent and is often guided by habit (table settings) and surrounding constraints (traffic). Observations of such scenes can give rise to a set of rules which describe the sequential ordering of events over time. These rules can be captured in a language, a grammatical model. In this chapter we make use of such models to control both the processing for database matching as well as the for the positioning of the sensor.

(a) Typical experimental scene with all objects present.



(b) Overall plan view of a tabletop scene with object locations plotted from 3D world position values.

Figure 10.1: Plan view and camera view of tabletop scene.

10.2 Breakfast Table Scenario

The experimental set-up we chose is that of the setting of a breakfast table. Two reasons for the choice of this scenario is that it is easy to reproduce in the laboratory and contains the type of objects easily recognised by our cylindrical object recognition engine. However, the main reason is that the scenario has a high degree of temporal structure which can be represented by a scene evolution model. Our main aim is to show the *principle* of control by such models. The same principles and methodology can be extended to different scenarios with different recognition engines. Figure 10.1a shows the full experimental scene with the objects used for

the setting of the breakfast table. The scene is divided up into three zones (figure 10.1b) which represent regions of interest of particular types of activity. The coordinates of the boundaries of the zones have been manually pre-defined. When objects are detected their world position is computed allowing assignment to the particular zone. The zones on either side (3 and 4) are place setting zones, where a single place setting, consisting of a plate, eggcup and cup and saucer are expected. The central zone (2) is the area where common objects are placed, in this case a milkjug and sugarbowl. For historical reasons zone 1 is the entire scene but is not explicitly used in the system.

10.3 Scene Evolution

In the current context a scene evolution model is a formal description of the order in which objects might appear in the scene. We assume that such a model is derived from many observations of a real scene which may evolve in different ways on different occasions. Therefore, the model incorporates *probabilities* for what the next object may be.

The temporal structure of scene events can be described by grammatical rules. A grammar describes a hierarchical structure of entities with high-level rules composed of lower-level rules. The rules at the lowest level include termination entities. In natural language those entities are actual words, the basic units of language. In the context of scene evolution grammars [70] the termination entities are recognised objects, indicating an actual event. For example, buying a beer is made up of the events *order beer, pour drink* and *pay money*, and can be described (in Backus-Naur Form) as follows,

```
<BUY_BEER>::=<ORDER>,<POUR>,<PAY>
```

with the *pour* rule consisting, roughly, of termination events (lower case),

```
<POUR>::=place(glass),open(tap),close(tap)
```

Such a description of evolution must include the following characteristics [20]: It should,

1. be based on observable features

2. describe individual objects
3. encompass relations between objects
4. include composition of objects

Such sets of rules can be captured in a regular grammar G;

$$G = (Q, \sum, P, q_0, Q_m)$$

where

Q is the set of states, or steps in the interpretation.

$\sum$ is the set of features that drive the interpretation. By nature these features must be discrete.

P is the set of productions that describe the evolutions from one state to another, given a particular feature is detected. $(P \subseteq (Q \times \sum \times Q))$.

q_0 is the initial state which is the entry point for the interpretation procedure.

Q_m is the set of terminal or marker states $(Q_m \subset Q)$, which indicate that the interpretation of a 'phenomenon' has been completed.

Q represents the states of the observed scene at particular points in the scene interpretation. The process is driven by the detection of features $\sum$ defining the current state and, in turn, the production, or set of productions, to invoke. Associated with the productions may also be action specifications allowing execution of specific actions at particular points as well as probabilities which indicate the priority of the elements of a production set.

The grammatical model of the scene evolution of the breakfast table scenario, for a single place setting and the general setting, is outlined as follows

```
<TABLE_SETTING>::=<PLACE_SETTING>,<GENERAL_SETTING>

<PLACE_SETTING>::=plate,<CUP_AND_SAUCER_SETTING>,eggcup

<PLACE_SETTING>::=plate,eggcup,<CUP_AND_SAUCER_SETTING>
```

```
<GENERAL_SETTING>::=milkjug,sugarbowl

<GENERAL_SETTING>::=sugarbowl,milkjug

<CUP_AND_SAUCER_SETTING>::=saucer,cup_and_saucer

<CUP_AND_SAUCER_SETTING>::=cup_and_saucer
```

Figure 10.2 shows the state transition network for the above rules. Each circle (node) represents a world state at various points throughout the evolution and the links indicate what objects are expected along with their probabilities, in percentages. In this case the first object expected for a place setting is always a plate. For example, the setting for a cup and saucer can occur either as a compound object, with both objects together, or with the saucer followed by the cup.

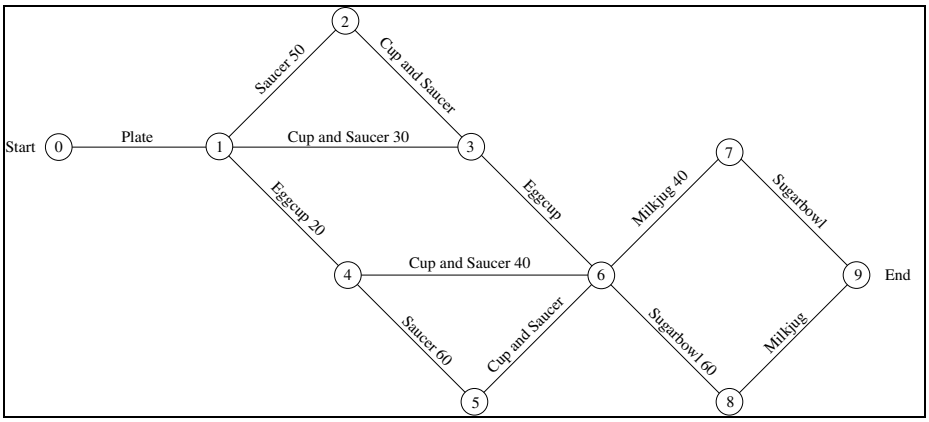


Figure 10.2: State transition network.

In the actual system we potentially allow an infinite number of place settings with one general setting, though the number must be known prior to execution. In our experiments we have two place settings.

In addition to the rules describing what is going to happen next we also have (separate) rules describing *where* the events are going to happen, in terms of zones. In order to keep things simple, at this stage, we have a simple rule which basically states that we pay attention to the place setting zones in clockwise order and the general setting when all the place settings are complete. In practice (as we only have two zones) this means alternating between the place setting zones. To avoid paying all our attention to one zone while there is activity in another we put a limit on the

amount of time spent viewing one zone. To add a touch of realism we also state that activity is more likely to happen in the same zone in which the last event occurred (unless the setting is complete). In our experiments it means that we spend 2/3 of our time in one zone and 1/3 in the other.

10.4 Scene Description

Maintaining and updating the description of the objects which are present in the scene as viewed by a static vision system is relatively straightforward. It is only necessary to examine the latest frame to build the description, discarding all previous descriptions. With an active vision system it is somewhat more complicated, as objects may be present in the scene but not the *current* view. Therefore, it is necessary to maintain a scene description which is independent of any particular viewpoint of the sensor. In our case we have two descriptions, one for the current view and one for the entire scene. Each time the sensor moves to a different view we detect the objects present to build the current view description which is passed to the scene description, which holds information about what objects are in each zone along with their world coordinates. The first time a view is visited we record the image of the empty view so that comparisons with subsequent frames give us the chromatic differencing information and regions of interest. With each subsequent visit, because we have the object positions noted in the global scene description we are able to confirm their presence without going through the frame comparison process.

The grammatical scene prediction model has been implemented in a software system for defining production systems, the C Language Integrated Production System (CLIPS). So that the CLIPS production rules of the scene evolution model have access to the current state of the world, the scene description is maintained as a set of CLIPS facts. For example, a fact about the presence of a plate in zone 3 is simply,

```
(Region (Num 8) (Zone 3) (Object plate))
```

which describes a region of interest with a globally unique identifier (Num), the zone and object present.

10.5 Experimental System Behaviour

Figures 10.3 to 10.13 show an example of the behaviour of our active vision system incorporating the scene evolution modelling, at various stages through the process. Each figure shows (clockwise from top left), the grey-level image of the current view, the binary image of the comparison with the base image, the list of predictions and probabilities for objects and zones, the plan view of the location objects and zones and the image of object and model outlines with region boundaries and numbers.

The explanation of each figure is as follows,

Fig. 10.3 The starting point of the program with zone 3 as the initial view, with no objects present. The first objects expected in zones 3 and 4 are plates (100% probability), though more likely in zone 3 (66%) than zone 4 (34%). If nothing happens for a certain period of time the robot/camera system moves to zone 4.

Fig. 10.4 A plate has been placed in zone 3. The world position, according to the computed coordinates is shown in the plan view. The binary image shows the region in which there has been significant change from the base image. The edge image shows the lines (white) extracted from the region of interest superimposed with the matched model (red). Now the predictions show that there are three objects which might appear next in the current view.

Fig. 10.5 An eggcup is also placed in zone 3. Expected next is a cup and saucer setting, which will be either the cup and saucer together or the saucer first.

Fig. 10.6 The cup and saucer setting did not occur within the time limit in zone 3 so that system moves to look at zone 4. Although not present in the current view the scene description maintains information about the plate and eggcup in zone 3 (see plan view).

Fig. 10.7 A plate is placed in zone 4.

Fig. 10.8 The place setting for zone 4 is completed. As indicated by the list of predictions no more events are expected in zone 4 (zone 3 probability is 100%).

Fig. 10.9 A saucer is placed in zone 3 and the last object, the cup, is predicted.

Fig. 10.10 The cup in zone 3 completes all place settings. We are now at the point where activity is expected only in the central zone (zone 2).

Fig. 10.11 The system moves to zone 2, waiting for either the milkjug (40%) or the or the sugarbowl (60%). It should be remembered that the functional purpose of the probabilities is to control the processing in relation to the search through the database when matching models against image outlines. There are 100 models in the database. Even though the object placed was not the one with the highest probability it means that the milkjug model was checked second, as opposed to its actual position in the database.

Fig. 10.12 The milkjug has been placed.

Fig. 10.13 The sugarbowl has been placed and the system reaches the end node in the scene evolution model (see 10.2).

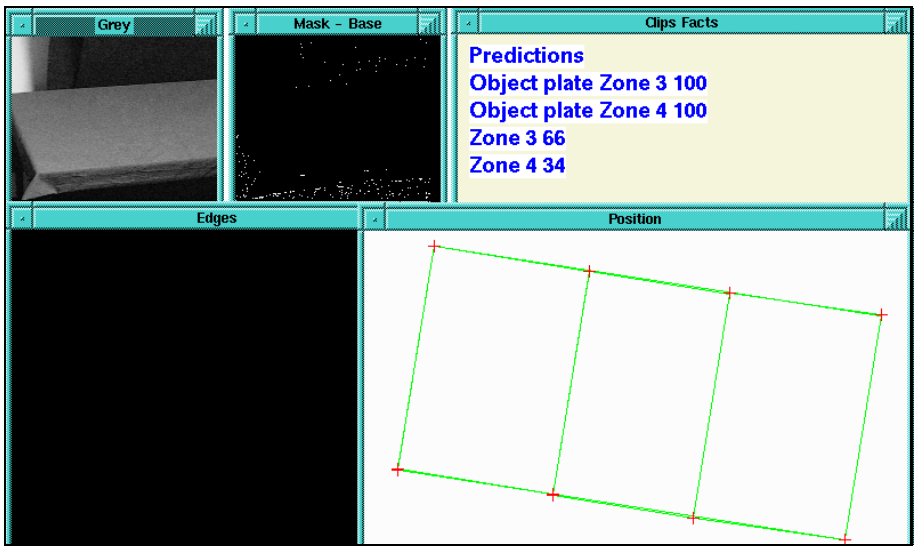


Figure 10.3: The initial view at zone 3.

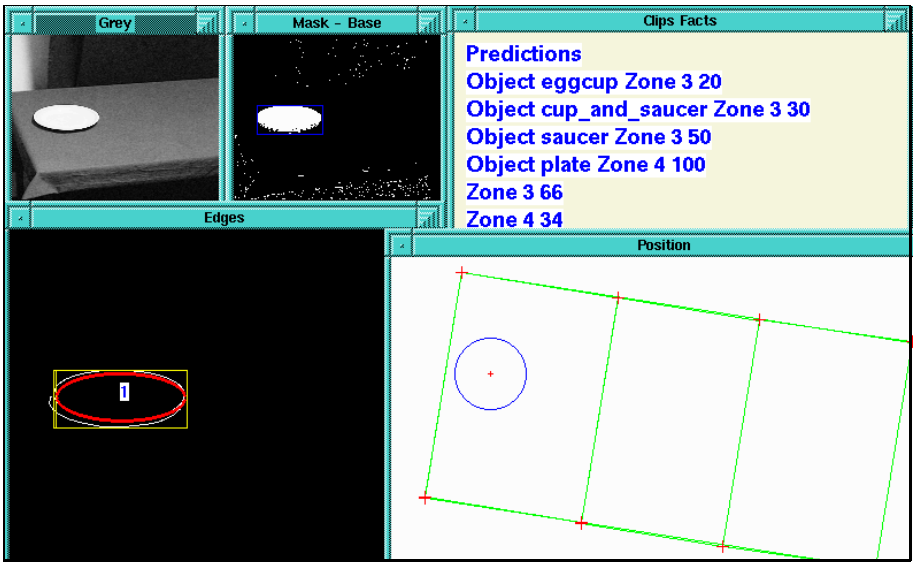


Figure 10.4: The first object, the plate, is placed.

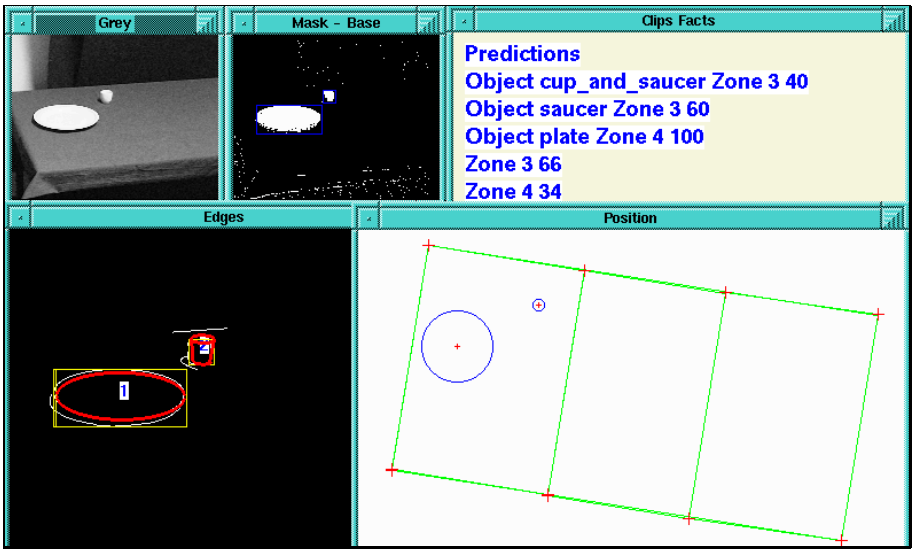


Figure 10.5: An eggcup joins the plate.

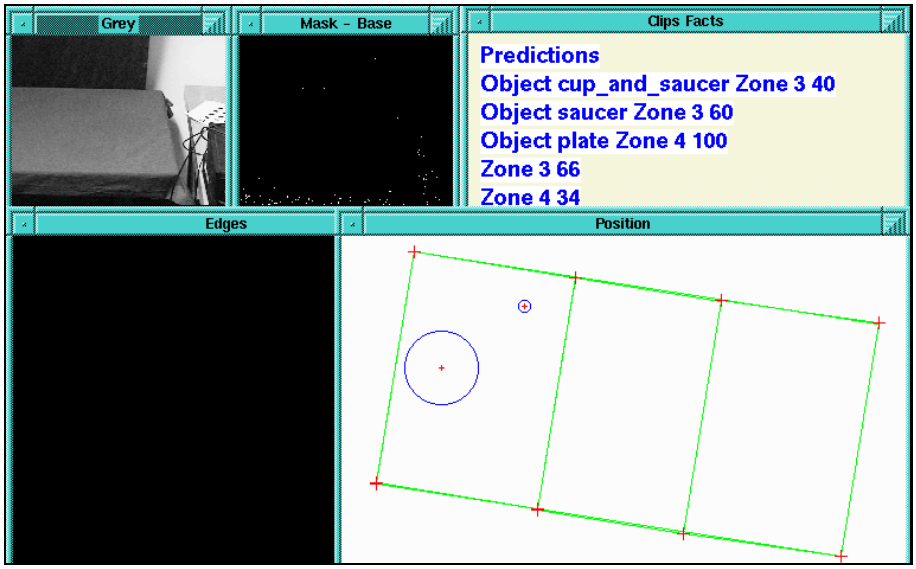


Figure 10.6: Attention moves to zone 4.

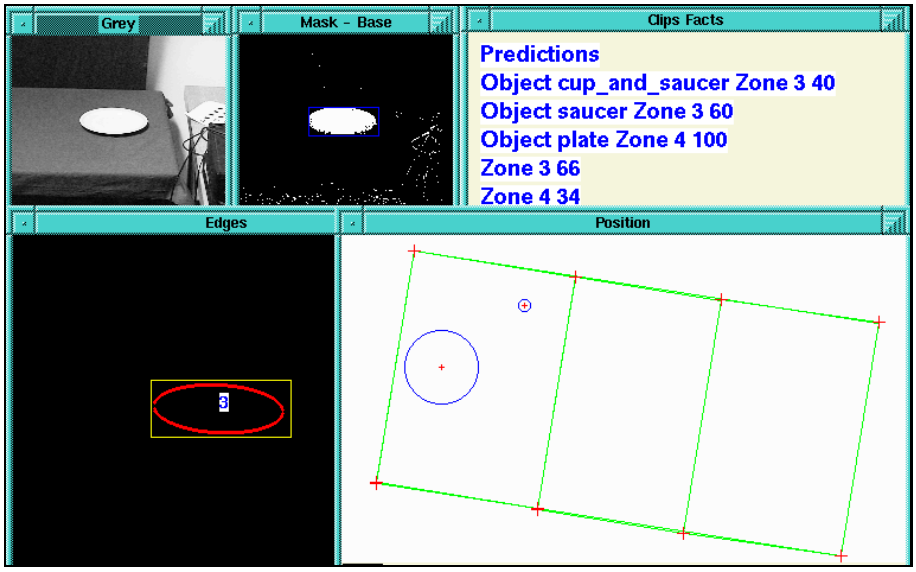


Figure 10.7: A plate is placed.

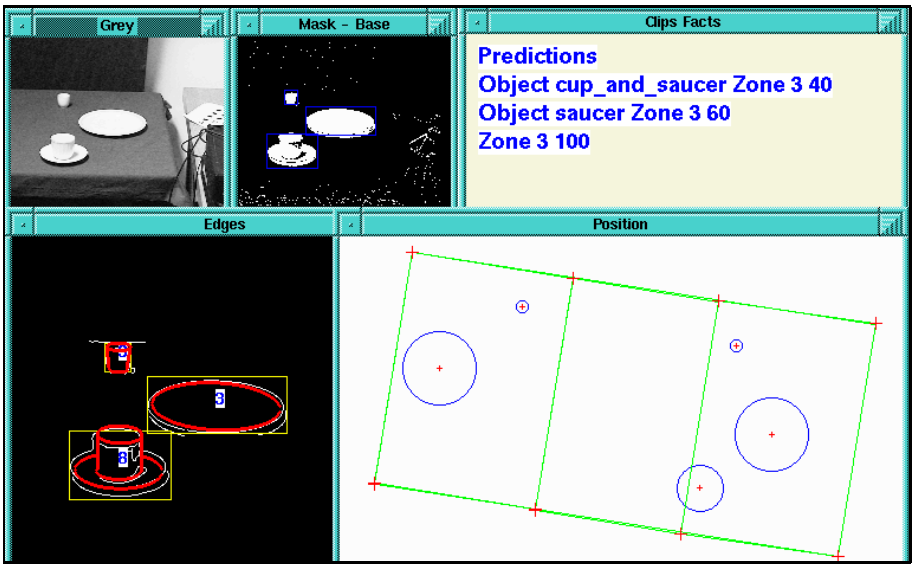


Figure 10.8: The zone 4 place setting is complete.

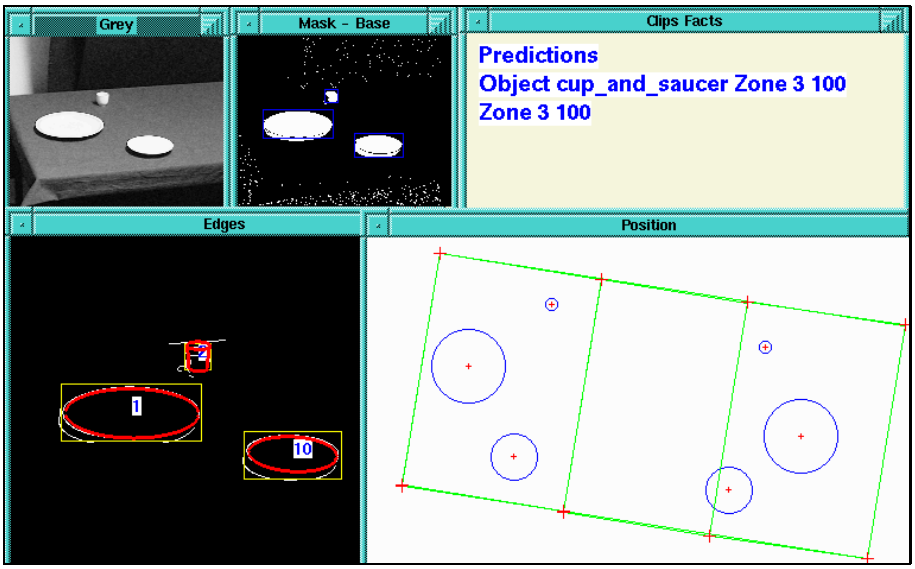


Figure 10.9: Attention returns to zone 3 where a saucer is placed.

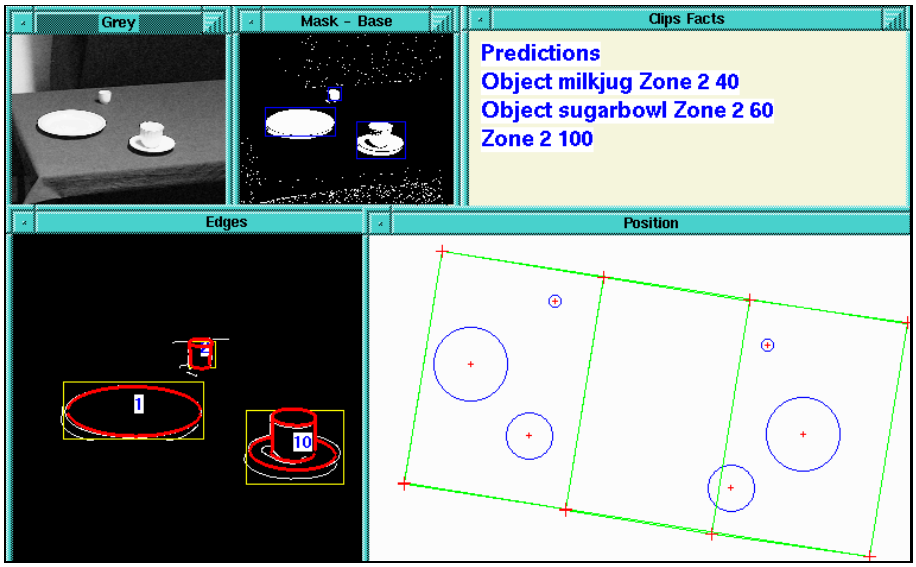


Figure 10.10: A cup is placed completing the zone 3 setting.

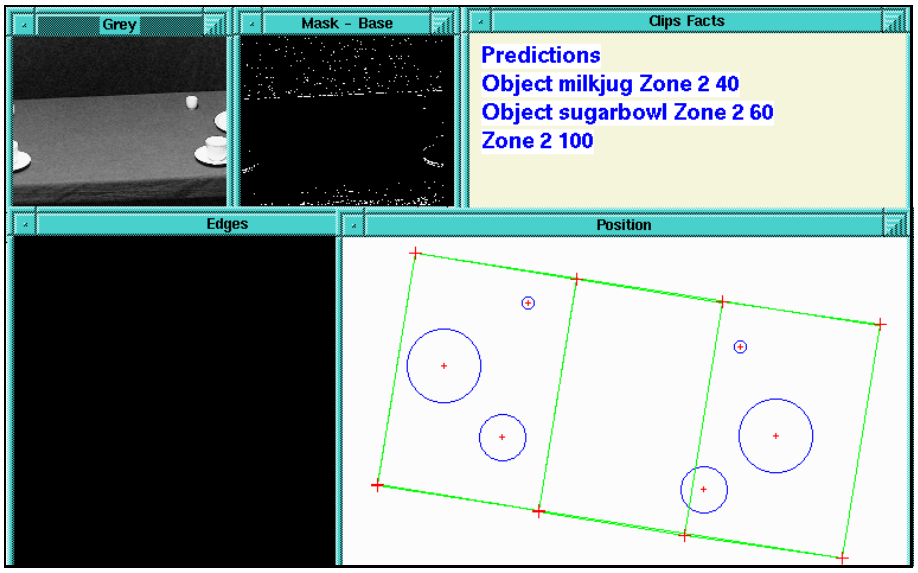


Figure 10.11: Attention moves to the last zone.

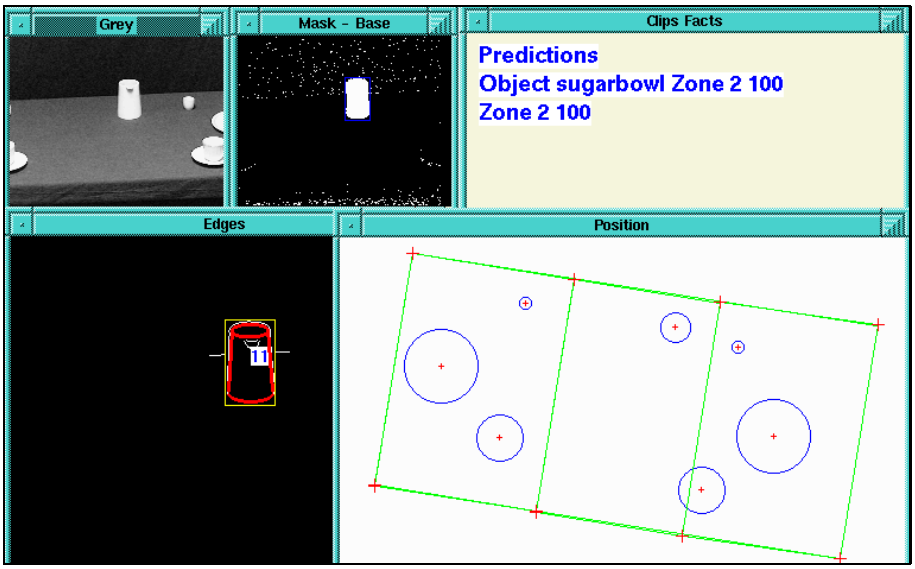


Figure 10.12: The milkjug is placed.

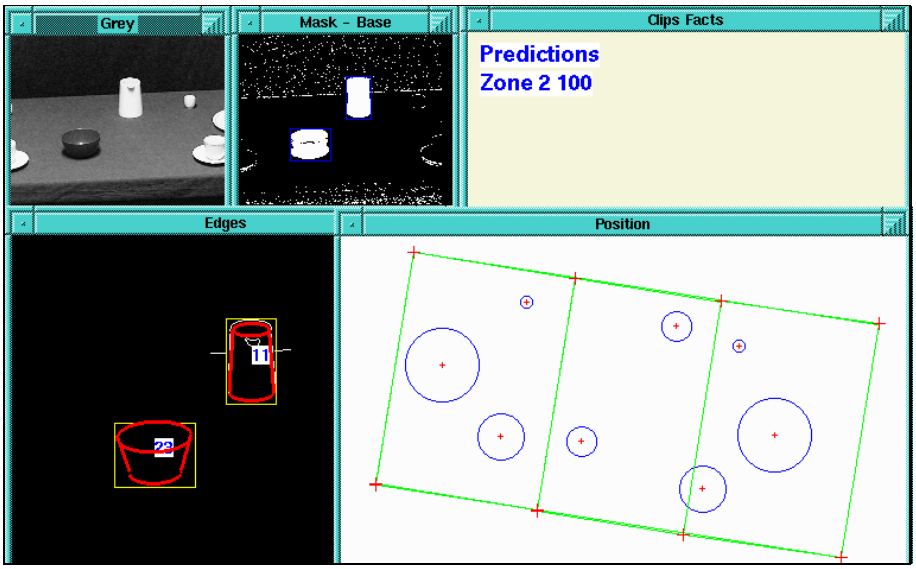


Figure 10.13: The sugarbowl is placed, completing the table setting

10.6 Results and Summary

In order to evaluate the benefits of using scene evolution models we ran our system with the predictive model previously described as well as with an unrestricted grammar model, which allowed random ordering of the objects. The processing times for the model matching were recorded for ten different routes through the two-place setting state transition network (figure 10.2) of the predictive grammar, and twice for the unrestricted grammar. The results are shown in table 10.1.

Model matching processing costs		
Number	Mean matches	Mean time
A	62.45	1.503
B	62.45	1.500
Mean	62.45	1.502
1	1.00	0.017
2	1.88	0.027
3	1.33	0.022
4	1.38	0.021
5	1.30	0.022
6	1.22	0.019
7	1.50	0.020
8	1.88	0.030
9	1.56	0.026
10	1.44	0.022
Mean	1.45	0.023

Table 10.1: Processing costs of model matching in a database of 100 objects. The middle column shows the average number of models checked, followed by the time in seconds, in the third column. A and B are from an unrestricted grammar and 1 to 10 from the predictive grammar shown in figure 10.2

With the unrestricted grammar model matching is performed by starting at the beginning of the database, of 100 objects, and matching each one until a suitable match is found. The processing is entirely dependent upon the order of models within the database. In this case an average of 62 (1.5 seconds) models had to be checked until the correct one was found. The result of the predictive grammar, on the other hand, is to define the order in which we access the models in the database.

Following the grammar results in significant improvements in the number of matches required (1.45) and corresponding processing times (0.02 seconds).

It should be noted, however, that these results represent what is probably the best case scenario, as all the objects placed in the experiments were legal with respect to the grammar model. As yet we have not implemented procedures for handling placement of objects which are in the database but not within the list of predictions. Such an extension would add to the robustness and realism of the system and would be a useful project for future work.

In this chapter we have described the function of scene evolution modelling along with an active vision system which uses such models. The system also integrates change detection, object detection, model matching and mobile camera control for processing complex dynamic scenes in real-time. The above results show the obvious benefits of modelling the temporal evolution of dynamic scenes.

Part IV

Conclusions

Chapter 11

Conclusions and Future Work

11.1 Introduction

In this thesis we have presented two aspects of a vision system. The rationale for the PCT-based functionality is partly to model and emulate a naturalistic theory of living systems. Currently this system is concerned with low-level visual and tracking behaviours, but is a potential module of any general vision system. The VAP inspired system is more application based and so is not concerned with biological plausibility. Instead it attempts to solve high-level visual problems for the purposes of scene interpretation. Although there are differences between the two methodologies there is a common aim in both areas of research, of the desire to construct artificial visual systems which are mobile, flexible, robust, autonomous and intelligent. We suggest that for the next generation of sophisticated artificial systems both approaches are necessary. The robustness is provided by the ability of the perceptual control systems to counteract unpredictable disturbances and keep low-level systems working, and the intelligence is provided by the high-level reasoning and interpretive abilities of the VAP-type system which guides the overall operation.

In this chapter we conclude with a summary of each area and some recommendations for future research. However, first we discuss a general point regarding the perspectives which are taken when attempts are made to describe and simulate complex systems, of which vision and visual behaviours are no exception.

11.2 Perspectives

Any complex system, or associated behaviour, whether it be an animal system, vision, language, problem solving or intelligence can be viewed from various perspectives. The usual research perspective which is taken is that of the observer. The descriptions which are then made are in terms of observable phenomena. Subsequent attempts to simulate these systems focus on these phenomena. We suggest that this is not the right perspective to take as such simulations do not get to the essence of the actual complex system, and as such will ultimately be limited in explanation and functionality. Both areas of research presented suffer, to some extent, from this malaise.

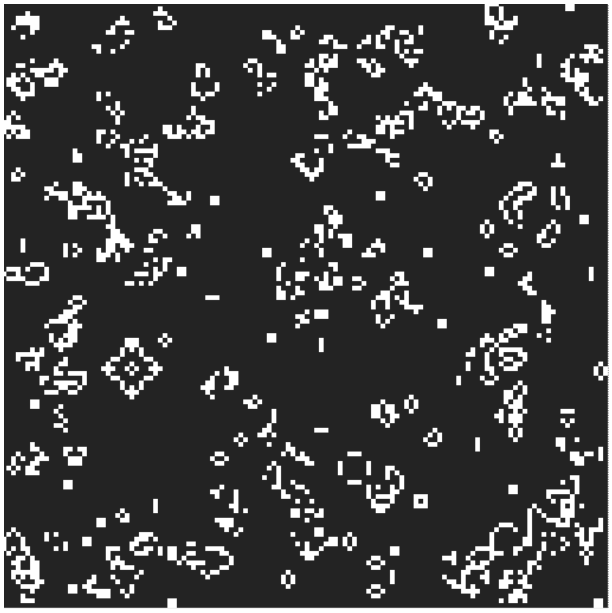
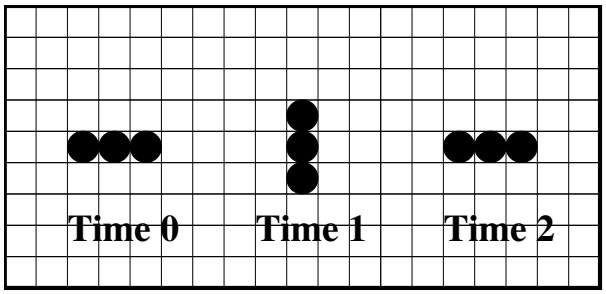


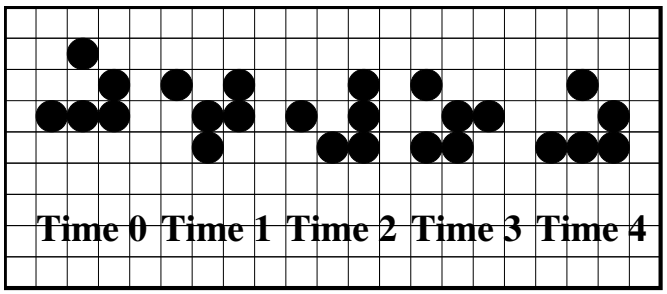
Figure 11.1: The Game of Life. "On" cells are white.

As illustration of this issue consider the *Game of Life* [86]. Figure fig 11.1 shows the state at one particular iteration of the Game of Life which is a dynamic, complex system consisting of a grid of elements where each cell can be either "on" or "off". From one iteration to the next the state of each cell may or may not change.

From the perspective of the observer there is a great deal of complex behaviour occurring. Many different kinds of patterns can be observed. Many rules can be devised which describe the transitions of parts of the Game of Life world from one state to the next. For example, a "blinker" (see figure 11.2) can be identified where an area alternates between three vertical and three horizontal "on" cells. Also a



(a) Blinker.



(b) Glider

Figure 11.2: Two examples of dynamic Game of Life patterns. "On" cells are black.

"glider" where an area of cells appears to *move* across the world. Many other shapes and configurations "appear" in the Game of Life including, the block, barge, aircraft carrier, spaceship, mango, canoe and pond [86]. From the perspective of the observer, rules can be written which specifically generate each of these patterns. If this observer-centric approach is taken the result will be an enormous amount of rules of varying levels of complexity from the very simple to the extremely complex. The Game of Life is actually implemented in a very different and extremely simple way. There are only three rules for the next state of each cell, which depend upon the states of the cell's eight neighbours,

- if a cell has 2 "on" neighbours, its state remains the same
- if a cell has 3 "on" neighbours the next state is "on"
- if a cell has 0, 1, 4, 5, 6, 7, or 8 "on" neighbours the next state is "off."

All of the patterns and transitions which can be observed in the Game of Life arise from these three simple rules. None of the behaviours are specifically implemented but *emerge* from the above rules. Here, then, lies the danger of taking the observers perspective,

- observed patterns or behaviours are *side-effects* of the system's operation
- simulations targeting specific emergent behaviours are unlikely to emulate the fundamental operation of the system
- the rules devised may never be a complete set as there may be other patterns not yet observed

To achieve a *true* simulation of any complex system the correct perspective to take would be *system-centric*. The target of simulations, then, should be the fundamental laws of operation themselves and not the emergent behaviour.

Next we summarise the two main parts of this thesis indicating to what extent the systems described suffer from the problems outlined in this section and suggest some recommendations for future research.

11.3 VAP

11.3.1 Summary and Results

In part III we presented a vision system for the purposes of high-level scene interpretation. The system integrates diverse modules for image based change detection, identification of regions of interest, shape outline detection, object recognition, scene evolution modelling, hypothesis management and mobile robot position control. In order to determine the 3D position of objects in the world it is necessary to first determine the relationship between the sensor and the world coordinate system. This process, camera calibration, is a major issue in computer vision. We investigated this problem and showed that calibration parameters derived from a single-view do not generalise well to other sensor viewpoints. To overcome this problem we devised a multi-view calibration method which significantly reduced the errors in the predicted position of objects from new views.

As well as the camera calibration problem and the technical difficulties involved in integrating diverse modules the other main problems investigated were control of database matching and of the sensor position. These two problems were addressed by modelling the temporal evolution of the experimental scene. The scene evolution models give predictions for what is going to happen and where it is going to happen, in terms of object placement. This knowledge allows us both to move the sensor to the required viewpoint as well as checking, in the object database, the predicted

object against the actual object. Prioritising the database search in this way along with experimentally determined match thresholds, avoids the requirement to search the entire database for the closest match. Modelling observed patterns of behaviour does, though, give rise to the problems alluded to in section 11.2. However, it is acknowledged that until the fundamental laws of such behaviour have been established the current approach is the only one which is practical.

The operation of the complete integrated system was described in chapter 10 and results presented showed the savings in processing costs due to the scene evolution modelling.

11.3.2 Future Work

Our recommendations for future work which extends this system are two-fold:

- we currently utilise one recognition engine for cylindrical objects. Others could be easily added to extend the recognition capabilities of the system.
- although the principle of the benefits to processing costs was demonstrated with the scene evolution model used we suggest, for the purposes of furthering the sophistication of the scenarios with which the system is able to cope, that the complexity of these models is increased.
- also a necessary part of a realistic system would be the ability to cope with errors in interpretation of object identities. This would involve the capability to backtrack through the evolution model to recover from invalid routes.

11.4 PCT

11.4.1 Summary and Results

In part II we introduced a little known theory of perception and behaviour within living systems, *Perceptual Control Theory*. In contrast to the conventional input/output view of perceptual processing, PCT suggests that living systems are made up of feedback control systems which control their *perceptual inputs*. All levels of behaviour follows from this simple premise. *Specific* actions are not computed by the system but are *varied* in order to maintain the input values. Observation and

measurement of the outputs of the system as performed by the psychological community are not a useful clue to internal processing and may merely be an indication of environmental disturbances.

Some simple simulations of basic perceptual control were presented. Among others the crucial function of the control system counteracting disturbances to maintain its input was demonstrated.

Leaving the theory behind we moved on to the elements required to construct a working example of visual perceptual control, that of an object tracker. For the purposes of computational efficiency as well as biological plausibility we used the foveal representation of the scene. After segmenting the target object we were able to extract a representation of the direction and magnitude of fixation which was the perceptual input of the control system, in other words, the controlled variable. Presented was a real-life control system in action. The system is able to easily and rapidly fixate and track moving objects of a single colour. Segmentation of grey-level or single-coloured objects was a relatively simple process. Our goal however, was to extend the abilities of the system to track more complex objects. For this purpose we introduced a method of encoding a target by a hierarchy of feature levels. The idea being that the higher levels would more specifically represent features belonging only to the target. The experiment for fixating multi-coloured faces showed that this was the case and successful fixation was made even though there were many distracting elements within the scene.

11.4.2 Future Work

The fixation experiments to the multi-column faces were performed on real but off-line images. Future work would benefit from testing the system on live scenes fully to test the robustness of the feature encoding and detection technique. Also recommended are different types of filtering techniques as well as introducing feature dimensions other than colour, such as motion, edges and texture. The more different kinds of dimensions involved the more discriminatory will be the segmentation process.

The control system approach is system-centric, as recommended in section 11.2. However, there is still a danger with manually designed control systems that the controlled variables used are those as perceived from the observers viewpoint and not those which actually emerge in real systems. We suggest, therefore, that the most beneficial direction for future PCT research is to investigate learning and reorganisation within perceptual control systems. This would ensure that the systems

which develop do so because they are successful at control as well as removing the design burden from the human researcher.

Bibliography

[1] J. (Y.) Aloimonos. Purposive and qualitative active vision. In *Image Understanding Workshop (Pittsburgh, Penn., Sep 11-13 1990)*, pages 816–828. Morgan Kaufmann, 1990.

[2] Y. Aloimonos. *Active Perception.* Lawrence Erlbaum Associates Ltd., 1993.

[3] P. Atkinson. *Feedback Control Theory for Engineers.* Heineman, London, 1972.

[4] R Bajcsy. Active percpetion vs. passive perception. In *Proc. 3rd IEEE Workshop on Computer Vision*, 1985.

[5] D. H. Ballard. Reference frames for animate vision. In *Joint Conference on Artificial Intelligence*, pages 1635–1641, 1989.

[6] Dana H. Ballard and Christopher M. Brown. Principles of animate vision. In Yiannis Aloimonos, editor, *Active Perception*, chapter 7, pages 245 – 282. Erlbaum, Hillsdale, New Jersey, 1993.

[7] Thierry Baron, Martin D. Levine, and Yehezkel Yeshurun. Exploring with a foveated robot eye system. In *ICPR-D*, pages 377–380, 1994.

[8] A. Blake and A. Yuille. *Active Vision.* MIT Press, Cambridge, MA, 1992.

[9] Margaret A. Boden. Autonomy and artificiality. *AISB Quarterly*, 87:22–28, Spring 1994.

[10] Rodney A. Brooks. A robust layered control system for a mobile robot. Technical Report A.I. Memo 864, Massachusetts Institute of Technology, Cambridge,MA, 1985.

[11] Rodney A. Brooks. A robot that walks: Emergent behaviors from carefully evolved networks. Technical Report A.I. Memo 1091, Massachusetts Institute of Technology, Cambridge,MA, 1989.

[12] Rodney A. Brooks. Intelligence without reason. In *International Joint Conference on Artificial Intelligence*, pages 569–595, 1991.

[13] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[14] Rodney A. Brooks and Anita M. Flynn. Fast, cheap and out of control. Technical Report A.I. Memo 1182, Massachusetts Institute of Technology, Cambridge,MA, 1989.

[15] Rodney A. Brooks and Lynn Andrea Stein. Building brains for bodies. Technical report, Massachusetts Institute of Technology, Cambridge,MA, 1993.

[16] C. Brown. Prediction and cooperation in gaze control. *Biological Cybernetics*, 63:61–70, 1990.

[17] Vicki Bruce and Patrick R. Green. *Visual Perception: Physiology, Psychology and Ecology.* Erlbaum, Hove, 1985.

[18] P. J. Burt. Attention mechanisms for vision in a dynamic world. In *ICPR*, pages 977–987, 1988.

[19] David J. Chalmers. Why fodor and pylyshyn were wrong: The simplest refutation. In *Proceedings of the Cognitive Science Society*, pages 340–347, 1990.

[20] H.I. Christensen, J. Matas, and J. Kittler. Using grammars for scene interpretation. In *IEEE International Conference on Image Processing, (16-19 September, Lausanne, Switzerland)*, pages 793–796, 1996.

[21] Andy Clark. Autonomous agents and real-time success: Some foundational issues. Technical report, Philosophy/ Neuroscience/ Psychology Program, Washington University, 1994.

[22] Andy Clark and Rick Grush. Towards a cognitive robotics. *Adaptive Behavior*, 7(1):5–16, 1999.

[23] Andy Clark and Chris Thornton. Trading spaces: Computation, representation and the limits of uninformed learning. Technical report, School of Cognitive and Computing Sciences, University of Sussex, 1994.

[24] D. T. Cliff. *Animate Vision in an Artificial Fly: A Study in Computational Neuroethology.* PhD thesis, Cognitive and Computing Sciences, University of Sussex, 1991.

[25] Dave Cliff, Philip Husbands, and Inman Harvey. Evolving visually guided robots. Technical Report CSRP 220, Cognitive and Computing Sciences, University of Sussex, 1992.

[26] T. S. Collett and M. F. Land. Visual control of flight behaviour in the hoverfly, *syritta pipiens l. Journal of Comparative Physiology*, 99:1–66, 1975.

[27] J. Crowley and H.I. Christensen, editors. *Vision as Process: Basic Research on Computer Vision Systems*, volume II. Springer-Verlag, New York, 1994.

[28] Sean M. Culhane and John K. Tsotsos. An attentional prototype for early vision. In *ECCV*, pages 551–560, 1992.

[29] Robert Desimone and Leslie G. Ungerleider. Neural mechanisms of visual processing in monkeys. In F. Boller and J. Grafman, editors, *Handbook of Neuropsychology*, chapter 14, pages 267–299. Elsevier, 1989.

[30] B R Draper, A R Hanson, and E M Riseman. Learning knowledge directed visual strategies. In *Image Understanding Workshop (San Diego, CA, Jan 26-29, 1992)*, pages 933–940. Morgan Kaufmann, 1992.

[31] Andrew P. Duchon, William H. Warren, and Leslie Pack Kaelbling. Ecological robotics. *Adaptive Behavior*, 6(3/4):473–507, 1991. Special Issue on Biologically Inspired Models of Spatial Navigation.

[32] *Computer Vision - ECCV '90*. Springer-Verlag, 1990.

[33] Rolf Eckmiller. Neural networks for motor program generation. In Rolf Eckmiller and Christoph von der Malsburg, editors, *Neural Computers*, pages 359–370. Springer-Verlag, Berlin, 1987.

[34] O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. *European Conference on Computer Vision*, pages 321–334, 1992.

[35] M. Fedor, J. Matas, L.M. Soh, and J. Kittler. Performance evaluation of a calibration chart detector. Technical report, UOS, 1998.

[36] N. Franceschini, J. M. Pichon, and C. Blanes. From insect vision to robot vision. *Phil. Trans. of the Royal Society London, Series B*, 337:283–294, 1992.

[37] G. F. Franklin, Powell J. D., and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Wokingham, 1986.

[38] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *Robotics.* McGraw-Hill, 1987.

[39] J. J. Gibson. *Ecological Approach to Visual Perception.* Houghton-Mifflin, Boston, 1979.

[40] W. E. L. Grimson, A. Lakshmi, P. A. O'Donnell, and G. Klamderman. An active visual attention system to "play where's waldo". Technical report, MIT AI Lab, Cambridge, MA, 94.

[41] E. Grosso and D. Ballard. Head-centred orientation strategies in animate vision. In *International Conference Computer Vision*, pages 395–402, 1993.

[42] Rick Grush. *Emulation and Cognition.* PhD thesis, University of California, San Diego, 1995.

[43] Min-Hong Han and Sangyong Rhee. Camera calibration for three- dimensional measurement. *Psychological Research*, 25(2):155–164, 1992.

[44] A. R. Hanson and E. M. Riseman. *VISIONS: A computer system for interpreting scenes.* Academic Press Inc., 1978.

[45] R.M. Haralick and Shapiro. *Computer and Robot Vision - Volume II.* Addison-Wesley, 1993.

[46] Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[47] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. *European Conference on Computer Vision*, A:471–478, 1994.

[48] Wayne A. Hershberger, editor. *Volitional Action: Conation and Control.* Elsevier, North-Holland,NY, 1989.

[49] P. Hoad and J. Illingworth. Recognition of 3d cylinders in 2d images by top-down model imposition. In *In: Crowley JL, Christensen HI ed. Vision as Process: Basic Research on Computer Vision Systems Berlin, Germany: Springer-Verlag*, pages 393–401, 1995.

[50] Paul Hoad and John Illingworth. Automatic control of camera pan, zoom and focus for improving object recognition by a moving observer. In *Appendices to PPR-3 of VAP II*, chapter D-2. Esprit Basic research Project 7108, 1995.

[51] G. A. Horridge. The evolution of visual processing and the construction of seeing systems. In *Proc. Royal Society London, Series B*, pages 279–292, 1987.

[52] R.J. Howard and H. Buxton. An analogical representation of space and time. *Image and Vision Computing*, 10(7):467–478, 1992.

[53] D. H. Hubel and T. N. Wiesel. Functional architecture of macaque monkey visual cortex. *Procs. R. Soc. Lond.*, 198(series B):1–59, July 1977.

[54] *First International Conference on Computer Vision, (London, England, June 8–11, 1987)*, Washington, DC., 1987. IEEE Computer Society Press.

[55] *Second International Conference on Computer Vision (Tampa, FL, December 5–8, 1988)*, Washington, DC., 1988. Computer Society Press.

[56] Minoru Ito and Akira Ishii. A non-iterative procedure for rapid and precise camera calibration. *Psychological Research*, 27(2):301–310, 1994.

[57] J. V. Kittler, J. Matas, M. Bober, and L. Nguyen. Image interpretation: exploiting multiple cues. In *IEE Conference Publication, 1995, 140, pp1-5*, 1995.

[58] Jim Z.C. Lai. On the sensitivity of camera calibration. *Image and Vision Computing*, 11(10):656–664, 1993.

[59] Michael F. Land. Similarities in the visual behavior of arthropods and men. In M. S. Gazzinga and C. Blakemore, editors, *Handbook of Psychobiology*, pages 49–72, London, 1975. Academic Press.

[60] Michael F. Land and Russell D. Fernauld. The evolution of eyes. *Annual Rev. Neuroscience*, 15:1–29, 1992.

[61] LD T Lawton, T S Levitt, and P Gelband. Knowledge based vision for terrestrial robots. In *Image Understanding Workshop (Palo Alto, CA, May 23-26, 1989)*, pages 933–940. Morgan Kaufmann, 1989.

[62] F. L. Lim, G. A. W. West, and S. Venkatesh. An investigation into the use of log polar space for foveation, feature recognition and tracking. Technical report, Curtin University of Technology, 1995.

[63] George F. Luger and Williamm A. Stubblefield. *Artificial Intelligence and the dDesign of Expert Systems.* Benjamin/Cummings, Redwood City, California, 1989.

[64] Q.T. Luong and O.D. Faugeras. Self-calibration of a camera using multiple images. *International Conference on Pattern Recognition*, A:9–12, 1992.

[65] Richard Marken. A science of purpose. *American Behavioral Scientist*, 34(1):6–13, September/October 1990.

[66] Richard S. Marken. The dancer and the dance: Methods in the study of living control systems. *Psychological Methods*, 2(4):436–446, 1997.

[67] David Marr. *Vision: A Computational Investigation into Human Representation and Processing of Visual Information*. Freeman, San Francisco, 1982.

[68] Maja J. Mataric. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents*, 9(2-3):323–336, 1997.

[69] J. Matas. *Colour-based Object Recognition*. PhD thesis, University of Surrey, 1995.

[70] J. Matas, J. Kittler, J. Illingworth, L. Nguyen, and H.I. Christensen. Constraining visiual expectations using a grammar of scene events. In I. Plander, editor, *Artificial Intelligence and Information-Control System of Robots '94 (Smolenice Castle, Slovakia)*, pages 81–93. World Scientific, 1994.

[71] Jiri Matas, Paolo Remagnino, Josef Kittler, and John Illingworth. *Control of Scene Interpretation*, chapter 20, pages 347–371. Volume II of Crowley and Christensen [27], 1994.

[72] Ruggero Milanese. *Detecting Salient Regions in an Image: From Biological Evidence to Computer Implementation*. PhD thesis, Department of Computer Science, University of Geneva, 1993.

[73] Daniel M. Wolpert nad Zoubin Ghahramani and Michael I. Jordan. An internal model for sensorimotor integration. *Science*, 269:1880–1882, September 1995.

[74] H.H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59–74, 1988.

[75] R. C. Nelson. Introduction: Vision as intelligent behaviour - an introduction to machine vision at the university of rochester. *International Journal of Computer Vision*, 7(1):5–9, 1991.

[76] R. C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1):33–46, 1991.

[77] B. Neumann. Natural language descriptions of time-varying scenes. In D. L. Waltz, editor, *Semantic Structures: Advances in Natural Language Processing*, pages 167–206. Lawrence Erlbaum, 1989.

[78] A. Newell and H. Simon. Computer science as emirical inquiry: Symbols and search. *CAM*, 19(3):113–126, 1976.

[79] Dan-E. Nilsson. From cornea to retinal image in invertebrate eyes. *Trends in Neurosciences*, 13(2):55–64, 1990.

[80] Dan-E. Nilsson and Susanne Pelger. A pessimistic estimate of the time required for an eye to evolve. *Procs. Royal Society London B*, 1994.

[81] Peter Nordlund and Tomas Uhlin. Closing the loop: Detection and pursuit of a moving object by a moving observer. In *Appendices to PPR-3 of VAP II*, chapter B-4. Esprit Basic research Project 7108, 1995.

[82] D. Osorio. Eye evolution: Darwin's shudder stilled. *Trends in Ecology and Evolution*, 9(7):241–242, July 1994.

[83] D. Osorio and J. P. Bacon. A good eye for arthropod evolution. *Bioessays*, 16(6):419–424, 1994.

[84] Kourosh Pahlavan and Jan-Olof Eklundh. A head-eye system - analysis and design. *CVGIP: Image Understanding*, 56(1):41–56, 1992.

[85] Raymond P. Pavloski, Gerard T. Barron, and Mark A. Hogue. Reorganization: Learning and attention in a hierarchy of control systems. *American Behavioral Scientist*, 34(1):32–54, September/October 1990.

[86] William Poundstone. *The Recursive Universe.* Oxford University Press, 1985.

[87] William T. Powers. *Behavior: The Control of Perception.* Aldine DeGruyter, Hawthorne, NY, 1973.

[88] William T. Powers. Quantitative analysis of purposive systems: Some spadework at the foundations of scientific psychology. *Psychological Review*, 85(5):417–435, 1978.

[89] William T. Powers. Arm with artificial cerebellum. *http://animas.frontier.net/ powers_w/INDEX.HTML*, 1998.

[90] William T. Powers. *How to write a letter ?* Personal communication, 1998.

[91] William T. Powers. *Making Sense of Behavior*. Benchmark, 1998.

[92] William T. Powers. A model of kinesthetically and visually controlled arm movement. *Int. J. Human-Computer Studies*, 50:463–479, 1999.

[93] P. Puget and T. Skordas. An optimal solution for mobile camera calibration. *European Conference on Computer Vision*, pages 187–198, 1990.

[94] Rajesh P. N. Rao. Top-down gaze targeting for space-variant active vision. *DARPA 94 Workshop*, 1994.

[95] P Remagnino, M Bober, and J Kittler. Learning about a scene using an active vision system. In *Machine Learning in Computer Vision*, pages 45–49, 1993.

[96] Elaine Rich and Kevin Knight. *Artificial intelligence*. McGraw-Hill, New York, 1991.

[97] G. Sandini and M. Tistarelli. Vision and space-variant sensing. In Harry Weschler, editor, *Neural Networks for Perception*, chapter II.11, pages 398–425. Academic Press, 1992.

[98] Giulio Sandini and Paolo Dario. Active vision based on space-variant sensing. In *5th Symposium on Robotics Research*, pages 75–77, 1989.

[99] Peter A. Sandon. Visual attention and manipulator control. In *13th Annual Conf. of the Cognitive Science Society*, pages 1092–97, 1992.

[100] Peter H. Schiller. The superior colliculus and visual function. In Ian Darian-Smith, editor, *Handbook of Physiology Section 1: The Nervous System*, chapter 11, pages 457–505. American Physiological Society, Bethesda, Maryland, 1984. neuro/motor.

[101] Eric L. Schwartz. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, 20:645–669, 1980.

[102] Aaron Sloman. On designing a visual system (towards a gibsonian computational model of vision). Technical Report CSRP 146, Cognitive and Computing Sciences, University of Sussex, 1989.

[103] Tim Smithers. What the dynamics of adaptive bahaviour and cognition might look like in agent-environment interaction systems. In Tim Smithers and Alvaro Moreno, editors, *Workshop Notes for DRABC94: On the Role of Dynam-*

ics and Representation in Adaptive Behaviour and Cognition, pages 134–153, December 9 and 10 1994.

[104] R. H. Smythe. *Vision in the Animal World*. Macmillan, London, 1975.

[105] L.M. Soh, J. Matas, , and J. Kittler. Robust recognition of calibration charts. In *IEE 6th International Conference on Image Processing and Its Applications*, pages 487–491, 1997.

[106] David L. Sparks. Translation of sensory signals into commands for control of saccadic eye movements: Role of primate superior colliculus. *Physiological Reviews*, 66(1):118–171, January 1986.

[107] M Spratling and R Cipolla. Uncalibrated visual servoing. In *BMVC*, pages 545–554, 1996.

[108] M. J. Swain and M. A. Stricker. Promising directions in active vision. *International Journal of Computer Vision*, 11(2):109–126, 1993.

[109] Massimo Tistarelli and Guilio Sandini. Dynamic aspects in active vision. *CVGIP: Image Understanding*, 56(1):108–129, 1992.

[110] A.F. Toal and H. Buxton. Spatio-temporal reasoning within a traffic surveillance system. In *European Conference on Computer Vision*, pages 884–892, 1992.

[111] F. M. Toates. *Control Theory in Biology and Experimental Psychology*. Hutchinson, London, 1975.

[112] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3:323–344, 1987.

[113] Daniel Y. Ts'o and Anna W. Roe. Functional compartments in visual cortex: Segregation and interaction. In Michael S. Gazzinga, editor, *The Cognitive Neurosciences*, chapter 20, pages 325–337. MIT Press, Cambridge, MA, 1995.

[114] Hilary Tunley and David Young. Dynamic fixation of a moving surface using log polar sampling. In *British Machine Vision Conference*, pages 579–588, 1994.

[115] Alan Turing. Computing machinery and intelligence. *Mind*, 59:433–460, Oct. 1950.

[116] S. Ullman. Against direct perception. *The Behavioral and Brain Sciences*, 3:373–415, 1980.

[117] Tim van Gelder. Its about time: An overview of the dynamical conception of cognition. In Robert Port and Tim van Gelder, editors, *Mind as Motion: Explorations in the Dynamics of Cognition*. Bradford/MIT Press, Cambridge MA, 1995.

[118] Tim van Gelder. What might cognition be if not computation? *Journal of Philosophy*, 92:345–381, 1995.

[119] Tim van Gelder. The dynamical hypothesis in cognitive science. *Behavioural and Brain Sciences*, 21(5):616–665, 1996.

[120] Carl F. R. Weiman and Richard D. Juday. Tracking algorithms using log-polar mapped image coordinates. In *SPIE Vol. 1192 Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, pages 843–852, 1989.

[121] Westelius95. *Focus of Attention and Gaze Control for Robot Vision*. PhD thesis, University of Linkoping, 1995.

[122] Michael Wheeler. Active perception in meaningful worlds. Technical Report CSRP 311, Cognitive and Computing Sciences, University of Sussex, 1991.

[123] Michael Wheeler. For whom the bell tolls? the roles of representation and computation in the study of situated agents. Technical Report CSRP 311, Cognitive and Computing Sciences, University of Sussex, 1994.

[124] R. Willson. Tsai camera calibration software. *http://www.cs.cmu.edu/People /rgw/TsaiCode.html*, 1995.

[125] Alexander Wolsky and Maria De Issekutz Wolsky. Structure first, function later: Evolution in arthropodan eye. *Biology Forum*, 84(2):229–240, 1991.

[126] D. Yang and J. Illingworth. Calibrating a robot camera. In E. Hancock, editor, *British Machine Vision Conference*, pages 519–528, 1994.

[127] D. Yang, J. Kittler, and G. Matas. Recognition of cylindrical objects using occluding boundaries obtained from colour based segmentation. In E. Hancock, editor, *British Machine Vision Conference*, pages 439–448, 1994.

[128] David Young. Logarithmic sampling of images for computer vision. In *Proceedings of the Seventh Conference of the Society for Study of Artificial Intelligence and Simulation of Behaviour*, pages 145–150, 1989.

[129] David Young. *Nerve Cells and Animal Behaviour.* Cambridge University Press, Cambridge, 1989.

[130] R. Young and J. Illingworth. A fixation and viewpoint measure for object-based gaze control. In *British Machine Vision Conference*, pages 570–579, 1997.

[131] R. Young, J. Kittler, and J. Matas. Hypothesis selection for scene interpretation using grammatical models of scene evolution. In *International Conference on Pattern Recognition*, 1998.

[132] R. Young, J. Matas, and J. Kittler. Active recovery of the intrinsic parameters of a camera. In *International Conference Control, Automation, Robotics and Vision*, December 1998.

[133] R. Young, J. Matas, and J. Kittler. On camera calibration for scene model acquisition and maintenance using an active vision system. In *International Conference on Computer Vision Systems*, January 1999.

[134] Rupert Young and John Illingworth. Towards a control model of object recognition. *Journal of Perceptual Control Theory*, 1, 1998. http://home.t-online.de/home/WZocher/jpctset.htm.