National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

Variable Resolution Vergence Control

By

Sergio Licardie ©

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Spring 1993

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-82207-4

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Sergio Licardie

TITLE OF THESIS: Variable Resolution Vergence Control

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1993

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) ____________

Permanent Address:
Av. Emiliano Zapata 319,
Tlaltenango,
Cuernavaca, Morelos
Mexico 62170

Date: Feb-22-1993

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Variable Resolution Vergence Control* submitted by *Sergio Licardie* in partial fulfillment of the requirements for the degree of Master of Science.

supervisor: A. Basu

Hong

examiner: H. Zhang (Computing Science)

Roger W Toogood

external: R. Toogood (Mechanical Engineering)

Date: Nov-25-1992

To my beloved wife Elena
and to my parents Elsa and Sergio

Abstract

Fast and reliable depth estimation is currently an important area of discussion in the field of computer vision. Relevant applications of depth information include hand-eye coordination, navigation, and obstacle avoidance. There are two main approaches to the problem — disparity analysis and vergence control. Both of these methods have been derived using anthropomorphic evidence, which also shows that the human visual system can be characterized as a variable-resolution system: foveal information is processed at very high spatial resolution whereas peripheral information is processed at low spatial resolution. Although the quantitative aspects of this variable-resolution processing are known quite precisely, its applications to different areas of vision have not been fully explored.

This thesis describes a method for performing fast and accurate vergence control using a variable-resolution framework. We show that this approach generates a matching function (with vergence angle as the free variable) which increases to a peak corresponding to the correct match and then decreases. The shape of the matching function helps in obtaining, quickly and reliably, correct vergence with respect to a given object. It is additionally shown that variable resolution images can be obtained by lenses similar to fish-eye. To validate the theoretical analysis, experimental results are presented, introducing a comparison between the matching functions generated by our approach with those generated by a similar vergence control approach that uses a uniform-resolution framework.

Acknowledgements

I would like to take this opportunity to sincerely express my appreciation to those who have helped me during the duration of my graduate studies. I am especially thankful to Dr. Anup Basu, my supervisor, for his guidance, encouragement and patience throughout this research. I also acknowledge with appreciation Dr. Walter Bishop from the psychology department, who acted as my co-supervisor during the first year of my research, for all his advise and guidance. I am grateful to the members of my committee, Dr. Hong Zhang, and Dr. Roger Toogood from the mechanical engineering department, for their valuable comments which made this work more complete. I am indebted to Steve Sutphen MSc., for all his help, extraordinary comments and support. I am thankful as well to Anne Nield, for the time she spent helping me proofreading this thesis. I would like to thank my friends: Gloria Chow, Don Murray and Manoj K. Jain, for the very constructive discussions that helped me clarify my ideas in this area. Finally, and most of all, I would like to thank my cherished wife Elena, for her understanding, sacrifice and encouragement throughout the completion of this dream.

Contents

1 **Introduction** **1**

1.1 Depth Perception 3

1.2 Thesis Objective 5

1.3 Thesis Organization 6

2 **Review of Previous Work** **7**

2.1 Disparity Analysis 7

2.2 Vergence Control 11

3 **Edge Detection and Matching** **16**

3.1 Edge Detection 16

3.1.1 Classification of Edge Detectors 18

3.1.2 Local Methods 18

3.1.2.1 Gradient Operators 19

3.1.2.2 Laplacian Operators 23

3.1.3 Regional Methods 24

3.1.4 Global Methods 25

3.1.4.1 Marr and Hildreth's Operator 25

3.2 Thresholding 28

3.3 Matching 30

4 **Fish Eye Transform** **33**

4.1 Modeling and Calibration of Fish Eye Lenses 36

5 Application of Variable Resolution in Vergence Control 43

6 Methodology and Experimental Results 47

6.1 Programming Environment and Equipment Layout 47
6.2 Image Platforms 48
  6.2.1 Graphics Environment 48
  6.2.2 Real Image Environment 49
6.3 Methodology 50
6.4 Experimental Results 55
  6.4.1 Experiment 1 (Real Environment) 55
    6.4.1.1 Effect of Changing the Distortion Factor λ 57
    6.4.1.2 Effect of Changing the Thresholding Value 58
  6.4.2 Experiment 2 (Graphics Environment) 59
    6.4.2.1 Matching Function for Different Depths 61
  6.4.3 Experiment 3 (Real Environment) 63
    6.4.3.1 Multiple Thresholding 63
  6.4.4 Experiment 4 (Real Environment) 67

7 Conclusions and Future Research 71

7.1 Future Research 72

Bibliography 74

List of Tables

3.1 Groups for Edge Classification 28

4.1 Focus of Distortion Estimate 38

6.1 Distortion and Scaling Factors 57

List of Figures

1.1 Disparity Caused by Difference in Perspectives. 3
1.2 Converging Views of a Cup. 4

2.1 Parallel Axis Imaging Geometry. 9
2.2 Top View of the Parallel Axis Imaging Geometry. 11
2.3 Vergence Control Imaging Geometry. 12
2.4 Top View of the Vergence Control Imaging Geometry. 14

3.1 Ideal and Realistic Step Edges 17
3.2 Roberts Mask 20
3.3 Prewitt and Sobel Masks 21
3.4 Kirsch Masks 22
3.5 Laplacian Masks 23
3.6 Hueckels Regional Masks 24
3.7 Histogram Thresholding 29

4.1 Variable Resolution Images 34
4.2 Fish Eye Grid Image and its Curvature 37
4.3 Graph of Fish Eye Distortion in Different Directions and the Average Value 39
4.4 Goodness of Fit for PFET Model 40
4.5 Goodness of Fit for FET Model 42
4.6 Distortion Compensated 42

5.1 Uniform resolution and Variable resolution ideal matching functions 45

6.1 Active Camera System 49
6.2 System Flowchart 51
6.3 Uniform Resolution Process 52
6.4 Variable Resolution Process 53
6.5 Image Sequence for Real Experiment 1 56
6.6 Matching Function for Real Experiment 1 56
6.7 Effect of Changing the Distortion Factor (small values) 57
6.8 Effect of Changing the Distortion Factor (large values) 58
6.9 Effect of Changing the Thresholding Value Variable Resolution . . 58
6.10 Effect of Changing the Thresholding Value Uniform Resolution . . 59
6.11 Image Sequence in Uniform Resolution for Artificial Experiment . 60
6.12 Image Sequence in Variable Resolution for Artificial Experiment . 60
6.13 Matching Function for Artificial Experiment 61
6.14 Matching Function in Uniform Resolution 62
6.15 Matching Function in Variable Resolution 62
6.16 Image Sequence for Real Experiment 2 64
6.17 Matching Function for Real Experiment 2 65
6.18 Multiple Thresholding Matching Function using Uniform Resolution 65
6.19 Multiple Thresholding Matching Function using Variable Resolution 66
6.20 Image Sequence for Real Experiment 3 68
6.21 Matching Function for Real Experiment 3 68
6.22 Multiple Thresholding using Uniform Resolution Real Experiment 3 69
6.23 Multiple Thresholding using Variable Resolution Real Experiment 3 69

Chapter 1

Introduction

Humans perceive the world that surrounds them in a wide variety of ways. Our brain acts as a multisensor interpreter, merging together information of different types and from different sources, and giving some meaning to it. Perception is, and always has been, essential for human beings to survive and interact with the environment. From control and manipulation of objects to simple navigation, we require a great deal of coordination between our senses and our actions. Of the five different ways we have to sense the objects and events around us (i.e., seeing, hearing, tasting, smelling, and touching), vision ranks as the most important one. This importance lies not only in the fact that vision represents the richest source of perceptual information, but also from the evidence of its being a dominant sense [34]. In other words, if other senses give us conflicting information about a certain object, we usually follow the information that was retrieved through our sense of vision.

Although vision has been extensively studied in fields like neurophysiology and psychophysiology, its research attention in the computing science field has been limited to the past 30 years. The importance of vision research in computing science comes from mainly two application areas: improvement of image data for human interpretation and usage, and automatic scene analysis for autonomous machine perception. In the first area of application, image processing techniques such as image enhancement, restoration and filtering [15] have been successfully used. In the second application area, high level vision tasks such as depth estimation, motion sensing and tracking, pattern recognition and identification, and obstacle avoidance need to be performed. Unfortunately, and in spite of considerable research in computer vision a general solution for the visual perception problem is far from being achieved. But even though a general solution is not yet feasible, computer vision has been successful in many domains [9]. This success is often due to limiting the scope of the problem so that it is reduced in a way that it becomes feasible to solve. "The chances of success are greatly increased by limiting the domain of application, simplifying the task to be performed, increasing the amount of image data used, and providing adequate computing power" A. Rosenfeld [9].

Among the problems to solve in the area of computer vision, recovering 3-D structure from a scene is one of the most important and challenging. Although biological evidence shows that humans and other advanced animals use their visual systems to perform this task in an automatic way, when the problem is approached from the computer vision point of view, it is far from simple. When a camera records a scene, the 3-D structure gets mapped into a 2-D image. This results in a loss of information which makes it almost impossible to reconstruct a 3-D model of the scene from a single image. Consequently, methods that have approached the problem from a general point of view, that is, without very detailed knowledge about the objects in the scene, use multiple cameras (stereo) to simultaneously observe the scene. They obtain in this way the means for recovering at least some of the 3-D information lost. A fast and accurate depth recovering system could be used in applications such as hand-eye coordination, navigation, and obstacle avoidance, and in other tasks which are hazardous or difficult for humans to perform.

1.1 Depth Perception

Depth perception, is usually referred to as the task of determining how far an object is from an observer. We use this kind of perception continuously throughout the day — for reaching for objects or for avoiding them, even for appreciating them better. Although reliable depth estimation arises from the interpretation of several different sources of information [34], one important source of evidence is the binocular visual information (*stereopsis*). Because our two eyes are located in the front of our head, we are provided with two views of the world from slightly different reference points, and our binocular field of view is almost completely duplicated. The slight differences (or *disparities*) between the views seen by our left and right eyes allows us to perceive depth. The magnitude of the disparity, depends on the distance between the objects in the scene and the observer. If an object is very close to the viewer, the resulting disparity will be large; if the object is far away from the observer, the disparity will be very small. Figure 1.1 illustrates this disparity for a cup placed very close to the observer. As we can see, there are noticeable differences between the two views.

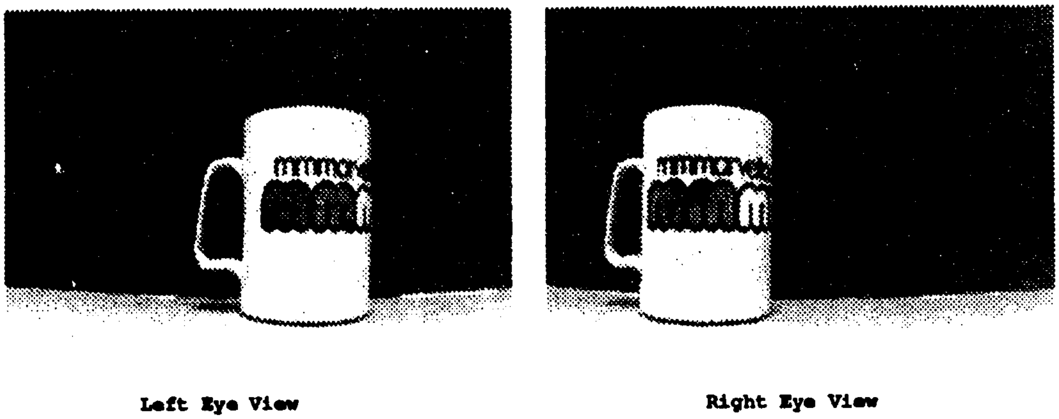


Figure 1.1: Disparity Caused by Difference in Perspectives.

Another important source of depth information is the oculomotor accommodation and convergence. This refers to the motor responses of the muscles in our eyes due to the distance between the object of interest and the observer. In order to clearly see an object, our eyes have to focus (accommodate) and converge on it.

Objects that are close to us require more focusing and convergence than objects that are farther away. Figure 1.2 shows the two views converging on a cup placed very close to the viewer.

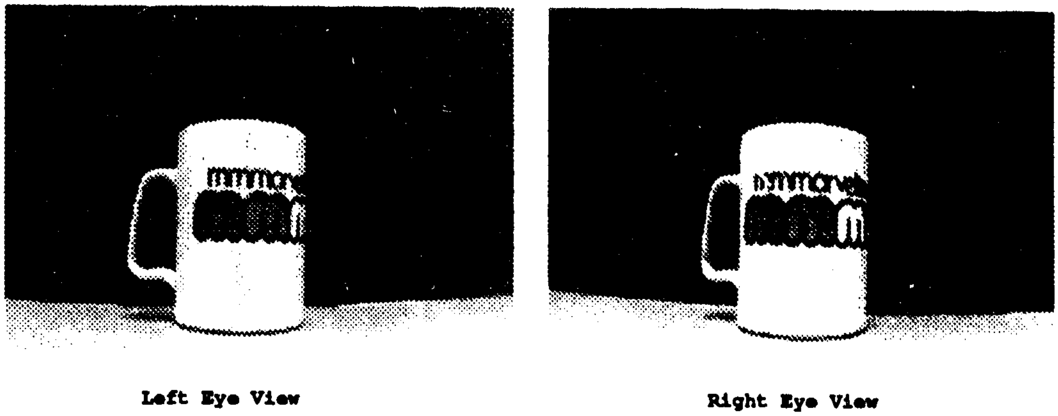


Figure 1.2: Converging Views of a Cup.

The two main approaches in computer vision to the problem of depth perception were derived from the evidence of disparity and convergence. The first one is a passive approach referred to as *Disparity Analysis.* It involves processing the information obtained with a static stereo pair of cameras which observe a scene from two different perspectives and attempt to establish correspondence between common locations in the scene. The second approach is an active one, known as *vergence control.* It transforms the passive analysis of the scene into an active interaction with it, by means of rotating one of the cameras in the stereo pair until both cameras "look at" the same object.

Current implementations of the disparity analysis approach have been successful to a certain extent [18, 11, 38]. However, this approach does not prevent problems related to occlusions and strong depth discontinuities. Additionally, it has very high computational cost to compute, and requires knowledge about the internal parameters of the camera system, mainly focal length.

The advantages of active approaches to solving several vision problems are well understood [1]. In the particular case of vergence control, the interaction with the scene not only provides the algorithm with a sequence of images to analyze but also simplifies the analysis. In order to actively observe a scene it is necessary that the eyes (cameras) have fast and accurate vergence control techniques. Eye movements are also closely linked with the process of *foveation*: information about a scene is obtained with high resolution in a small region around the point of attention of the eye, while the resolution drops continuously as we move into the periphery [33].

Although this problem has been approached in different ways, i.e., using uniform resolution, multiresolution, and log-polar transform, there is not yet an approach that clearly meets all of the following criteria :

(i) Not dependent on window size, but rather has some way of reducing the importance of the region depending on the distance from the fovea.

(ii) Has translation invariance properties in the foveal region.

(iii) Has a matching function (with vergence angle as the free variable) which increases to a peak corresponding to the correct match and then decreases.

(iv) Is very simple to design and implement.

In this thesis, we attempt to satisfy the above criteria in a single vergence control scheme, henceforth called Variable Resolution Vergence Control (VRVC). Our algorithm is based on an approximation to images obtained by fish-eye lenses, combined with a simple correlation function. The algorithm does not need to suppress the periphery; on the contrary it uses the periphery to guide the vergence process.

1.2 Thesis Objective

The objective of this work is to design a methodology for performing fast and reliable vergence control using a camera system with panning capabilities. The experiments will be conducted with real images, captured using an active camera system, and artificial images, created using a graphics environment with predefined models and a perspective projection scheme. A detailed description of both environments can be found in Chapter 5. We will focus the experimental work on the analysis and comparison of the matching function, which helps in obtaining (rapidly and reliably) the vergence angle for a given object in the scene.

1.3 Thesis Organization

The organization of the thesis is as follows: Chapter 2 briefly describes some of the previous research on passive and active stereo matching algorithms.

In Chapter 3, a summary of some of the more popular feature extraction, thresholding, and matching techniques is presented. A detailed description of the methods used in this thesis is shown.

Chapter 4 explains the Fish Eye Transform (FET), the variable resolution approach that we applied to the vergence control problem, and the significance of some of its parameters. It also discusses the modeling and calibration of fish eye lenses.

A system overview is presented in Chapter 5. The environments used to produce the images, and the process involved in the vergence control analysis are described.

Chapter 6 discusses the utility of the FET in designing a simple and fast vergence control scheme.

In Chapter 7, the experimental results of our vergence control algorithm are shown. Comparisons with a uniform resolution vergence control algorithm and discussions about the different scenes are also presented.

The conclusions and future research are presented in Chapter 8.

Chapter 2

Review of Previous Work

In this chapter, we discuss some of the different approaches to the stereo matching paradigm. We start by presenting a brief description of the so called "passive" algorithms, and then we move on to the active approaches.

2.1 Disparity Analysis

Let us first summarize the major steps of the passive stereopsis process [22] [13]. There are three main steps involved in measuring stereo disparity :

(i) A particular location on a surface in the scene must be selected from one image.

(ii) That same location has to be identified in the other image(s).

(iii) The disparity between the two (or more) corresponding image points is then measured.

In order to establish corresponding points between the two images, a preprocessing step to obtain well defined feature characteristics is often used. Lately, edge features (including not only location, but also strength and direction) have been widely used in the matching process [16] [18]. However, this has not always been the case. Earlier approaches to stereo vision used area-based matching schemes in which patches from two images were paired. Several feature extractors have been used. Among the most popular are the global edge detectors (Marr and Hildreth [23] and Canny [8]), and the window gradient edge detectors (Roberts, Sobel and Prewitt [3]).

Linear edge segments have also been considered as matching elements for stereo [26, 2]. In the segment-based matching algorithm, edge points are extracted, usually using a template (window) based edge detector, and then aligned and connected using a group of line models. The description of each edge segment is stored for the subsequent step of matching, saving information about the start and end points, orientation, and average grey-level intensity.

Once the features have been extracted from the two (or more) images, correspondence among homologous features needs to be established. This matching step is the most important stage in the process of stereopsis, a problem that is far from simple.

Earlier approaches, such as area-based and some initial feature-based algorithms, used cross correlation for matching. These approaches have several shortcomings, as pointed out by [13] [22]. Most significantly, the area-based techniques have the disadvantage of using the intensity values at each pixel directly, making them sensitive to changes in absolute intensity, contrast, illumination, and perspective caused by differences in the viewing positions.

To alleviate some of these deficiencies, several other methods [16, 17, 26, 2, 18, 6], including those using multi-resolution techniques, were developed. All these methods use either edge points or edge segments as their matching primitives, thus making them more stable towards changes in contrast and ambient lighting. Furthermore, matching among edge features makes the comparison simpler. However, these algorithms are susceptible to ambiguity in the correspondence [22], that is, a local feature or group of features in one image may match equally well with a number of features or groups of features in the other image(s).

To overcome this problem, primarily two constraints have been introduced in the literature. The first one enforces the disparities of features in a window to have similar values. That is if, for a single feature, more than one match occurs within a region (window), then the one having disparity closest to the dominant disparity in the region is accepted. Grimson's [16] initial implementation of the Marr and Poggio [22] theory uses this disambiguation criterion. Also the algorithms by Medioni and Nevatia [26] and Ayache and Faverjon [2] use the same idea applied to segment-based matching. The second constraint that has been used is the so-called *figural continuity* [25]. Figural continuity is an extension of the continuity constraint described above. It assumes that edges due to surface limits or surface markings are to be continuous, thereby resulting in continuity of disparity along the figural contours. Mayhew and Frisby [25] use this constraint to solve matching ambiguities along edge segments. Grimson [17], in his modified implementation of the Marr and Poggio theory, uses the figural continuity constraint along the edges at a coarse edge density in order to avoid ambiguity at finer edge densities.

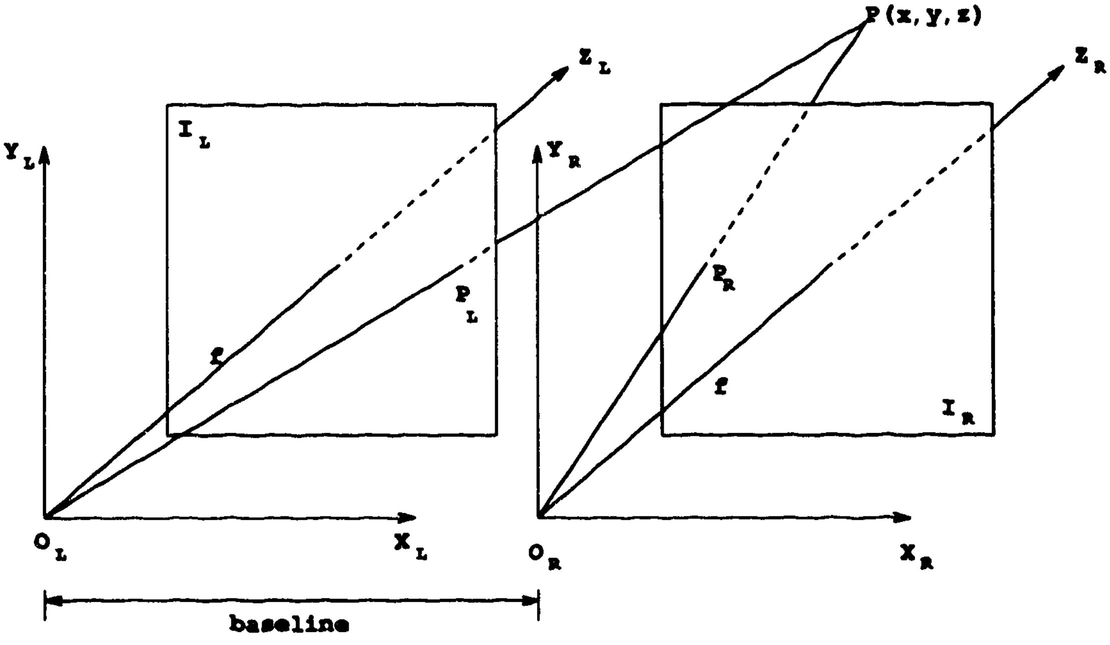


Figure 2.1: Parallel Axis Imaging Geometry.

Once the correspondence has been established, we obtain a disparity value d for every matched pair of points $P_L(X_L, Y_L)$ and $P_R(X_R, Y_R)$ as, $d = X_L - X_R$. Using the parallel axis stereo geometry model (see Figure 2.1) and the disparity information obtained, we are now able to reconstruct the 3-D position of the point $P(x, y, z)$ that originated the pair of points P_L and P_R. Figure 2.1 shows the pinhole approximation models of the two cameras with their image planes, I_L and I_R, reflected about their centers of projection, O_L and O_R. The focal length of each camera is f, and the separation between them (stereo baseline) is b. Using similar triangles, the relative x position of the point $P(x, y, z)$ with respect to O_L and O_R will be given by

$$\begin{aligned} x_L &= \frac{z_L X_L}{f} \\ x_R &= \frac{z_R X_R}{f} \end{aligned}$$

Knowing that the separation between the cameras is b, and because they are located in the same Z plane, then

$$\begin{aligned} b &= x_L - x_R \\ z &= z_L = z_R \end{aligned}$$

$$\begin{aligned} b + x_R &= \frac{z X_L}{f} \\ x_R &= \frac{z X_R}{f} \end{aligned}$$

Solving for z we will obtain

$$\begin{aligned} z &= \frac{bf}{X_L - X_R} \\ z &= \frac{bf}{d} \end{aligned}$$

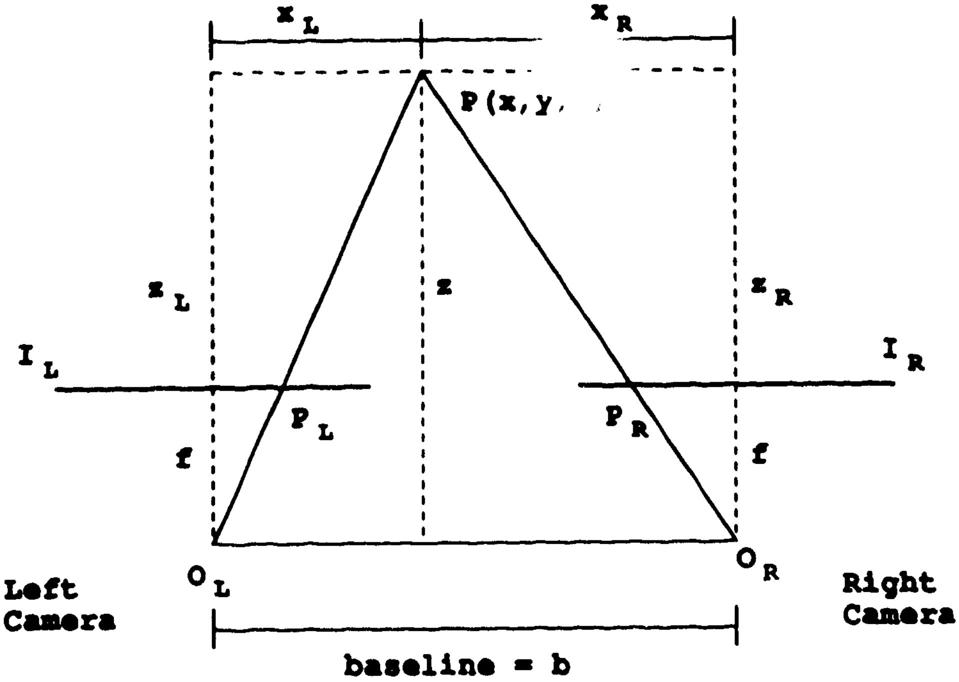


Figure 2.2: Top View of the Parallel Axis Imaging Geometry.

Assuming that the world coordinates X, Y and Z coincide with the coordinate axis of the left camera X_L, Y_L and Z_L, then the x and y positions for point P will be

$$\begin{aligned} x &= \frac{bX_L}{d} \\ y &= \frac{bY_L}{d} \end{aligned}$$

2.2 Vergence Control

The active vision model of stereopsis differs from the passive model described above in that it uses a dynamic pair (or more) of cameras. This kind of system [27] [10] is based on the model described by [30, 7]. A head has two cameras which can be tilted (rotated about the horizontal axis) together and panned (rotated about the vertical axis) independently. An additional movement, called "gaze" in the literature, pans the two cameras together in a movement equivalent to turning the neck. The problem of disparity measurement becomes one of vergence angle estimation. As described before, vergence is the process of adjusting the angle between the eyes (cameras in our system) in order that the stereo pair "looks at" the same object. Figure 2.3 shows a simplified model of the imaging geometry used for vergence control. The left camera in this figure remains orthogonal to the X axis, while the other pans attempting to locate the object of attention. A more general approach, shown in Figure 2.4, incorporates panning capabilities to the left camera as well.

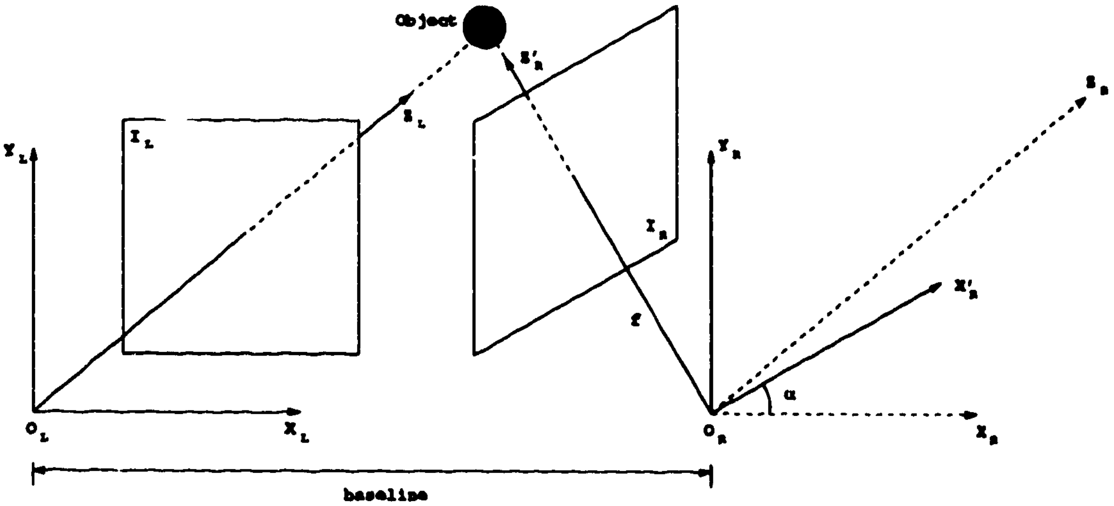


Figure 2.3: Vergence Control Imaging Geometry.

The vergence process is usually modeled as in [27]. One of the cameras is assumed to be fixed (static camera), the other pans (active camera) until both "look at" the same object. As in the passive approaches, a preliminary feature extraction step has to be executed in order to improve the performance in the matching stage. The active approaches to vergence control described in the literature [27, 10, 19] use mainly edges (strength and direction) or edge segments as their matching features.

The use of non-linear image transformation in disparity analysis was discussed in [39]. The technique developed, known as Cepstral filtering, was later applied to vergence control by various other researchers [27]. There are two problems with algorithms based on Cepstral filters. First, the window size to which the method is applied must be carefully determined, so that the object of interest lies within the windows in the stereo pair. Also the windows should not be so large that peripheral objects are included. Thus the window size is very much scene dependent. That is, in order to obtain the correct window one must have detailed knowledge of placement of objects in the three dimensional space. The second problem with the Cepstral filter is computational cost.

Other approaches like the log-polar transform [32, 37] and those that use variable resolution schemas like the one proposed in this thesis, are based on physical descriptions of the human visual system. These descriptions are the result of extensive psychophysical and neurophysiological studies that show that the mapping from the retinal space into the striate cortex is not homogeneous and that it can be summarized by the cortical magnification factor. The cortical magnification factor can be described as "the ratio of the distance moved across the surface of the cortex to the corresponding distance moved by a spot across the surface of the retina" [33]. A first approximation of this factor can be expressed as an inverse linear function of eccentricity in the retinal space, i.e., $M \propto E^{-1}$ where M is cortical magnification and E is eccentricity in the retina [29, 31, 33]. These studies also show that the retino-cortical mapping can be approximated by a complex logarithm [33]. More specifically, let (r, θ) denote the polar coordinates of the retinal image and $[u(r,\theta), v(r,\theta)]$ denote the cartesian coordinates for the cortex. We can then define the mapping using the complex variables $z = re^{i\theta}$ for the retina, $w = u(z) + iv(z)$ for the cortex, and the mapping function $w = \log(z) = \log(r) + i(\theta + 2k\pi)$ for integer k. Finally, it is emphasized that this approximation is intended to be applied only to the central $20 - 30°$ of the visual field.

The log-polar transform mapping scheme uses the idea described above, but without considering the visual field restriction. It has often been used to simplify some visual tasks [32, 37, 24, 19, 40]. This transformation is rotation invariant, i.e., radial lines (concentric circles) in a uniform resolution image get mapped onto horizontal lines (vertical lines) in the transformed image. Rotational invariance allows rotated versions of an object to appear translated on the angular axis [32, 37]. Even though the log-polar transform has certain desirable properties, its application to vergence (or disparity) estimation has been limited [35]. There are two main weaknesses in this transform. The first one lies in the fact that lines get mapped onto curves, making recovery of vergence (or disparity) extremely difficult. The second consists in the need for specialized hardware to produce such images in real-time.

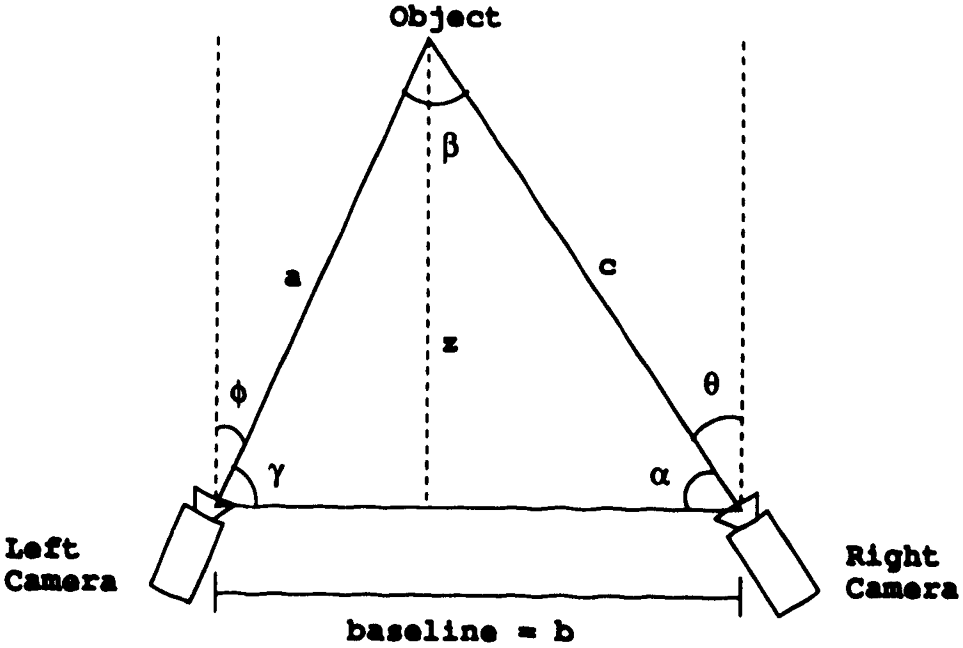


Figure 2.4: Top View of the Vergence Control Imaging Geometry.

Once the vergence angle has been estimated, we can use the law of sines to calculate the depth.

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$$

from where we obtain

$$a = \frac{b\sin\alpha}{\sin\beta}$$

$$\begin{aligned} c &= \frac{b \sin \gamma}{\sin \beta} \\ z &= a \sin \gamma = c \sin \alpha \end{aligned}$$

We will now review some of the most popular feature extraction, thresholding, and matching techniques. Later on we will consider an alternative image transform, namely the Fish Eye Transform, and study its properties.

Chapter 3

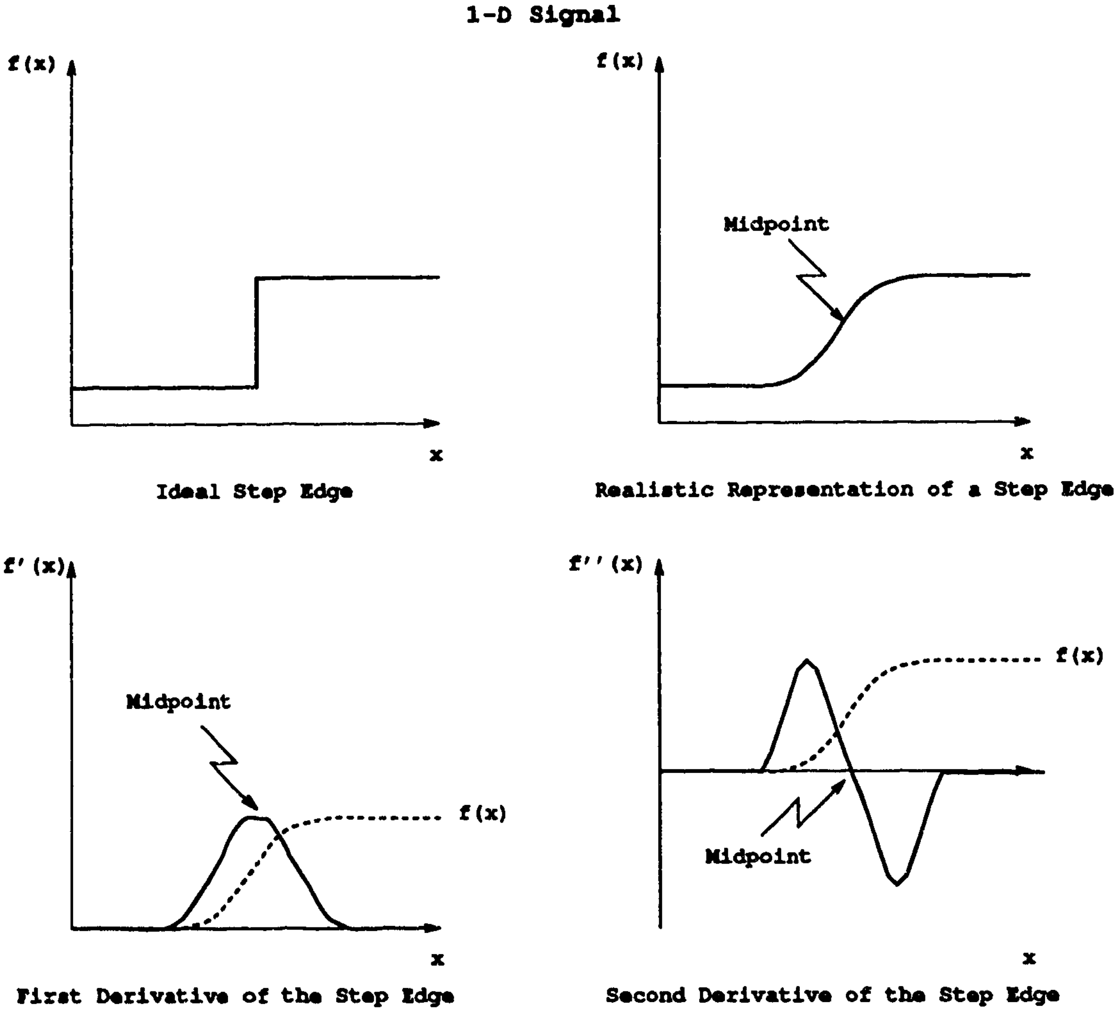
Edge Detection and Matching

Feature extraction and matching are the two main components in the process of vergence control. The feature extraction step makes the matching stage simpler and makes the whole process less sensitive to the changes in illumination, contrast, intensity and perspective caused by changes in the viewer position. Edge points and segments are image attributes which have been widely used for image analysis and classification in a wide range of applications, including disparity analysis and vergence control. In this chapter we describe some of the most popular edge detection, thresholding and matching techniques in the literature and describe in more detail the ones that we use to implement our vergence control.

3.1 Edge Detection

Because of its usefulness in high level vision tasks, edge extraction has become an important problem to solve in image processing. A large number of approaches have been presented in the literature, some of which are summarized in this chapter. Unfortunately, edge extraction is a difficult problem both to solve and to define. For this reason, we will start with a definition of the concept "edge". *An edge is defined to be an abrupt change of gray levels in an image* and its location is defined to be the midpoint (inflection point) of the edge slope [28].

Figure 3.1 shows an ideal step edge and a more realistic representation of it. It also illustrates the two most popular ways of locating edges: finding the local maxima using the first derivative of f(x), or looking for the zero crossings in the second derivative of f(x). Step edges are by far the most common type of edges present in images, but other types like roof and spike edges [12] are also present.


1-D Signal

Ideal Step Edge

Realistic Representation of a Step Edge

First Derivative of the Step Edge

Second Derivative of the Step Edge

Figure 3.1: Ideal and Realistic Step Edges

Edge extraction, as a low level vision tool, is important because it simplifies the analysis of images by drastically reducing the amount of data to be processed. Additionally, it preserves useful structural information about object boundaries so that much of the original scene information can be recovered from an edge image.

3.1.1 Classification of Edge Detectors

Many ways of classifying edge detectors have been proposed. The most popular appears to be the one that classifies them from the point of view of their originating approach [20], [12]. This classification criteria divides them in the following ways :

- Local Methods

  These methods involve convolution of the original image with a set of templates, which are based on a digital approximation of an operator that is originally applied on continuous functions.

- Regional Methods

  These methods involve the best fit of a function to a given image. Generally, the process involves fitting a functional model of an edge to an area of the original image. The best fit is formed by minimizing the error between the model and the actual image.

- Global Methods

  This is a very different approach used by experts in signal analysis and digital filtering. The edge extraction problem is viewed as one of filtering the image so that only the edges remain and all the rest is eliminated.

3.1.2 Local Methods

These methods are the oldest ones [20], [12]. They usually involve convolution of the image with a set of templates (windows), tuned to different orientations in order to identify variations of the intensity levels in such orientations.

Generally, the set of templates used is based on an approximation of an operator which is originally applied to continuous functions and must be adapted to the case of digital images.

3.1.2.1 Gradient Operators

Gradient operators are first-order derivative operators adapted to the conditions of digital images. The gradient for a continuous two dimensional function is defined by the vector :

$$\nabla f(x,y) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Its magnitude will be

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

and its orientation,

$$\alpha = \tan^{-1}\left(\frac{\partial f/\partial y}{\partial f/\partial x}\right)$$

In the discrete case, x, y and f(x,y) are positive integer numbers, so the partial derivatives involved in the gradient operator can be approximated with finite differences along the orthogonal directions x and y. Thus the x and y gradient components for the discrete case can be written as :

$$\begin{aligned} \nabla_x f(x,y) &= f(x,y) - f(x-1,y) \\ \nabla_y f(x,y) &= f(x,y) - f(x,y-1) \end{aligned}$$

and the magnitude will be

$$|\nabla f(x,y)| = \sqrt{\nabla_x f(x,y)^2 + \nabla_y f(x,y)^2}$$

it can also be approximated by

$$|\nabla_x f(x,y)| + |\nabla_y f(x,y)|$$

or by

$$\max(|\nabla_x f(x,y)|, |\nabla_y f(x,y)|)$$

Finally the orientation can be calculated as above

$$\alpha = \tan^{-1}\left(\frac{\nabla_y f(x,y)}{\nabla_x f(x,y)}\right)$$

The magnitude of an edge (also called strength of an edge) represents how fast the signal changes from a high to a low intensity level or vice versa. Its direction (only meaningful in 2-D), points to the maximum slope (maximum magnitude) of the edge.

It is important to mention at this point that the edge map of the image is usually thresholded using the magnitude of the edges as a reference, in order to eliminate noise (weak edges) as well as to obtain a more precise localization.

Roberts

Roberts operator uses a very similar concept to the one mentioned above. This approximation computes the finite differences about an ideal element (pixel) located at $(x+\frac{1}{2}, y+\frac{1}{2})$. So the finite differences are not calculated in the horizontal (x) and vertical (y) directions with respect to a particular pixel, but instead diagonally. Figure 3.2 shows the masks used by the Roberts operator.

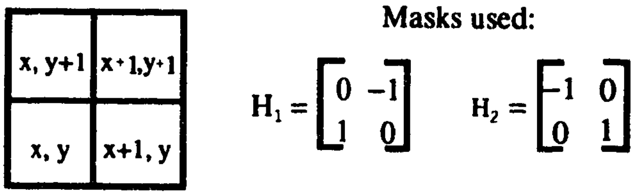

x, y+1
x+1,y+1
x, y
x+1, y
Masks used:
$H_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
$H_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

Figure 3.2: Roberts Mask

Because of the way the finite differences are calculated, this method will not provide the gradient components with respect to x and y, so the magnitude could be approximated by:

$$max(|f(x,y) - f(x+1,y+1)|, |f(x+1,y) - f(x,y+1)|)$$

Prewitt and Sobel

Prewitt and Sobel operators use the same gradient principle, with the difference of computing it over a 3 by 3 neighborhood. Convolving the original image with each of the masks illustrated in Figure 3.3 will result in obtaining the x and y gradient components, which can be combined (using the square root or the maximum) to approximate the magnitude of the gradient of the image.

Masks Used:

x-1,y+1	x, y+1	x+1,y+1
x-1, y	x, y	x+1, y
x-1,y-1	x, y-1	x+1,y-1

Prewitt

$$\nabla_x f(x,y) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \nabla_y f(x,y) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Sobel

$$\nabla_x f(x,y) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \nabla_y f(x,y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figure 3.3: Prewitt and Sobel Masks

Prewitt

$$\begin{aligned} \nabla_x f(x,y) &= \sum_{i=y-1}^{y+1} f(x-1,i) - \sum_{i=y-1}^{y+1} f(x+1,i) \\ \nabla_y f(x,y) &= \sum_{i=x-1}^{x+1} f(i,y-1) - \sum_{i=x-1}^{x+1} f(i,y+1) \end{aligned}$$

Additionally, as shown in Figure 3.3, the Sobel operator introduces weights in its summation of the values of the elements :

Sobel

$$\begin{aligned} \nabla_x f(x,y) &= \sum_{i=y-1}^{y+1} f(x-1,i) + f(x-1,y) - \sum_{i=y-1}^{y+1} f(x+1,i) - f(x+1,y) \\ \nabla_y f(x,y) &= \sum_{i=x-1}^{x+1} f(i,y+1) + f(x,y+1) - \sum_{i=x-1}^{x+1} f(i,y-1) - f(x,y-1) \end{aligned}$$

The weights are introduced in order to enhance the computation over the central pixel of the window while giving a smoothing effect over the rest.

Kirsch

Kirsch operator is another example of gradient based edge extractors using a 3 by 3 neighborhood, the main difference being that it computes the finite differences in 8 possible directions. To illustrate better the way the operator is computed we show in Figure 3.4, 3 of the 8 different masks used, with orientations of 0°, 45°, and 90°.

Masks used:

$$G_1 f(x,y) = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix} \qquad G_2 f(x,y) = \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix} \qquad G_3 f(x,y) = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \qquad \text{etc.}$$

Figure 3.4: Kirsch Masks

The equations for calculating the gradient components in the 8 different directions are;

$$\begin{aligned}
G_1(x,y) &= 3*[f(x,y+1) + \sum_{i=y-1}^{y+1} f(x-1,i) + f(x,y-1)] - 5 * \sum_{i=y-1}^{y+1} f(x+1,i) \\
G_2(x,y) &= 3*[\sum_{i=y-1}^{y+1} f(x-1,i) + \sum_{i=x}^{x+1} f(i,y-1)] - 5*[\sum_{i=y}^{y+1} f(x+1,i) + f(x,y+1)] \\
G_3(x,y) &= 3*[f(x-1,y) + \sum_{i=x-1}^{x+1} f(i,y-1) + f(x+1,y)] - 5 * \sum_{i=x-1}^{x+1} f(i,y+1) \\
G_4(x,y) &= 3*[\sum_{i=x-1}^{x+1} f(i,y-1) + \sum_{i=y}^{y+1} f(x+1,i)] - 5*[f(x,y+1) + \sum_{i=y}^{y+1} f(x-1,i)] \\
G_5(x,y) &= 3*[f(x,y-1) + \sum_{i=y-1}^{y+1} f(x+1,i) + f(x,y+1)] - 5 * \sum_{i=y-1}^{y+1} f(x-1,i) \\
G_6(x,y) &= 3*[\sum_{i=y-1}^{y+1} f(x+1,i) + \sum_{i=x-1}^{x} f(i,y+1)] - 5*[\sum_{i=y-1}^{y} f(x-1,i) + f(x,y-1)] \\
G_7(x,y) &= 3*[f(x+1,y) + \sum_{i=x-1}^{x+1} f(i,y+1) + f(x-1,y)] - 5 * \sum_{i=x-1}^{x+1} f(i,y-1) \\
G_8(x,y) &= 3*[\sum_{i=x-1}^{x+1} f(i,y+1) + \sum_{i=y-1}^{y} f(x-1,i)] - 5*[\sum_{i=x}^{x+1} f(i,y-1) + f(x+1,y)]
\end{aligned}$$

The magnitude of the gradient will be obtained by

$$|\nabla f(x,y)| = \max(|G_i(x,y)|)$$

and the orientation will correspond to the orientation of the $G_i(x,y)$ that was found to be maximum. The Sobel operator has also been approximated with this kind of directional mask.

The use of larger gradient operator masks has the advantage of increasing smoothing, thus reducing the noise sensitivity, but computing over larger neighborhoods is computationally more expensive.

3.1.2.2 Laplacian Operators

The Laplacian is a second-order derivative operator defined in the continuous case as:

$$\mathcal{L}[f(x,y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

and can be approximated on the discrete case by:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1+y) + f(x,y+1) + f(x,y-1) - 4 * f(x,y)$$

Masks used:

$$\nabla^2 f(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \nabla^2 f(x,y) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 3.5: Laplacian Masks

The laplacian operator does not give the x and y gradient components. Instead, after convolving with the mask, the zero crossings will indicate the precise location of the edges according to the point of inflection criteria described above. Figure 3.5 shows two different masks that have been used to approximate the continuous Laplacian operator.

Gradient and Laplacian methods can be combined to form a more robust edge detector operator, overcoming the weaknesses of one method with the strengths of the other (i.e. Laplacian will give the localization of the edge and Sobel will provide the magnitude).

3.1.3 Regional Methods

The use of regional methods involves the best fit of a function to a given image [20, 12, 28]. The methodology involves fitting to an n by n neighborhood a surface of degree $m < n^2$. The best fit is formed by minimizing the error between the surface of degree m and the actual image (usually the least square criteria is used).

Hueckel's approach

Hueckel considers how a theoretical edge should look (within a circular region) and then finds, for each region in the original image, how much difference exists with respect to the model, by using the least squares best fit criteria.

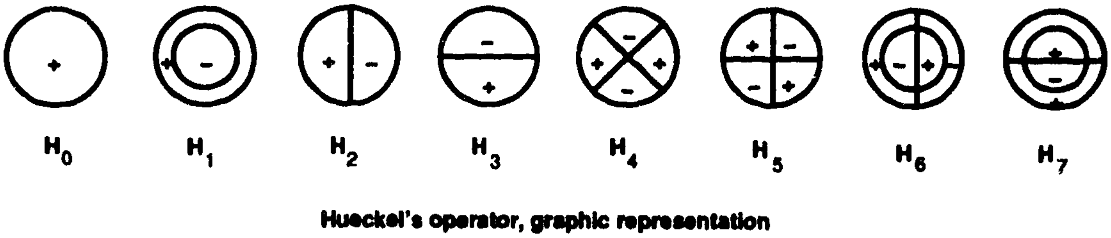


Figure 3.6: Hueckels Regional Masks

In this method, two grey levels are assumed to exist in the evaluated area and an abrupt change between them will indicate the presence of an edge. Fitting this area with the base edge operators shown graphically in Figure 3.6 will identify the candidate edges. The main disadvantage of this method is that each image has to be fitted with several different edge models in order to identify a reasonable number of edges, therefore making this approach very expensive to compute.

3.1.4 Global Methods

The use of global methods involves a completely different process. It is used in signal analysis and digital filtering. The edge extraction problem is viewed as one of filtering the image so that only the edges remain and all the rest is eliminated.

Edge detectors designed on this basis have proved to be quite efficient for a wide range of images. Unfortunately, these methods are much more computationally expensive than the gradient operators mentioned above. The process involves a convolution in the time domain of the digital image with digital approximations of the analog filters to be used. Performing the convolution in the frequency domain has been widely used to speed up the performance. As is well known, a convolution of two functions in the time domain is equivalent to the multiplication of their independent representations in the frequency domain (fourier transforms). Examples of these methods that have proved to be very effective are Marr and Hildreth's [23] zero-crossing operator, and Canny's [8] local maxima operator.

In general, the advantages of using global methods for edge extraction are :

- Image independence. They can be considered general methods that perform reasonably well over almost any kind of image.
- Noise immunity. They are considerably less noise sensitive than the other methods described above.

The main disadvantage of them is that they are computationally expensive.

3.1.4.1 Marr and Hildreth's Operator

Marr and Hildreth's operator involves a convolution of the image being processed with the laplacian of the gaussian and finding the zero-crossings of the resulting image. A zero-crossing is the place where the value of a function changes sign, i.e., passes from negative to positive or vice versa.

$$Z^{(2)}(x,y) = I(x,y) \circledast \nabla^{(2)} G(x,y)$$

In the formula above, I(x,y) represents the N × M image intensity values, G(x,y) is a two-dimensional gaussian distribution, with a standard deviation σ, of the form :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

and $\circledast$ denotes the convolution operator, which in the discrete case can be obtained by [15],

$$I(x,y) \circledast \nabla^{(2)} G(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \nabla^{(2)} G(x-m, y-n)$$

for $x = 0, 1, 2, \ldots, M-1$, $y = 0, 1, 2, \ldots, N-1$.

The Laplacian (or second derivative) of the Gaussian $\nabla^{(2)}G(x,y)$ is a circularly symmetric Mexican-hat-shaped operator whose distribution in two dimensions may be expressed by the formula

$$\begin{aligned} \nabla^{(2)}G(x,y) &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) G(x,y) \\ &= \frac{-1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

But why convolve with $\nabla^{(2)}G(x,y)$? There are two main reasons for this. The first is that the Gaussian distribution will blur the image, effectively removing all structure at scales much smaller than the space constant σ. Therefore, an appropriate value of sigma will help in removing noise from the image. An additional advantage of the Gaussian is that the blurring will be done smoothly according to the shape of the operator, making it less likely to introduce changes that were not present in the original image.

The second reason is related to using the Laplacian of the Gaussian $\nabla^{(2)}G(x,y)$ as an operator. As mentioned before, the second derivative is a very good criterion for identifying the location of an edge, in addition it is less expensive to compute than the local maxima criteria for edge localization. A zero crossing in the resulting image $Z^{(2)}(x,y)$ will represent the inflection point of an edge in the original image $I(x, y)$. Unfortunately this process only produces information about the location of the edges, but no knowledge is obtained about their magnitude or direction.

In our implementation of this method, an additional step is used for the purpose of classifying edges and removing noise. The source image is additionally convolved with the first derivatives of the gaussian with respect to x and y, thus finding the x and y ($Z_x(x, y)$, $Z_y(x, y)$ respectively) components of the gradient, so

$$\begin{aligned} Z_x(x,y) &= I(x,y) \circledast \nabla_x G(x,y) \\ Z_y(x,y) &= I(x,y) \circledast \nabla_y G(x,y) \end{aligned}$$

where $\nabla_x G(x, y)$ and $\nabla_y G(x, y)$ represent the first derivatives of the Gaussian with respect to x and y

$$\begin{aligned} \nabla_x G(x,y) = \frac{\partial G(x,y)}{\partial x} &= \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ \nabla_y G(x,y) = \frac{\partial G(x,y)}{\partial y} &= \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

Using the x and y gradient components, the magnitude and direction of each edge are calculated. The adaptive thresholding technique described in the next section is applied to remove noise (low magnitude edges). A classification of the remaining edges is implemented, with the purpose of avoiding to a certain extent false alignments during the matching stage of our vergence control process. The edges are separated in four broad groups according to their gradient orientation, two groups of horizontal edges defined by the intervals [15°, 165°), and [195° , 345°), and two groups of vertical edges defined by [345°, 15°), and [165°, 195°).

Each group is assigned a different grey-level value in order to identify later edges belonging to different groups (see Section 3.3). That is, only edges of the same group in both images will be considered as a match. Table 3.1 shows the four groups, the grey-level values assigned, and their intervals. The original idea of

	value	Start [	Finish)
$\mathcal{G}_1$ Group I	0	345°	15°
$\mathcal{G}_2$ Group II	80	15°	165°
$\mathcal{G}_3$ Group III	140	165°	195°
$\mathcal{G}_4$ Group IV	200	195°	345°

Table 3.1: Groups for Edge Classification

classifying edge pixels according to its orientation, comes from disparity analysis, where only the vertical edges are considered for matching. Furthermore, only edges with similar characteristics should be matched, i.e., edges due to changes from dark to light in the image, should not be matched with edges due to changes from light to dark.

3.2 Thresholding

Thresholding by itself covers a complete area of image processing. In this section the thresholding concept is reviewed briefly. In general, thresholding can be considered a separator function that will classify the elements of a set — in this case a set of edge pixels from an image — into two sets (could be more) separable using a threshold value.

Usually thresholding [3], [15] is represented by a binary function like

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{otherwise} \end{cases}$$

where T is the Threshold value applied to f(x,y). When applied to the edge map of an image, the threshold value is usually compared to the magnitude of the edge pixels, obtaining in this way two sets of edge points, those whose magnitude is larger than the threshold value (retained), and those whose magnitude is smaller than the threshold value (weak edges, eliminated).

The approach above, although one of the most popular, is highly image dependent and noise sensitive. Other approaches like histogram thresholding, an adaptive threshold method, have shown better general performance and image independence. In this thresholding technique, a percentage of the edge pixels are retained. The threshold value is calculated using a histogram of the magnitudes of the edges in the image, and the required percentage of edges that are to be kept.

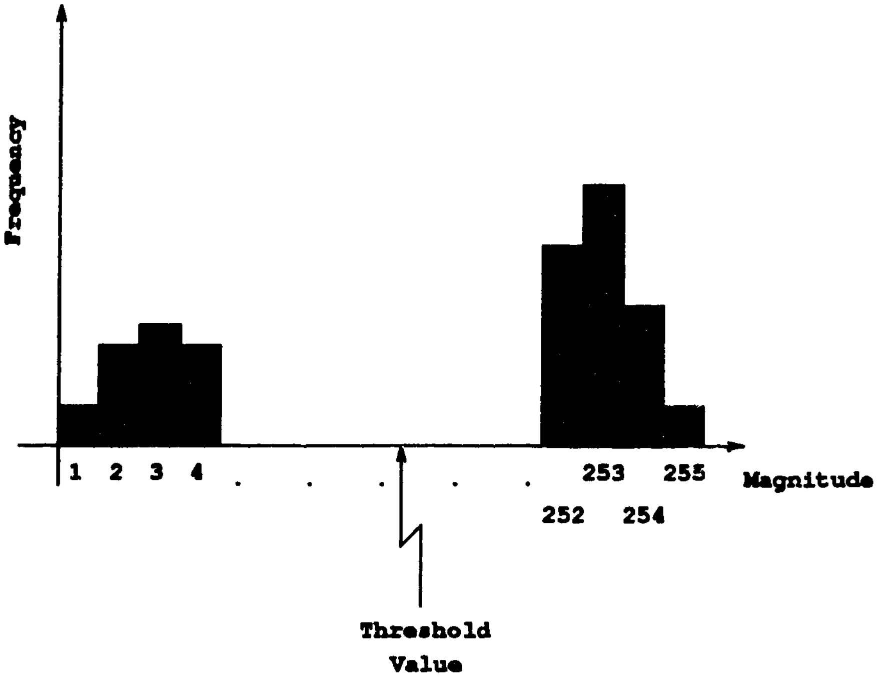


Figure 3.7: Histogram Thresholding

Figure 3.7 shows an example of the edge magnitude histogram. Once the histogram has been computed, the total number of edges to be kept will be determined by

$$K_{\text{edges}} = \frac{T_{\text{edges}} * K\%}{100}$$

where K_{edges} represents the total number of edges to be kept, T_{edges} represents the total number of edges present and $K\%$ is the percentage of edges that the user wants to keep. The threshold value will be obtained by adding the frequencies from higher to lower magnitude until the number of edges is larger or equal to K_{edges}.

Thresholding is an important approach to performing image segmentation. In the case of edge detection, it is useful to filter the magnitudes obtained from the gradient operators above, generally filtering out the weak (low magnitude) edges and thus removing most of the noise of the image.

3.3 Matching

Matching, within the context of disparity analysis and vergence control, can be defined as the process of establishing correspondence among common features in two or more images. The problem has been approached in several different ways and in this section we will describe briefly the most popular ones. Let us start by describing the matching approaches used by the disparity analysis methodology.

Earlier approaches to disparity analysis, such as area-based and some initial feature-based algorithms, used cross correlation for matching. The main idea behind this methodology is to find the highest correlation value for a patch or subimage $w(x,y)$ of size $m \times n$, taken from the left image $L(x,y)$, within the right image $R(x,y)$ of size $M \times N$. The correlation between $R(x,y)$ and $w(x,y)$ is defined by [15],

$$C(s,t) = \sum_{s=0}^{M-1} \sum_{t=0}^{N-1} R(s,t)w(x+s,y+t)$$

where $x = 0,1,2,\ldots,M-1$, $y = 0,1,2,\ldots,N-1$, and the summation is evaluated over the region where R and w overlap. This methodology is not only very expensive to compute but also, as we pointed out in Chapter 2, using patches of intensity values to perform the matching process makes it very sensitive to changes in absolute intensity, contrast, illumination, and perspective caused by differences in the viewing positions.

To make the systems more stable towards changes in illumination and contrast, other methods [16, 17, 26, 2, 18, 6], started using features like edges or edge segments as their matching primitives. Furthermore, matching among edge features makes the comparison simpler. An additional simplification of the problem, called *epipolar constraint*, makes the matching process less complicated. This simplification is due to the use of parallel imaging geometry, i.e. the pair of cameras in the system have their optical axes mutually parallel (see Figure 2.1 in Chapter 2). Since the displacement between the optical centers of the two cameras is purely horizontal, the position of corresponding points in the two images can differ only in the horizontal component, therefore restricting the search space for matching to a line that is called the epipolar line. Consequently, our matching problem has been reduced to that of finding for each feature in the left image its corresponding pair along the epipolar line in the right image. However, these algorithms are susceptible to ambiguity in the correspondence [22], that is, a local feature or group of features in one image may match equally well with a number of features or groups of features in the other image(s).

To overcome the ambiguity problem described above, various approaches have been proposed in the literature. Initially, additional information about the edge features, like strength and direction or average strength for edge segments, was used. That is, only features with similar strength and direction were considered to match, reducing in this way the probability of spurious matches. Unfortunately this kind of restriction is not enough to cope with the ambiguity problem, and more robust constraints had to be designed. The two main constraints reviewed in the literature use information about the disparity values of the neighbor features to solve the ambiguity. The first constraint enforces the disparities of features in a window to have similar values. That is if, for a single feature, more than one match occurs within a region (window), then the one having disparity closest to the dominant disparity in the region is accepted. The second constraint that has been used is the so-called *figural continuity* [25]. Figural continuity is an extension of the continuity constraint described above. It assumes that edges due to surface limits or surface markings are to be continuous, therefore resulting in continuity of disparity along the figural contours.

Now let us summarize the matching approaches used in vergence control, approaches which are considerably more simple than the one described above for disparity analysis. Being an active approach, vergence control interacts with the environment by panning the active camera in the system until it "looks at" the object of attention of the passive camera. This panning process is analogous to the search performed along the epipolar line in the disparity analysis methods, therefore the matching problem is reduced to that of evaluating the proportion of match between the two current images. In other words, without doing any search, the corresponding features between the two images are computed. Consequently, the matching process is performed iteratively comparing the original image obtained from the static camera with each of the images captured during the panning process of the active camera.

Using the same conjectures as above, vergence control uses edges or edge segments as matching primitives. Direction and/or strength are also used to reduce the ambiguity problem. Strictly speaking, for a given vergence angle α, a feature $f(x,y)$ in the static image $I_s(X,Y)$ is considered to have a match if there is a feature $g(x,y)$ with similar characteristics (similar direction and/or similar strength) in the same position in the active image $I_\alpha(X,Y)$. This consideration though is too restrictive and sensitive to errors due to perspective and feature extraction. Our implementation considers a tolerance window for the match to occur. Additionally the edge features need to belong to the same group (see Table 3.1). That is, feature $f(x,y)$ will be considered to have a match if there exists a feature $g(i,j)$ in the active image $I_\alpha(X,Y)$ such that

$$f(x,y) \in \mathcal{G}_m \text{ AND } g(i,j) \in \mathcal{G}_m \text{ AND } x-\frac{d}{2} < i < x+\frac{d}{2} \text{ AND } y-\frac{d}{2} < j < y+\frac{d}{2}$$

where d defines the size of the tolerance window, and $\mathcal{G}_m$ denotes the orientation group to which $f(x,y)$ and $g(i,j)$ belong.

Figural continuity constraints could be added to the matching process in order to further reduce ambiguity problems, but this approach has not been addressed in the literature, nor was it implemented as part of the matching process in this thesis.

Chapter 4

Fish Eye Transform

Only recently have variable-resolution imaging systems which imitate biological systems such as the human eye been used in computer vision research. The advantages of these methods have been briefly discussed earlier. The fish eye transform (FET) introduced in this chapter, describes a variable resolution mapping function that generates an image with a high resolution fovea and a non-linearly decreasing resolution towards the periphery. The FET is based on a simplification of the complex logarithmic mapping described by Schwartz in [33]. This mapping is an approximation of the cortical magnification factor existent in humans and other primates.

Figure 4.1 (a) shows an image and an approximation of the retino-cortical mapping described by Schwartz (Figure 4.1 (c)) [33]. As can be seen, the foveal region is projected at very high resolution, and resolution decreases continuously in the periphery. The log-polar transform approximation which has been applied to several visual tasks is also shown (Figure 4.1 (d)).

One problem of applying Schwartz's method directly is that image continuity across the vertical meridian is lost, and cannot be recovered easily. In our approach we use a simplified variable-resolution (VR) projection method of the following form : Let (r, θ) denote the polar coordinates of the retinal image, i.e., if (x, y) represent the cartesian coordinates in the retina, $r = \sqrt{x^2 + y^2}$ and

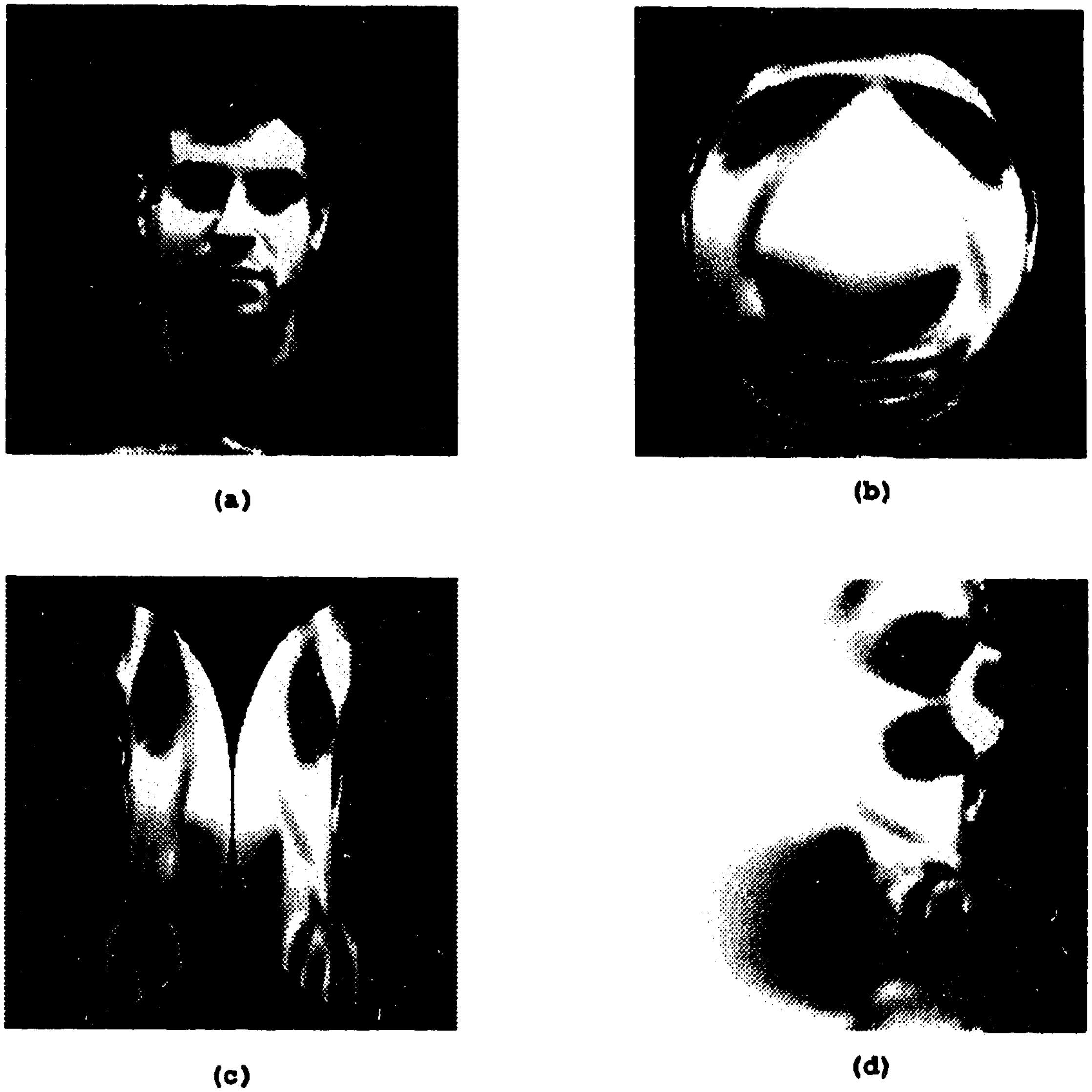


Figure 4.1: (a) Input image, (b) Our approximation,
(c) Schwartz approximation (d) Log-Polar approximation.

$\theta = \tan^{-1}(\frac{y}{x})$, then the cortical polar coordinates (ρ, θ^*) can be obtained as

$$\rho = s\log(1+\lambda r) \tag{4.1}$$

$$\theta^* = \theta \tag{4.2}$$

and the corresponding cortical cartesian coordinates (x^*, y^*) are given by

$$x^* = \rho\cos\theta^* \tag{4.3}$$

$$y^* = \rho\sin\theta^* \tag{4.4}$$

where s is a simple scaling factor, and λ controls the amount of distortion over the whole retinal image. The inverse mapping is given by

$$\begin{aligned} \rho &= \sqrt{x^{*2}+y^{*2}} \\ \theta^* &= \tan^{-1}(\frac{y^*}{x^*}) \end{aligned}$$

$$\begin{aligned} r &= \frac{(e^{(\frac{\rho}{s})}-1)}{\lambda} \\ \theta &= \theta^* \\ x &= r\cos\theta \\ y &= r\sin\theta \end{aligned}$$

Figure 4.1 (b) shows the resulting cortical image after using this simplified projection method ($\lambda = 0.5$, $s = 26.3385$). As we can observe, the image continuity is preserved. The disadvantage of the simplified mapping function is that it produces a strong anisotropic distortion in peripheral regions for large values of the distortion parameter λ.

It is important to note that our approximation can be implemented simply by using a special lens, such as wide angle or fish-eye. This is a major advantage of the simplified VR mapping.

4.1 Modeling and Calibration of Fish Eye Lenses

In order to better illustrate how our approximation can be implemented using simple lenses, we present in this section the experimental results of modeling and calibrating the distortion caused by fish eye lenses. Two modeling alternatives are evaluated, one using the FET (Equations 4.1 - 4.4), and the other using a polynomial function to approximate the distortion. That is, the polynomial fish eye transform (PFET for short) differs from the FET (Equation 4.1) in that $\rho = G(r)$ where $G(r)$ is a polynomial in r.

To be able to measure the amount of distortion caused by a fish eye lens (for the purpose of modeling it later), we need to first determine the optical center of distortion (henceforth referred to as focus of distortion or FD). The focus of distortion, unfortunately, does not usually correspond to the center of the digitized image, therefore a methodology for establishing it accurately was developed. The main steps of this technique are :

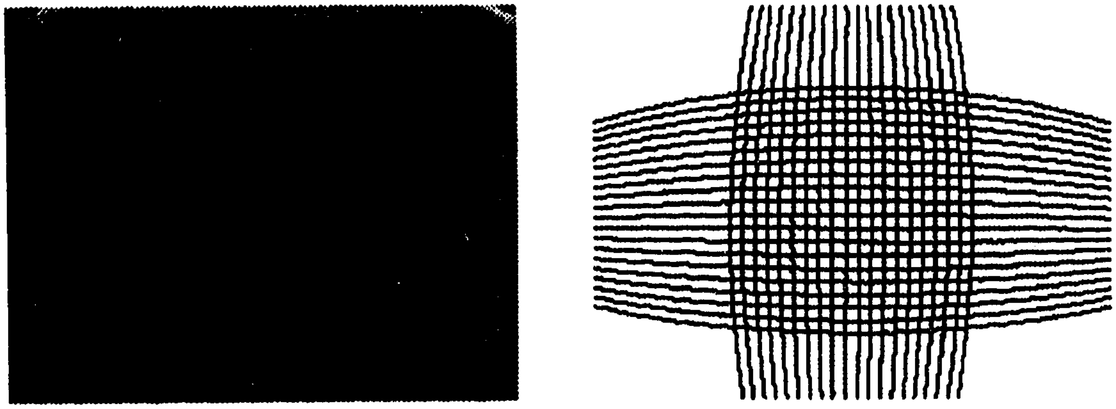
(i) A set of pictures is taken using a grid.

(ii) The curvature of each line in the grid is estimated and the lines with minimum curvature are chosen. In the event of ties an average is used.

(iii) The intersection of the minimum curvature lines in the two orthogonal directions indicates the approximate center of distortion of the picture.

An average of several estimates of the center of distortion is evaluated in order to obtain a better estimate. In Figure 4.2 we show a picture of the grid (left) and the line pattern that was used to determine the curvature (right). Table 4.1 shows the data collected after running the process over 10 grid images, resulting in an estimated focus of distortion of (254.3, 222.9). Once the focus of distortion has been calculated, the distortion over any direction can be evaluated. Considering the FD as the point of reference, the distortion is similar in all directions.

Using the same grid images described above, we obtained a measure of the distortion in four different directions (up, down, left and right) and averaged



Fish-eye Grid Image

Line Extraction

Figure 4.2: Fish Eye Grid Image and its Curvature

their values to obtain a more reliable estimate. The distortion is expressed in terms of radial distance (in pixels) between a pixel of the grid and the focus of distortion, and is determined by comparing the expected position of such a pixel with its measured position in the image. Figure 4.3 shows how the distortion propagates in each direction, and its average.

The objective now is to find a function that represents the average distortion calculated above. As mentioned previously, we demonstrate two approaches to this problem. The first one fits a polynomial function to the average distortion data using the least squares method. The second one uses the same idea to match the FET described above.

The least squares method for curve fitting [14, 36] is a standard procedure for determining the coefficients of a nonlinear function so that it represents a set of recorded data. The accuracy is determined by minimizing the error between the values predicted by the function and the actual data.

The polynomial (PFET) model, is similar to the FET except that $\rho = G(r)$. Where r is the radial distance for an undistorted image, ρ is the measured one, and $G(r)$ has the form

$$G(r) = a_0 + a_1 r + a_2 r^2 + \cdots + a_n r^n = \sum_{j=0}^{n} a_j r^j$$

	x	y
image0	262	221
image1	252	221
image2	253	221
image3	253	221
image4	250	222
image5	254	222
image6	250	216
image7	260	229
image8	260	228
image9	249	228
Average	254.3	222.9
Standard Deviation	4.45	3.91

Table 4.1: Focus of Distortion Estimate

Using this polynomial function and a set of N data pairs (r_i, ρ_i), the objective of the least squares method is to minimize an error function χ

$$\chi = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (\rho_i - G(r_i))^2$$

which is equivalent to

$$\chi = \sum_{i=1}^{N} (\rho_i - \sum_{j=0}^{n} a_j r_i^j)^2$$

For the minimum value, all the partial derivatives $\partial\chi/\partial a_0, \partial\chi/\partial a_1, \cdots, \partial\chi/\partial a_n$ vanish, so we obtain

$$\begin{aligned}
\frac{\partial\chi}{\partial a_0} &= \sum_{i=1}^{N} 2(\rho_i - \sum_{j=0}^{n} a_j r_i^j) \quad (-1) &= 0 \\
\frac{\partial\chi}{\partial a_1} &= \sum_{i=1}^{N} 2(\rho_i - \sum_{j=0}^{n} a_j r_i^j) \quad (-r_i) &= 0 \\
&\vdots \\
\frac{\partial\chi}{\partial a_n} &= \sum_{i=1}^{N} 2(\rho_i - \sum_{j=0}^{n} a_j r_i^j) \quad (-r_i^n) &= 0
\end{aligned}$$

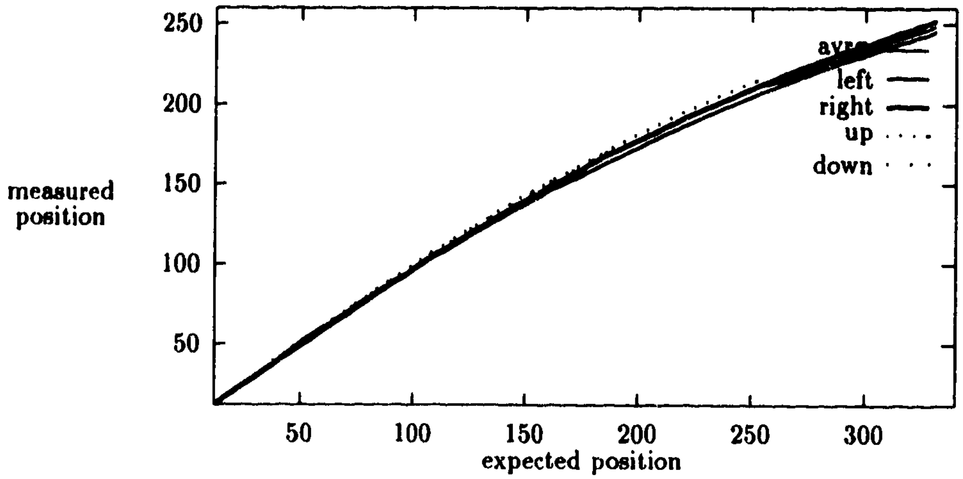


Figure 4.3: Graph of Fish Eye Distortion in Different Directions and the Average Value

Dividing all the equations by -2 and reordering we obtain the following set of $n+1$ equations with $n+1$ coefficients.

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=0}^{n} a_j r_i^j & & = \sum_{i=1}^{N} \rho_i \\
\sum_{i=1}^{N}\sum_{j=0}^{n} a_j r_i^j & \quad (r_i) & = \sum_{i=1}^{N} \rho_i r_i \\
& \vdots & \\
\sum_{i=1}^{N}\sum_{j=0}^{n} a_j r_i^j & \quad (r_i^n) & = \sum_{i=1}^{N} \rho_i {}^n
\end{aligned}$$

To get a solution to the set of equations above, we can use almost any method for solving a set of linear equations. In our case we used the Gauss-Jordan Elimination method.

The problem now is deciding what degree of polynomial represents accurately the recorded data. A standard way of solving this problem is to use the variance of the fitted model with the observed data. One increases the degree of the polynomial as long as there is a significant decrease in the variance σ^2, which is

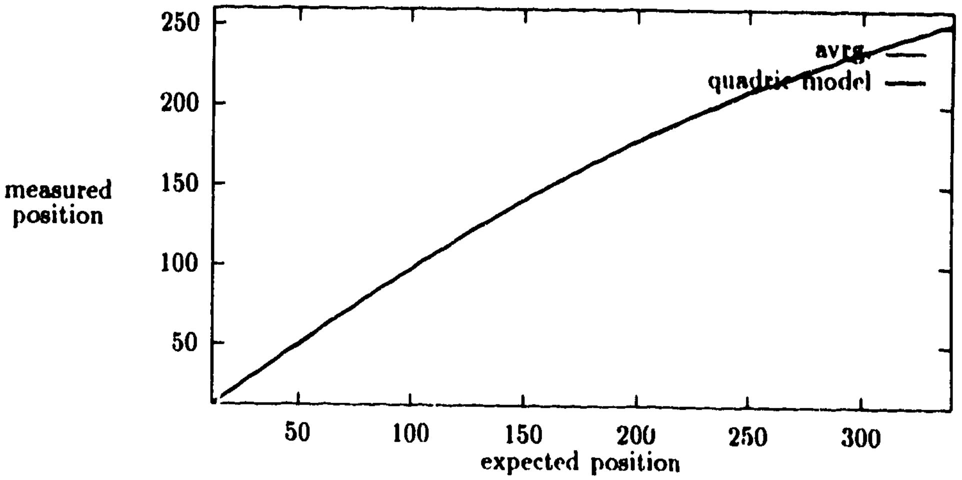


Figure 4.4: Goodness of Fit for PFET Model

computed by

$$\sigma^2 = \frac{\chi}{N-n-1} = \frac{\sum_{i=1}^{N} e_i^2}{N-n-1}$$

For the PFET model described above, the quadric polynomial

$$0.3801167 + 0.972645r + 5.6469E - 4r^2 - 6.778E - 6r^3 + 9.1484E - 9r^4$$

was found to be a reasonably good approximation of the average distortion, having a variance of 0.192238. Although it was not the best approximation, it was chosen because the decrease in the variance for the polynomials of higher degree was not significant. Figure 4.4 shows the recorded average distortion and a plot of the polynomial model above. Figure 4.6 (left) shows the grid picture compensated using this model.

Now let us move to the analysis of the FET model. The FET model can also be expressed in terms of the radial distances r and ρ described above. The function $\rho = H(r)$ in this case is

$$\rho = H(r) = s\log(1 + \lambda r)$$

Using the same concept of least squares we try to minimize a function χ of the form

$$\chi = \sum_{i=1}^{N} (\rho_i - s\log(1+\lambda r_i))^2$$

At the minimum, the two partial derivatives $\partial\chi/\partial s$ and $\partial\chi/\partial\lambda$ vanish, so we obtain

$$\begin{aligned}\frac{\partial\chi}{\partial\lambda} &= \sum_{i=1}^{N} -2(\rho_i - s\log(1+\lambda r_i))(\frac{s r_i}{1+\lambda r_i}) = 0 \\ \frac{\partial\chi}{\partial s} &= \sum_{i=1}^{N} -2(\rho_i - s\log(1+\lambda r_i))(\log(1+\lambda r_i)) = 0\end{aligned}$$

Unfortunately, the unknown coefficients are not in a linear relation to the equations as they were before, thus methods like Gauss-Jordan elimination cannot be applied. Instead, successive evaluation techniques, and Newton's method [14] can be used. Using such techniques, we obtained the following coefficients :

$$318.177564\log(1+0.0036612r)$$

Figure 4.5 shows the average distortion function and a plot of the FET model above. Figure 4.6 (right) shows the grid image after using this model for compensation.

Although a better approximation can be achieved using the polynomial model, the number of coefficients involved is much larger (in this case 5) and their interpretation is unclear. In the case of the FET model, the amount of distortion can be easily changed by modifying the "λ" parameter and the scaling factor "s" accordingly, while in the polynomial model this change will imply a simultaneous modification of many coefficients. Furthermore, an inverse mapping to restore the original image can be obtained using the inverse function of the simplified FET model. There is no simple way to obtain the inverse transform for the polynomial model.

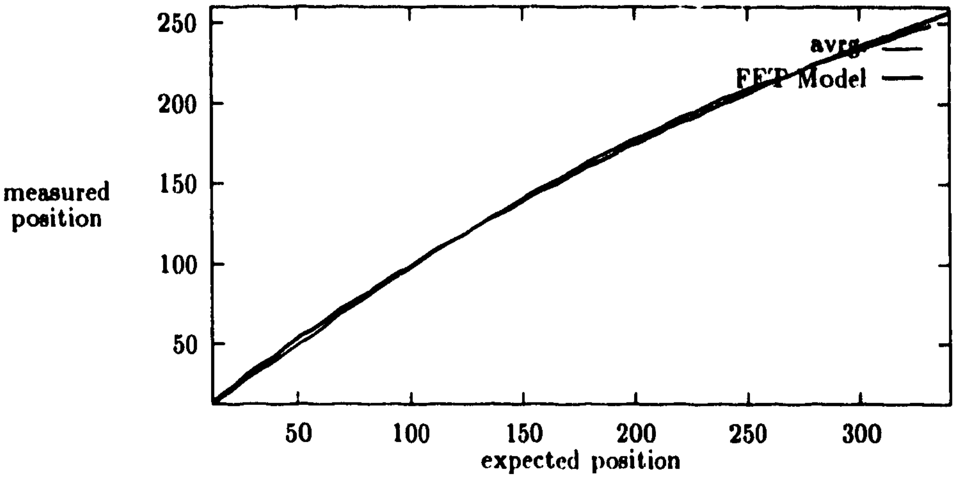


Figure 4.5: Goodness of Fit for FET Model

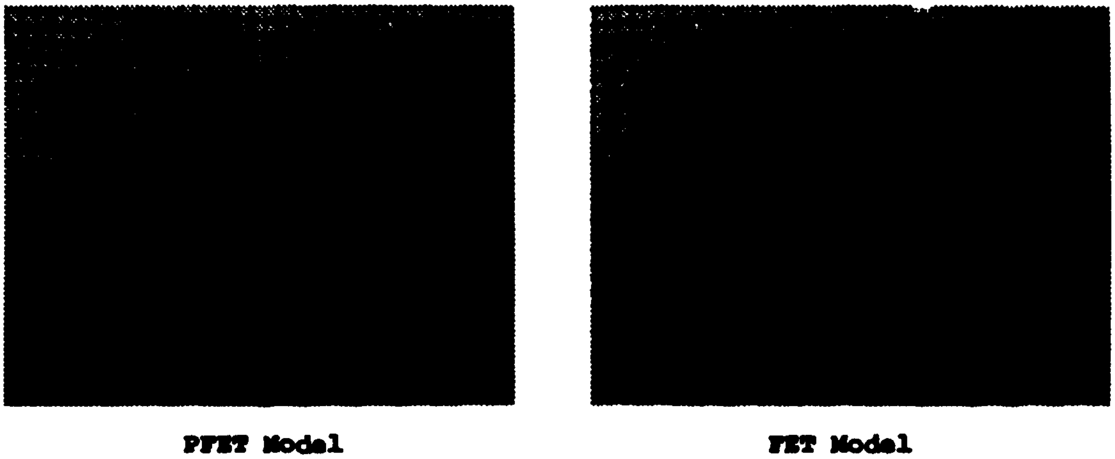


Figure 4.6: Distortion Compensated

Chapter 5

Application of Variable Resolution in Vergence Control

In this chapter we describe how variable resolution helps in the vergence process. We show, using theoretical analysis, the difference between the matching functions for the uniform and variable resolution cases. A short description of this work can be found in [5].

Vergence estimates are obtained by applying a simple correlation function to the left and right edge images. For simplicity in analysis we assume the following model (the actual matching technique used is described in Chapter 3). For a given vergence α, a pixel (i,j) in the left image is considered to have a match if $U_l(i,j) = 1$ and $U_r(x,j) = 1$, $i - c/2 \leq x \leq i + c/2$, where c denotes the range in which an edge match is accepted, and U_l, U_r represents the edge maps in the left and right images respectively. Vergence estimates are obtained by summing edge matches between the left and right images, centered at the point of attention in both cases. The correlation function $C^{\alpha}(i,j)$ is expected to have a local maximum for the correct vergence angle $\alpha = \alpha_0$.

We will now consider the behavior of the matching function from a statistical point of view. Let p denote the probability of having an edge at a pixel in a given window. For the correct vergence angle the value of $C^{\alpha}(i,j)$ is expected to be p, since the edges match in the left and right window. However, when the angle is outside the acceptance range of one pixel of length c, $C^{\alpha}(i,j)$ is expected to be p^2, i.e., the probability that two edge points in the left and right window match by chance. In other words,

$$C^{\alpha}(i,j) = \begin{cases} p & \text{if } \alpha_0 - \frac{c}{2} \leq \alpha \leq \alpha_0 + \frac{c}{2} \\ p^2 & \text{otherwise} \end{cases}$$

If resolution is increased, the acceptance range c decreases correspondingly. Hence if the resolution is increased by a factor r with respect to the unit of resolution in the previous equation, then:

$$C^{\alpha}_{r}(i,j) = \begin{cases} p & \text{if } \alpha_0 - \frac{c}{2r} \leq \alpha \leq \alpha_0 + \frac{c}{2r} \\ p^2 & \text{otherwise} \end{cases}$$

The above equation relates the correlation function to the resolution in an image. A sketch of the matching function C^{d}_{r} is shown in Figure 5.1 (a). As can be seen, the matching function provides no clue about the location of α_0 outside the matching range $[\alpha_0 - \frac{c}{2r}, \alpha_0 + \frac{c}{2r}]$.

Consider now the case of variable resolution and its effect on the shape of the matching function. Assume that resolution varies between R_1 and R_2 and for simplicity further assume that $R_1 = 0$ and $R_2 = R$. To find the matching function, $V^{\alpha}(i,j)$, in this case we integrate the matching function $C^{\alpha}_{r}(i,j)$ over r varying in the range $[0, R]$. That is,

$$V^{\alpha}(i,j) = \int_0^R C^{\alpha}_{r}(i,j)dr$$

or,

$$V^{\alpha}(i,j) = \int_0^R pI(\alpha \in (\alpha_0 \pm \frac{c}{2r}))dr + \int_0^R p^2 I(\alpha \notin (\alpha_0 \pm \frac{c}{2r}))dr$$

(Here I denotes the indicator function. That is, $I(A) = 1$ if condition 'A' is true, and is equal to 0 otherwise.)

$$= E_1 + E_2$$

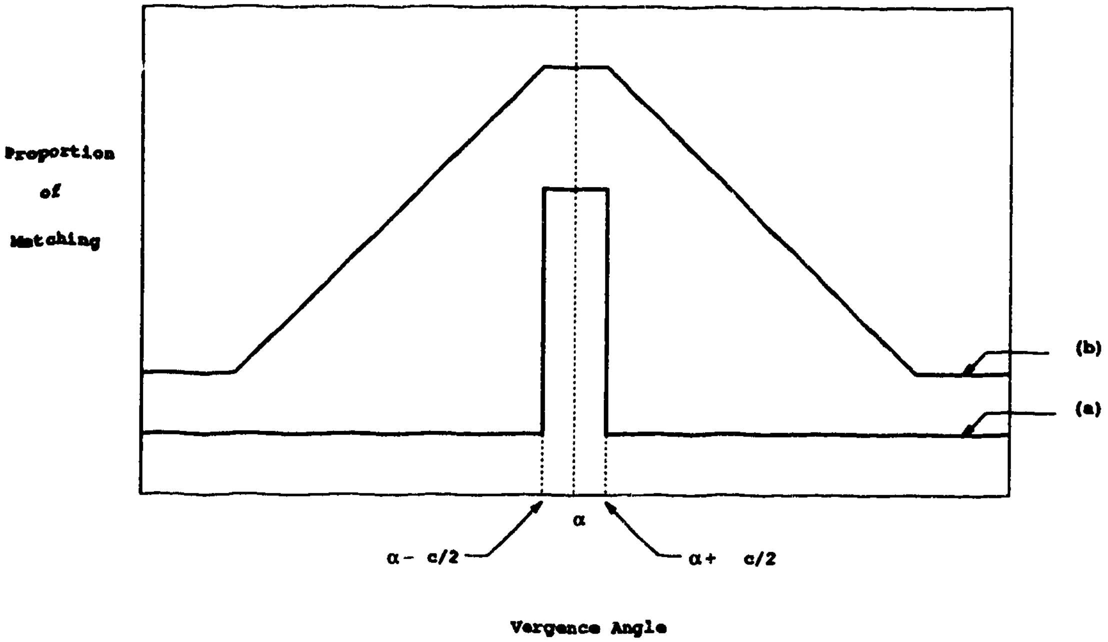


Figure 5.1: Uniform resolution and Variable resolution ideal matching functions

Now,

$$E_1 = \begin{cases} \int_0^R p dr & \text{if } \frac{c}{2|\alpha_0-\alpha|} > R \\ \int_0^{\frac{c}{2|\alpha_0-\alpha|}} p dr & \text{if } \frac{c}{2|\alpha_0-\alpha|} \leq R \end{cases}$$

or,

$$E_1 = \begin{cases} pR & \text{if } \frac{c}{2|\alpha_0-\alpha|} > R \\ p\frac{c}{2|\alpha_0-\alpha|} & \text{if } \frac{c}{2|\alpha_0-\alpha|} \leq R \end{cases}$$

Similarly,

$$E_2 = \begin{cases} 0 & \text{if } \frac{c}{2|\alpha_0-\alpha|} > R \\ \int_{\frac{c}{2|\alpha_0-\alpha|}}^R p^2 dr & \text{if } \frac{c}{2|\alpha_0-\alpha|} \leq R \end{cases}$$

That is,

$$E_2 = \begin{cases} 0 & \text{if } \frac{c}{2|\alpha_0-\alpha|} > R \\ p^2(R - \frac{c}{2|\alpha_0-\alpha|}) & \text{if } \frac{c}{2|\alpha_0-\alpha|} \leq R \end{cases}$$

Thus,

$$V^{\alpha}(i,j) = \begin{cases} pR & \text{if } \frac{c}{2|\alpha_0-\alpha|} > R \\ p\frac{c}{2|\alpha_0-\alpha|} + p^2(R - \frac{c}{2|\alpha_0-\alpha|}) & \text{if } \frac{c}{2|\alpha_0-\alpha|} \leq R \end{cases}$$

Or,

$$V^{\alpha}(i,j) = \begin{cases} pR & \text{if } \alpha \in (\alpha_0 \pm \frac{c}{2r}) \\ p(1-p)\frac{c}{2|\alpha_0-\alpha|} + p^2R & \text{if } \alpha \notin (\alpha_0 \pm \frac{c}{2r}) \end{cases}$$

Intuitively the integral is obtained as follows. If the vergence is very close to the actual value, then at any resolution the value of the matching function is p, thus the value pR in the first case. When the value of the vergence (α) is not very close to the true value, then the integral is a weighted sum of p and p^2 with the weights being determined by how close α is to the true value.

The shape of the VR matching function is shown in Figure 5.1 (b). Note that the plateau of the VR matching function is determined by the maximum resolution R. As R increases the plateau becomes narrower. Comparing the shapes of the uniform-resolution matching function (Fig. 5.1 (a)) and the VR matching function (Fig. 5.1 (b)), we note an important difference. The uniform-resolution function is a step function, and the values outside the correct matching range provide no clue about the location of the peak. Moreover, at higher resolutions the matching range becomes narrower. This is not the case for VR matching. The gradual slope of the VR matching function allows a relatively simple search technique for locating the peak. This is further illustrated in Chapter 6.

Chapter 6

Methodology and Experimental Results

In the previous chapters, we have discussed different approaches to the problems of feature extraction, thresholding, matching and variable-resolution. A detailed description of the methods that were implemented as part of this vergence control system was given. In this chapter we describe the tools and equipment that were used, how the different modules interact with each other, and a group of experiments that will show the performance of our method. We start by describing the programming environment and the equipment layout, followed by the specifications of the two environments used to generate the images. A complete explanation of the methodology is presented next, finally we present four experiments that will help us illustrate better the advantages of using our technique.

6.1 Programming Environment and Equipment Layout

All the modules described in this thesis were implemented in "C" language. The user interfaces were implemented using tools from the X11/XView libraries, with the exception of the user interface of the graphics environment, which was implemented using tools from the VOGLE graphics library over a suntools/sunview environment. Excluding the real image environment, all the software was executed using a SPARC IPC Sun Workstation. A SUN3 with a Data-Translation DT1451 frame digitizer board, connected via VME bus, was used for capturing images in the real image environment. A CCD camera (NEC TI-23A) with standard video output was used as the input device connected to the DT1451. A description of the camera mounting is given in Section 6.2.2.

6.2 Image Platforms

The experiments are conducted using images generated by two types of environments. In the first one, artificial scenes are created using a graphics interface. In the other, real images are captured by a camera system with panning capabilities. The separation between the two viewing positions (cameras), and the angle rotated in each pan step (manual for the real image environment), are configurable parameters in both environments.

6.2.1 Graphics Environment

The graphic environment provides a mechanism for creating artificial scenes based on predefined object models (chair, cube, shelf and desk) that can be placed at any position within the limits of the scene. This platform works by providing different perspective projections (from two different view points along the horizontal axis) of the artificially created scene. The different perspective projections simulate the positions of the left and right cameras, as well as the panning process involved in vergence. The basic add, move, and delete object tools are provided, and special tools like changing the viewer position or executing the vergence process are also available. During the vergence process, a sequence of images (raster file format) are generated. One from the left perspective and a sequence from the right perspective, each after performing a panning step (predefined angle). Finally, the scene description can be saved for further usage (repeat the experiment) or reference.

6.2.2 Real Image Environment

The camera system, along with the software written for it, provides a mechanism for capturing images of a real scene from two different viewing positions along the horizontal axis. The system has panning capabilities, thereby providing the means for performing the vergence process. Unfortunately, rotations equivalent to tilt or gaze are not available.

Right Viewing Position **Left Viewing Position**

Figure 6.1: Active Camera System.

The camera system, consists of one camera mounted in a sliding device with preset locking positions (6.5 and 13 inches), which will define the baseline separation of the simulated stereo system. The bottom of the camera is attached to a gear that allows it to rotate about the "y" axis (pan), providing in this way the means of performing the vergence process. All the movements of the camera are currently performed manually. Figure 6.1 shows the camera system and the left and right viewing positions.

A user interface was developed to provide a high level tool for capturing images. It makes use of the data-translation library (libdt.a), which provides mechanisms for accessing the DT1451 memory via VME bus. The interface is mouse driven and runs over a X11/Xview environment. The images captured by the interface are 512 pixels width × 480 pixels height and are stored in standard raster format assuming a 256 grey-scale color map.

6.3 Methodology

Images from two view points are generated, either created artificially or captured by the camera system. The left or right camera, henceforth called the static camera, is adjusted so that it "looks at" the desired object, i.e., has the object over which we will perform the verging process in the center of the image. The opposite camera (left or right), henceforth called the active camera, will start from a viewing position that does not have the object in question in the center, and will execute a sequence of panning steps. The panning is performed until the camera has rotated beyond having the object in the center of the image. An image is captured after every pan step. Image sequences that will further illustrate the process, using both image platforms can be found in Section 6.4.

The analysis stage consists of three steps: resizing, feature extraction, and matching. Figure 6.2 shows a flowchart displaying the interaction of the main modules of the analysis stage of the system. The resizing step scales the original image to a size that would be less computationally expensive to process (the processing size is a configurable parameter of the system). In variable resolution, the images are resized by calculating an appropriate scaling factor s according to the desired distortion factor λ and the processing image size (See FET in Chapter 4). In the case of uniform resolution, the images are scaled by subsampling. Figure 6.3 (top) shows the left and right images in uniform resolution after being scaled to a 256 by 256 size. Figure 6.4 (top) shows the left and right images in variable resolution after being scaled to a 256 by 256 size using a scaling factor s of 52.677 and a distortion factor λ of 0.15. The resizing step is the only one during the analysis that will handle images in uniform resolution and in variable resolution differently.

The feature extraction step is responsible of obtaining the edge map of the image being processed. As pointed out earlier, matching using the grey-level

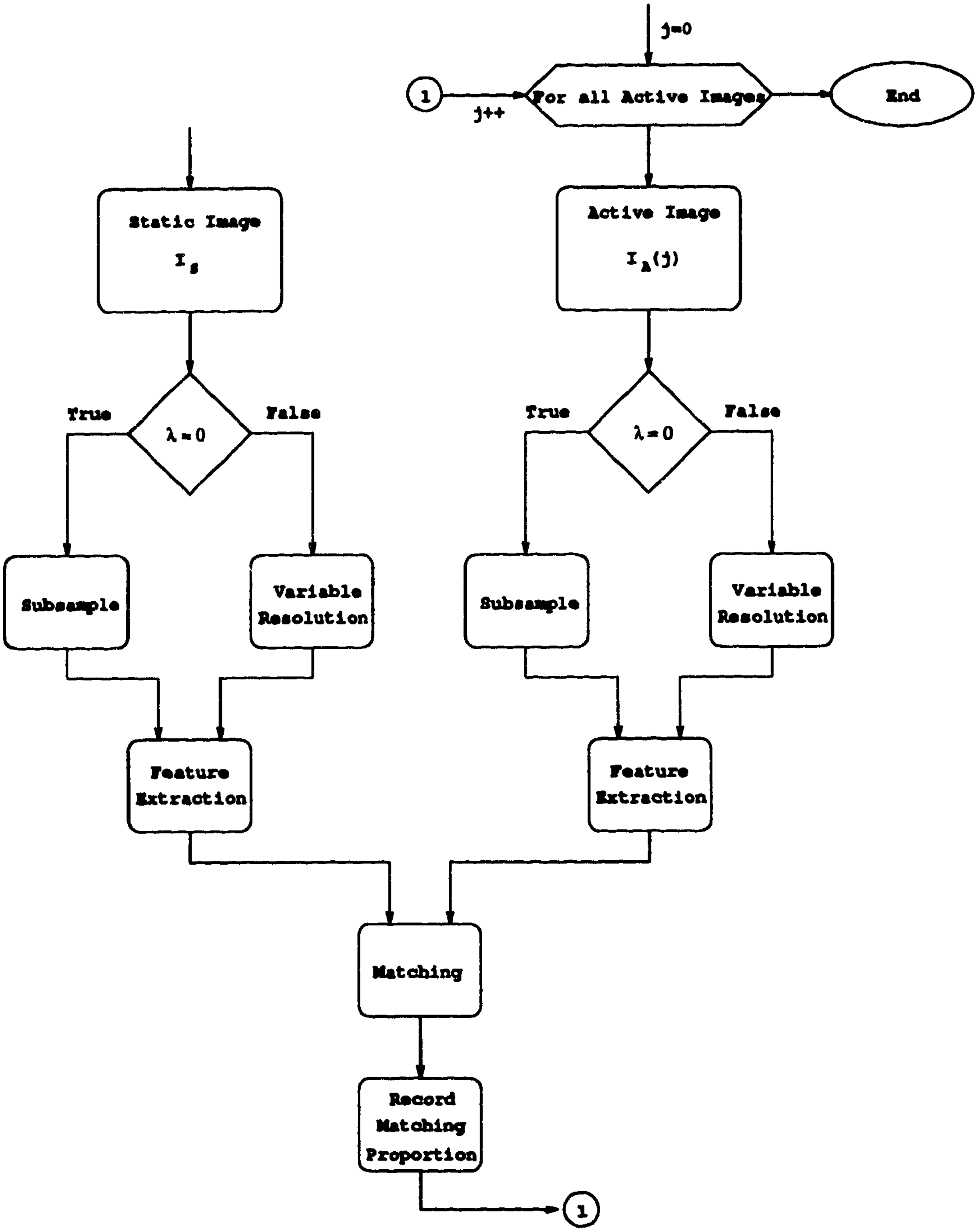
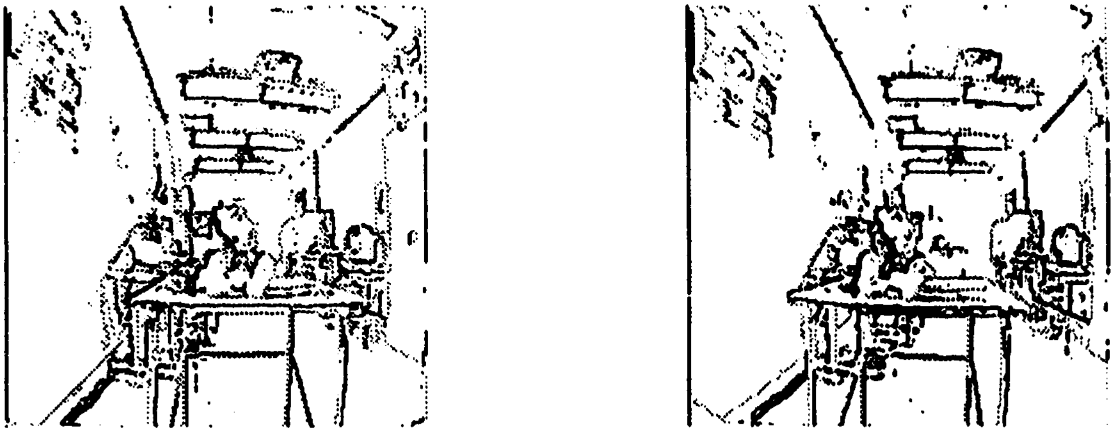


Figure 6.2: System Flowchart

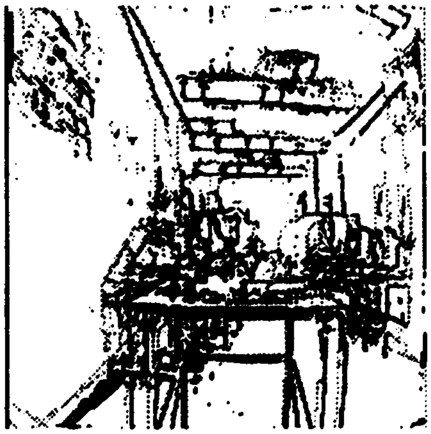
(a) Left Image

(b) Right Image



(c) Left Image Edge Map

(d) Right Image Edge Map

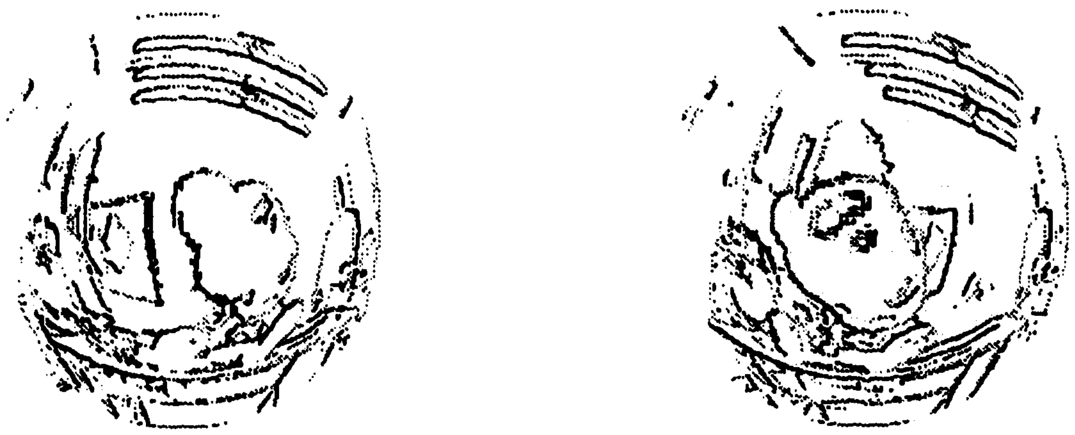


(e) Visual Representation of Matching

Figure 6.3: Uniform Resolution Process

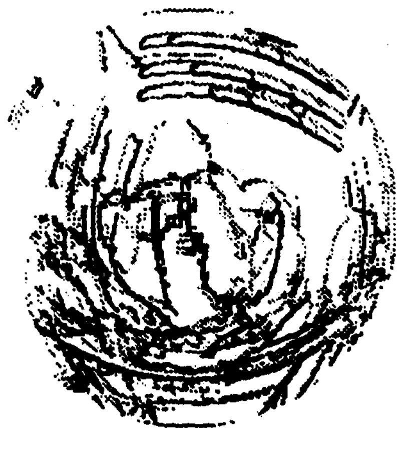
(a) Left Image

(b) Right Image



(c) Left Image Edge Map

(d) Right Image Edge Map



(e) Visual Representation of Matching

Figure 6.4: Variable Resolution Process

values has several shortcomings, thus feature matching is recommended. In the experiments presented in this chapter, Section 6.4, edge features are used in the matching process. Edges, as well as their gradient direction and magnitude, are obtained using the modified version of the Marr-Hildreth operator [23] described in Chapter 3 Section 3.1.4.1. The edges with low magnitude (weak edges) are eliminated using the adaptive thresholding technique described also in Chapter 3 but Section 3.2. The remaining edges are classified into 4 broad groups according to their gradient direction (see Table 3.1). A different grey-level value is assigned to each group to identify edges belonging to different groups. This classification helps in reducing the false alignments during the matching process, i.e., edges of one group in one image cannot be matched with edges of other groups in the other image. Figure 6.3 (middle) shows the left and right images after the feature extraction stage during a uniform resolution analysis. Figure 6.4 (middle) shows the left and right images after the feature extraction stage during a variable resolution analysis.

The matching process is performed iteratively, comparing the original image obtained from the static camera with each of the images captured during the panning process of the active camera. As described in Chapter 3 Section 3.3, the matching is evaluated using a very simple correlation function that takes into account a matching window and the group to which the edge belongs. Only edges of the same group within the tolerance window in both images are matched. This simple correlation function allows us to avoid, to a certain extent, false edge alignments. Figures 6.3 (bottom) and 6.4 (bottom) show a visual representation of the matching process in uniform resolution and variable resolution respectively. The total number of edge matches is divided by the total amount of features in the static image and then recorded for each correlation performed. The results are plotted to show the performance. Examples of the graphs generated by both methods are illustrated in the next section.

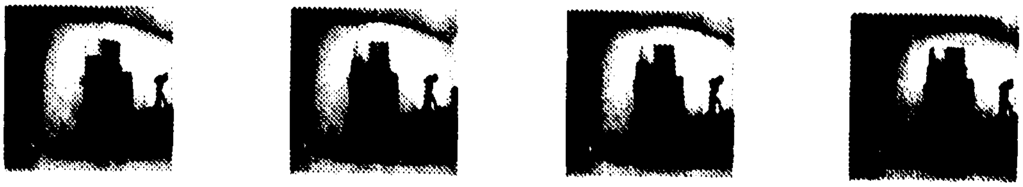
6.4 Experimental Results

We now present four experiments that will show the performance of our method in the two different environments. In the experiments presented through this section, two different approaches using the same source images are compared, our approach using the variable resolution (FET) scheme introduced in Chapter 4, and an analogous approach that uses a uniform resolution scheme. Along with the experiments, we present four subsections that will clearly describe the effect of modifying each key parameter individually in the system.

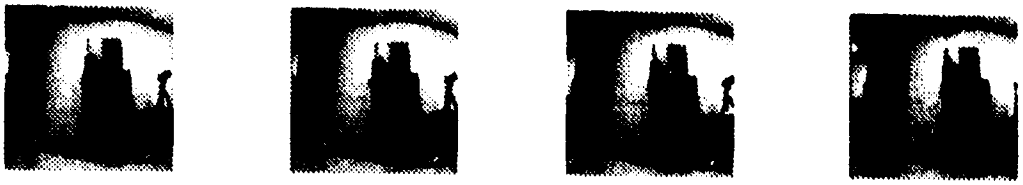
6.4.1 Experiment 1 (Real Environment)

In this first experiment presented we use images from the real camera system. A real scene was captured from the two viewing positions. For this experiment the filing cabinet in the left image (see Figure 6.5 (a)) is the object of interest. A sequence of 16 images was taken from the right viewing position (after slidding the camera 6.5 inches), each after performing one panning step of approximately 1.29° (see Figure 6.5 (b) to (h)), achieving in this way a total rotation of 19.40°. The analysis over these images was performed in both uniform resolution and variable resolution. The processing image size used was 128 × 128 pixels, obtained by subsampling for the uniform resolution analysis, and by using a distortion factor λ of 0.5 and a scaling factor s of 13.17, the thresholding value applied to the images during the analysis was 40 %. Figure 6.6 shows the matching performance of both experiments. Observe that the peak of the VR function is obtained for 10 rotation steps. This indeed appears to be the correct vergence for the filing cabinet (comparing Figures 6.5 (a) and 6.5 (e)).

Additional experiments were performed with the objective of showing the effect of changing the distortion factor λ and the thresholding value used for edge detection. The two following subsections describe how these changes affect the shape of the matching function in the system.



(a) Left Image (b) Right Image (c) 3.88 Deg Rotation (d) 7.76 Deg Rotation



(e) 10.35 Deg Rotation (f) 12.94 Deg Rotation (g) 15.52 Deg Rotation (h) 19.41 Deg Rotation

Figure 6.5: Image Sequence for Real Experiment 1

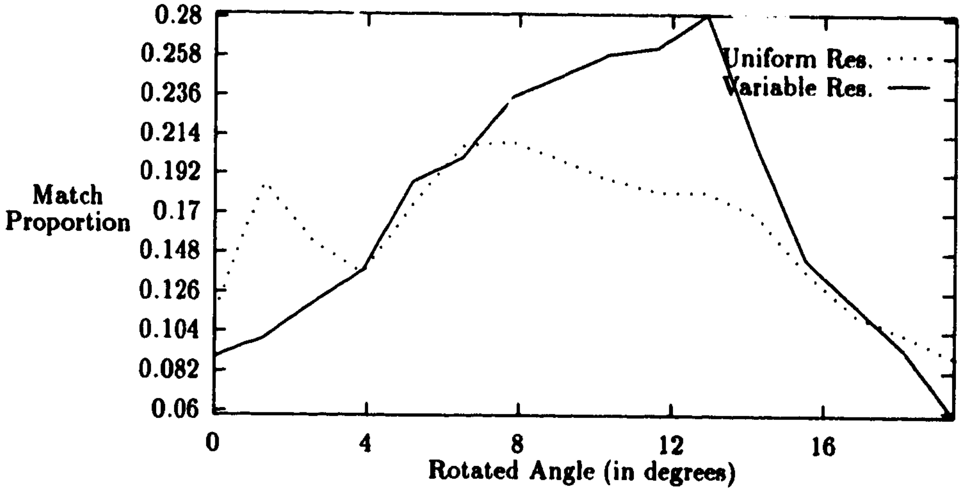


Figure 6.6: Matching Function for Real Experiment 1

6.4.1.1 Effect of Changing the Distortion Factor λ

In this section, we use the same image sequence of the experiment above (see Figure 6.5) to illustrate how changes in the distortion factor λ affect the shape of the matching function. We performed the analysis using the following distortion (λ) and scaling (s) factors,

λ	0.01	0.05	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
s	50.40	24.38	19.50	16.18	14.70	13.80	13.17	12.69	12.32	12.01	11.75

Table 6.1: Distortion and Scaling Factors

to produce images of 128 × 128 pixels.

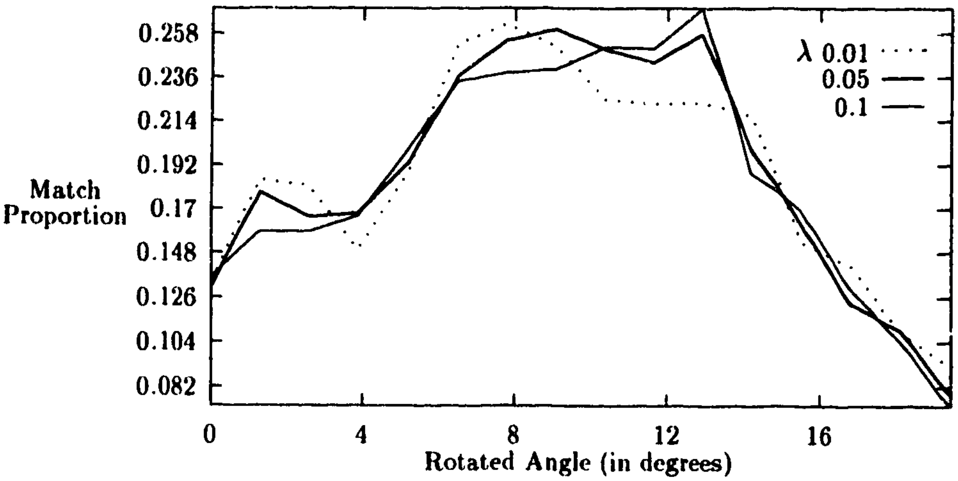


Figure 6.7: Effect of Changing the Distortion Factor (small values)

As can be observed from figures 6.7 and 6.8, the graph grows smoother as we increase the value of the distortion factor λ. With a very small distortion factor like 0.01 the graph shows several local maxima, making it difficult to search for the global maxima that will indicate the proper match. With a large λ like 0.5 or 0.7 the graph grows more evenly, making it much easier to search for the global maxima.

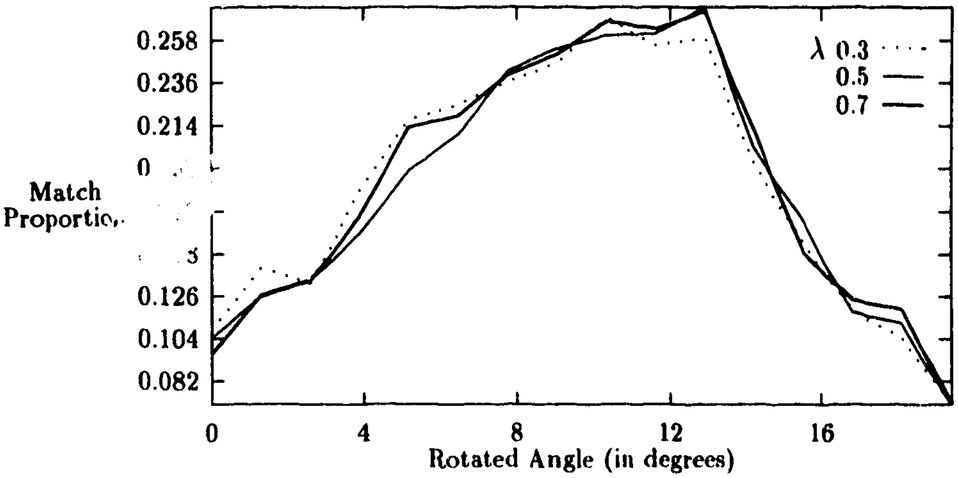


Figure 6.8: Effect of Changing the Distortion Factor (large values)

6.4.1.2 Effect of Changing the Thresholding Value

We now present how changes in the thresholding value affect the matching function. Using the source images shown in Figure 6.5 we performed the analysis in variable resolution (using $\lambda = 0.5$ and $s = 13.17$) and in uniform resolution, for different thresholding values (25%, 30%, 35%, 40%, 45%, and 50%).

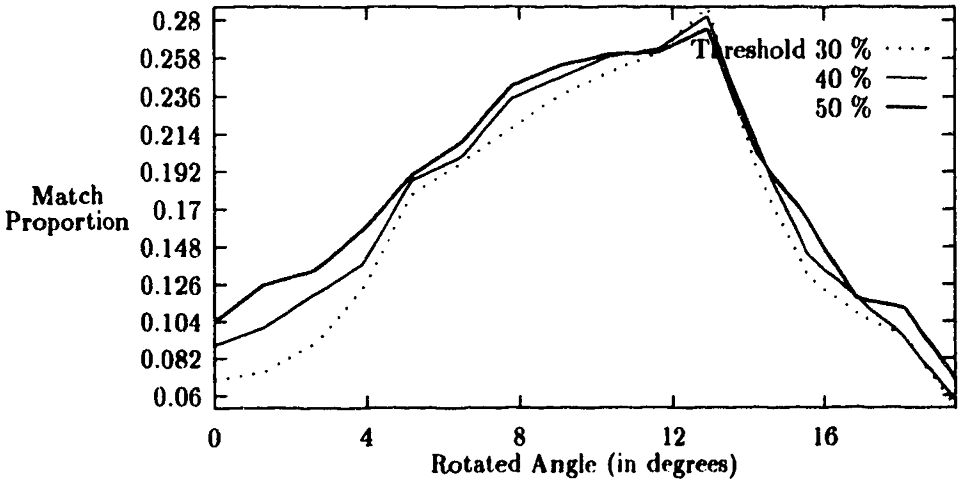


Figure 6.9: Effect of Changing the Thresholding Value Variable Resolution

Graph 6.9 shows the thresholding analysis using variable resolution, we can

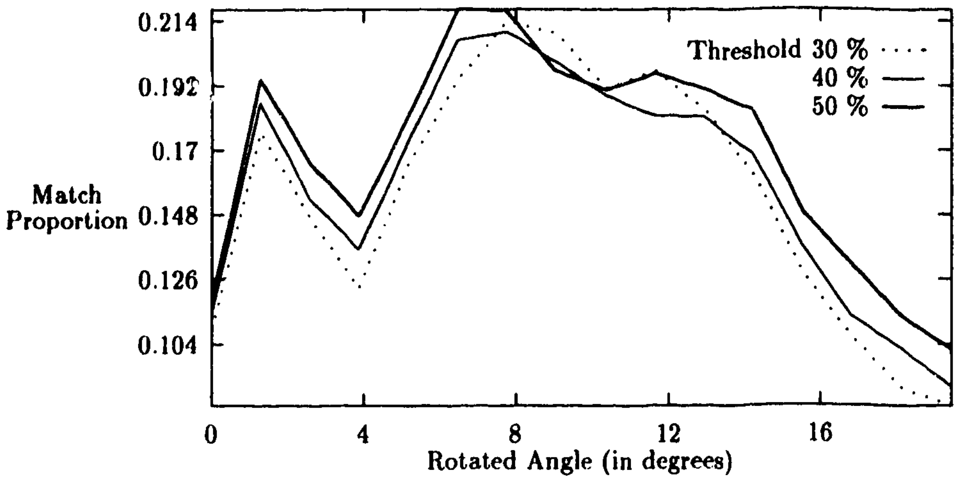


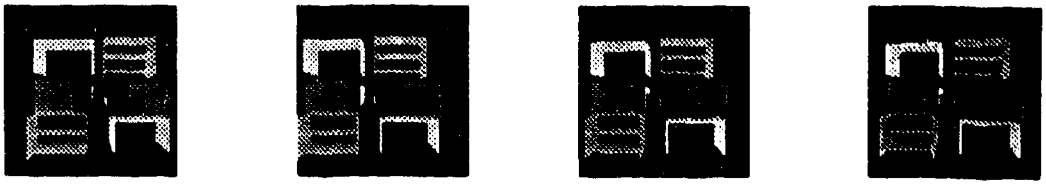
Figure 6.10: Effect of Changing the Thresholding Value Uniform Resolution

observe that the performance is very similar, with a slight smoothing effect caused by the reduction in the percentage of edges to be kept (from 50 % to 30 %). Even though the thresholding analysis in uniform resolution (see Figure 6.10) shows a similar smoothing effect, this is not enough to eliminate the erroneous peaks that appear at 1.5° and 6.5° degrees of rotation.

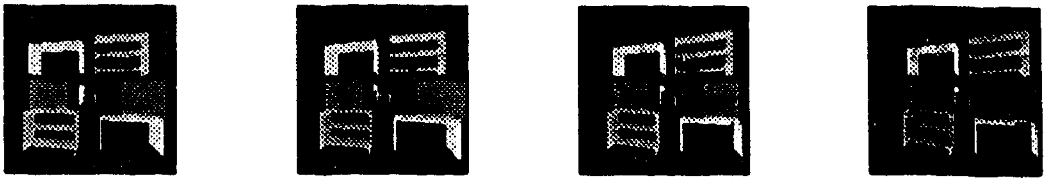
6.4.2 Experiment 2 (Graphics Environment)

The next experiment presented was conducted using the graphics platform. A scene with 7 objects, all at different depths, was created. For this experiment the chair in the center of the left image (see Figure 6.11 (a)) was chosen to be the object of attention. A sequence of 42 images was taken from the right viewing position, each after performing one panning step of 0.5° for a total rotation of 21° (see Figure 6.11 (b) to (h)). As before, the analysis over these images was performed in both uniform resolution and variable resolution (using a distortion factor of 0.5). Graph 6.13 shows the matching function for the two methods.

This experiment demonstrates the usefulness of the VR scheme for clut-



(a) Left Image (b) Right Image (c) 2 Deg Rotation (d) 4.5 Deg Rotation



(e) 7 Deg Rotation (f) 9.5 Deg Rotation (g) 12 Deg Rotation (h) 19.5 Deg Rotation

Figure 6.11: Image Sequence in Uniform Resolution for Artificial Experiment

(a) Left Image (b) Right Image (c) 2 Deg Rotation (d) 4.5 Deg Rotation

(e) 7 Deg Rotation (f) 9.5 Deg Rotation (g) 12 Deg Rotation (h) 19.5 Deg Rotation

Figure 6.12: Image Sequence in Variable Resolution for Artificial Experiment

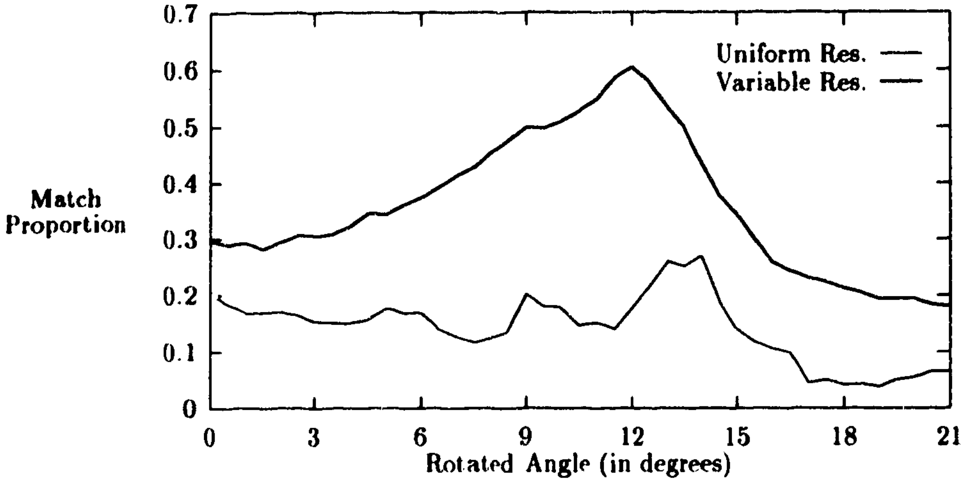


Figure 6.13: Matching Function for Artificial Experiment

tered environments. In uniform resolution the fovea is not emphasized, which creates many erroneous peaks in the matching function. The VR function increases to a unique peak and then decreases. Taking advantage of the flexibility that the graphics platform provides, we conducted an experiment that will show the changes in the matching function for an object at different depths. The results of such experiment are presented in the following subsection.

6.4.2.1 Matching Function for Different Depths

Three scenes with a single object (bookshelf) were created using the graphics environment, with the only difference of having the object at different depths (1 m, 1.1 m, and 1.4 m). The separation between the two viewing positions was set to 22 cm, and a sequence of 42 images was taken from the right viewing position (21° of total rotation). Figures 6.14 and 6.15 show the results of the analysis of the three scenes in uniform and variable resolution respectively. As expected, the required rotation to obtain the appropriate match for objects that are far away is less than that required for objects that are closer. We can observe this phenomenon in Figures 6.14 and 6.15.

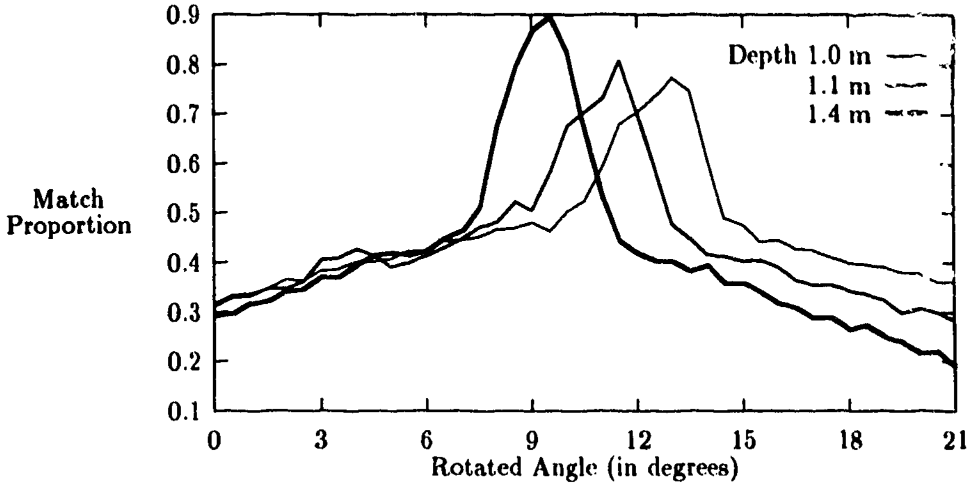


Figure 6.14: Matching Function in Uniform Resolution

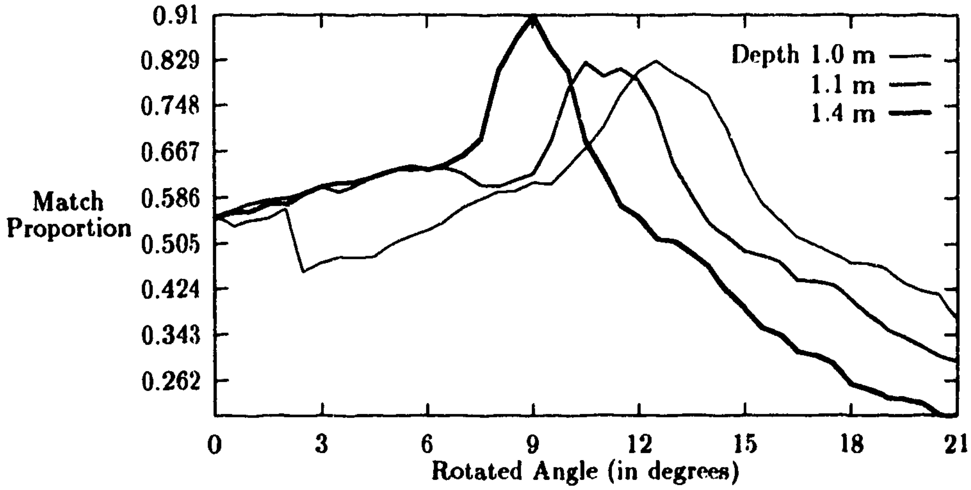


Figure 6.15: Matching Function in Variable Resolution

6.4.3 Experiment 3 (Real Environment)

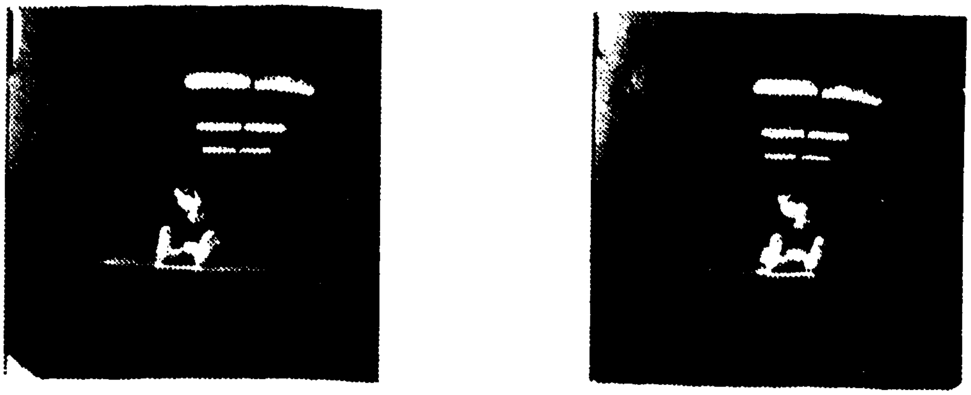
Images captured with the real camera system were used in this experiment. Two viewing positions separated by a baseline of 6.5 inches were used to obtain the left and right perspective views of the scene. In this case, the toy in the center of the right image (top left Figure 6.16) is the object of attention of the analysis. A sequence of 24 images were captured from the left viewing position, each after one panning step of approximately 0.59° for a total rotation of 13.58°. Figure 6.16 shows some of the images of the sequence, the complete sequence is not presented to save space. The analysis was performed in variable ($\lambda = 0.5$, $s = 13.17$) and uniform resolution, using a processing image size of 128 × 128 pixels.

Figure 6.17 shows the matching function for the real experiment 2. Observe that the VR graph has a much more prominent peak, compared to the uniform resolution graph, for the correct vergence angle. The peak near 0° rotation is caused by the peripheral objects, all of which are far away, therefore almost no rotation is required for them to match.

6.4.3.1 Multiple Thresholding

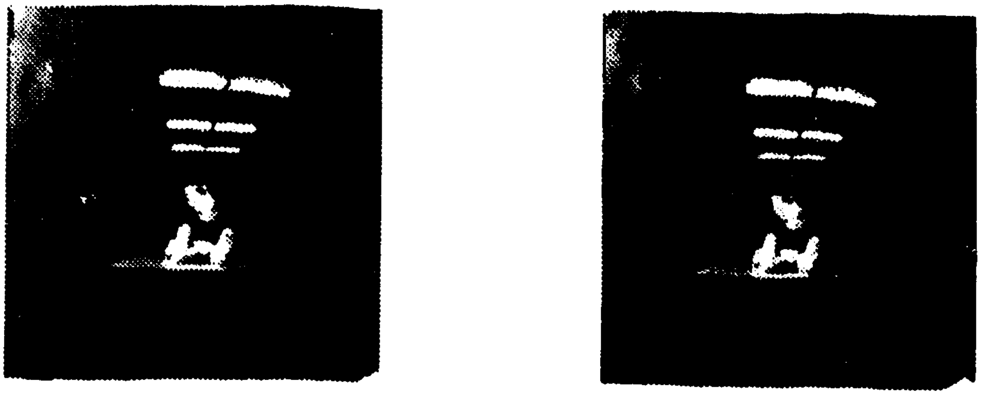
When most of the peripheral objects are located very far from the object of interest, i.e., strong depth discontinuities between the object of attention and the peripheral objects exist within the scene, the matching function describes two peaks, as shown in Figure 6.17 — one due to the objects in the periphery and the other due to the object of attention in the fovea. The variable resolution approach to vergence control, by itself, cannot avoid this type of matching problem, so additional processing needs to be performed.

To compensate and improve the performance of our method under the circumstances described above, a modification of the thresholding step in our methodology was deviced. This modification considers the usage of different threshold percentages for the periphery and for the fovea, trying in this way to highlight the importance of the features corresponding to the object of attention while reducing it for those corresponding to the objects in the periphery.



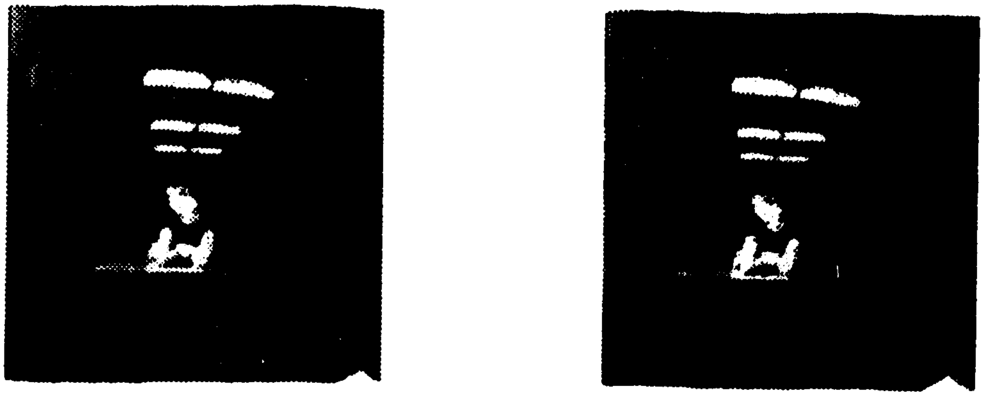
(a) Right Image

(b) Left Image



(c) Left Image after 1.72 deg rotation

(d) Left Image after 4.59 deg rotation



(e) Left Image after 7.45 deg rotation

(f) Left Image after 10.32 deg rotation

Figure 6.16: Image Sequence for Real Experiment 2

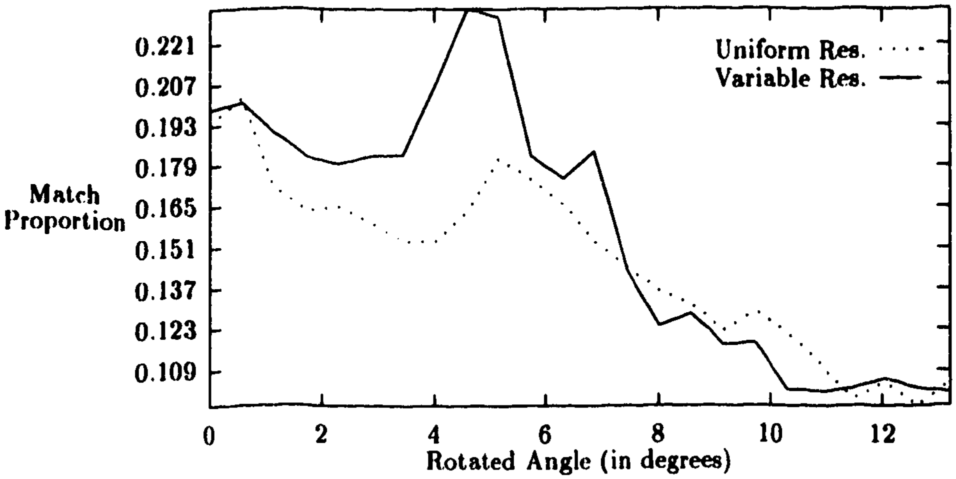


Figure 6.17: Matching Function for Real Experiment 2

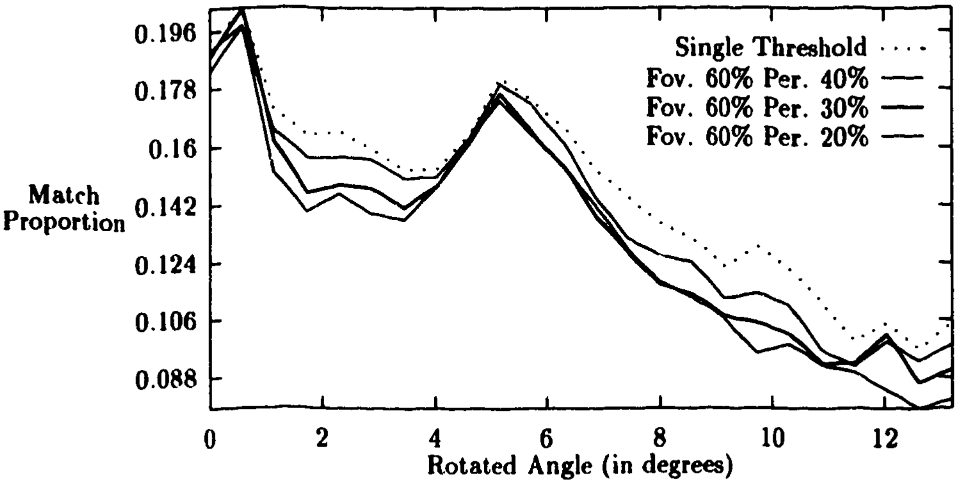


Figure 6.18: Multiple Thresholding Matching Function using Uniform Resolution

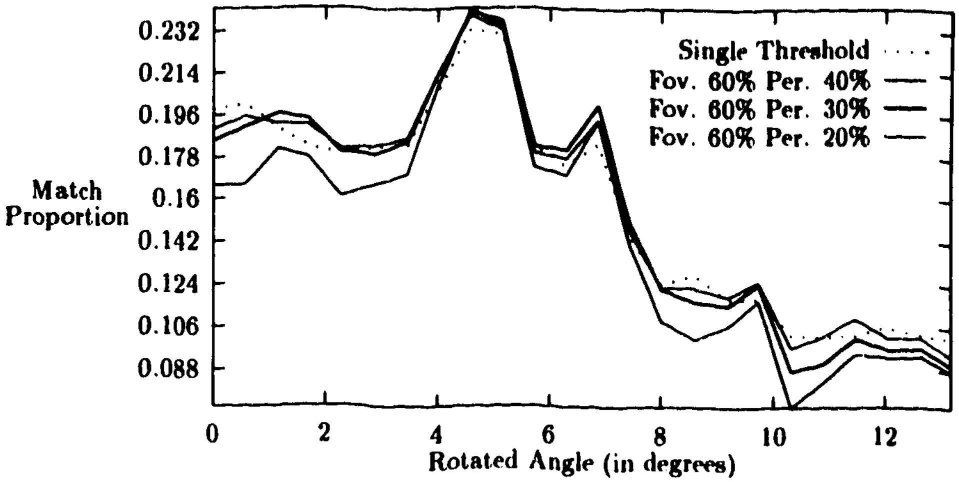


Figure 6.19: Multiple Thresholding Matching Function using Variable Resolution

In the case of the variable resolution analysis the boundary between the foveal and peripheral regions is defined by the FET mapping function. When the radius of the original image turns larger than the radius of the variable resolution image, the shrinking process starts, together with the periphery. That is, when a pixel p(x,y) in the uniform resolution (UR) image gets mapped into a pixel q(x^*, y^*) in the variable resolution (VR) image, and the distance from p(x,y) to the center of the UR image is larger than the distance from q(x^*, y^*) to the center of the VR image the mapping a shrinking process is taking place, this indicates that this pixels belong to the periphery of the VR image. In the uniform resolution analysis, the modification was also implemented to provide a comparison measure. Although in uniform resolution there is no foveal or peripheral regions, the foveal threshold was applied to one third of the total area in the central region of the image, and the peripheral threshold was applied to the rest. Figure 6.18 shows a comparison between the uniform-resolution matching functions. As can be observed from the multiple threshold analysis, there is no sensible improvement in the shape of the matching function. Figure 6.19 shows the analogous comparison between the variable-resolution matching functions. In this case, the multiple thresholding technique helped in reducing the height of the peak due to peripheral matches while slightly enhancing the peak due to matches in the fovea.

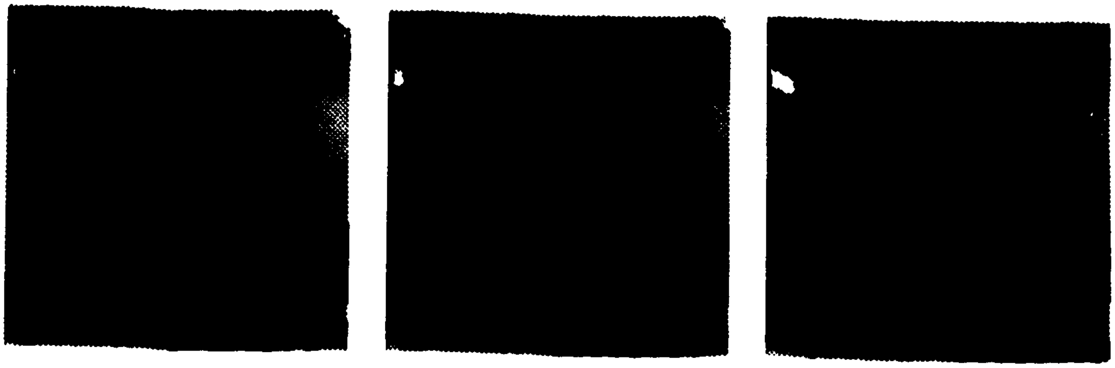
6.4.4 Experiment 4 (Real Environment)

The final experiment was conducted using images captured by the real camera environment. For this experiment the cup in the center of the left (static) image (see Figure 6.20 (a)) is the object of attention. A sequence of 21 images was taken from the right viewing position, each after performing one panning step of approximately 0.99° for a total rotation of 19.45° (see Figure 6.20 (b) to (f)). Figure 6.21 illustrates the matching performance of the experiments conducted in uniform resolution (dashed line) and variable resolution (solid line). The distortion factor used to generate the variable resolution images was 0.5.

As reviewed in Section 6.4.3.1 the matching function shows two peaks due to strong depth discontinuities between the object of attention and the background. The peak obtained after 15° of rotation is due to the proper match of the object in the fovea. It is understandable that the peak is not as prominent as the one shown in Figure 6.17, because as can be noted in Figures 6.20 (a) and (e) the background around the cup has changed from black in Figure 6.20 (a) to almost white in Figure 6.20 (e). This change in the background surrounding the cup changed drastically the strength of the edges due to the contour of the cup, therefore making them very susceptible to be removed during the adaptive thresholding stage. Changes in the direction of the edges due to radically different backgrounds were observed in other experiments. The uniform resolution matching function, barely shows the peak resulting from matching the object of attention. This is because of the lack of feature information about the contour of it, as well as the strong depth discontinuities.

A multiple thresholding analysis was computed, using threshold values of 10%, 20%, and 30% for the periphery and 50% for the fovea, to try to improve the shape of the matching functions. Figure 6.22 shows the matching functions for

(a) Left Image (b) Right Image (c) 4.96 Degree Rotation



(d) 9.93 Degree Rotation (e) 14.90 Degree Rotation (f) 19.86 Degree Rotation

Figure 6.20: Image Sequence for Real Experiment 3

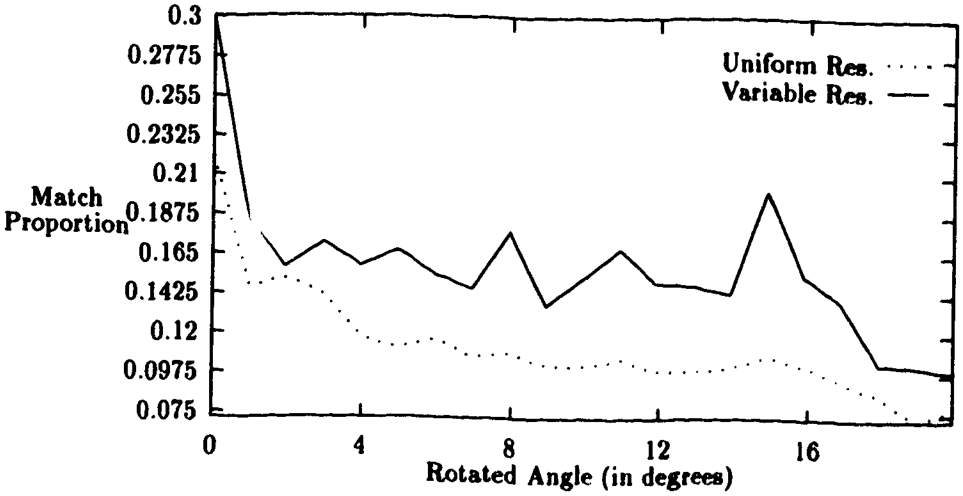


Figure 6.21: Matching Function for Real Experiment 3

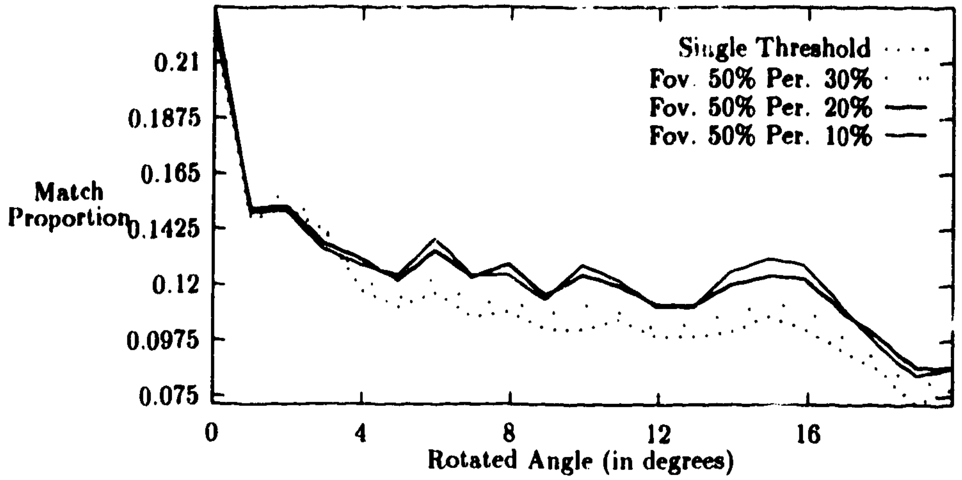


Figure 6.22: Multiple Thresholding using Uniform Resolution Real Experiment 3

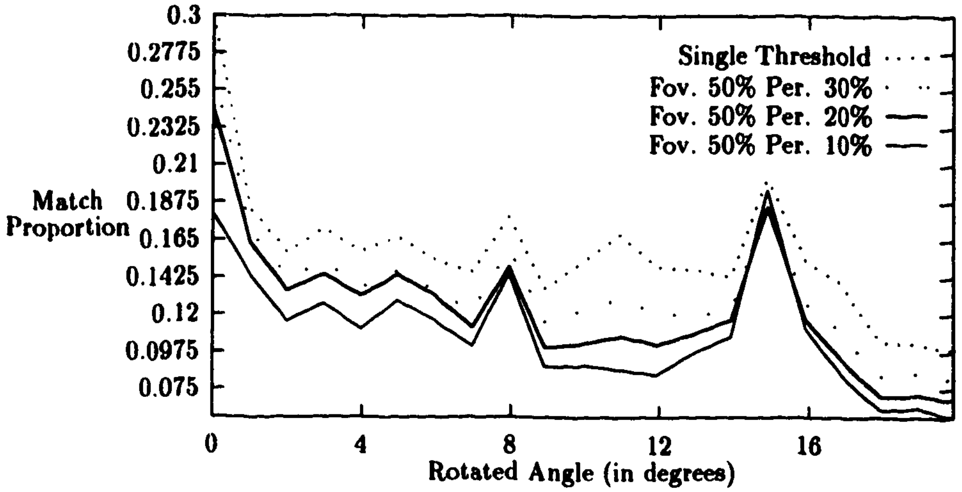


Figure 6.23: Multiple Thresholding using Variable Resolution Real Experiment 3

the multiple thresholding analysis using uniform resolution. Although, a slight improvement can be observed in the peak due to the match of the object of attention, no change can be noted for the peak at 0° rotation. Figure 6.23 shows the analogous analysis using variable resolution. In this case, a reduction in the height of the peak at 0° rotation can be identified; unfortunately, there is also a slight reduction for the peak at 15° of rotation.

Figures 6.17, 6.13, 6.6, and 6.21 demonstrate the validity of the theoretical analysis in Chapter 5. Variable resolution clearly assists the vergence process by creating a smoother matching function with a clear and high peak. This peak can be used to derive the angle between a stereo pair of cameras, so that both "look at" the same object. The vergence angle determines the depth of the object in the fovea (see Section 2.2).

Chapter 7

Conclusions and Future Research

In this thesis we have presented a method for controlling vergence movements using a variable-resolution representation of the input image. The method relies on a very simple scheme for matching edges of the left and right views, combined with constraints for eliminating false edge alignments. The power of this method is derived solely from the variable-resolution approach, and it could be further enhanced by using better edge matching schemes. The method described is most useful for solving problems in stereo vergence, although it could be used for determining disparity for any object in a pair of left and right images. The advantages of the variable resolution scheme are lost in scenes containing small objects with large depth discontinuities. It should be pointed out, however, that it is these situations that also create problems for the human visual system. To cope with such situations, multiple thresholding techniques applied to the fovea and the periphery were briefly presented. A variable distortion factor λ that grows with the vergence angle, or a multiresolution scheme may have to be used.

We also presented two methods for modeling and calibration of fish-eye lenses. The first model (FET) is simple and efficient, however the second (PFET) model fits real fish-eye lenses more accurately. Experimental results demonstrating the validity of the two models were also presented.

Simplified vergence control is not the only advantage of variable-resolution vision. Recently it has been shown that the complexity of character thinning and boundary following [21, 4] problems can be greatly reduced by using templates with a high-resolution center and a low-resolution periphery. At the same time, the performance of such algorithms has been demonstrated to be superior compared to uniform resolution methods. In general, variable-resolution schemes can reduce the complexity of many perceptual tasks. This is achieved at the cost of restricting detailed vision to a small part of the visual input. In order to obtain elaborate and detailed descriptions of a whole scene, multiple variable-resolution images must be integrated across fixations, and thus control of head and eye movements becomes a crucial part of the perceptual process.

7.1 Future Research

At present, the methodology has been implemented over a real image environment which has only one camera, that can be shifted about the x axis to simulate the different perspectives obtained with a stereo system. A complete stereo system, including an accurate way of estimating the panning angle of the cameras, needs to be acquired in order to improve the reliability and performance of the method. The stereo system could be used as well for performing accuracy of depth estimation tests and the results compared with those of other vergence control methods.

Although a real-time implementation (strictly speaking) using the methodology presented, is difficult to conceive, the use of a dedicated real-time platform such as the Maxvideo20 pipeline image-processor, will certainly speed up the process considerably.

It was found that edge direction and strength features are very sensitive to background changes caused by the camera rotation. As mentioned in Section 3.3, the use of a better matching scheme, utilizing constraints like figural continuity, will help in solving this problem as well as improve the still existent ambiguity. The use of a variable distortion factor λ that grows with the vergence angle, to improve the performance of the methodology for scenes with strong depth discontinuities, could be studied. In addition, it would be interesting to investigate the repercussions of changes in the distance between the optical centers of the cameras (baseline), and how errors in estimating this parameter would affect t accuracy of depth estimation.

Fish-eye lenses whose distortion has not been optically compensated could be used to obtain the variable-resolution images, reducing in this way the computational cost of the algorithm. Analysis of the performance of our methodology using such lenses will have to be evaluated.

In the long term this methodology could be integrated with tracking technology to produce a more complete system. In such environment, vergence control will not only provide additional information to the tracking process making it more accurate, but working together with it, will make possible to keep continuous depth information about an object in the scene even if it moves. This is particularly important for performing higher level tasks like Hand-eye coordination, navigation, or obstacle avoidance.

Bibliography

[1] J. Y. Aloimonos and A. Bandyopadhyay. Active vision. In *Proceedings of the IEEE 1st International Conference on Computer Vision*, pages 35–54, London, December 1987.

[2] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, pages 107–131, 1987.

[3] D. A. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.

[4] A. Basu, M. Jain, and X. Li. Variable resolution techniques for character thinning and boundary detection. Technical report, University of Alberta, Computing Science Department, 1992.

[5] A. Basu and S. Licardie. Variable resolution vergence control. Technical report, University of Alberta, Computing Science Department, 1992.

[6] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.

[7] C. M. Brown. Gaze controls with interactions and delays. *IEEE Transactions on SMC*, 20(1), 1990.

[8] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[9] R. Chellappa and A. Rosenfeld. Computer vision: Attitudes, barriers, counseling. In *Proceedings of Vision Interface '92*, pages 1–7, University of Maryland, Maryland, May 1992.

[10] J. Clark and N. Ferrier. Modal control of visual attention. In *Proceedings of the International Conference on Computer Vision*, pages 514–531, Tarpon Springs, Florida, December 1988.

[11] S. D. Cochran and G. Medioni. Accurate surface description from binocular stereo. In *Proceedings of the International Workshop in Image Understanding*, pages 857–869, Palo Alto, California, USA, May 1989.

[12] L.S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3):248–270, 1976.

[13] U. R. Dhond and J. K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, November 1989

[14] C. F. Gerald and P. O. Wheatley. *Applied Numerical Analysis*. Addison-Wesley, third edition, 1985.

[15] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[16] W. E. L. Grimson. A computer implementation of a theory of human stereo vision. *Phil. Transactions of the Royal Society of London*, B292:217–253, 1981.

[17] W. E. L. Grimson and E. C. Hildreth. Comments on digital step edges from zero-crossings of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):121–126, January 1985.

[18] W. Hoff and N. Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136, February 1989.

[19] R. Jain, S. L. Bartlett, and N. O'Brien. Motion stereo using ego-motion complex logarithmic mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):356–369, May 1987

[20] S. Levialdi. Edge extraction techniques. *INRIA-CREST Course on Computer Vision*, 1992.

[21] X. Li. and A. Basu. Variable resolution character thinning. *Pattern Recognition Letters*, 1991.

[22] D. Marr. *Vision*. W. H. Freeman Company, 1982.

[23] D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond.*, pages 187–217, 1980.

[24] L. Massone, G. Sandini, and V. Tagliasco. "form-invariant" topological mapping strategy for 2d shape recognition. *Computer Vision, Graphics and Image Processing*, 30:169–188, 1985.

[25] J. E. W. Mayhew and J. P. Frisby. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17:349–385, 1981.

[26] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing*, 31:2–18, 1985.

[27] T. J. Olson and D. J. Coombs. Real-time vergence control for binocular robots. *International Journal of Computer Vision*, pages 881–888, 1991.

[28] T. Peli and D. Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20:1–21, 1982.

[29] J. S. Pointer. The cortical magnification factor and photopic vision. *Biological Reviews of the Cambridge Philosophical Society*, 61:97 119, 1986.

[30] D. Robinson. Why visumotor systems don't like negative feedback and how they avoid it. *Vision, Brain and Cooperative Computation*, 1987.

[31] J. Rovamo and V. Virsu. Visual resolution, contrast sensitivity, and the cortical magnification factor. *Experimental Brain Research*, 37:475 494, 1979.

[32] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14:365-372, 1980.

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

[34] R. Sekuler and R. Blake. *Perception*. McGraw-Hill, second edition, 1990.

[35] M. Tistarelli and G. Sandini. Dynamic aspects in active vision. *Computer Vision, Graphics and Image Processing : Image Understanding*, 56(1):108 129, July 1992.

[36] S. A. Teukolsky, W. H. Press, B. P. Flannery and W. T. Vetterling. *Numerical Recipes in Pascal*. Cambridge University Press, 1989.

[37] C. F. R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *Computer Graphics and Image Processing*, 11:197-226, 1979.

[38] J. Weng, N. Ahuja, and T. S. Huang. Matching two perspective views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):806-825, August 1992.

[39] Y. Yeshurun and E. L. Schwartz. Cepstral filtering on a columnar image architecture: A fast algorithm for binocular stereo segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):759-767, July 1989.

[40] Y. Yeshurun and E. L. Schwartz. Shape descriptions with a space-variant sensor: Algorithms for scan-path, fusion and convergence over multiple scans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1217-1222, November 1989.