Algorithms for Low Cost VLSI Stereo Vision Systems, with Special Application to Intruder Detection

Kevin William John Findlay

Submitted for the degree of Doctor of Philosophy

Department of Electrical Engineering
University of Edinburgh

January, 1993

EX BIBL. UNIV. EDINBURGH

Abstract

The work described in this thesis is concerned with the development of hardware efficient, image processing and machine vision algorithms for implementation, using recently developed low cost CMOS cameras. These allow the integration of processing on the same silicon substrate as the imaging sensor. The general approach differs from other image processing research in that algorithms are being developed for a target architecture, rather than hardware being developed for a particular image processing function. A particular application, namely intruder detection and tracking, has been chosen, to demonstrate this approach.

The use of image processing in alarm systems has many advantages over active electronics: the main ones being installation costs and reliability. In particular, stereo vision has the potential of providing an invisible wall and estimates of intruder significance. However it is also desirable that alarm systems have wide angle lenses. Wide angle lenses create particular problems for stereo vision, in relation to pixel quantisation. Techniques to provide a low cost sub-pixel estimate of disparity are presented. Further, an original stereo matching algorithm is described which solves the stereo correspondence problem, in a computationally simple manner. Adaptations are also made to the low level segmentation stages which would allow an efficient implementation using CMOS sensors and processing. Other savings have been made by eliminating digital floating point calculations, multiplications and divisions at the lower levels of processing. Also, due to the reduced data rates required for global frame to frame computation, higher level, calculations can be performed on an associated microprocessor. Thus, a Kalman tracking filter has been applied to integrate the three possible disparities from three cameras, with experimentally calculated error covariance matrices. A results chapter describes the extraction of these matrices, together with simulations of the algorithms applied to twelve different sequences. These show that the system could be effective as an alarm system. Also described, at various stages in the thesis, are possible hardware implementations of the algorithms and partitions between analogue and digital circuitry. The thesis finishes with some general conclusions.

Declaration

I declare that this thesis has been completed by myself and that, except where indicated to the contrary, the research documented in this thesis is entirely my own.

Kevin W.J. Findlay

Acknowledgements

I would like to thank my supervisor Dr. David Renshaw and second supervisor, Professor Peter B. Denyer, for their considerable, and valued, advice and financial support throughout the last three years. Thanks are also due to members of the ISG and VVL Ltd. for their help and friendship. Finally, thanks must be given to my father, for proof reading the text, and my mother, Rachel and Elspeth for their support over the years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aims and Objectives

The purpose of the research reported in this thesis is to demonstrate methods and techniques suitable for the design of commercial stereo vision systems. An original algorithm will be presented which would allow a low cost implementation of a stereo vision application using CMOS sensors. The algorithm takes advantage of the fact that a known application is being considered; an intruder alarm. Thus, it is not an aim to imitate the human vision system and an "engineering approach" has been taken. In this respect, a theme of the algorithms described, will be reductions in the required processing power for a final implementation. Thus, implementation using CMOS sensors, with on-board processing, would be a feasible option. Another feature is the adaptation of existing image processing techniques to this application. In many research projects, the different machine vision functions are implemented as "black boxes". However, in this system the segmentation stages, of the algorithm, have been developed to produce only those edges which matching requires. In effect, the *correspondence problem*[1] has been treated as one of segmentation. As a result, computational improvements have ensued in both segmentation and stereo matching.

1.2 Background

The computer vision and image processing areas of research have developed, over the years, from projects directed at both specific problems and general machine understanding. There has been considerable interplay between the two subject areas. As a rough rule, image processing is normally thought of as the study of lower level operations including compression, edge detection and thresholding, whereas vision research, emanating from the artificial intelligence community, has tended

[1] This is described later in this chapter.

to concentrate on the general problems of making machines see. This usually involves consideration of higher level data structures compared to those normally associated with image processing. Such higher level processes are currently less well understood and on a poorer theoretical basis than the initial stages of vision.

The different subject areas of the entire vision problem tend to obscure the fact that the divisions are very rarely clear cut. For example, movement can be useful in an object's recognition, as well as the more obvious parameters such as shape and colour. The interdependence of different research areas is particularly true in the case of 3D depth perception. The major problem in stereo vision is *correspondence*; finding features in one view of a scene and matching them to those of another view of the same scene. Clearly, the matching procedure will be dependent, to some extent, on how the initial features are extracted. In terms of recognition using 3D information, errors are likely to be dependent to some extent on the input information from the matching algorithm. In these terms, stereo vision could be classified as an intermediate process between edge detection and object modelling. It is neither at the bottom nor at the top of any hierarchy.

1.2.1 The Imaging Hierarchy and Correspondence

General machine vision can be viewed as the pyramid shown in Figure 1–1. As one proceeds upwards, towards the pinnacle, each layer employs larger data objects and more heuristic algorithms will be applied. Several vision architectures and data structures, based on this principle, have been proposed. For example, Marr [51], citing biological evidence, suggests segmentation algorithms using multiple spatial channels of different bandwidth. Processing starts with information from the smoother, low frequency, channels, which is then used to constrain the results from the higher frequency channels. A different approach is suggested by Burt [13] where the lower levels of the pyramid are used to survey the complete scene for regions of interest. Higher processes in the algorithm can then "home in" on the appropriate areas. Pyramidal organisation can also be found in hardware architectures for image processing. Often processing elements are arranged in a master-slave arrangement with the master distributing tasks and processing the results on a global level. Tregidgo et. al. [82] and Howlett [38] provide examples of this type of architecture.

It is not the purpose of this thesis to describe generalised processing algorithms and architectures as the ones just suggested. However, hierarchy can usually be applied to a specific vision application, where data is transformed from low level pixel primitives to higher level object features. Application specific decisions can then be made on these objects at a global level. The above discussion would suggest that such hierarchical structures would result in a reduction in computational requirements, at higher levels in the pyramid. This is not always the case. Indeed if one is trying to construct groups of features based on the strength of their connections, then solutions often become impossible within the resources available. An example would be recognition based on a search for maximal cliques in a general graph [4]. Graph matching, in this manner, is an NP-complete problem [25] and such a search is likely to lead to a combinatorial explosion. Recognition algorithms usually have to restrict the possible search space by employing additional constraints. Similar space constraints exist when calculating generalised Hough transforms for non-standard shapes. Often the Hough solution will use an infeasible number of variables, and therefore dimensions, to parameterise a shape.

Problems arise, with the above and other techniques, when the object under study is not rigid. Its shape will vary between frames and a stationary matching model cannot be used. Hogg [33] describes techniques for modelling non-rigid objects, such as humans, using ellipses and posture as matching parameters. This has had some success but has not been extensively tested in a wide range of situations. In view of the computation involved it is unlikely that such a system would be efficient in a current sensor implementation. Further, for the applications being considered in this thesis, ie. intruder detection and tracking, it seems that height and size would be more useful parameters. Thus, for this application, there is no requirement for a complete body model. Such non-rigid model matching is a current area of research and may be feasible in the future, given improved commercial processing abilities.

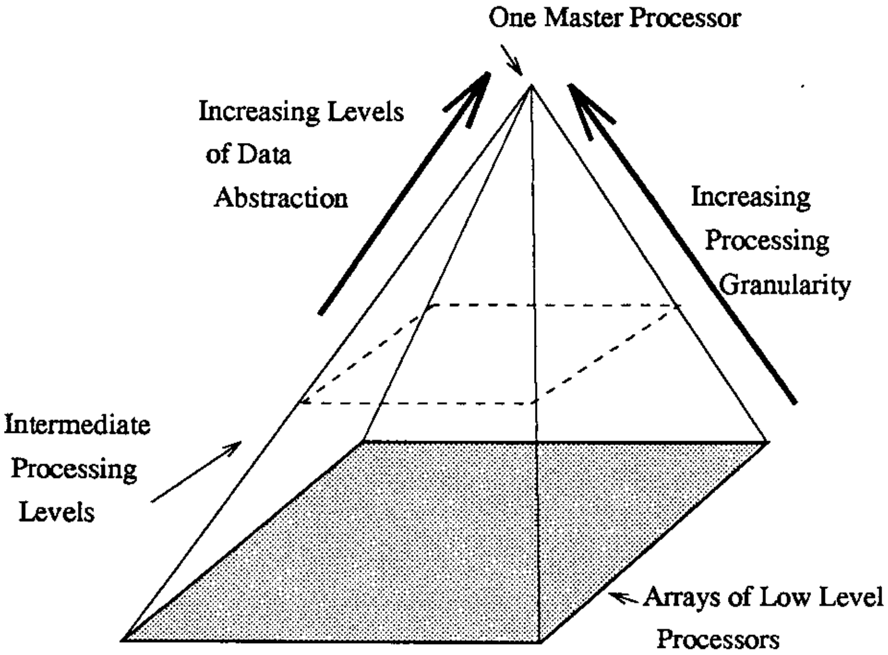

One Master Processor
Increasing Levels of Data Abstraction
Increasing Processing Granularity
Intermediate Processing Levels
Arrays of Low Level Processors

Figure 1–1: The Imaging Pyramid

In this work, computation is reduced by the development of algorithms which suit a particular application. We chose intruder detection and tracking as an area where,image processing techniques could improve on existing systems. Depth and disparity information would be extremely useful in detecting genuine intruders and reducing the number of false alarms. We thus need to study techniques which can extract depth information from a scene.

Vision algorithms, which attempt to extract 3D information, can be divided into two groups; active and passive. In the first category, light of known source is projected into the scene and the resultant images recorded and analysed using triangulation. Different combinations of camera can be deployed. For example, it is possible to extract light using a single camera with two incoherent light sources switching on and off alternatively. As an alternative, a single camera with some form of patterned light source can also be used to project structure onto the scene. Jarvis [41] provides a survey of techniques which have been suggested in the research literature. Such range finding techniques are generally considered more reliable and accurate than passive stereo but suffer from several problems in the type of application considered here. Firstly, normal background light may interfere with the projected light causing distortions and mismatches. It is desired that the system function in normal daylight as well as artificial light. Thus in active vision systems, the projection of patterned beams of light over large scenes during daylight may be difficult due to light dispersion. The second problem, for large scenes, is the difficulty in projecting beams of light onto surfaces of unknown reflectivity. For example, a white shirt may reflect a beam which a black jacket would absorb. The final reason for rejecting active vision systems is that the correspondence problem remains. Features from one image, no matter how it is lit, will have to be matched with those from another image. Thus the basic principles for stereo matching remain for both active and passive systems. The passive stereo algorithms described in this thesis could be extended to use active light.

Stereo matching and disparity estimation is the main area where computation has been reduced. The stereo correspondence problem is one of a number of similar problems in general machine vision. It is closely allied to time matching, where attempts are made to track features and objects through scenes. The problem to be solved, in both tasks, is that of finding the same features in a number of different images. Before the correspondence problem is solved, a decision must be made as to what type of feature to match. The most obvious technique is that of correlation between areas of the two images. However, correlation has serious problems when used in this manner. When there is little luminosity gradient it becomes increasingly difficult to differentiate between adjacent patches. Areas of constant texture also cause problems together with the obvious fact that different views of the same scene *will* be different. These constraints together with the probable expense in correlation computation have resulted in stereo matching algorithms based on features, for example edges. These do not occur in isolation and are usually part of a larger object which can be utilised to provide further constraints on matching. For this and the reasons described above, an edge based stereo algorithm has been developed in this thesis. Further, this part of the correspondence problem has been transferred to the segmentation stages of processing, simplifying matching. Chapters Four and Five describe an approach based on extracting and grouping edges into larger objects before matching. Also discussed are the problems found when calibrating two or more cameras. However, for the present, a description of the basic types of data representation will be given.

1.2.2 Image Representation and Hardware Restrictions

Images are normally represented as a matrix of grey levels extracted from some kind of sensor. The sensor can either be a vacuum tube or an array of CCD's or CMOS diodes. All current sensors produce analogue signals and the images are normally digitised before processing. This has the obvious disadvantage of noise but also the many processing advantages of the digital domain. The majority of research in image processing is described in terms of digital raster arrays of pixels.

An issue which arises in nearly all vision systems is that of error and noise control. This is of particular importance in stereo vision systems where direct measurements are being extracted. Hardware restrictions will normally impose a minimum pixel size which together with poor calibration and lens distortion may cause false disparity estimates. These errors are determined by the quality of hardware employed in the equipment and are unavoidable. It is important that such errors are recognised when attempting to assess the possible uses and failings of a system. The work described in this thesis uses a technique based on disparity histograms to estimate the magnitude of errors empirically and also to reduce the actual effects of quantisation noise. These issues will discussed in Chapters Three, Five and Six. In considering, algorithm design, it is also important to understand the limitations of currently available VLSI and sensor hardware. This must be done in the light of a practical final implementation.

Research conducted at Edinburgh [17] [72] in recent years has been directed at cost-effective implementations of VLSI vision technology. A camera sensor, an example of which is shown in Figure 1–2, has been developed. The design can be manufactured using a standard CMOS process. CMOS fabrication allows processing to be conducted on the same silicon substrate as the sensor array. This has been demonstrated by Anderson [1] in a single chip fingerprint recognition

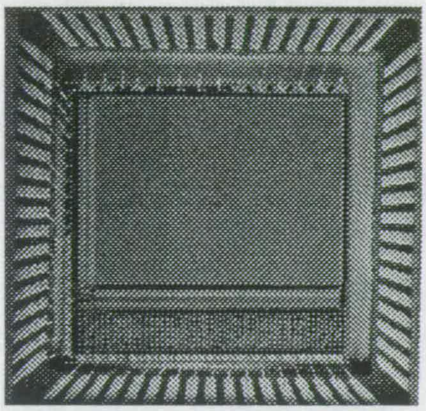


Figure 1–2: The ASIS1011 Single Chip Video Sensor

system. It is the intention of this work to design algorithms suitable for this type of target architecture.

Physical space restrictions place limitations on what calculations can be utilised in an implemented algorithm. For reasons of cost, floating point calculations have to be restricted to those which can be performed in real time on associated microprocessors. Constraints must also be placed on general multiplications and divisions which would also have to be placed off-chip. At this stage choices have to be made between performing calculations in the digital or analogue domain. Here there is little choice as the stereo algorithm needs digital information to work. The analogue to digital conversion must be performed before stereo matching[2]. In contrast, edge detection can be performed using analogue circuits, which is particularly easy when performing lateral differentiation.

1.2.3 Applications

The main thrust of the thesis is the development of vision systems and algorithms which could be implemented as commercial products. Although parallel algo-

[2] This is true for the algorithm in this thesis. Analogue stereo algorithms do exist one of which is considered in Chapter Two.

rithms and architectures have solved many image processing speed problems it is highly unlikely, at current prices, that large parallel machines will be employed to perform processing in any volume product. Basic image processing modules must therefore be tailored to specific applications.

Stereo vision has many possible applications when combined with other vision modules such as tracking. Here we consider human body detection for alarm systems and door opening devices. A machine which can passively determine the presence and distance that a moving object has in relation to the camera would have many advantages over the active electronics currently used. From the depth information the size of the object could be estimated and used to test its significance. Tracking would allow the use of "invisible walls". Thus, if a disparity threshold is crossed for a number of frames then the alarm can be sounded. Similar principles apply to door opening systems.

1.3 Thesis Plan and Objectives

The overall theme of the thesis is the design of vision algorithms aimed at creating a stereo vision system which could be employed in alarm systems and other range and detection applications. This thesis will explain the techniques and finally present some results, discussion and conclusions. The last chapter will also include some ideas for future research. In order to set the scene, and gain a better understanding of the problems encountered in developing machine vision applications a literature review was conducted. This has been divided into two chapters. Chapter Two will provide a review of current theory and practice in image processing with a discussion of the trade-offs involved in edge detection, segmentation and thresholding. Also, there will be a brief presentation of common recognition techniques. The last part of Chapter Two will review current hardware techniques and vision applications. This is of particular importance in understanding the restrictions that would be imposed by a final implementation. Algorithms can be then tailored accordingly. The third chapter will consider some possible stereo vision algorithms in detail. Calibration and accuracy are examined in light of their relevance to this application. Chapter Four will describe the lower level image processing applied in the system, while Chapter Five will deal with the stereo matching algorithm and higher level analysis. For convenience, the system described will be referred to as DETECT, throughout the thesis. Experimental results, from the system, will be presented in Chapter Six together with a description of the equipment employed. The final chapter will draw some general conclusions and describe future lines of research.

Chapter 2

The Image Processing and Hardware Background

2.1 Introduction

It has been instructive for the purposes of this work to survey some of the standard techniques proposed in other research. This section will describe the general algorithms such as edge detection, thresholding and segmentation. These are to be found in most machine vision applications. In addition to this survey, a section on current implementation techniques is also included. As described in the introduction, an aim of this work is the development of algorithms for an efficient hardware machine vision implementation. An understanding of current image processing hardware was therefore important. The hardware survey will start with some of the more general architectures, such as parallel machines and arrays, and then progress to a survey of VLSI architectures. Finally some specific applications will be briefly described, including a finger print recognition system developed at Edinburgh University, a circuit to calculate the centre of mass and an analogue implementation of a stereo algorithm.

Stereo vision algorithms are discussed in the next chapter. This division between segmentation and stereo is purely organisational and is not intended to imply that they should be separate in practice. It is the author's view that vision modules usually have considerable interdependencies. Errors and strengths in one module can have effects on the efficiency of another. This is particularly true in the relationship between stereo and edge detection, where matching problems can be reduced by extracting relevant edges. Edge detection can be adapted, possibly reducing the computation, to suit the stereo algorithm.

The review begins with a detailed discussion of edge detection. Edges are of prime importance in the current theories of human vision and are normally assumed to represent object boundaries.

2.2 Edge Detection

It is known that the mammalian eye performs a form of edge detection and that this constitutes an integral part of human vision [51]. In fact, it is thought that the human recognition system attaches far more importance to luminosity changes than to colour boundaries.

Based on the human model, the problem of edge detection can be stated as the extraction of luminosity gradients and the construction of higher elements to represent changes in intensity across a two dimensional image. Using these data structures, higher level algorithms such as model and stereo matching can be applied, while at the same time reducing the required processing. Thus, in view of the wide spread use of edge detection in applications and biological systems, an understanding of the trade-offs involved is required.

There are two classes of edge detection algorithm. Zero crossing detectors attempt to find the spatial second derivative of the image, while maximum gradient operators attempt to find the steepest part of the luminosity variation. Marr and Hildreth [52] suggest zero crossings based on evidence that humans apply this technique. In contrast, the Canny Operator [16] employs a maximum gradient technique which is optimal with respect to the criteria about to be discussed. Many other operators have been described in the literature [7][24][34]. Thus apart from Canny's methodology, two zero crossing techniques are also briefly described; Marr and Hildreth[52], and Vliet and Young[49].

2.2.1 Assessment Criteria

We require to be able to compare edge operators. Canny [16] has defined three criteria for comparison. These are:

1. **Good detection:** There should be a clear difference between true and false edges. In signal processing terms this can be simply expressed as maximising the signal to noise ratio.

2. **Good localisation:** Points marked should be close to the true centre of the edge.

3. **Limited number of maxima:** The number of marked responses to a particular edge should be restricted to one, (noise can cause several).

For these desirable qualities, measures were derived, for the one dimensional situation. These are shown in Equations 2.1, 2.2 and 2.3.

$$SNR = \frac{\left| \int_{-w}^{+w} G(-x) f(x) dx \right|}{n_o \sqrt{\int_{-w}^{+w} f^2(x) dx}} \tag{2.1}$$

$$Localisation = \frac{\left| \int_{-w}^{+w} G'(-x) f'(x) dx \right|}{n_o \sqrt{\int_{-w}^{+w} f'^2(x) dx}} \tag{2.2}$$

$$x_{sep} = 2\pi \left(\frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''^2(x) dx} \right)^{\frac{1}{2}} \tag{2.3}$$

Equation 2.1 can be used to calculate the signal to noise ratio (SNR) for a spatial filter, $f(x)$, applied to a luminosity function $G(x)$. The filter has an impulse response limited by $(-w, w)$ where n_o is the RMS grey level noise per pixel. For localisation, Equation 2.2 increases with the expected distance between a marked and true edge[1]. This parameter is inversely proportional to the degree of smoothing in the applied filter. The final constraint, Equation 2.3, is a limitation on the number of false peaks within a specified width, w. x_{sep} is the mean distance, between the first derivative peaks, in the response of of $f(x)$. The distance between maxima will be $2x_{sep}$ and we can expect $\frac{w}{x_{sep}}$ noise peaks in the filter response.

[1] Here we are working in the continuous domain. It is assumed, that for the spatial frequencies found in a real image, the pixel size will be sufficiently small to prevent aliasing errors. Pixel size is therefore ignored in these calculations.

2.2.2 The Canny Operator

The above allows us to measure the quality of a filter and define an optimal operator, for the above criteria, for a particular type of edge. Canny employed numerical techniques to maximise the product of Equations 2.1 and 2.2. The third constraint, Equation 2.3, was implemented as a penalty function. Thus when the desired distance, between first derivative peaks, was violated the penalty had a non-zero value.

The numerical optimisation procedure was employed to estimate a filter for unit step edges. The ideal was found to be close to the first derivative of the Gaussian curve[2], as shown in Figure 2–1. A selection of operators, suitable for the different types and directions of edge can then be calculated for the particular types of edge found in an application. Further extensions include the use of noise estimates over a sequence of images. The filter masks can then be adjusted accordingly.

Calculations such as the ones just described above would normally be considered impractical for every image in a particular application. A sub-optimal adaptation is now described as a compromise between performance and computation.

A Realistic Implementation of the Canny Operator

Figure 2–1 shows the first derivative of a one dimensional Gaussian curve, suggested as a filter in the last section. A 2D approximation can be derived from the application of two 1D curves in the x and y directions. Simpler processing is the result. Thus a practical near optimal step edge operator can be implemented as two one dimensional Gaussian smoothing curves followed by adjacent pixel differencing in both the X and Y directions. The vertical and horizontal

[2] If we reduce the ability of the filter. with respect to one criteria improvements can be made in another. This will change the optimal shape. It is unlikely to be Gaussian.

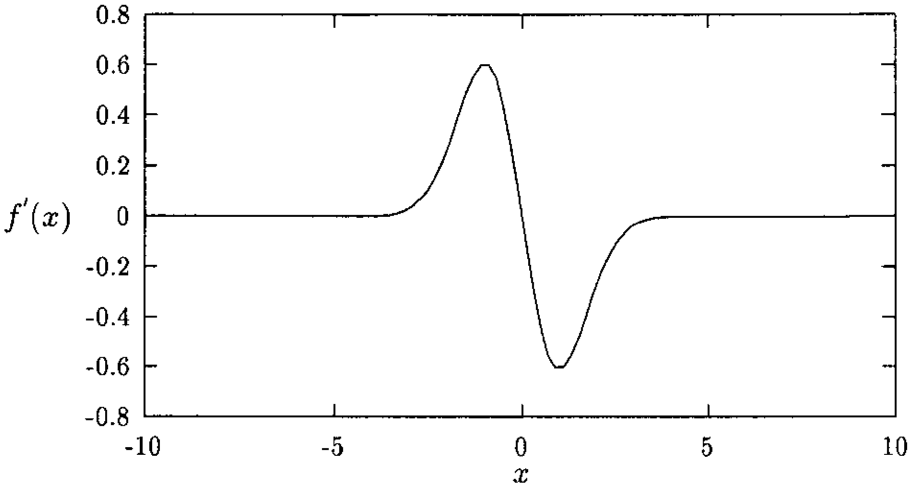


Figure 2–1: The First Derivative of a Gaussian Curve

components can then be combined to provide a direction and strength normal to the edge. This allows an estimate of the likelihood of a particular pixel being a true peak. Once the above calculations have been performed, a 1D representation of the edge can be obtained by tracking peaks. One problem that occurs when tracking is "streaking": an edge will fall below some predefined noise threshold. Hysteresis thresholding based on noise estimates from the edged image can reduce this problem. Canny calculates a noise estimate from the edged image using the second derivative of an impulse function. The two thresholds can then be extracted as some percentile of the noise histogram.

Problems still arise as to when, and where, edges begin and end. Edges can be broken at points of maximum or minimum curvature [3] and also when the overall strength, over a number of pixels, fall below minimum thresholds. Length can also be used. The above approximation to the Canny operator was implemented in software. A typical result image is shown in Figure 2–2

[3] These are more stable when an object moves[39].

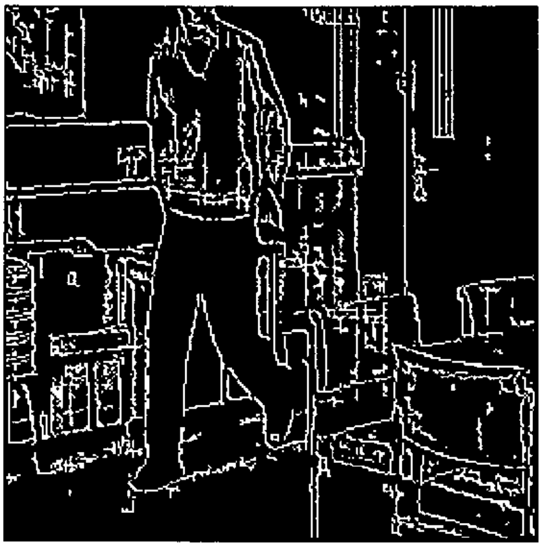


Figure 2–2: A Canny Edge Detected Scene

2.2.3 Other Edge Detectors

The Marr-Hildreth Operator

Marr and Hildreth [52] suggest edge detection using several filter channels of differing spatial frequency. The operators are based on the Laplacian of a Gaussian ($\nabla^2 G$), as shown in Figure 3–2. Zero crossings are tracked, instead of the gradient maxima, as in the Canny technique. The usual problems of tracking apply here, as in other operators, with one important difference. Zero crossings are not tracked on the basis of their strength. Thus edges will always connect to themselves, the edge of the image or to another object. However, multi-resolution spatial filters can be used to constrain the overall extraction of segmented information in a coarse-to-fine strategy. As will be discussed in the next chapter the coarse to fine approach corresponds with ideas of the human visual system and with some stereo matching algorithms.

The Non-Linear Laplace Operator

Vliet and Young [49] describe another zero-crossing operator which adapts to the spatial gradient within a pixel neighbourhood. Each neighbourhood is searched for the largest and also the smallest grey scale value. Then, two functions, called *gradmax* and *gradmin*, are calculated. *Gradmax* is the difference between the largest, grey level value and the central value and *gradmin* is the difference between the smallest grey level, and the central pixel. The output value, called NNLAP(X,Y), is then calculated from the sum of the two functions *gradmax* and *gradmin*.

The result, of the above processing, is an image composed of positive, negative and zero regions. Zero crossings are extracted from the joins between the two types of region. A problem arises when there are areas of zeros. Where is the true positive to negative transition? Before crossings can be found these zero regions must be assigned to their nearest positive or negative area. A distance transform can be used to calculate the nearest region to a zero pixel. Vliet and Young suggest the Borgefors [9] method. This computes, in two passes, paths to the nearest region. Tracking techniques can then be applied to the zero crossings, as in the Marr-Hildreth operator.

2.2.4 Discussion of Edge Detection

Although edge detection is simple in concept, it is rarely so in practice; many problems can arise due to noise, closely spaced edges and poor thresholding. It has been suggested by Torre and Poggio [81] that edge detection is "ill posed". In essence, all edge detectors perform some form of differentiation, thus amplifying noise. It is for this reason that detection normally begins with smoothing. Canny showed that the optimum low pass filter for step edges is close to the Gaussian curve. However, in practice sampling will ensure only an approximation to this ideal. Further, as noted by Horn [35] other types of formalism can provide similar valued weights. Apart from sampling, decisions must be made about the size of an operator. Both accuracy and computational complexity must be considered; computational complexity, because the number of pixel multiplications increases as the square of the mask width, and accuracy, because if an operator is truncated too much its performance will fall. A slightly less obvious tradeoff is that between localisation and detection as defined in Equations 2.1 and 2.2. An operator which provides a high signal to noise ratio will act as low pass filter, smooth the image, and reduce positional accuracy. The opposite applies if a filter provides accurate location.

Overall "Canny edge detection" has become widely used in the image processing and machine vision community. However, for efficiency reasons, such a stand alone module has not been utilised in the work, described in later chapters, where only parts of the above edge detection theory are applied. For example, if only vertically orientated edges are required, tracking can be reduced to downwards searches. Also, differentiation can be restricted to the horizontal direction. Alterations such as these can reduce the required computation without a reduction in performance and show that individual machine vision modules should not be considered in isolation. Thus in an overall system, functions such as stereo matching, edge detection and thresholding will be interdependent. Thresholding is described in the next section.

2.3 Thresholding

Thresholding is one of the more common techniques used in image segmentation and is often termed a pixel classification problem. Sahoo, Soltani and Wong [77] group thresholding algorithms into three classes: point dependent, region dependent and local. A point dependent algorithm classifies pixels solely from it's individual grey value. In contrast, region dependent algorithms take account of the neighbourhood of a particular pixel. Local thresholding is the application of global techniques to smaller sub-images. Here, smoothing is often used to limit discrepancies, caused by threshold variation between areas.

2.3.1 Point Dependent Methods

Sahoo, Soltani and Wong [77] list seven different point dependent methods which will be summarised here:

The Ptile Method

This simple method assumes that the percentage area of the object is known. A threshold is chosen to provide that percentage of object pixels. It must also be known, *a priori*, whether the object is darker or lighter than the background.

The Mode Method

The histogram of the difference image is extracted and assumed to consist of background and foreground peaks. The threshold can be based on the valley between the peaks. Problems occur when valleys are flat, when peaks are unequal and when the difference between the peaks is small.

Ostu's Method

Ostu [63] describes a method based on minimising the ratio, η, from Equation 2.4,

$$\eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{2.4}$$

where σ_T^2 is the variance of all grey levels in the entire image and σ_B^2, a joint variance, calculated from,

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 \mu_2)^2, \quad \omega_0 = \sum_{i=0}^{t} p_i \; and \; \omega_1 = 1 - \omega_0, \tag{2.5}$$

μ_1 and μ_2 are means of the grey levels above and below a particular threshold. p_i is the probability of a particular grey level, i. From the above, the parameters σ_T^2 and σ_B^2 can be calculated from the difference histogram and used to evaluate η for each possible threshold. The lowest η indicates the optimal threshold.

Histogram Concavity Analysis

It is often the case that there is no clear valley in the histogram and the ideal threshold is on the shoulder of a histogram. After calculating the smallest convex polygon which covers the histogram, possible thresholds can be selected at maxima of the difference between the true histogram and the convex curve. This is illustrated in Figure 2–3.

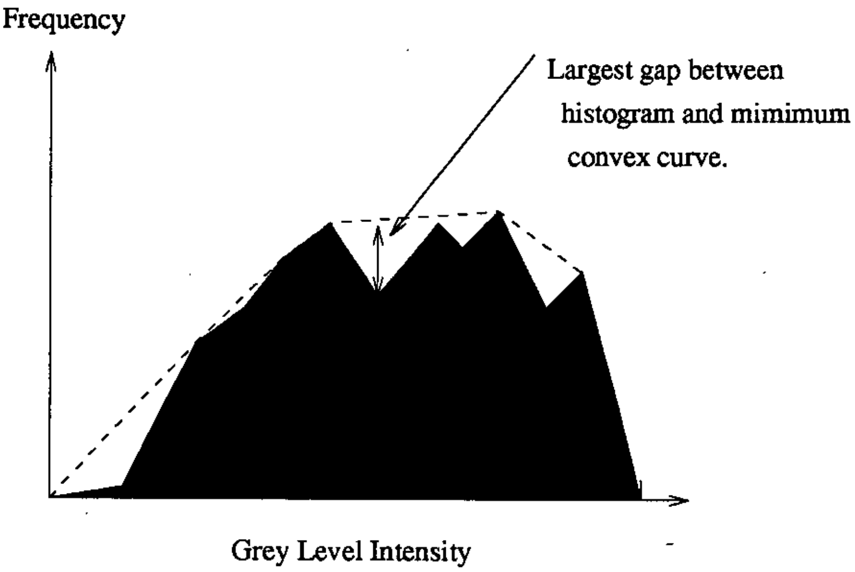


Figure 2–3: Histogram Concavity Analysis

Entropic Methods

Entropic methods utilise information theory to make a threshold decision. Several methods have been proposed [69][40][45] which attempt to maximise an equation similar to 2.6.

$$H' = H'_b + H'_w \tag{2.6}$$

H'_b and H'_w are normally calculated directly from the histogram using some form of Equations 2.7 and 2.8.

$$H'_b = -\sum_{i=0}^{t} p_i ln(p_i) \tag{2.7}$$

$$H'_w = -\sum_{i=t+1}^{l-1} p_i ln(p_i) \tag{2.8}$$

where the threshold, t, is chosen from l, possible, grey levels.

2.3.2 Region Dependent Methods

With the techniques described above, thresholds are dependent, solely, on global image statistics. No allowances are made for regional information. Apart from adaptively thresholding sub-images, local information can be used to improve the characteristics of the histogram and make the point dependent techniques more accurate.

Histogram Improvement

Theoretically, edges are likely to be at the boundary between the background and foreground of a difference image. Using this assumption, edge information can be used to calculate weights for histogram values. After applying an edge operator, pixels which have high values can be weighted least in the calculation of the new histogram from the original image. The threshold can then be chosen using one of the techniques described above. As a variation, Weska and Rosenfeld [88] suggest that actual thresholds be chosen from peaks in the histogram of pixels which have been extracted as edges.

One other method, of region based histogram improvement, uses quadtrees[90]. A particular difference image, or sub-image, can be recursively divided into blocks, according to whether the standard deviation, of all the pixels within that block, exceed a predefined limit. The limiting standard deviation can be altered as the hierarchy is descended[4]. Thus, by the end of the division a particular block should represent a roughly homogeneous region. The pixels, within that block, are then replaced by its mean.

The division is illustrated in Figure 2–4. Due to the homogeneity of each block the overall histogram will have deeper valleys and sharper peaks which, in some cases, might allow better thresholding. Code was written to implement the quadtree operation and example difference images are shown in Figure 2–5. Histograms are also shown. Clearly, the histogram of the quadtree image has sharper peaks and deeper valleys than that of the original difference image, making the choice of a threshold, based on the histogram, simpler. Another advantage

[4] If the standard deviation is increased blocks are less likely to overlap an image feature as the hierarchy is descended.

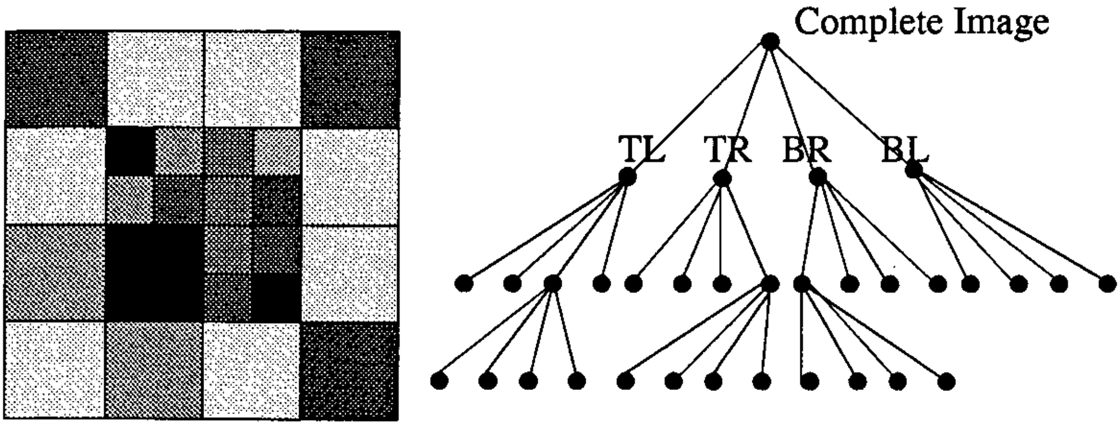


Figure 2–4: The Quadtree Method of Segmentation: Each node represents a quadrant of the picture. TL = top left, TR = top right, BR = bottom right and BL = bottom left. Quadrants represent an area of the image with grey level values within a defined standard deviation.

of quadtrees is that they provide a conveniently segmented and easily accessible data structure for further processing.

Relaxation Techniques

Relaxation methods are only briefly mentioned here, as they often require considerable computation and are unpredictable in the time taken to converge. Pixels are initially classified according to a very rough threshold. Pixels are then altered according to the surrounding neighbourhood. A black pixel in white neighbourhood will likely be classified as black and vice versa. The process is repeated until convergence.

2.3.3 Discussion of Thresholding Techniques

In this short review, only the most general ideas in binarisation have been covered. Many, more specialised, algorithms have been developed for particular applications. For example, several thresholding algorithms have been developed specifically for character recognition [89]. None of the above will work in all situations and

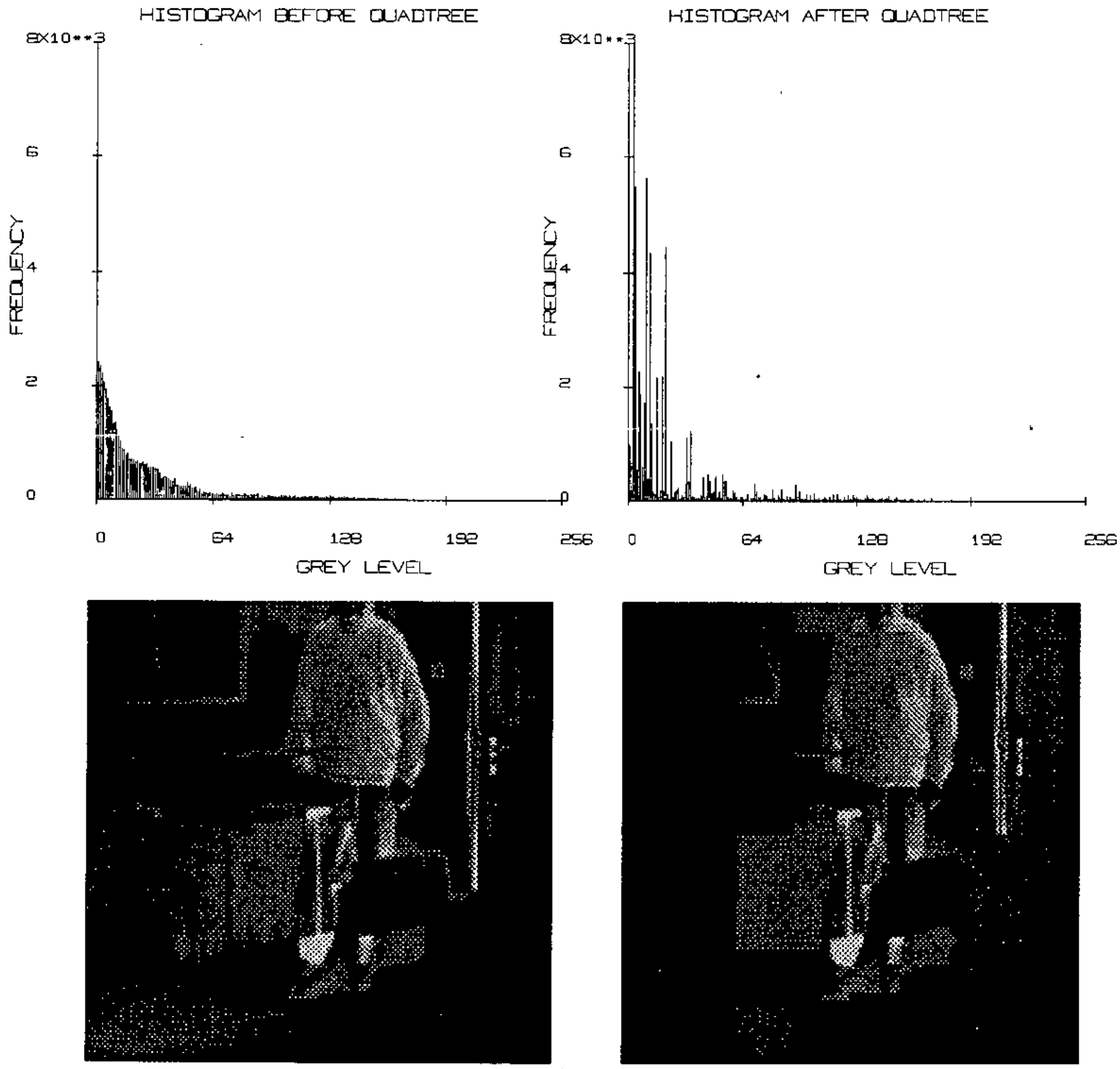


Figure 2–5: Original (left) and quadtree (right) images and respective histograms.

techniques can be combined to satisfy a wider range of situations. As described in Section 2.3.2, thresholding can be related, closely, with edge detection. However, there are also close relationships with other problems such as model matching, finding regions of interest and background estimation. The next sections will consider some techniques related to building larger features from the basic ones just described.

2.4 Segmentation

The problem of separating objects, from each other and from backgrounds, is known as segmentation. An exact definition of how this broad aim should be achieved is hard to come by and may or may not include thresholding and edge detection, as described above. A general segmentation algorithm, such as that present in the human vision system, will require the integration of many different sources of information. For example human beings are capable of recognising and extracting most unknown objects from a mixed bin of parts. Here, it seems logical that humans use knowledge of the physical world to extrapolate the few visible edges and determine the location of the object in the bin. In this case, physical knowledge of how solid objects react in the world will be integrated with visual knowledge. Although, the development of such a system is beyond the scope of this thesis, it should be noted that attempts have been made to solve the "bin picking" problem. For example, work at Sheffield University has been directed at building robots which perform this task[67].

In practical terms, segmentation means the construction of larger primitives from lower features; edges can be built into outlines and clusters into surfaces. This section will deal with some common techniques used to find connections between objects, including the Hough transform and graph matching.

2.4.1 Hough Transforms

As Figure 2–2 shows, edge detectors do not provide perfect outlines or boundaries. There will always be breaks where the luminosity gradient falls below the set threshold. A technique often employed to extract edges from unreliable and discontinuous data is the Hough transform [37]. It's basic form attempts to find straight line edges in terms of gradient and offset. Each pixel in the edged image can then be assigned a particular gradient and distance which are used as coordinates in the Hough space. Thus the Hough space is divided in terms of gradient and offset coordinates, with peaks corresponding to lines in the original image. The problem with this formulation is that it is only sensitive to straight lines.

A generalised Hough transform was developed by Ballard [5] and is capable of detecting arbitrary shapes. Here votes for each accumulator [5] are cast according to a predicted centre which is calculated on the basis of a pixel s spatial gradient. Figure 2–6 shows an irregular shape with an arbitrary reference point chosen in the centre of the object. Before the transform is applied a model of the shape, under study, is used to calculate values of the R-table. This table records all the possible radii for each spatial gradient. Thus the R-table is constructed from the orientation, ϕ, at each boundary point and the radius, r, from that point to the central reference. When a new image is processed the spatial gradient for each boundary pixel allows access to all the radii for that orientation. Thus all possible centres can be accumulated by drawing a circle of votes in Hough space.

The generalised Hough transform has all the advantages of the basic version, in that it is very robust with incomplete and noisy data. However, problems occur when the orientation of the object in the scene is unknown. Unfortunately this is, probably, the majority of situations. Object orientation has to be introduced as an extra dimension resulting in an increase in required computation. Computational complexity is a problem often found when implementing Hough transform

[5] The Hough space is usually divided into an array of cells. Each edge pixel will cause a particular accumulator to increment.

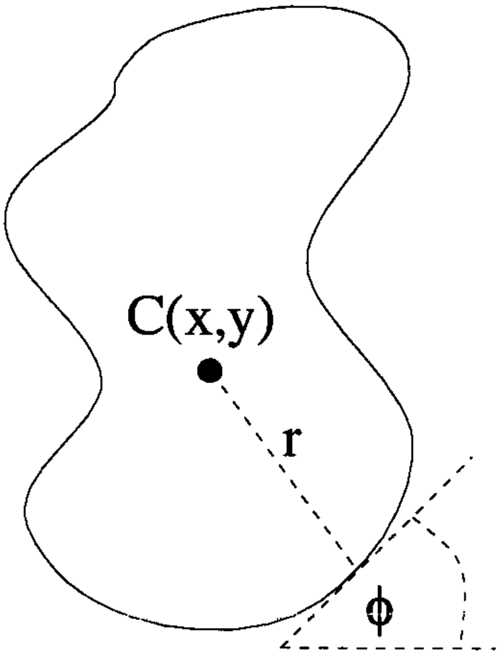
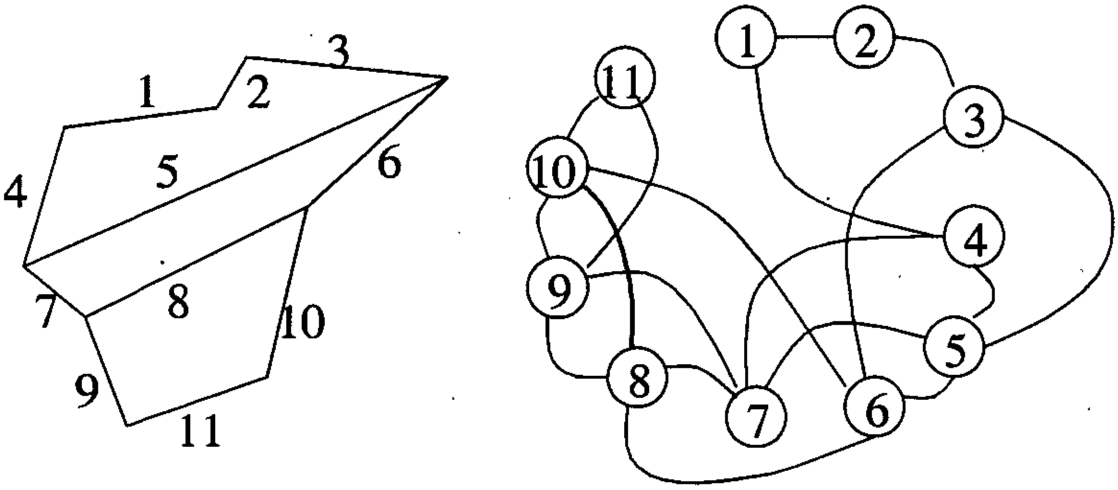


Figure 2–6: The Generalised Hough Transform

Angle of Orientation	Set of Possible Vectors
ϕ_1	$r_1^1, r_2^1, \ldots, r_{n_1}^1$
ϕ_2	$r_1^2, r_2^2, \ldots, r_{n_2}^2$
.	.
.	.
.	.
ϕ_m	$r_1^m, r_2^m, \ldots, r_{n_m}^m$

Table 2–1: The Hough Transform R-Table



Structure

Connection Graph

Figure 2–7: Graph Matching

algorithms. Every extra parameter requires a new dimension in the search space. Although maximum and minimum bounds can be applied to a model's search space, the Hough transform is likely to be too complex for the work considered later in this thesis. Problems also arise if a particular object rotates in the scene. In this situation, multiple Hough models must be extracted and then matched. Again computation is likely to be unreasonable. One technique which, sometimes, is resilient to rotation, is graph matching. However computational problems still arise in a different form.

2.4.2 Graph Matching

In many applications there is a known relationship between different features. A common approach to segmentation, or grouping, is to take a set of extracted features and calculate interconnection parameters. For example, the distance between the centroid of two edges can be one connection, as can the relative feature directions. These parameters, and many more, can be represented using a graph where nodes represent features and arcs represent connection strengths. A diagrammatic example is shown in Figure 2–7.

For each incoming image a set of incomplete features and their relationships is extracted and compared to the stored graph. This procedure requires the discovery of *maximal cliques.* If one matched clique is discovered within another it is likely that the smaller clique is an incomplete representation of the larger. Thus, as with the Hough transform, this method allows for incomplete data. However, unless the search is pruned, computational problems will also arise with graph matching. General graph matching of this sort is known to be NP-complete [87].

Apart from matching models, graph matching data structures can be used to combine fragmented edges. A seed edge can be defined as a node and other edges, in the same neighbourhood with similar orientation, can be connected with varying degrees of confidence. The problem can be reduced to the pixel level and edges extracted by choosing a path through the graph. Heuristic cost functions can be defined to evaluate each potential edge segment or pixel. Examples of heuristic parameters include edge strength, curvature and distance.

Graph matching has also been used to find corresponding features[6] between stereo images[61]. However, as with model matching, stereo graphs are likely to be too computationally complex for the low cost alarm applications considered here.

2.4.3 Discussion of Segmentation

Both the above techniques have a wide range of possible applications which are not restricted just to image processing and machine vision. The Hough transform is useful in that it can detect specific types of feature, such as circles and ellipses, whereas the graph matching can be used to establish relationships between different features. Unless the search space is restricted, both suffer from complexity problems.

Segmentation should also be considered in the light of this application. For example, the problem of estimating the background and foreground parts of an

[6] The stereo correspondence problem is considered in the next chapter.

image is interlinked with that of thresholding and segmentation. In effect it is a classification problem. As such, it is unlikely to be perfect for every pixel. Also of importance is the way that background extraction, thresholding and feature grouping are related.

In the imaging hierarchy, described in Chapter One, the stages above edge detection are often referred to as clustering or grouping. It is at this level that typical heuristic decisions are often made. However the decisions are based on information from the lower processes such as thresholding and edge detection. Also lower level processing can be improved, in a feedback loop, by information from higher levels. Features are often proposed by low level processing and "filtered" by higher level grouping and object construction . Such integrated vision algorithms are usually more application specific and less well understood.

The next section is included as a review of current machine vision architectures and, in particular, specialised VLSI systems. As has been suggested above the practicalities of an implementation have considerable bearing on the overall design.

2.5 Hardware Review

Researchers have designed many systems for image processing, although, for obvious reasons of cost and time, there have been far fewer hardware implementations. It is the author's view that the limitations of hardware have considerably reduced the effectiveness of practical research from the acquisition stage through to the image analysis stage. Figure 2–8 shows the typical information flow for an image processing system. Most of the existing specialised hardware is directed at lower level processing. This is where the current bottleneck is and inherent parallelism at its most obvious. Images stored as arrays of pixels can easily be mapped onto parallel processors. An assumption often made is that higher level operations are inherently less parallel. Weems [73] has countered this view saying that it is wrong to assume that the computational bottleneck always lies with pixel based operations. He cites graph matching as an example where there is significant com-

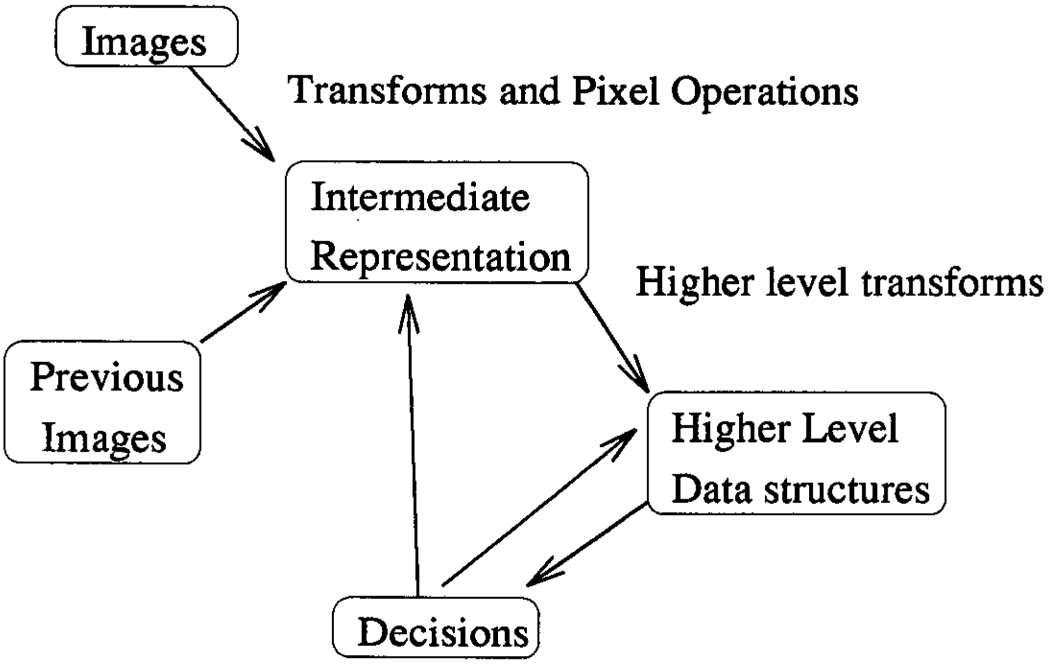

Images
Transforms and Pixel Operations
Intermediate Representation
Higher level transforms
Previous Images
Higher Level Data structures
Decisions

Figure 2–8: Information Flow in a Typical Imaging System

putation at higher levels. In a hardware implementation of such algorithms the main problems tend to be centred on efficient communication and the mapping of processes to processing elements. In this respect, it is likely that the real reason most specialised hardware is orientated at early processing, is that low level functions are the best understood and have the most obvious parallelism.

This section will present an overview of a few of the more important architectures proposed in recent years. Such a study is highly relevant in this work where we are attempting to build cost effective and practical systems. In this context it is necessary to understand what architectures would be practical and useful for this type of application. Thus the section will begin with some general architectures, designed to satisfy a number of processing functions, and progress towards more application specific hardware. The first architectures considered are array processors.

2.5.1 General Parallel Processing

Early image processing is ideally suited to implementation on parallel networks of processing elements. Sub-images can usually be mapped directly onto the processing array with small boundary overlaps and the problems of communication kept to a minimum. As a result many research projects have been conducted on arrays of Transputers or machines such as the ICL DAP where advantages of speed can be gained using general purpose machines. Apart from general parallel hardware, specific image processing languages have been developed [29] which allow machine vision problems to be expressed in a higher level form.

Array Architectures

The two dimensional nature of basic image processing has led many researchers to propose processing arrays. The most common, in the British research community, is the Transputer [21][74][92]. The Transputer was introduced in 1985 and has been used to build many multiprocessors of the SIMD and MIMD types. It is programmed in OCCAM which directly reflects the parallelism inherent in the architecture. The current versions of the Transputer have on-chip memory and are capable of connection through parallel I/O ports to other Transputers. Usually Transputer arrays are arranged with a master processor distributing tasks to it's multiple slaves. The array will be serviced by a host such as a PC or workstation.

Morrow and Perrott [59] describe several low level algorithms implemented using Transputers. An entropy based edge operator was built using three processors connected in the pipeline shown in Figure 2–9. The first processor takes in the nine values of the present pixel s neighbourhood and calculates probability values, P_i. The second processor calculates nine values of $P_i log P_i$ whereas the third calculates the sum and normalises the final value. Results are returned to the host. The division of an algorithm in this way is often referred to as task parallelism. It is a technique common in current high performance scientific computers and RISC microprocessors. Systolic arrays also employ such task parallelism.

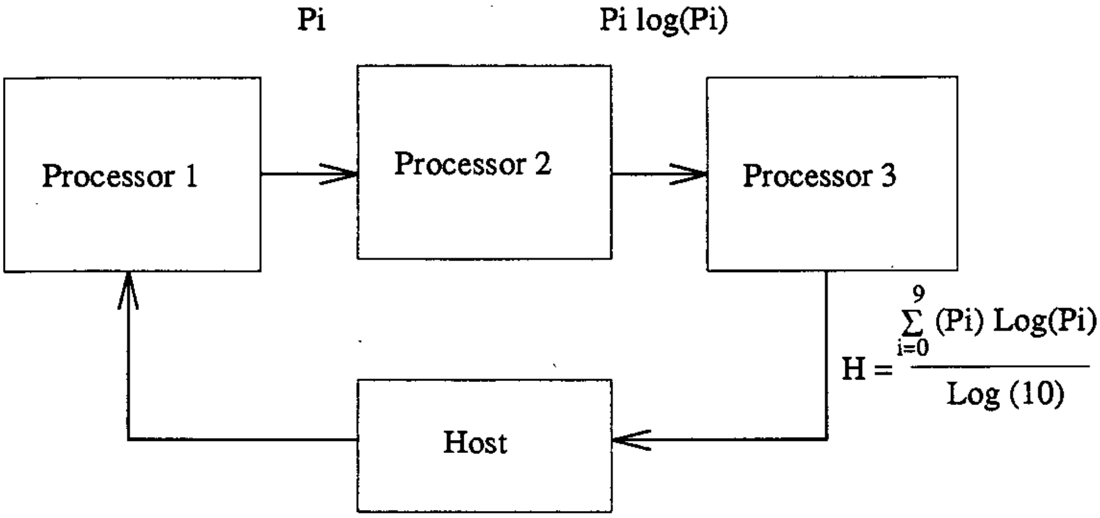


Figure 2–9: Entropy calculations using Transputers

Systolic Arrays

Systolic arrays were first proposed in the early eighties by H.T. Kung [47] of Carnegie Mellon University based on the idea of task parallelism described above. The term systolic comes from the human heart and processing arrays are meant to resemble opening and shutting of heart valves. Effort has also gone into developing compilers and efficient optimisation tools to generate the arrays and interconnect. [3][18]. Kung proposed a machine called Warp [2] based on a linear array of processing elements connected to a host through an interface unit as shown in Figure 2–10. The design was specifically aimed at image processing problems although other applications were programmed.

Although linear arrays only have two PE's to communicate with the host an increased I/O bandwidth is possible due to the separation of function along the array. Every warp processor has it's own program memory of 8k combined with 32k words of data memory. A larger data memory can allow more computation for the same I/O bandwidth for some algorithms. The architecture of each cell is shown in Figure 2–11.

Individual cells can receive data from either of its two neighbours. Also, data can pass both ways along the array. For some algorithms this can ensure that all

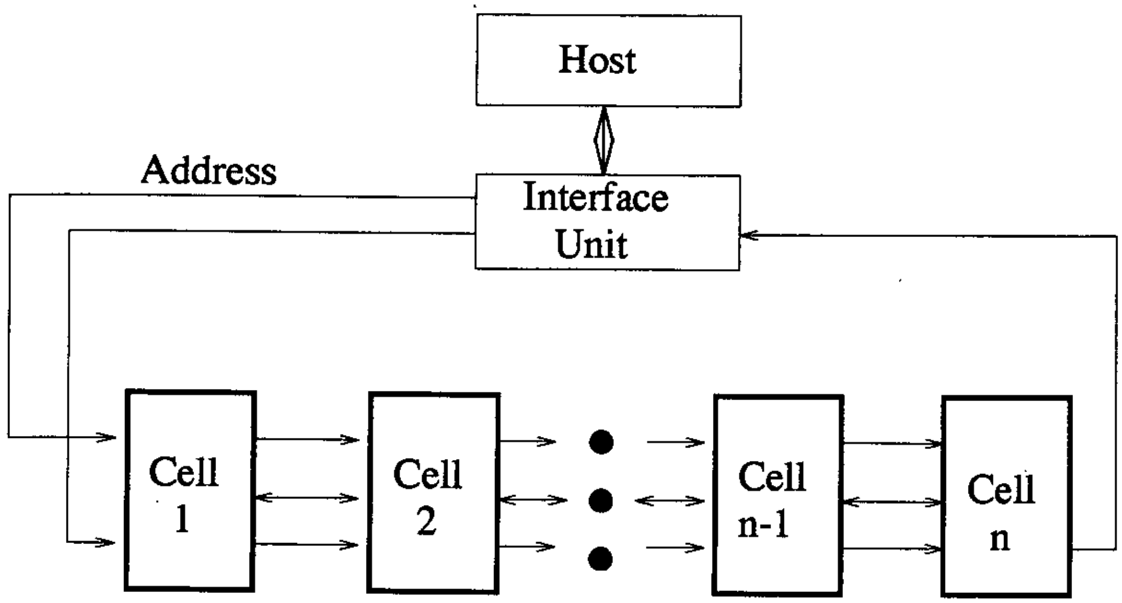


Figure 2–10: Warp Processor Array

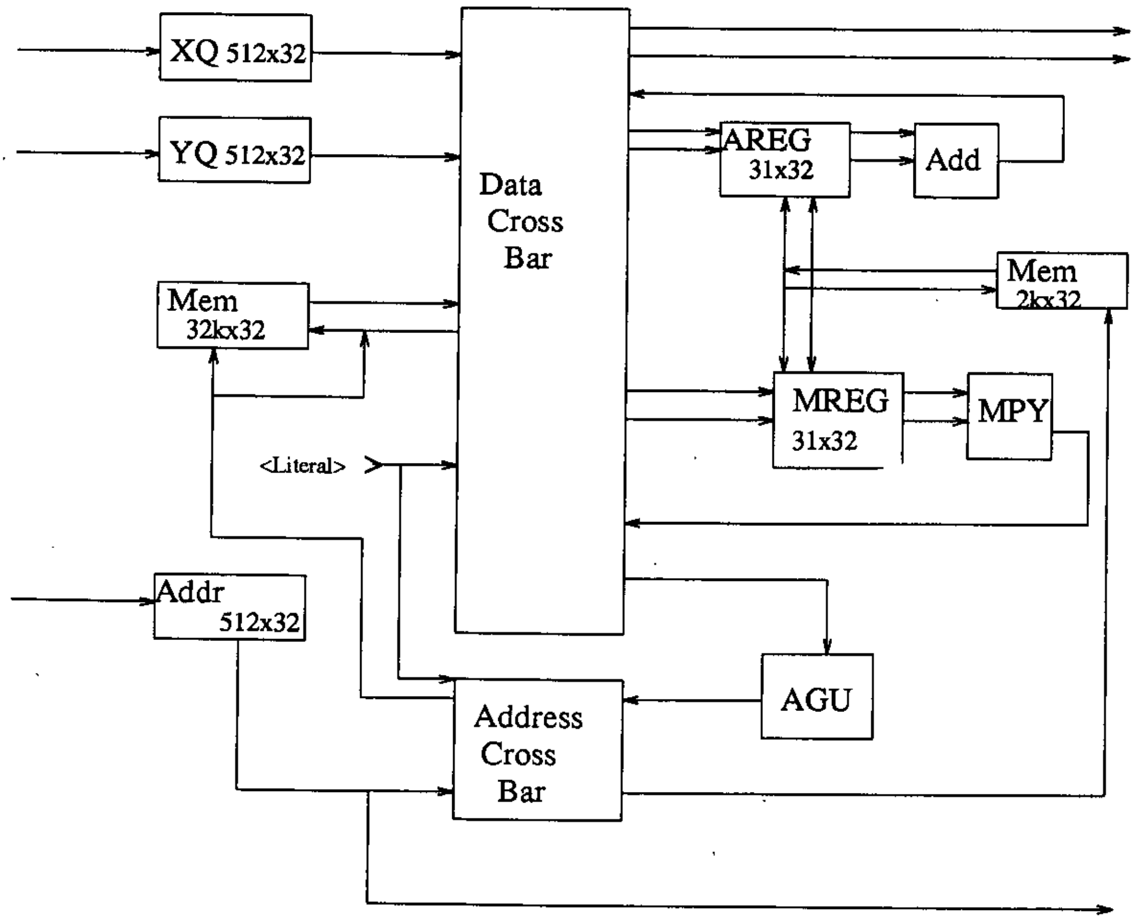


Figure 2–11: Warp Cell Data Path

cells are processing making better use of the array. Each cell communicates with its left and right neighbours through two data and one address link. All three links have a 512 word queue at their inputs. This is large enough to buffer one row of a 512x512 image. Hardware control ensures that one cell cannot write to another cell when its queue is full and cannot read from its own queue when it is empty. With this type of I/O individual cells can block the passage of data and care must be taken in programming to ensure that data flows evenly through the array. To allow this type of dataflow, clocking signals must cross chip boundaries. Speed restrictions may ensue.

Warp cells have one floating point multiplier and one floating point adder which are pipelined within themselves. Obviously effective programs must supply these arithmetic pipelines with uninterrupted data. Thus, algorithm implementation is restricted to those with regular data sequences and few interrupts. The next section describes some of the problems normally associated with programming such machines.

Parallel Programming

Parallel programming is complicated by the problem of program partition. Annaratone et. al. [3] propose three methods:-

1. **Input Partitioning:** Each processing element computes only a portion of the input data and its corresponding output. Most low level vision algorithms can be efficiently implemented using input partitioning. Often the problem with this arrangement is downloading the various sub-images to each processor's individual memory. However if the array is big enough and the operations well ordered in time and space the array has only to be loaded once and the results can be stored at each processor. A new operation is then demanded by the master processor as a SIMD instruction. This type of algorithm is likely to be inefficient at higher levels where disparity information has to be brought together from other parts of the image.

2. **Output Partitioning:** Each PE processes the entire input data but only produces a section of the output. Histogram processing and image warping are examples where output partitioning is efficient.

3. **Pipelining:** Pipelining is typical of systolic arrays where each cell performs one part of the computation. Annaratone et.al [3] provides, as an example, a solution of the partial differential equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \tag{2.9}$$

The system is solved by recursively calculating the values of u on a two dimensional grid using the Equation 2.10:

$$u'_{i,j} = (1-\omega)u_{i,j} + \omega\frac{f_{i,j} + u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j}}{4} \tag{2.10}$$

where ω is a constant parameter. Each cell performs one of the above relaxations. While cell k is performing on raster i, the preceding cell, k-1, is computing row i+2 and the following cell, k+1, is computing row i-2. The process is repeated until convergence is achieved..

2.5.2 Pyramid Architectures

Some researchers have noted that the human vision system is dynamic and does not process entire scenes at any one time. The eye can concentrate on a particular part of a scene and only be peripherally aware of other parts of the scene. P.J. Burt [13], among others, propose a more active architecture around the general idea of "smart sensing". There are a number of ideas which constitute smart sensing:-

1. **Controlled Resolution:** Clearly, it is difficult to alter camera resolution during operation. However low-band pass filters and sub-sampling can be employed to reduce data rates to the minimum that is required.

2. **Restricted Windows:** It is desirable that only windows of current interest be extracted from the sensor. Thus an architecture should be able to access specific regions of the image individually.

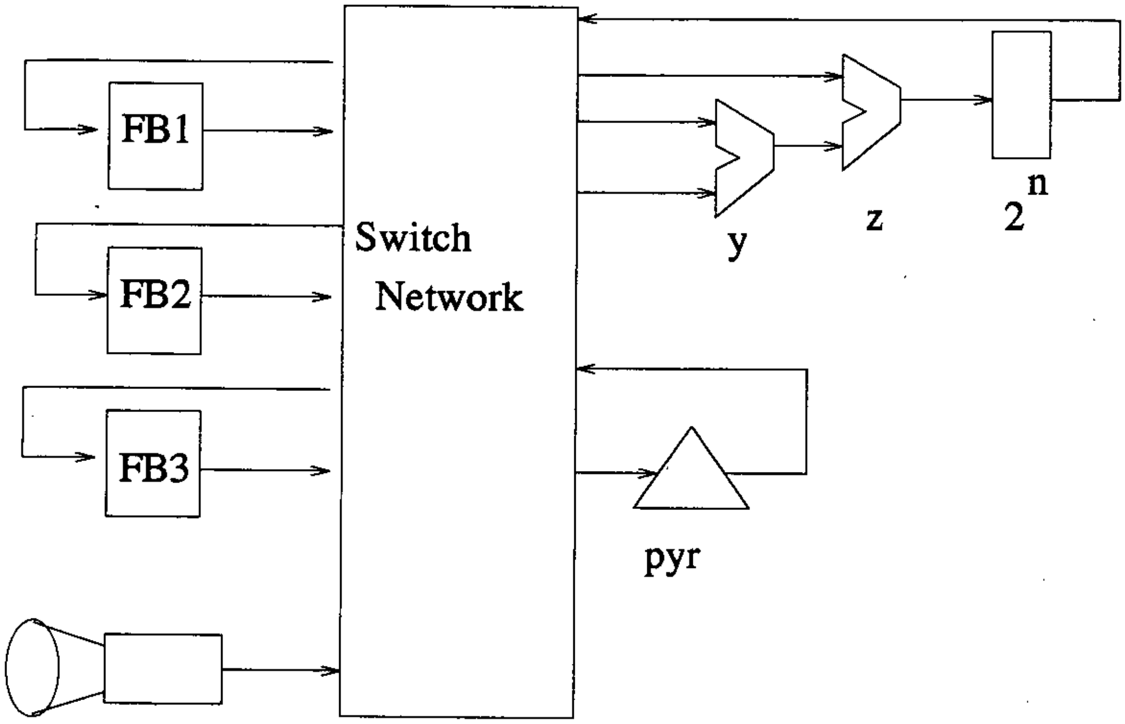


Figure 2–12: Pipelined Pyramid Machine

3. **Feature Extraction:** The general extraction of edges and other measures of image structure require flexible window sizes and hardware capable of convolution. Look-up tables may also be necessary.

4. **Compressed Range:** Another suggestion by smart sensor proponents is the compression of grey level resolution using high-band pass filters followed by a log function.

The Pipeline Pyramid Machine (PPM) proposed by Burt, and shown in Figure 2–12, consists of a number of special purpose functional units connected through a switch network. The flexibility provided by the switch network allows configurations and algorithms to be dynamically changed during operation. This is important when specific image regions are being processed and allows more functional units to be added to the system.

Burt gives several examples of the system in operation including detecting flaws in television screens and smart surveillance. The latter is of particular interest in this research as the application considered, in later chapters, is vision alarm systems. Based on difference images between successive frames, a decision is made as to whether there is motion in the scene. This is then decomposed into a set of spatial bandpass channels by constructing a Laplacian pyramid [14]. Laplacian pyramids are built by taking the difference between two Gaussian outputs , one of which provides a smoother, lower pass, response. G_0 and G_1 represent the original and low pass filtered images respectively. A difference image, $L_0 = G_0 - G_1$, is calculated. As L_0 is a difference image, fewer bits are required. Also, G_1 can be sub-sampled on the basis that it has been low pass filtered. The same procedure can be repeated through several Gaussian channels to achieve a sequence, $L_0, ..., L_n$. This is known as a Laplacian pyramid and can be used to completely reconstruct the original image.

Such data structures can be used to detect events and regions of interest at a specific spatial frequency. Reduced processing is attainable due to the reduced resolution and data compression. The PPM machine's switch network, with connected and variable functional units, is suited to running such algorithms.

2.5.3 VLSI Architectures

The above descriptions of hardware and related algorithms have all been directed at solving general vision problems. They are capable of computing more than one algorithm. Many VLSI architectures are aimed at individual image processing functions such as edge detection, correlation and filtering. Other processing chips have been directed at slightly more specific functions such as stereo vision. For cost reasons, VLSI application specific image processing systems have been less common and tend to be directed at commercial products. Initially, this section will cover some of the more recent research VLSI architectures targeted at solving low level image processing functions. Following this some analogue architectures will be described including moments calculation and a CCD/CMOS stereo sensor.

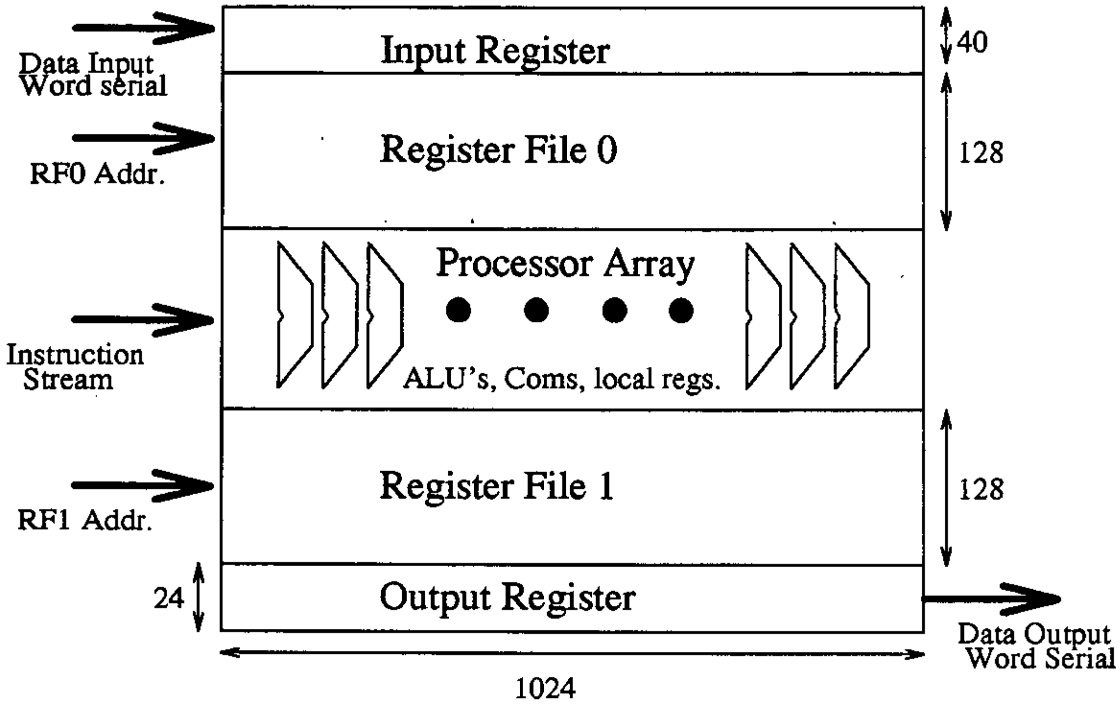


Figure 2–13: The SVP 20MHz processor

General VLSI Processors

Texas Instruments have developed a chip called the SVP or Serial Video Processor. This has 1024 bit processing elements combined with input and output register files. A feature common to most of the current image processors is the closeness of the memory to the processing elements and the techniques used to access memory appear to be of increasing importance. Another feature of these processors is their SIMD nature. They are directed at low level functions where the same operation has to be performed many times.

Figure 2–13 shows the layout of the SVP processor. Data flows in at the top left hand corner and out at the bottom right corner. Both input and output are serial and could be in the form of raster scans. Individual lines can then be built up in the input register and then moved downwards. There are two dual ported register files, one for input and one for output. Addresses and instructions are provided by a controller. Thus some form of address generation must be considered when using this element.

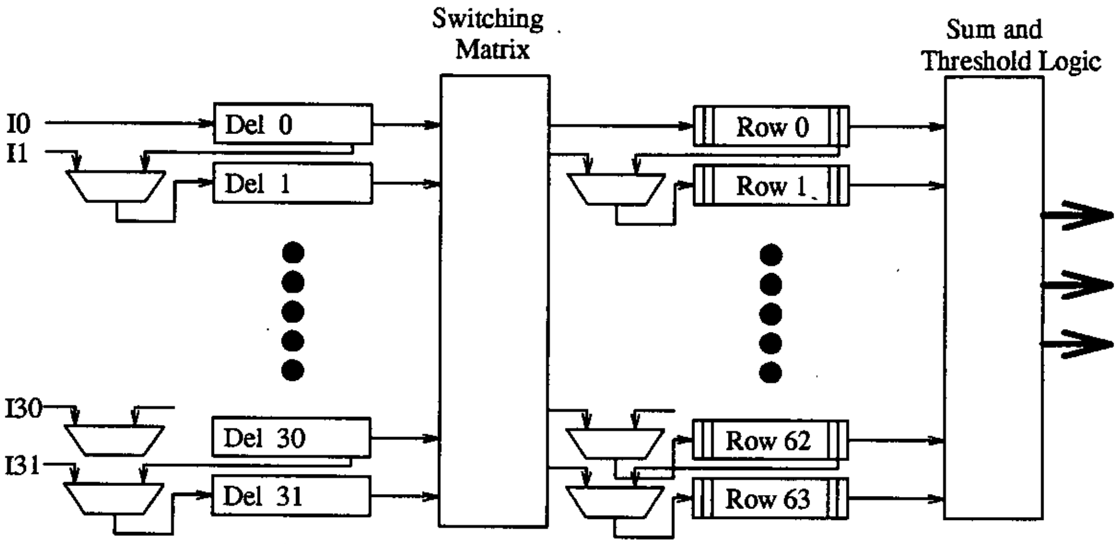


Figure 2–14: The IRIS Video Processor

A second architecture is shown in Figure 2–14 and differs from the SVP architecture in that it includes a general purpose switching array. There are two memory buffers either side of the switching array with the processing elements restricted in their function, only being able to accumulate and threshold. However many image processing functions, eg. Fourier Transform, can be performed using switch arrays combined with comparators.

Of interest in both these architectures is the use of input and output memory to compile raster scans into data which can be acted on in a regionally specific manner. In the SVP processor the input buffer is 1024 pixels, whereas in the IRIS chip it is 512 pixels wide. Also of interest is the use of a switch array, in the IRIS processor, to provide some of the functionality of the ALU's of the SVP processor. The general conclusion that can be drawn from these examples is that memory organisation and data storage is of crucial importance. Indeed, for some functions, eg. the Fourier Transform, data shifting and register organisation is a major part of the processing.

Application Specific VLSI Processors

There now follows three architectures directed at application specific tasks. The first computes the Canny edge detector described earlier in this chapter. The second was designed over a number of years in Edinburgh as a finger print security system. The third is a stereo matching architecture developed for a CCD/CMOS implementation.

The Canny design [75] is divided, as in the algorithm, into four blocks and was designed to edge detect at 25 frames per second. Initial smoothing is performed using two identical 1-D convolvers as an approximation to the Gaussian function. If a design was implemented, further silicon space savings would be obtained by halving the mask, reversing the data stream and then adding the results together. As Figure 2–15 shows, Ruff computes the two masks using a buffered memory between two convolvers. While the X-convolver is writing to one memory buffer in row/column format the Y-convolver reads from the other buffer in a column/row manner. Access to the memory is then switched. Also shown in Figure 2–15 is a more detailed circuit of the half Gaussian filter. Input and output data is eight bits wide and internal calculations range between 8 and 21 bits. Gradient magnitudes and directions are calculated for the smoothed image on a 3x3 neighbourhood input through FIFO buffers. These are then used to interpolate, to sub-pixel acuity, the expected values of gradient either side of the central pixel. Next, non-maximal suppression is applied by marking, as edges, those pixels where the central value is greater than the two interpolated pixels.

The next stage of a Canny edge detector is tracking. As edge tracking is unpredictable, and the system pipelined, the Canny hysteresis thresholding algorithm must be adapted to work with neighbourhood data. A technique based on edge growing has been implemented. Each pixel neighbourhood from the suppression module was thresholded twice according to an upper and lower threshold. The lower threshold bitmap was then compared to the upper. Any low thresholded pixels adjacent to upper thresholded pixels were marked for output at the next iteration. Overlaps between neighbourhoods were also implemented to ensure edge connection.

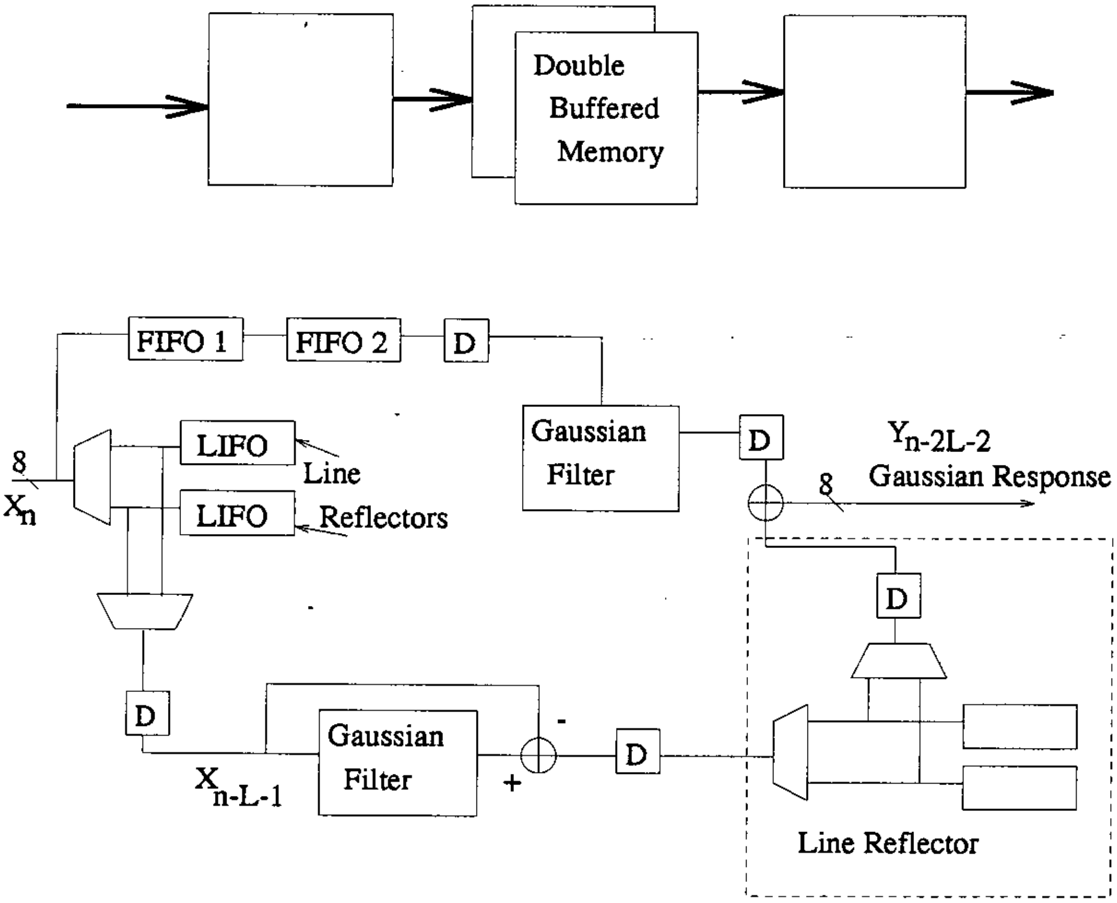


Figure 2–15: VLSI Separable Gaussian Implementation

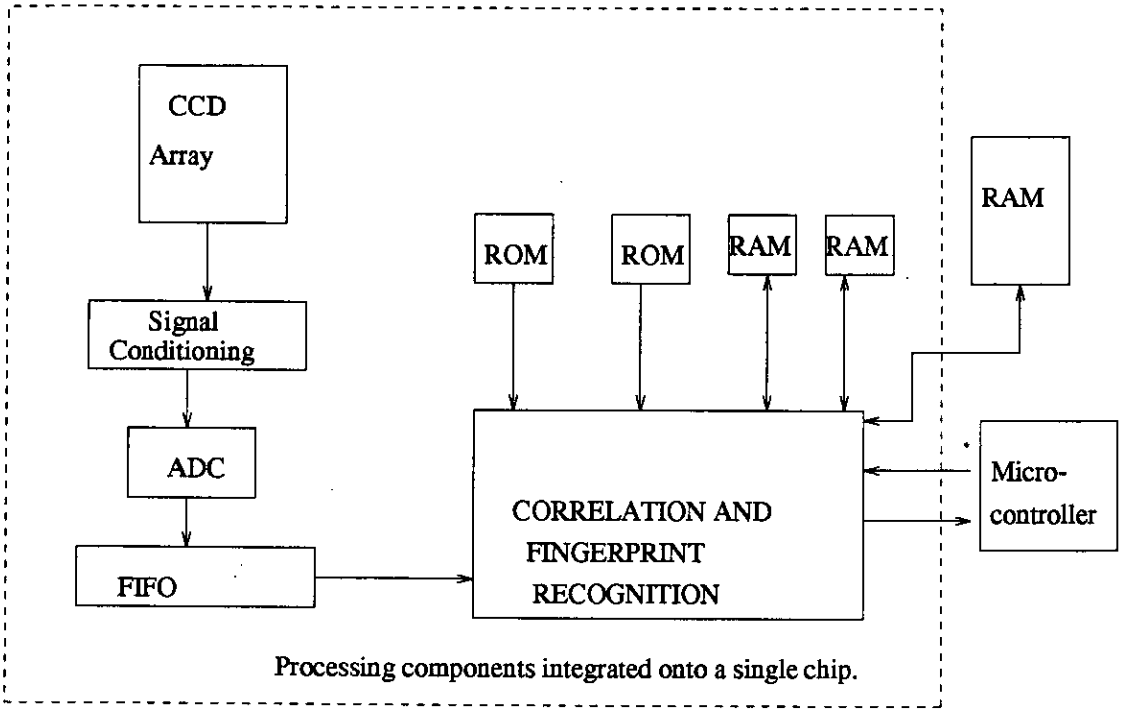


Figure 2–16: A Single Chip Image Sensor for Fingerprint Verification

The final design considered in this review is a finger print recognition system developed by Anderson et al. [1]. A block diagram of the system is shown in Figure 2–16. A particular feature of this design is the integration of the sensor onto the same substrate as the image processing functions. With the exception of the micro-controller and some RAM, all functions were integrated onto a silicon substrate. To avoid the full cost of an ADC a thresholding operation is combined with the ADC using a DAC and comparator. Possible thresholds are fed into the DAC and compared with the analogue output from the sensor. If the ratio of black to white pixels is wrong then the threshold is adjusted accordingly. Using this technique, the requirement for an expensive framestore is eliminated. The thresholding and normalisation function is shown in Figure 2–17.

Compared to other image processing functions relatively few VLSI designs have been developed specifically for stereo. Mahowald and Delbruck[50] implemented the Marr-Poggio algorithm whereas Hakkarainen[28] implemented the Marr-Poggio-Drumheller(MPD) [54] algorithm. As the Hakkarainen design was aimed at integration of processing and sensing on the same CCD chip, this is

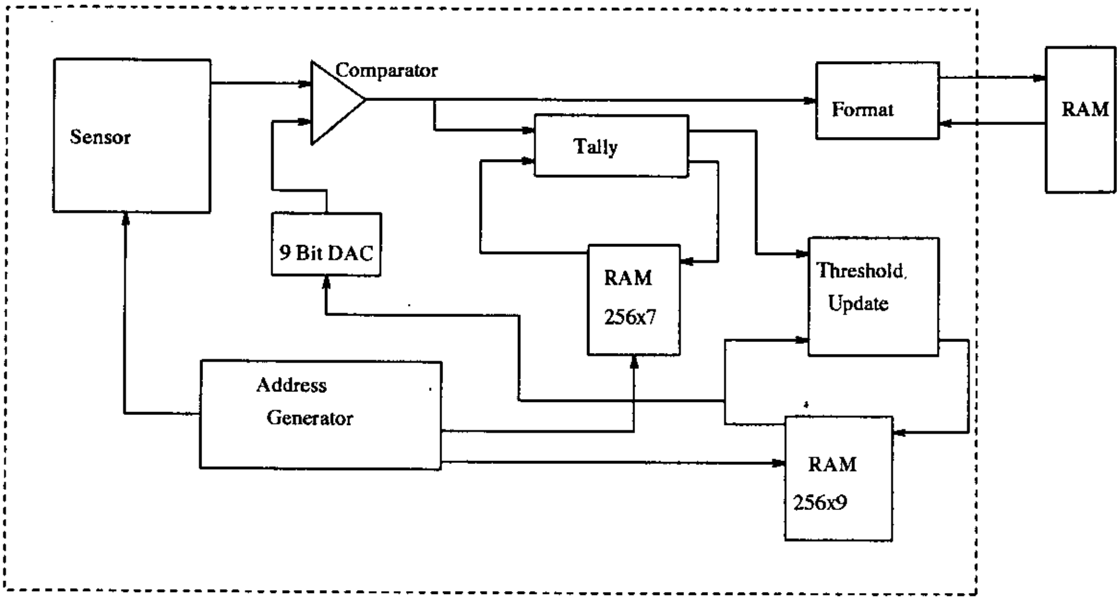


Figure 2–17: Combined Thresholding and Analogue to Digital Conversion

the one presented here. Figure 2–18 shows a block diagram of the algorithm and architecture.

Pre-processing in this algorithm implies the application of a Laplacian of Binomial (LoB) spatial bandpass filter masks. This is simply a smoothing filter followed by an edge detector operation. Hakkarainen did not implement the LoB function himself and other CCD analogue architectures from related work were suggested [46] as possible solutions to this problem. Thus although the architecture has been designed as a complete system, the only part fabricated specifically for this project was the match generator. This was tested by interfacing the AVD module to a computer and performing the LoB operation, the local support operation and the decision module in software.

A schematic of the AVD (Absolute-Value-Difference) module is shown in Figure 2–19. Data, or charge, from the pre-processed left and right images is entered into CCD shift registers (L_i, R_i) before corresponding rows are differenced in parallel. For each pixel, the result is inverted to provide a measure of similarity. Following this, the next set of candidate matches are calculated by shifting either input row along by one pixel. This is repeated for each possible disparity within a predefined range and the results, for each disparity, stored in a third shift register (A_i) before being read out. After this the candidate matches for each disparity shift and pixel strengths would be supplied to the, software implemented, local support module.

In Hakkarainen's results the local support module assumes that the disparities across a scene vary smoothly [7]. Using the local support module, each candidate match score is recalculated, by taking a weighted sum, from the neighbouring disparities and the results fed into a decision support module. The decision algorithm maintains the highest scores for each pixel together with its associated disparity. In a working system this would be continually updated as the best matches, for each pixel, were found.

Such a CCD implementation is efficient in area and would fit in well with current commercial sensors. However there are practical problems associated with this type of approach. As suggested by Skifstad and Jain [78] stereo matching becomes difficult, or impossible, for parts of an image where there is no luminosity gradient. However, in this work, the assumption is made that disparity varies smoothly over the image and, using a neighbourhood support scheme, an attempt is made to find a match at every pixel. The neighbourhood support is most likely to cause errors at areas of the image where there are sudden changes in disparity. These areas are likely to be edges. Thus this algorithm will provide poor performance in the areas of the image with the easiest matches, ie. large luminosity changes or edges. Despite this such an algorithm can have success in textured scenes. Also Hakkarainen's design shows that stereo algorithms can be implemented in analogue VLSI. The next section considers some other analogue image processing circuits.

[7] As is discussed in the next chapter this assumption is not always true.

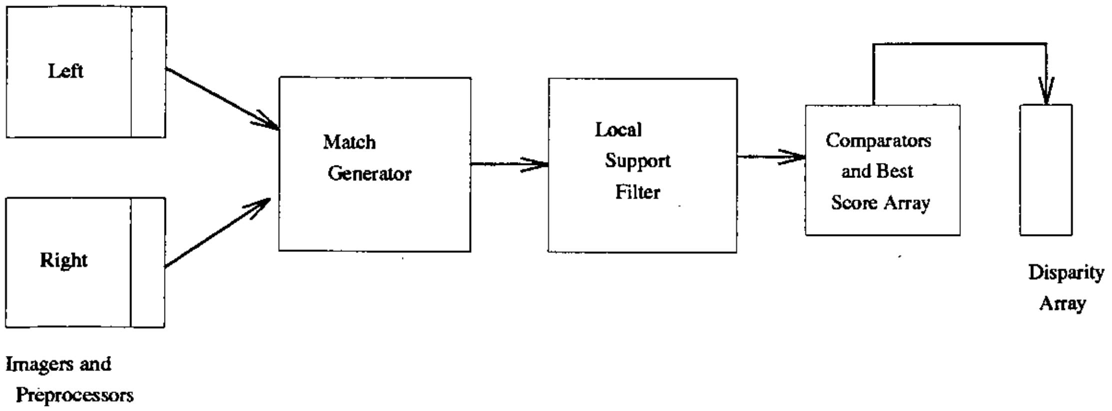


Figure 2–18: Block Diagram of CCD Stereo Architecture

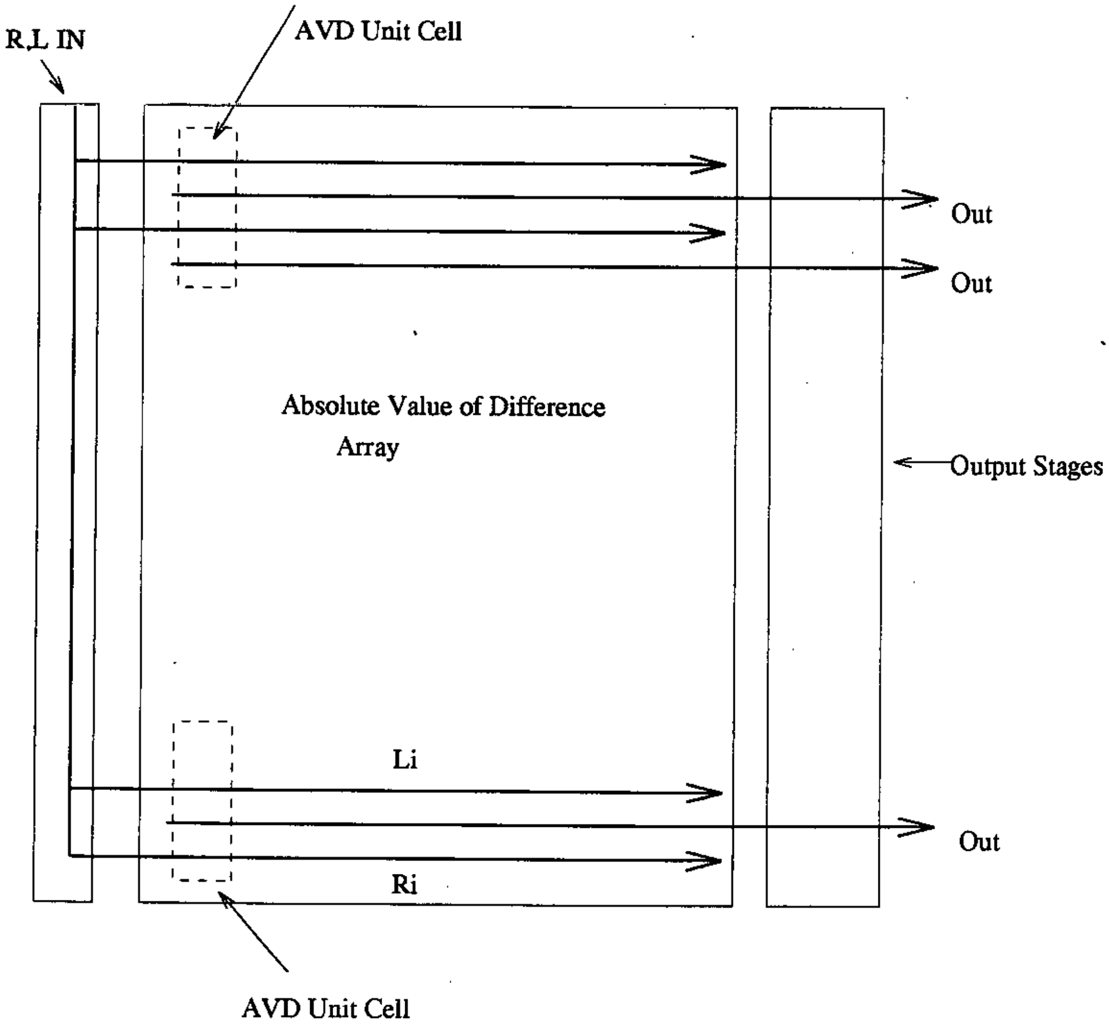


Figure 2–19: Schematic of Absolute Value Difference CCD Processor

Analogue Architectures

Horn [34] states that although parallel digital networks are ideal for research they are large and expensive and he proposed experimentation with analogue VLSI networks. Examples include edge detection, Gaussian and binomial filters and moment calculation. Several analogue chips have been built to implement these functions and moments calculation is discussed here as an example.

DeWeerth and Mead [19] designed an analogue chip to calculate the centre of mass of a thresholded object in the scene. The calculation is performed by noting that the inertia about an axis perpendicular to the image plane is minimised at the centre of mass. In effect we must minimise the value of Equation 2.11.

$$E = \int\int_D ((x-\overline{x})^2 + (y-\overline{y})^2)b(x,y)dxdy, \qquad (2.11)$$

where $b(x,y)$ is the thresholding function for each pixel. Finding the minimum of Equation 2.11 can be solved by finding the zeroes of the following two derivatives.

$$\frac{d\overline{x}}{dt} = \alpha \int\int_D (x-\overline{x})b(x,y)dxdy \qquad (2.12)$$

$$\frac{d\overline{y}}{dt} = \alpha \int\int_D (y-\overline{y})b(x,y)dxdy, \qquad (2.13)$$

where α is a gain factor which controls the speed of adjustment of estimates of $\overline{x}$ and $\overline{y}$.

The implementation proposed by DeWeerth and Mead [19], employs a bus for each of the above equations. The voltage on this bus is proportional to the current estimates $\overline{x}$ or $\overline{y}$. Every pixel injects, onto the bus, a current proportional to the difference between its x or y coordinate and the bus potential. This is only done if the pixel exceeds the threshold, $b(x,y)$. When equilibrium has been achieved the injected currents into the bus add up to zero and the voltages on the busses represent true estimates of $\overline{x}$ and $\overline{y}$.

2.5.4 Discussion of Image Processing Hardware

The above hardware survey is aimed at providing a broad overview of current research. It ranges from the general machines, such as the Transputer, to the application specific analogue and digital processors described above.

With the exception of the PPM machine, relatively few parallel architectures seem directed at solving lower level image processing problems, such as smoothing and feature extraction. Problems such as graph matching, model matching and perception have not been seriously studied in terms of hardware. A main reason for this is that these processes tend to be less well understood and are considered irregular. Of particular interest to this work is the implementation of stereo algorithms in CCD analogue VLSI.

In terms of the general processors, two characteristics appear to be common. Firstly all have on-chip memory to allow raster scan data to be assembled into 2-D neighbourhoods. Secondly the multi-function processors are nearly all designed to be cascadable. The processor can then be employed in a "dataflow" arrangement and its function altered with respect to the desired algorithm. Cascadable processors are at their most obvious in systolic architectures. The designers of these machines have emphasised the problems of I/O bandwidth across chip boundaries. They state the inefficiencies inherent in a design if one part of the overall architecture is faster than another. Thus computation should be balanced with I/O speed.

Analogue hardware was also covered and it is clear that there is potential for some very efficient implementations in a final system. The calculation of moments by DeWeerth and Mead [19] provide one example as does the combination of thresholding and analogue to digital conversion described by Anderson [1]. However problems still remain in image storage and in the combination of information from different parts of the scene.

2.6 Conclusions

This chapter started with a description of current research and practice in low level image processing covering the areas of edge detection, thresholding and segmentation. What is evident from these distinct areas is the interdependence, in a final system, of the different modules. For example, thresholds can be chosen from an edge detected image, edges can be built from segmentation techniques such as the Hough transform and, as will be shown in the next chapter, stereo vision errors can be dependent on extracted edges.

Such low level functions can be used to build up an imaging system for a particular application. For example, an estimation of the background and foreground parts of an image can be performed using threshold and segmentation techniques. A similar and, again, interlinked problem is that of grouping features into one object. Basic functions can be built into a larger system to provide a solution to this problem. The solution will very rarely be perfect due to unpredictable changes in the scene and it is important that all the available information is combined. This approach will be described in later chapters where a simplified edge detector is combined with a simplified thresholding function to extract only those edges which are necessary for the stereo matching algorithm. Combination of information in this manner can also reduce the overall computational cost and avoid the use of intensive algorithms such as the Hough transform and graph matching.

The last section of this chapter was a survey of current hardware techniques. It is clear from the survey that, in a practical CMOS/sensor implementation, the design should be conducted in the light of the following considerations:

1. The cost of floating point arithmetic.
2. The cost of low level processing compared to possible advantages to be had from using larger data structures. Thus a speedy transition from a pixel based description to a higher level segmented description of the image would reduce the data processing required at the more functionally complex stages of the system. This is also important in terms of

the I/O bandwidth problem described in the section on systolic arrays. If the sensor can be integrated on the same chip as early processing the problem of pin capacitances and drive circuits is largely resolved. Only significant features such as edges need be extracted from the chip.

3. Early image processing should be specialised to the application in mind. Savings can be made by altering the algorithm.

4. Memory is of considerable importance. All the processing arrays described above had some form of on-chip memory to arrange data in a spatial manner.

These considerations have been applied in the DETECT system described later where the algorithm has been adapted to be suitable for a CMOS/Sensor implementation.

Chapter 3

Stereo Vision Techniques

EX LIBR. UNIV. EDINBURGH

3.1 Introduction

This chapter is concerned with the examination of current stereo vision techniques and their application to the problem of tracking human beings using minimal hardware. Stereo vision can provide us with estimations of distance and size. Such estimates would be of considerable use in reducing the occurrence of false alarms.

An initial statement of general stereo geometric principles will be followed by a brief description of recent biological research and computational algorithms, designed to model the human system. Marr's matching constraints, which can be applied to solve what is commonly called the *correspondence problem*, are also described. Of particular interest is recent research [64] suggesting that the human eye only attempts matching for a limited number of points in the scene. In the work described in later chapters complete matching is not attempted for every edge feature in the scene. It seems pointless to do so and would generate an unnecessary computational load. Section 3.4 provides a review of current developments in stereo algorithms and discusses them in the light of possible VLSI implementations. This discussion will link with Chapters Four, Five and Six which describe stereo vision and image processing algorithms aimed at a specific application, ie., the detection and tracking of intruders. Later sections, of the chapter, will deal with calibration and error analysis, followed by some conclusions.

3.2 Geometric Principles

Depth estimates of a particular scene, feature or object can be based on spatially separated views, using a stationary camera, or temporally separated views when the camera or subject is moving. Estimates of distance are inversely proportional to the differences, or disparities, of separate views of the scene. This inverse nature of the problem has led to stereo being labelled "ill-conditioned". Small errors are amplified by the process of inversion. Such errors are considered in section 3.5.2.

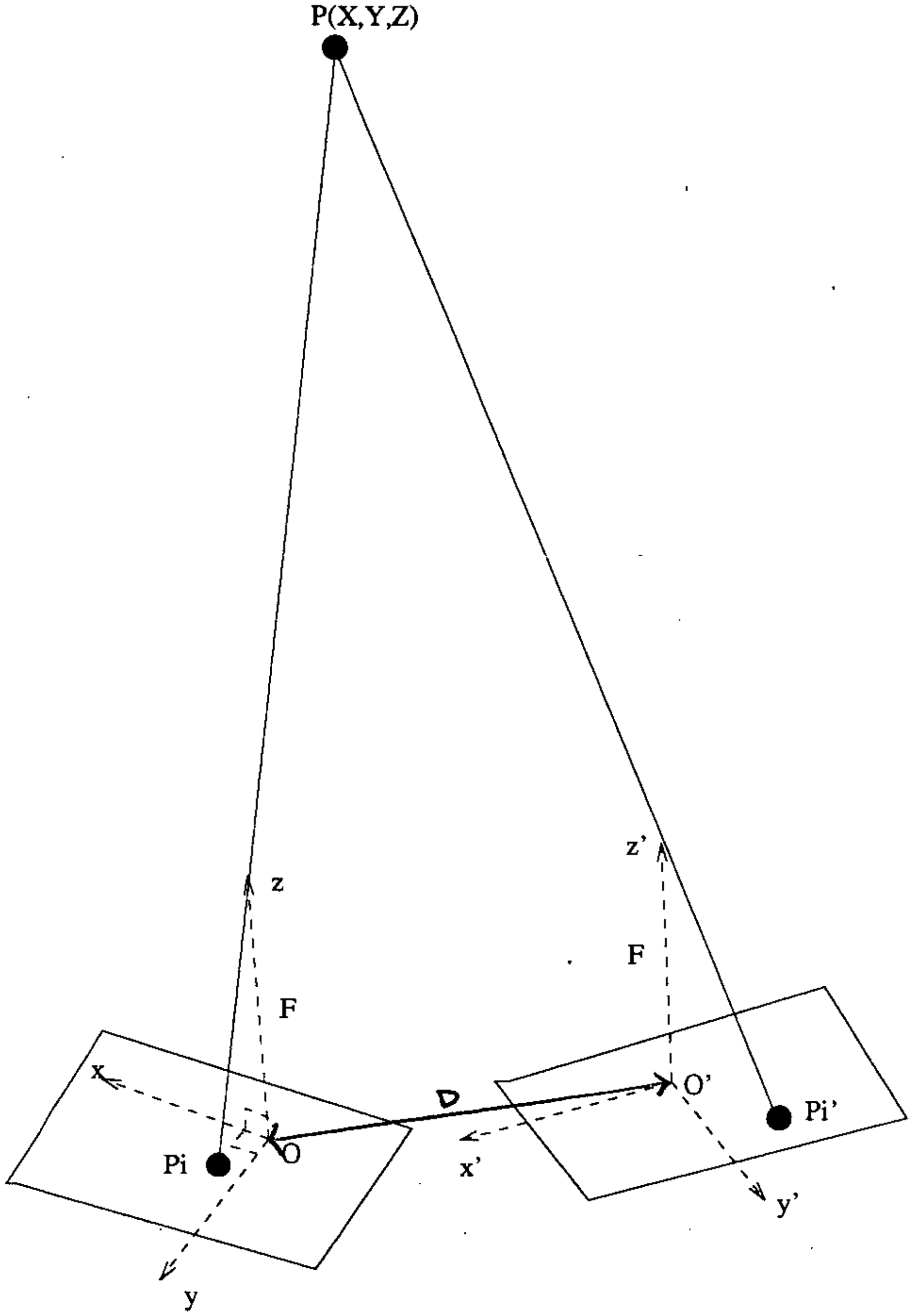


Figure 3–1: General Two Camera Stereo Rig

Figure 3–1 shows a generalised, two camera stereo rig. Knowing the interocular distance, D, focal length, F, pixel size, P and the transformation relating the two coordinate systems, the world coordinates of a point P can be found from the left and right camera views. The two coordinate systems can be represented in terms of the rotation, **R** and translation, **T**, shown in Equation 3.1.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathbf{T} \tag{3.1}$$

The above situation can be simplified such that cameras are laterally separated and on the same imaging plane, Figure 3–12. In this situation rasters from the two cameras will correspond and the epipolar [1] constraint applied. The following relationships can be defined. Disparity, δ, Equation 3.2, is defined as the difference in X position on each of the two imaging planes once their local coordinate origins are aligned.

$$\delta = \mid X_l - X_r \mid \tag{3.2}$$

where X_l and X_r are the left and right lateral feature positions on the imaging plane with respect to their local origins. We also get an equation relating depth, Z, and disparity, δ, with the product of the focal length, F, and the interocular distance, D.

$$Z = \frac{FD}{\delta} \tag{3.3}$$

Thus, disparity is inversely proportional to depth and can be used to estimate depth if a particular camera geometry is known and similar features in both images are known. The next section covers current ideas about how the human vision system solves this correspondence problem.

[1] The epipolar constraint simply means that searches for corresponding matches can be restricted to scans along the relevant rasters in the two images.

3.3 Biological Systems

The ability of the human eye to judge distance is remarkable and a digression to consider current theories of human correspondence is worthwhile. Much of stereo vision research has been conducted in artificial intelligence organisations where biological systems have provided the main inspiration for the most important theories. Marr gives a good summary in Vision [51] which describes researchers attempts to imitate the human eye's own stereo algorithm. The exact algorithm by which the eye solves the correspondence problem is not known. However, it would seem obvious that humans' memory and image understanding play important roles, together with explicit depth extraction from lateral or temporal motion.

It is known that the eye performs edge detection early on in an image's interpretation. It is also known that that the receptive fields of the eye consist of a central excitatory region surrounded by an inhibitory area. This is thought to result in a $\nabla^2 G$ operator being applied to incoming light. The $\nabla^2 G$ function is shown in Figure 3–2 and is obtained by taking the Laplacian of a Gaussian curve. Psychologists [15] believe that four of these $\nabla^2 G$ operators provide four spatial channels, at different scales, as an input to further processing.

Human stereo vision is thought to use the zero crossings as matching features. A coarse to fine strategy, starting from the largest spatial channel and proceeding downwards is then used to constrain matching and feature detection. Higher bandwidth channels are more detailed and, therefore, correspondence more problematic, whereas lower bandwidth channels will have poorer localisation of features but better matching.

Marr describes a computational theory of stereopsis and proposes three constraints in order to determine a unique correspondence between two images. These are:-

1. **Compatibility:** Obviously only items which arise from the same object and have similar features should be considered for matching.

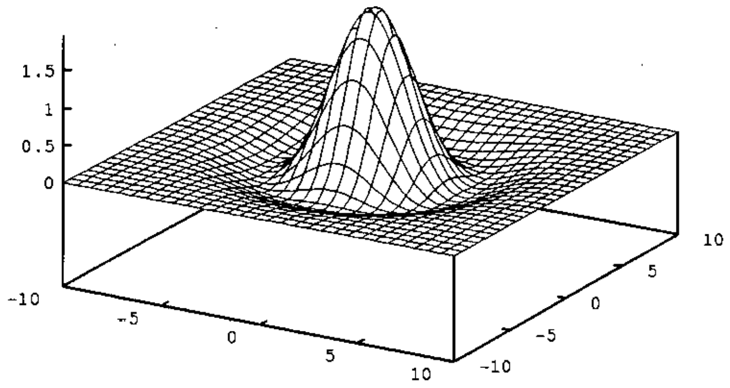


Figure 3–2: The $\nabla^2 G$ Operator

2. **Uniqueness:** A feature from one image can only match *one* feature from the other image.

3. **Continuity:** The disparity of correct matches should vary smoothly over the majority of the image. Obviously this will not apply everywhere, but as objects are usually continuous it is most likely that its depth will be continuous.

3.4 Stereo Algorithms

Based upon the above, Marr described two algorithms which satisfy the above constraints and are therefore possible imitations of human stereopsis. The first is a cooperative algorithm.

3.4.1 Marr's Cooperative Algorithm

The following description refers to Figure 3–3 which shows a binary array with excitatory connections along the diagonals, and inhibitory connections in the vertical and horizontal directions. Each diagonal line represents a particular disparity with the central positive diagonal being zero. Binary features from the corresponding epipolar rows of a left and right camera are extracted. The disparity array is initialised by setting individual bits to one, if the corresponding pixels in the left and right epipolar lines are both one, and to zero for all other combinations. A correspondence is said to be found if a disparity array bit is one. Clearly the initialisation of the array will generate a number of false matches, violating the uniqueness constraint. False matches are eliminated by applying the continuity constraint. Each bit in the disparity array is updated by summing the weighted bits from the surrounding neighbourhood. Weights in the positive diagonal direction will excite, whereas all others will inhibit. Further to this, a threshold function was applied to the neighbourhood sum to define whether an individual pixel should be one or zero. Over a number of time-steps, parts of the epipolar line with similar disparities, and which are near each other, will reinforce and reduce the number of false matches. Convergence is achieved when the difference between successive time steps is reduced to an acceptable minimum.

Variations on the above approach are possible. Instead of having an on-off binary array, accumulators could be employed at every possible disparity match. In this situation it would be important to ensure that the system is stable and that both the inhibition and excitation are within reasonable limits. Hardware implementations have been proposed to implement the above algorithms and an

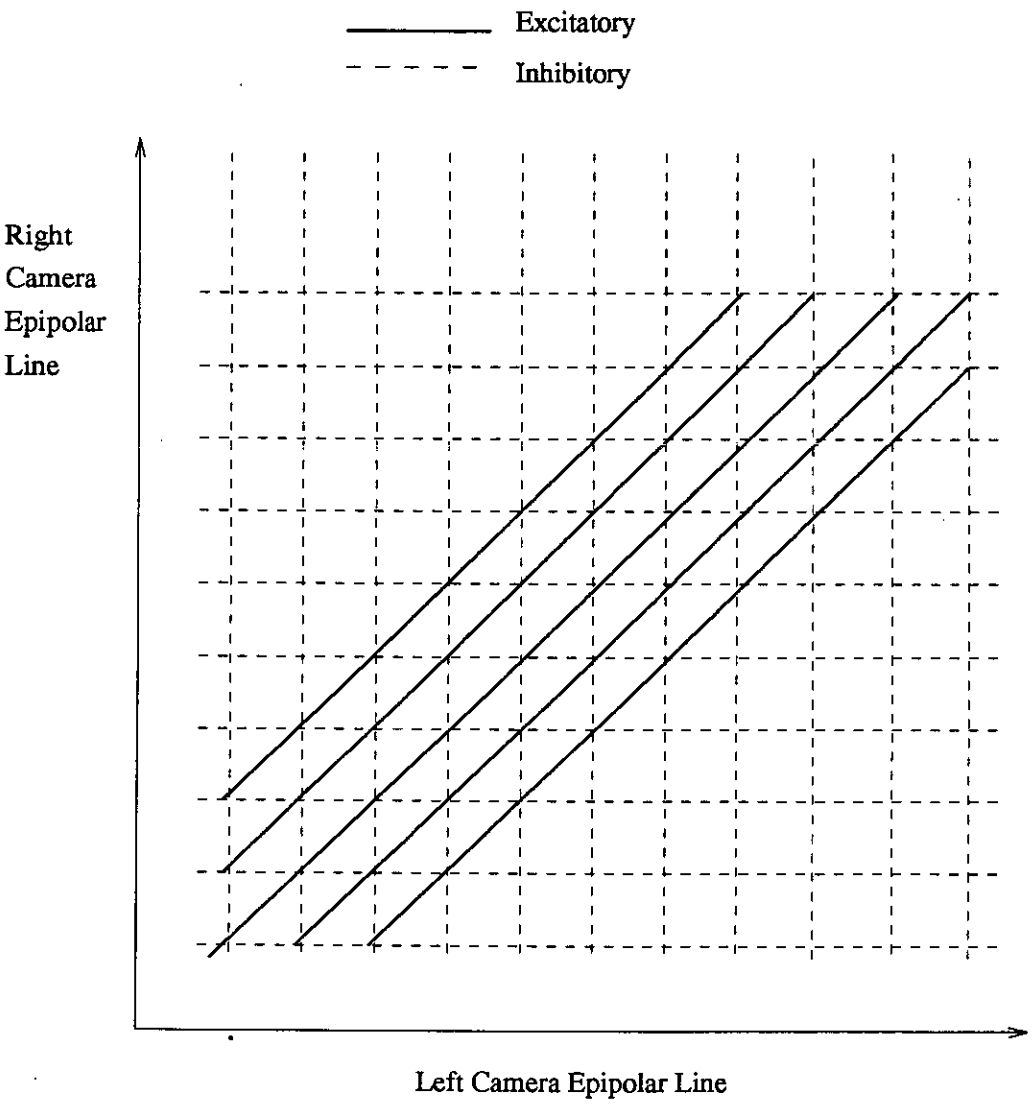


Figure 3–3: Marr's Cooperative Algorithm

analogue network, with connections implemented using variable resistances has been described by M.A. Mahowald and T. Delbruk [50].

3.4.2 Marr's Biological Algorithm

A second algorithm, suggested by Marr and Poggio [53] and implemented by Grimson[27], is based directly on the limited knowledge of the human matching system: ie., a four channel coarse to fine strategy. As described above, zero crossings, with their sign and orientation, are extracted from the output of a $\nabla^2 G$ operator at different spatial resolutions. Crossings from the two separated views are matchable if their signs are similar and their orientations within 30 degrees. This was done across the filtered image on a pixel by pixel basis. The coarsely filtered images are matched first, reducing the number of potential matches and the disparity search range in the higher bandpass filters is restricted according to the results from coarser channels. Thus, after successive filters and matches have been applied, reasonable edge localisation and match accuracy can be achieved.

3.4.3 Other Approaches and Constraints

There are many other approaches to finding correct disparity matches [6] [42] [32] [70] [62]. Often a brute force correlation technique has been utilised where a grey level patch from one image is compared with successive patches from the other image. However, problems arise with the size of correlation block and from luminosity variations between different camera views. Often attempts will be made to match blocks where there are minimal luminosity variations. As human snow blindness shows, it is impossible to extract disparities from featureless information [78].

Another question, which often arises, is whether depth information should be calculated before or after recognition has been performed. Clearly, knowledge about the shape and structure of the object under consideration would be of advantage in both reducing the number of false matches and increasing the accuracy of the measurement. As the problem of recognition and model matching is not being considered in this work, we will confine the discussion, in this and coming chapters, to general matching principles which can be applied without recognition.

The most fundamental feature which can be used for matching is the edge as this represents a change in luminosity. Further to this, is the fact that edge strengths are relative. It is more likely that edge strengths from two views of the same scene will be similar. This is not the case when one considers absolute luminosities. The next section discusses the possibilities of edge based matching.

3.4.4 Edge Based Methods

Edges have several parameters which can be used in matching:

1. **Strength:** The usual understanding of an edge in a grey level image is that of a sharp change in luminosity gradient. If the two views are reasonably close then corresponding edges should have luminosity gradients of the same order. Edges should also be of the same polarity, ie. a positive gradient edge should not match a negative gradient edge.
2. **Direction:** Spatial direction has been used as a constraint. For example, Grimson[27] eliminated any candidate matches if they diverged by more than 30 degrees. The threshold angle will be, to some extent, dependent on the camera geometry.
3. **Position:** Scene knowledge and camera parameters can be used to restrict the image area in which candidate matches will be considered. In effect maximum and minimum disparity limits can be applied to reduce the chances of false matches.
4. **Length:** If the focal lengths of the two cameras are equal and are at approximately the same depth, then the features should be of similar size. It is unlikely that size can be used when matching lower level features, due to fragmentation of the initial segmentation.
5. **Disparity Gradient:** The concept of disparity gradient, $\nabla\delta$, is taken from experiments which appear to indicate that humans find difficulty

in matching features where $\nabla\delta > 1$. In computer vision the disparity gradient has proved to be of considerable use in discriminating between correct and false matches. Burt and Julesz [65] defined the disparity gradient, between two points, as the difference in their disparities divided by their separation in distance. Figure 3–4 illustrates. For the upper matching feature, the shape and, therefore, disparity is constant, as the edge is tracked from top to bottom.

$$\delta_1 = \delta_2 = \delta_3 \tag{3.4}$$

This edge would be accepted as a correct match. The second feature would probably fail a disparity threshold test and be eliminated as an incorrect match. The quality of a point to point match can therefore be. determined by both the surrounding matches and changes in disparity. In addition, disparity gradient thresholds can be used to segment an edge into separate components. Any sudden jumps in disparity of an edge are likely to indicate the end of one feature and the start of another.

3.4.5 The PMF Algorithm

Pollard, Mayhew and Frisby [76] have developed a stereo vision system within the Sheffield TINA environment[67]. This algorithm directly applies a disparity gradient limit to the matching problem. Figure 3–5 shows the algorithmic details.

Potential matches from the feature maps are initially selected according to the epipolar constraint, the sign and edgel strength. An initial strength is computed using the product of the two candidate pixels. The strongest matches can then be proposed as seeds from which other matches can be derived. Starting with edge pixels, from the left image, all potential matches along the epipolar line in the right image are considered. For each potential match, support within a circular area is calculated to provide an estimate of match strength. The support is weighted according to the disparity gradient with the central match. Those initial matches

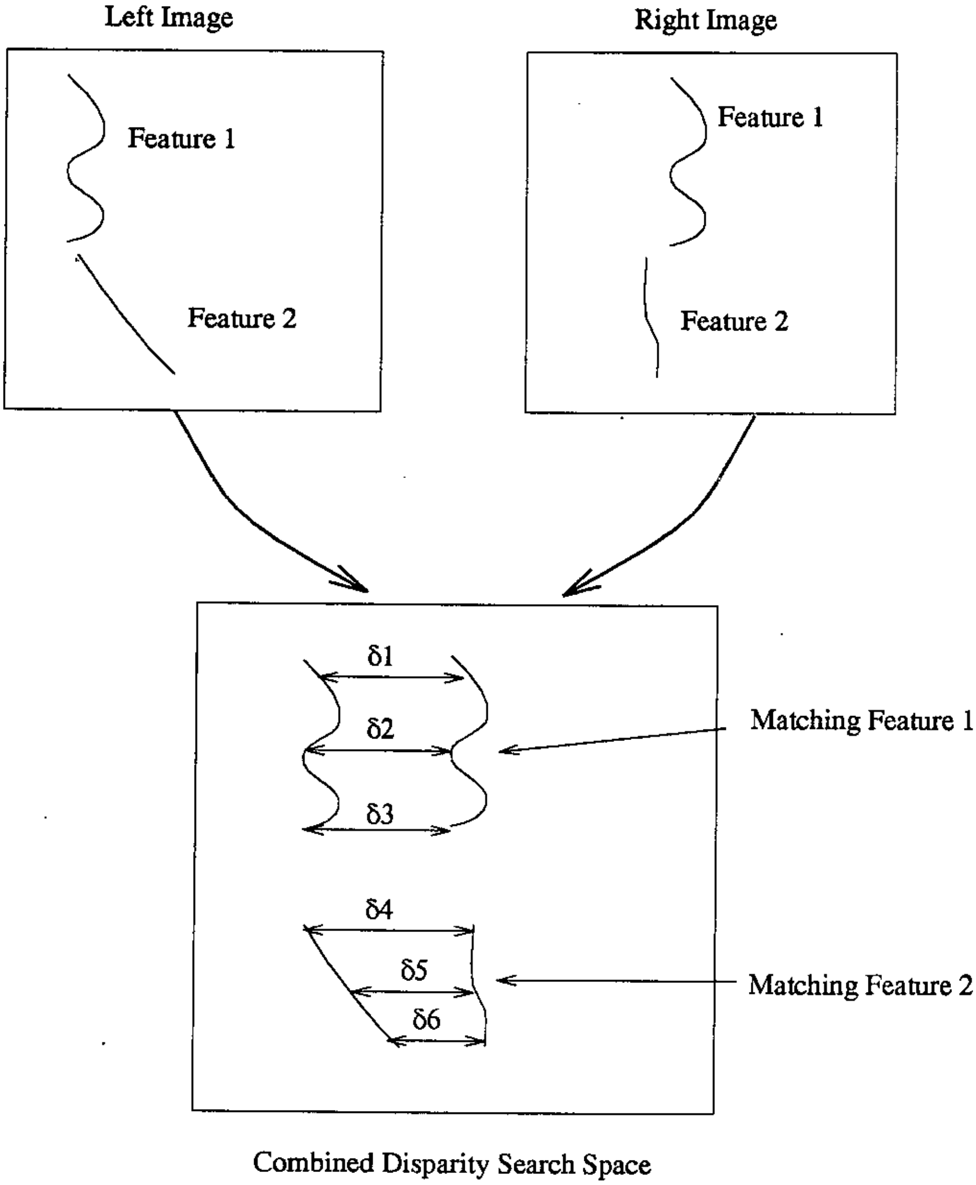


Figure 3–4: The Disparity Gradient

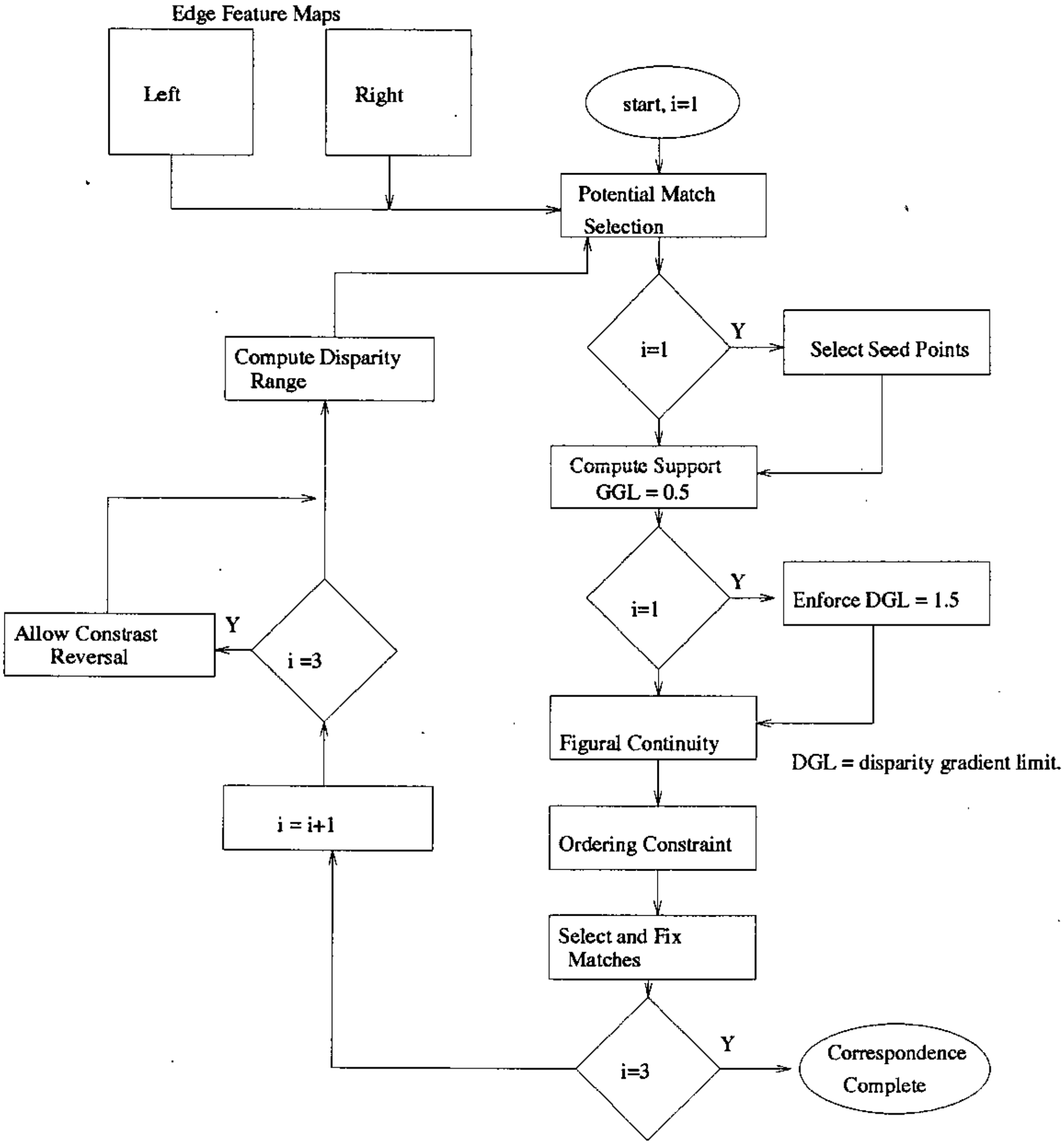


Figure 3–5: The PMF Algorithm

which have sufficient support are proposed as input for the next iteration.. The cycle continues three times.

3.4.6 Phase Based Stereo

Another way of considering disparity is to represent it as a phase difference. Several authors [44][20] have proposed disparity extraction techniques using phase difference. As with correlation techniques, a dense disparity map, for the entire scene, can be extracted. Phase differences can be calculated, in the frequency domain, from the output of multiple bandpass filtered images. The technique is illustrated, for the one dimensional case, in Figure 3–6. Frequency dependent phase information is calculated for each spatial channel and for both the left and right images. The difference in phase, between the left and right images, may be used as an estimate of disparity. Often several possible disparities will be extracted and further constraints such as ordering and the disparity gradient can then be applied.

Using the notation presented by Jenkin [43], perfectly filtered left and right luminosity images can be represented, in the x dimension, by Equations 3.5.

$$I_l = Asin(\omega_l x + \theta_l) \quad I_r = Bsin(\omega_r x + \theta_r) \tag{3.5}$$

where A and B are amplitudes, ω_i frequencies and θ_i phase angles. In phase based disparity estimation, the assumption, that $\omega_l = \omega_r$, is made. Equation 3.6 indicates the disparity $d(x)$ as a function of the phase difference ϕ.

$$d(x) = \frac{1}{\bar{\omega}}\phi(x) \tag{3.6}$$

where

$$\bar{\omega} = \frac{1}{2}(\omega_l + \omega_r) \tag{3.7}$$

and

$$\phi(x) = (\omega_l - \omega_r)x + (\theta_l - \theta_r) \tag{3.8}$$

Using the above representation phase information can be extracted from the Fourier Transform and the difference between the left and right images calculated.

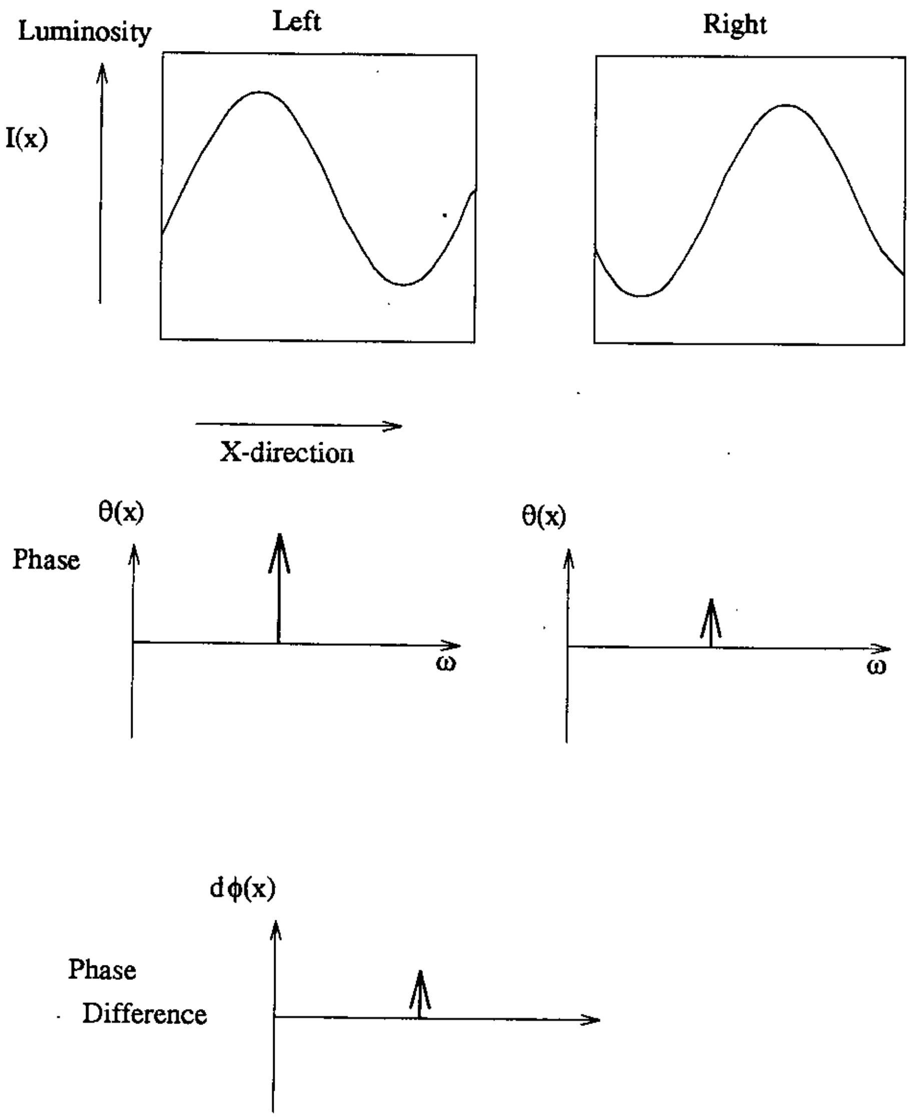


Figure 3–6: Disparity Estimation Using Phase Differences

These techniques have not been applied in this work due to the computational complexity required to implement each spatial filter and the subsequent Fourier transforms.

3.4.7 Neural Algorithms

Considering the current interest in neural networks it is surprising that there is not more published work on the explicit application of neural networks to the stereo vision problem. Perhaps this is due to the conceptual similarities between the cooperation algorithms, one of which was discussed in section 3.4.1, and neural networks. Both involve processing elements, which sum inputs, and both use connections which can be inhibitory or excitatory. An alternative approach, described by Nasrabadi and Choo [60], calculates correspondence, on edge elements, using a Hopfield network to optimise the solution.

The initial features are extracted using a Moravec operator [58] and Marr's constraints represented as a cost function to be minimised. Figure 3–7 represents an N_l x N_r array of neurons where N_l and N_r are the *total* number of interesting points in the left and right images respectively. Neurons are on or off, indicating the possibilities of matches between the left and right images. Thus a suitable initialisation of the network would be setting all possible matches along an epipolar line, and within a certain disparity, to one. The network update functions, described by Nasrabadi, are:

$$V_{ik} \rightarrow 0 \; if - \left[\sum_{j=1}^{N_l}\sum_{l=1}^{N_r}(C_{ikjl} - \delta_{ij} - \delta_{kl})V_{jl} + 2\right] < 0 \tag{3.9}$$

$$V_{ik} \rightarrow 1 \; if - \left[\sum_{j=1}^{N_l}\sum_{l=1}^{N_r}(C_{ikjl} - \delta_{ij} - \delta_{kl})V_{jl} + 2\right] > 0 \tag{3.10}$$

$$no\ change\ if - \left[\sum_{j=1}^{N_l}\sum_{l=1}^{N_r}(C_{ikjl} - \delta_{ij} - \delta_{kl})V_{jl} + 2\right] = 0 \tag{3.11}$$

where

$$C_{ikjl} = \frac{2}{[1 + e^{\lambda(X-\theta)}]} - 1 \tag{3.12}$$

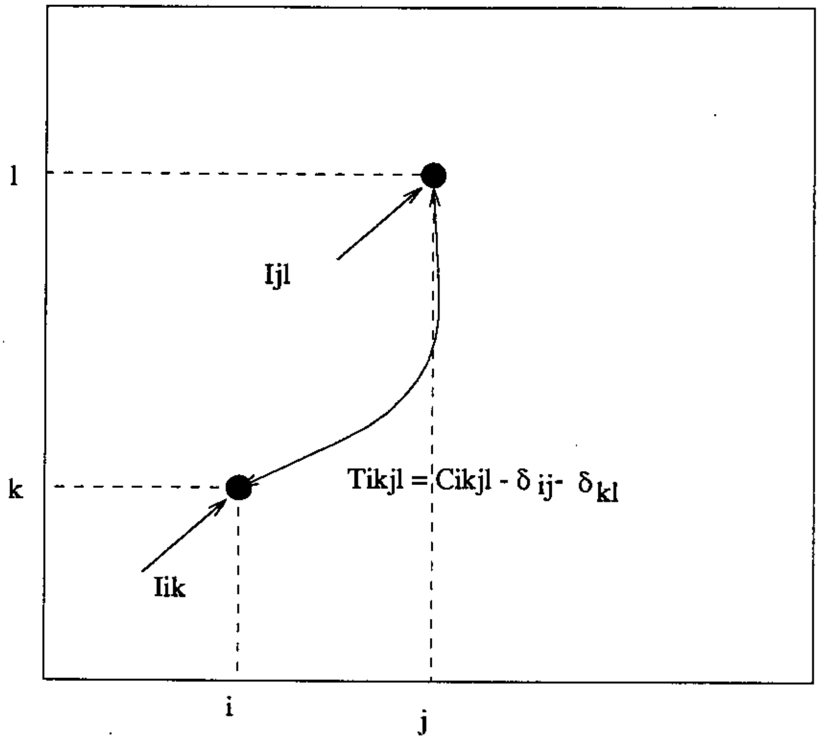


Figure 3–7: Disparity Estimation Using a Hopfield Network

and

$$X = [W_1\,|\Delta d| + W_2\,|\Delta D|] \tag{3.13}$$

C_{ijkl} is a measure of compatibility between features. A graph of C_{ijkl} is shown in Figure 3–8. Its value varies between -1, a poor match, and +1, a good match. In Equation 3.12 the values of δ_{ij} and δ_{kl} are used to prevent correspondences between impossible matches. Thus δ_{ij} is 1 when i = j and 0 for all other combinations of i and j. λ is a scaling factor and θ controls where the function crosses the X-axis. Δd is the difference in the disparities of the matched pairs (i,k) and (j,k) and ΔD is the difference between the distances i to j and from k to l. W_1 and W_2 are constant weight factors and satisfy the relationship $W_1 + W_2 = 1$. Experimental values for all the above parameters can be found in the paper.

The network can be run by random update of individual neurons until the change in energy becomes minimal. Obviously, local minima problems can arise and these are exaggerated by the binary nature of the network. A continuous network would reduce this problem at the expense of computational complexity.

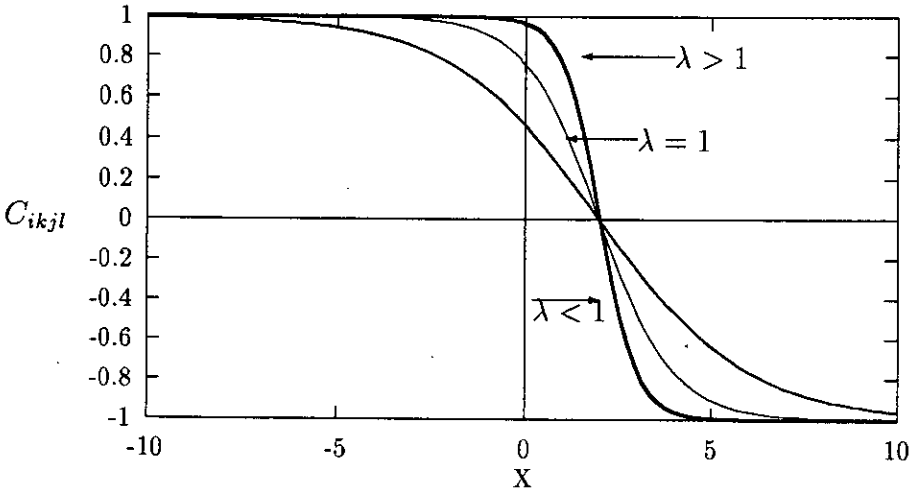


Figure 3–8: The Compatability Function

The authors present results for the network applied to several images for which the network took 1000 iterations to find the correspondences. Again, for computational reasons, the above neural approach is unsuited to the objectives of this thesis.

3.5 Error Analysis

An important part of stereo matching is an analysis of the probability of errors. The types of error which can occur in a stereo rig can be described as follows.

1. Stereo mismatching and false edge extraction.
2. Pixel quantisation noise.
3. Aliasing noise.
4. Digitisation noise.
5. Physical camera misalignment and lens distortion.
6. Grey level noise.

The above list gives an indication of possible sources of error. These, with the exception of stereo matching, are present to a greater or lesser degree in every image processing system. This section will not concern itself with the the last two items due to their physical nature. The noise caused by converting an analogue signal to a digital signal depends on the quality of the ADC circuit and the variability of the sampling clock. Although not discussed here, camera misalignment is also inevitable. Sections 3.6 and 5.6 discuss techniques to calculate an unknown camera geometry.

3.5.1 Stereo Mismatching and False Edge Extraction

Correspondence must be solved before depth can be extracted from a stereo vision system. It is worth considering the types of errors which can occur when matching two pictures. For the purposes of this section, it is assumed that the epipolar constraint is satisfied and that we have a perfect pin hole camera. We also assume that edge elements are the features being used for matching.

Mohan et. al [56] classify incorrect matches into the two catagories:

1. **Type 1 (local) errors:** Figure 3–9 shows correct pixel matches between segments AB and CD. Also shown are incorrect matches to two other segments. There are more correct matches between AB and CD than false matches to any other segments. These types of error can be corrected on the basis of figural continuity. [2]

2. **Type 2 (global) errors:** Figure 3–10 shows an erroneous match for which it is impossible to correct. Here there are more matches to the wrong segment EF than to the correct segment CD. Type 2 errors cannot be detected or corrected, whereas type 1 errors can.

Having defined the extent to which matching errors can be both corrected and detected we now discuss the likely causes of errors. Edge detection will never be

[2] Figural continuity simply implies that the edge CD is connected and not split into several elements.

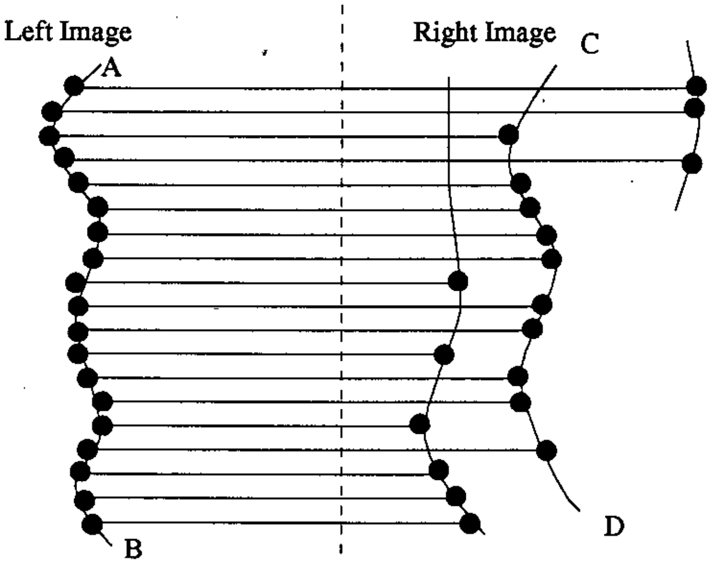


Figure 3–9: Type 1 (local) Errors

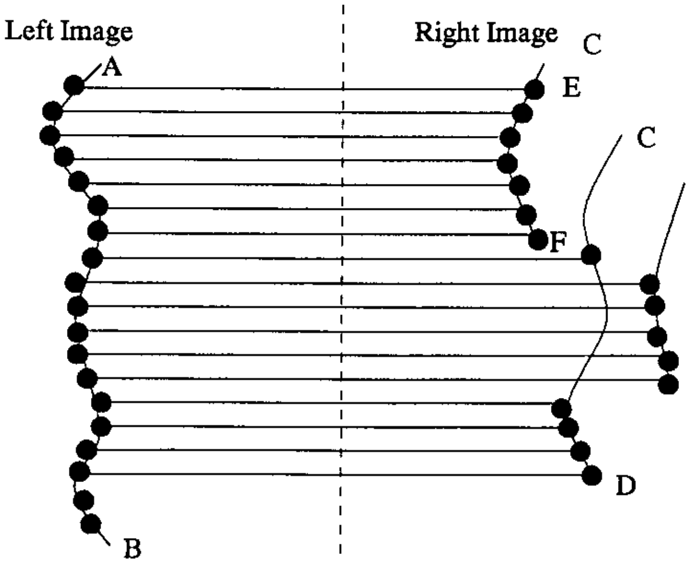


Figure 3–10: Type 2 (global) Errors

perfect and streaking may cause problems by breaking the edges into insignificant segments. A detector may simply generate spurious features. Clearly, these errors are dependent on the initial segmentation and the luminosity in the scene. Other errors can be caused by the physical structure of the scene in relation to the two cameras. Occlusion is the most obvious where an edge in one view of a scene is also not present in the other. More subtle problems can occur with transparent objects and with reflections and shadows. Finally the correspondence technique may simply make mistakes.

The analysis of the likelihood of error for a correspondence technique will depend on the algorithm itself. However, a simple technique, proposed by Mohan, to calculate a percentage error is to count the number of Type 1 corrections that require to be made. This will give a rough indicator of erroneous mismatches without having to work out by hand, or by an active matching algorithm, the true correspondences.

Further to the above, Thacker and Courtney [79] published a technique to estimate errors for a specific corner matching algorithm. The details of this analysis are specific to the particular matching algorithm proposed in the paper and are therefore not repeated here. Thacker rightly criticises empirical approaches, employed to estimate error, as data dependent. However, his approach assumes that matching errors are independent of the detection process. This is clearly not the case in the algorithm described later in this thesis and is not a valid assumption for many other stereo techniques. As suggested by Thacker, current comparison methods for different stereo algorithms are unsatisfactory and require more research.

As an alternative to the mathematical approach experimental results can be extracted over a range of input images from various scenes. An empirical technique which estimates the combined likelihood of error, from both the matching and feature detection algorithms, described in this thesis, is presented in Chapter Six. This allows a confidence to be assigned to a particular measurement.

3.5.2 Geometric Errors

This section will confine itself to discussing the fundamental types of error likely from the camera geometry. Other sources of geometric error such as camera misalignment will be discussed in Section 3.6. The two sources of error considered here, pixel quantisation and aliasing, are fundamental to the imaging sensor and therefore hard to correct, using calibration techniques. Referring to Figure 3–11, and assuming perspective projection, pixel quantisation and aliasing are dependent on the following parameters:

1. Pixel size, P
2. Feature depth and position in the scene, Z
3. Focal Length, F

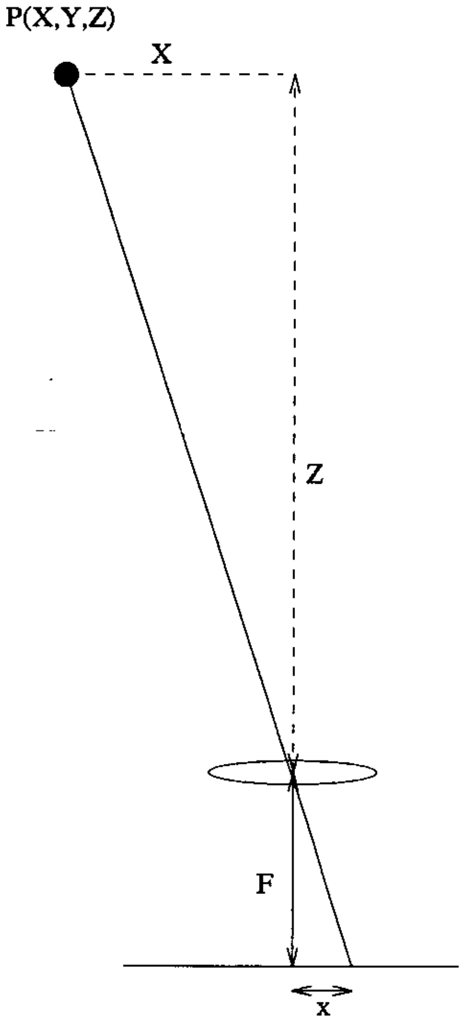
Similar triangles from Figure 3–11 gives us Equation 3.14 and Equation 3.15, where $\hat{X}$ is the measured value of X such that $\hat{X} = X + \Delta X$ and Δx the quantisation noise.

$$\frac{\hat{X}}{Z} = \frac{(x + \Delta x)}{F}. \tag{3.14}$$

$$\frac{X}{Z} = \frac{x}{F}. \tag{3.15}$$

$$\Delta X = \frac{Z \Delta x}{F} \tag{3.16}$$

Equation 3.16, derived from Equations 3.14 and 3.15, shows how ΔX varies with distance and localisation error, Δx. It is clear that Δx increases with depth and decreases with focal length. Thus, to reduce the effects of pixel quantisation, we require to have a long focal length and restrict the maximum allowed distance from the camera. This situation arises in most of the current imaging applications such as industrial work benches and robotics. However it is *not* the case for alarm systems, where large distances are common place and wide angle lenses, ie. short focal lengths, are important. We will now discuss the effects of quantisation noise on stereo measurements.



Sensor with pixel width P.

Figure 3–11: An Ideal Pinhole Camera

Stereo Errors

Figure 3–12 shows an idealised stereo rig in which the imaging planes are parallel, the focal lengths are equal, epipolar lines coincide and perspective pin-hole projection is assumed. The disparity (δ) is defined as the difference between the x positions with respect to local origins as in Equation 3.17.

$$\delta = x_l - x_r \tag{3.17}$$

From this equation, the measured disparity, $\hat{\delta}$, with quantisation noise is shown in Equation 3.18.

$$\hat{\delta} = x_l + \Delta x_l - x_r - \Delta x_r \tag{3.18}$$

We also define the disparity error, $\Delta\delta$, as $\hat{\delta} - \delta$. Thus the limits on $\Delta\delta$ are $\pm P$. This gives us an absolute value on the error, the significance, of which, depends on the value of the disparity. A relative error, ΔR, can be defined as in Equation 3.19. The relative error is inversely proportional to disparity.

$$\Delta R = \frac{\mid \Delta\delta \mid}{\mid \delta \mid} \tag{3.19}$$

The next problem is to estimate how disparity error affects estimates of depth information. Equation 3.22, derived from Equations 3.20 and 3.21, shows that the range error is inversely related to the product, DF.

$$\Delta z = \hat{z} - z \tag{3.20}$$

$$z = \frac{DF}{\delta} \tag{3.21}$$

$$\Delta z = \frac{-z^2 \Delta\delta}{DF + z\Delta\delta} \tag{3.22}$$

Equation 3.22 still does not provide an estimate of how error varies with respect to distance. For this we need a measure of relative error, $\frac{|\Delta z|}{|z|}$. Thus, we now have Equations 3.23 and 3.24 in terms of depth and disparity respectively.

$$\mid \frac{\Delta z}{z} \mid = \frac{z\Delta\delta}{DF + z\Delta\delta} \tag{3.23}$$

$$\mid \frac{\Delta z}{z} \mid = \frac{\Delta\delta}{\delta + \Delta\delta} \tag{3.24}$$

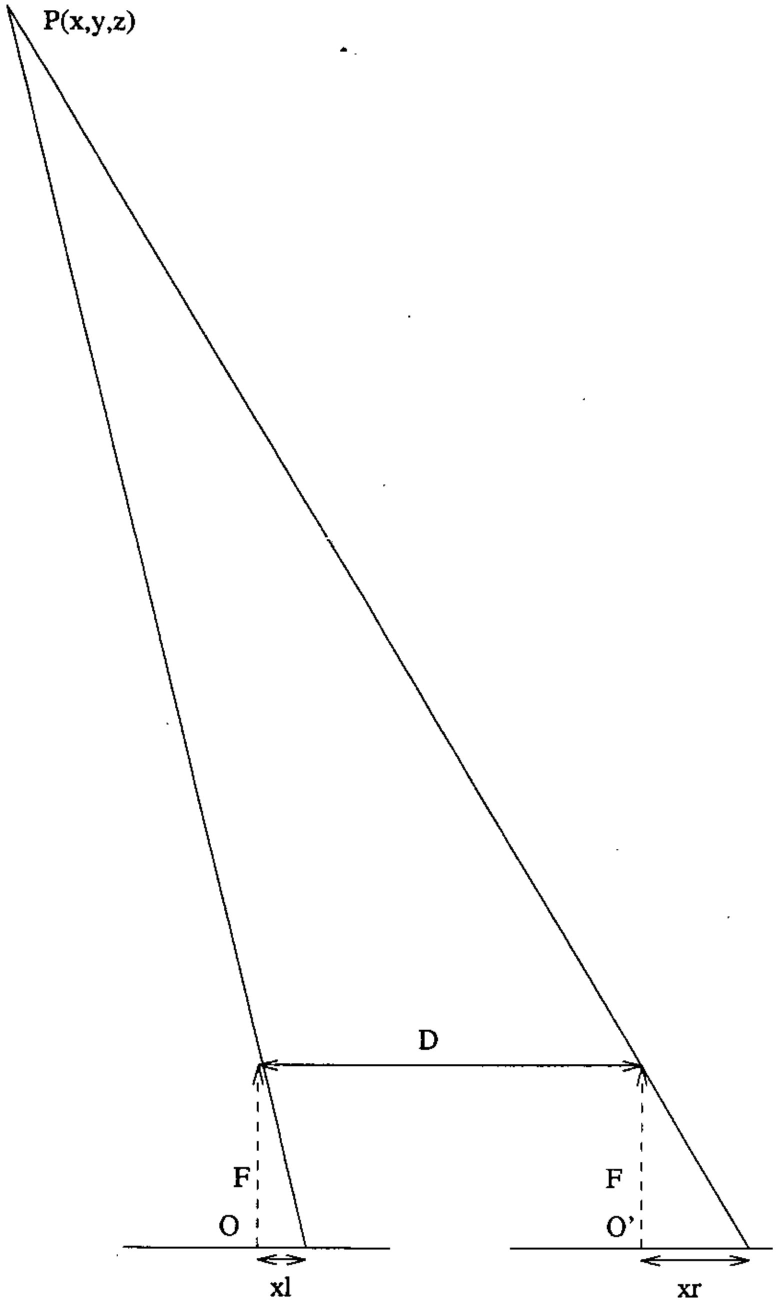


Figure 3–12: An Ideal Stereo Camera

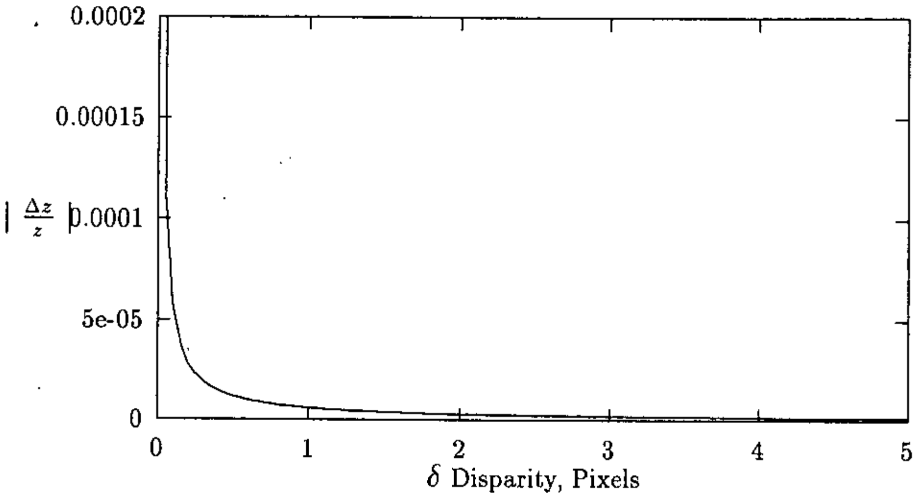


Figure 3–13: Relative Z Error Against Disparity

The relative error, in depth, plotted against disparity and distance are shown in Figures 3–13 and 3–14, respectively. For both graphs a pixel size of 20 μm is assumed. Thus if the luminosity gradient across the pixel is 1 then the expected disparity quantisation noise, Δx, is $\frac{20 \times 10^{-6}}{\sqrt{3}}$. The focal length was 14mm and the interocular distance 10cm.

As an extension to the above Blostein and Huang [8] have derived an equation giving the probability of a depth error (ϵ_z) being less than a specified tolerance (τ_z). This is is repeated here in Equation 3.25.

$$\hat{P}_{|\epsilon_z|}(| \epsilon_z < \tau_z |) = \begin{cases} 1 - (1 - \tau_z \delta)^2 & \tau_z < \frac{1}{\delta} \\ 1 & \tau_z \geq \frac{1}{\delta} \end{cases} \tag{3.25}$$

Equation 3.25 is accurate as long as the disparity is greater than some small number of pixels. Unfortunately this is not necessarily the case in a wide angle lens alarm system.

Equations 3.23 and 3.24 provide estimates of percentage error for a particular depth or disparity. Equation 3.25 provides us with a method of specifying a desired depth tolerance and calculating the probabilities of error for various disparities. What has not been defined is a probability density function or a measure of error

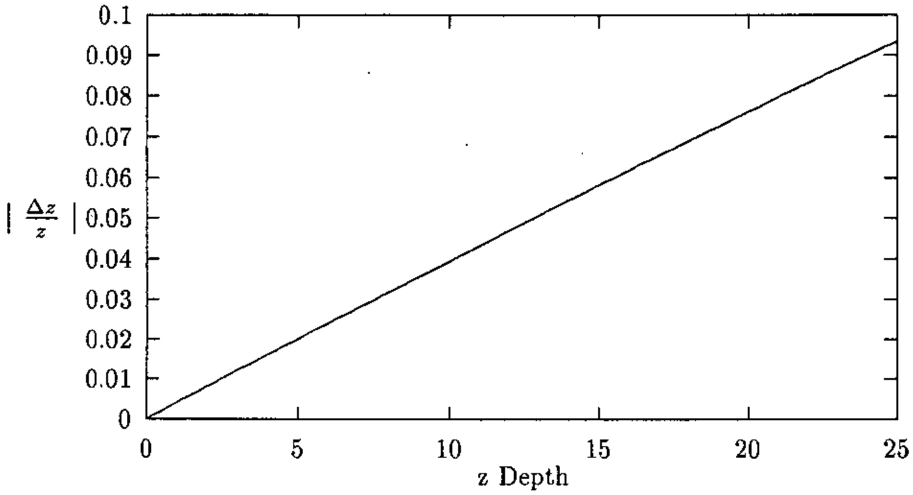


Figure 3–14: Relative Z Error Against Depth

within a defined depth band. Figure 3–15 shows the PDFs, for two and three camera rigs, for Δz within a specified range. The error is calculated by randomly generating points in a scene and projecting them onto a simulated stereo rig. The depth is then recalculated. The difference between the recalculated depth and the true depth is the error. Obviously, the errors are likely to be less for a three camera stereo rig. Due to pixel quantisation, this is especially important for the short focal length lenses likely in an alarm system. As a result, the experiments in Chapter Six are conducted for a three camera rig. Such an analysis would also be useful in estimating the likelihood of error in any final alarm system or installation.

3.5.3 Discussion of Errors

Clearly there is a trade-off between accurate feature matching and accurate range estimation. On the one hand we wish to avoid as much occlusion as possible and therefore require a low baseline-focal length product, whereas on the other we require the opposite for accurate 3D rectification. As the sampling interval is always restricted, a compromise must be made between accurate feature matching and accurate rectification. Added to this is the problem of segmentation. Often

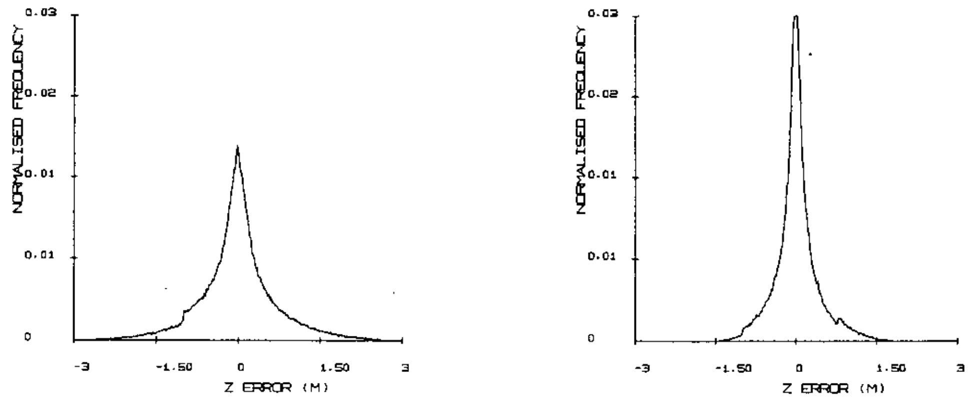


Figure 3–15: Probability Density Function of the Error Δ z

the errors associated with feature extraction, such as edge localisation, noise and streaking, will have a considerable affect on a matching algorithms effectiveness. All this makes comparison between different, stereo algorithms extremely difficult with results being data dependent.

Although not discussed here, errors can also vary with relative camera angles. Borghese and Ferrigno [10] show that the quantisation errors are likely to be at their minimum when both cameras are at 45 degrees to the main depth axis. However, for practical reasons a commercial system is likely to have both cameras laterally separated on the same imaging plane. This is the system used in the experiments described in Chapter six.

The above section dealt with the fundamental, and usually unavoidable, causes of error in a stereo system. In any practical system it is unlikely that the camera geometry will be known accurately, *a-priori*, due to mechanical inaccuracies. The next section will deal with the problem of estimating the camera parameters from known scene geometry or feature matches.

3.6 Calibration

All the previously described stereo matching theories and techniques have studiously ignored the question of rectifying or extracting depth information, in recognised units, from estimated disparities. This was done for two reasons. The first is that that it is a distinct issue and has no direct bearing on the problems of correspondence and the second is that, for alarm systems, it is possibly not required. For an individual camera setup we can conduct comparisons in raw inverse disparities. The problems only arise when we consider the possible errors around a particular disparity value.

The problem of camera calibration is the accurate 3D determination of internal camera geometries and optical properties with as little a priori scene knowledge as possible. Estimating the translational and rotational parameters between two or more stereo cameras is a similar problem to that of estimating the motion of a moving camera. We wish, only, to use the correspondences between features in spatially separated views of a scene. For the rest of this section, we will assume that the correspondence problem has already been solved.

Until about 1980, it was unknown how many correspondences were required to ensure a unique estimate of motion. Further, the only techniques available for finding camera geometries from matched correspondences required the iterative solution of third order simultaneous equations. In 1981, Longuet-Higgins [48] solved both the above problems and proposed a non-iterative scheme. A similar algorithm was published around the same time by Tsai and Huang [85][86]. Both authors showed that eight independent correspondences were necessary and sufficient to uniquely determine 3D motion. The eight point method proposed by Tsai et. al. and Longuet-Higgins is now summarised.

3.6.1 The Longuet-Higgins/Tsai Calibration Algorithm

The scene coordinates of a point P are (X_1, X_2, X_3) in the left camera coordinate system and (X'_1, X'_2, X'_3) in the right camera coordinate system. We can project these points onto the imaging planes, as in Equations 3.26 and 3.27.

$$(x_1, x_2) = (\frac{X_1}{X_3}, \frac{X_2}{X_3}) \tag{3.26}$$

$$(x'_1, x'_2) = (\frac{X'_1}{X'_3}, \frac{X'_2}{X'_3}) \tag{3.27}$$

It is more convenient to work in homogeneous coordinates and we therefore set $X_3 = 1$, $X'_3 = 1$ and define

$$x_\mu = \frac{X_\mu}{X_3} \quad x'_\nu = \frac{X'_\nu}{X'_3} \tag{3.28}$$

where $(\mu, \nu = 1, 2, 3)$. We can now define a relationship between the coordinate systems of the left and right hand images where **R** is the rotation matrix and **T** a translation vector.

$$X'_\mu = R_{\mu\nu}(X_\nu - T_\nu) \tag{3.29}$$

Longuet-Higgins shows that by using a matrix Q such that,

$$\mathbf{Q} = \mathbf{RT} \tag{3.30}$$

and where

$$\mathbf{T} = \begin{bmatrix} 0 & T_3 & -T_2 \\ T_3 & 0 & T_1 \\ T_2 & -T_1 & 0 \end{bmatrix} \tag{3.31}$$

we can obtain relationships between the image coordinates such that

$$x'_\mu Q_{\mu\nu} x_\nu = 0 \tag{3.32}$$

Thus, if eight corresponding points are known, x'_μ and x_ν, in the scene, then the coefficients of Q can be found by solving eight simultaneous linear equations. We do this by dividing the LHS and RHS by $Q_{3,3}$. This has the effect of providing a square 8x8 matrix on the L.H.S. with a 3x1 matrix on the R.H.S. and can be solved for every ratio using L.U. decomposition and back-substitution. We now have an un-scaled value for every element in Q and can proceed to calculate the values of the translation and rotation matrices.

The values and relative signs of the translation vector can be obtained from Equation 3.33.

$$\mathbf{QQ^T} = \begin{bmatrix} 1-T_1^2 & -T_1T_2 & -T_1T_3 \\ -T_2T_1 & 1-T_2^2 & -T_2T_3 \\ -T_3T_1 & -T_3T_2 & 1-T_3^2 \end{bmatrix} \tag{3.33}$$

and the rotation matrix from Equation 3.35 by defining the new matrix, **W**, shown in Equation 3.34. Each row is calculated individually and α, β and γ are permutations of (1,2,3).

$$\mathbf{W}_\alpha = \mathbf{Q}_\alpha \times \mathbf{T} \tag{3.34}$$

$$\mathbf{R}_\alpha = \mathbf{W}_\alpha + \mathbf{W}_\beta \times \mathbf{W}_\gamma \tag{3.35}$$

The final stage of the procedure is to alter the signs of **T** and **Q**. This is done by calculating estimates of X' and X using the Q matrix. If the signs are not correct then the relevant rows and columns of Q and T require to be multiplied by -1. The details of this procedure can be found in the papers.

The Longuet-Higgins, Tsai technique is straight forward and well understood and depends on all the points in the scene being independent. Tsai and Huang [86] also describe combinations of points which must be satisfied to ensure that the eight point algorithm provides a unique solution. These can be found in Reference [86].

The above provides a non-iterative approach to calibrating stereo cameras. However, there are doubts about accuracy and choice of feature points. Yasumoto and Medioni [91] suggest that the problem of estimating 3D motion is an *ill-defined inverse problem* and results would be improved if regularisation techniques were employed. The ill-poised nature of the problem was also expressed in error estimates provided by Tsai and Huang [86]. These results are repeated in Table 3–1 for image plane points shifted in a random directions by 1%. This simulates errors in feature matching and extraction. It is of interest to note that the error for R, the rotation matrix, decreases as the number of points escalates. In contrast,

Number of Points	8	9	20
Error of Q	47.13%	18.74%	2.32%
Error of R	14.32%	3.68%	0.83 %
Error of $\frac{\Delta x}{\Delta z}$ and $\frac{\Delta x}{\Delta z}$	53.97 %	3.52 %	10.09 %

Table 3–1: Calibration Error Versus Number of Points for 1 % Perturbation of (X',Y')

the error for translation, $(\frac{\Delta x}{\Delta z}, \frac{\Delta y}{\Delta z})$ first decreases and then increases to 10.09%. These types of errors were confirmed in software simulations by the author. There is clearly a problem in accurately extracting translations at the same time as good rotations. The next section discusses ways of improving on these results.

3.6.2 Improvements in Calibration Accuracy and Other Calibration Techniques

It is clear, from the previous section that calibration accuracy can be severely impaired by pixel quantisation and feature localisation errors. Considering the inverse nature of camera calibration, Yasumoto and Medioni [91] proposed the use of additional constraints to restrict the number of acceptable solutions. They search the surface of an error function calculated for each matched point. Yasumoto proposes regularisation in order to smooth the extracted error surface and reduce the number of false minima.

As an alternative, or adjunct, to Yasumoto's work, researchers have attempted to improve results by integrating the calibrations from successive images. Thacker and Mayhew [80] published one example which uses a Kalman filter to track rotation and translation variables through time. In addition to this they employ a form of variational regularisation first suggested by Trivedi [83].

Initially an estimate of the error is calculated from the positional changes required, in x' and x, to satisfy Equation 3.32, for a particular Q matrix. Thacker and Mayhew calculate estimates of these shifts explicitly and use the results to calculate an overall error function, Equation 3.36.

$$E = \sum_i \delta E_i = \sum_i (\delta x_{'} S^{-1} \delta x_{'} + \delta x_i'^T S^{-1} \delta x_i') \tag{3.36}$$

where δx_i and $\delta x_i'$ are the required shifts to satisfy Equation 3.32 and S is the a 3x3 error covariance matrix derived from the feature detection algorithm. The minimal solution to Equation 3.36 was found, by adjusting the Q matrix, in terms of a standard algorithm such as the "Downhill Simplex Method" [68].

3.6.3 Discussion of Calibration Techniques

The techniques just reviewed are all computationally intensive and are certainly not candidates for a commercial CMOS implementation. If accurate Cartesian coordinates are required in a final stereo installation, the most likely configuration is that of a portable computer with an interface to the image sensors. As an alternative it is not always necessary to perform a complete calibration to establish correspondences or relative depth. In many applications depth need only be estimated in terms of disparity. For example, as described later in this thesis, it may only be necessary to determine a disparity threshold for an alarm system. If the intruder crosses the threshold for a certain number of frames then the alarm can be activated. Thus, using relative depths, higher level algorithms can be developed which still employ depth information. In this respect, Mohr and Arbogast [57] show one technique for extracting depth information without knowing the camera's geometry.

Overall, accurate camera calibration is a serious problem for stereo vision systems if we wish to establish the Cartesian world coordinates of points in a scene. The inverse nature of the problem amplifies small errors in the coordinates being used. Automatic camera calibration is an area which requires more research.

3.7 Conclusions

This chapter started with review of algorithms which solve the correspondence problem in the light of the biological evidence. There are many different approaches, ranging from brute force correlation to specific feature and object matching. It has been decided, for reasons of computational complexity, that a feature matching approach is most suitable for the intruder problem described in the introductory chapter. Employing a feature based approach allows a reduction in the data rates required, by matching only those parts of the image which are interesting [3]. An important point, not made by many authors, is that the task of feature detection is integral to the matching algorithm employed. Most researchers appear to concentrate only on the correspondence problem, or only on the feature detection problem without considering the trade-offs involved between the different stages. Clearly if one alters the feature extraction process to extract one type of edge, simplifications can be made to the correspondence algorithm. This is the approach taken later in this thesis.

Simplifications in the algorithm can also mean reductions in the hardware required. There have been relatively few stereo algorithms implemented in specialised hardware, two of which were mentioned in the previous chapter. Apart from cost, several reasons appear to have caused this situation. The first is the dependency of stereo algorithms on the input features. It is therefore difficult to build a general piece of hardware which would be useful in a sufficient number of applications. This dependency is not just restricted to the input data. Different applications may desire various forms of output data. Thus in this work, the general approach of building a stereo module was avoided. Effort was directed to altering the various image processing modules to take advantage of interdependencies.

[3] In any case, it is difficult, if not impossible, to match areas of the image where there is no luminosity gradient.

Also considered, in Section 3.5, was an analysis of the likely errors which will be encountered in a typical stereo system. These were divided into errors dependent on the matching algorithm and those dependent on the camera and scene geometry. As Figure 3–15 shows there are clear advantages in using a three camera stereo rig. Such calculations would be of particular interest in an alarm system where wide angle lenses and longer ranges can be expected.

The final sections dealt with the thorny problem of calibration. The inverse nature of stereo vision has resulted in this being an area of vision where errors can be large. The algorithms reviewed here were all computationally intensive and therefore infeasible for this application. However it is not always necessary to extract depth in metric units and valuable information can be gained using relative disparity measurements.

Chapter 4

The DETECT System and Initial Segmentation

4.1 Introduction

The aim of this and the next chapter is to describe a hardware efficient system capable of tracking a human moving in a stationary background in a large scene. For the remainder of this thesis this software will be referred to as the DETECT system. The main functional problem associated with such a stereo vision system is in the short focal lengths necessary to view the required area. In addition, the hardware constraints imposed by any cost effective implementation require to be satisfied. The last two chapters have considered algorithms in the light of possible and real hardware implementations. In particular, this work is aimed at an implementation using the CMOS sensors [1] described in Chapter One. Achievement of this goal requires that limits be placed on allowed arithmetic at the pixel level of the algorithm. In effect, at the pixel level, floating point digital calculations must be eliminated together with general multiplications and divisions. Due to the reduced data rates, higher level processing, such as explicit disparity calculation, time domain filtering and global threshold calculation would be performed using associated microprocessors. Such arithmetic efficiency has been achieved by developing software which uses three major constraints, specific to stereo vision, and the expected application. These are:

1. **The vertical nature of the human form:** A standing human has very few horizontal edges. Also, the lateral separation of the stereo cameras used in this work makes it difficult to match horizontal edges anyway. This makes it sensible to extract only vertical edges, reducing edge detection to a lateral differentiation, followed by a vertical track.
2. **Only outline edges are extracted for matching:** This transfers the stereo correspondence problem to the lower stages of segmentation and reduces the computation required for matching to a simple scan.
3. **Only a disparity threshold is required to implement an invisible boundary.** Thus if an object continually exceeds that threshold, through time, the alarm can be sounded. This disparity threshold could

be calculated, on site, by a person moving around at the desired distance. Alternatively, if metric distances are desired, a more accurate calibration could be provided at installation using an interface to a portable computer. In the latter case, the errors mentioned in the last chapter should be considered.

Figure 4–1 gives an overview of the entire DETECT system and the various sections of the algorithm. It is not intended that these modules be considered in isolation from one another; implementation advantages can be gained by considering the problem in its entirety. It is the author's view that treating individual vision problems in isolation often hides many interdependencies. In creating a high level intelligent vision system, an understanding of the likely errors and failings of the lower level "image processing" is required. Allowances and limits of operation can then be defined. Despite this, it is necessary to consider the different sections of the system in isolation and then highlight their dependencies.

This chapter will start with a section considering the low level algorithmic principles and techniques used in the system. Throughout Section 4.2 reference will be given to possible hardware implementations and trade-offs. Section 4.3 will give a brief description of the software used to test the algorithms discussed in section 4.2 and the next chapter. A more detailed explanation, of the software function, can be found in Appendix B. The chapter will finish with some general conclusions.

4.2 The DETECT Stereo Algorithm: Level Segmentation

Attention is now given to the individual DETECT modules shown in Figure 4–1. Figure 4–2 shows the earlier stages of processing in more detail. The techniques used to provide this segmentation are all differential to allow reductions in computation, and correlation methods have been avoided. This has included the elimination of an initial smoothing filter for edge detection. In the trials described

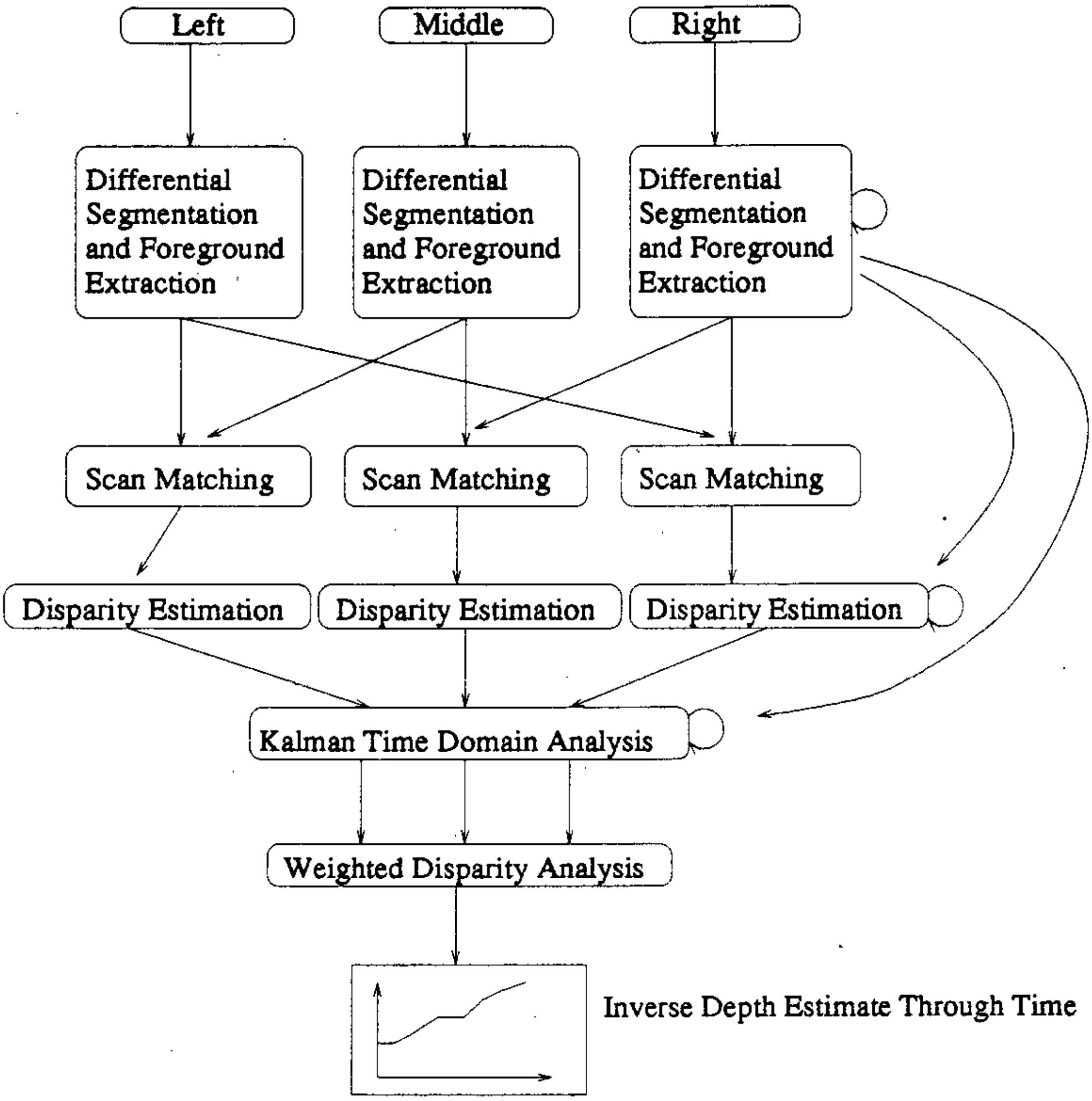

Left
Middle
Right
Differential Segmentation and Foreground Extraction
Differential Segmentation and Foreground Extraction
Differential Segmentation and Foreground Extraction
Scan Matching
Scan Matching
Scan Matching
Disparity Estimation
Disparity Estimation
Disparity Estimation
Kalman Time Domain Analysis
Weighted Disparity Analysis
Inverse Depth Estimate Through Time

Figure 4–1: Overview of DETECT System

in Chapter Six, it was found that this made little difference to the overall disparity estimate. If, in a larger trial, a smoothing filter was necessary then low cost analogue implementations of such functions exist[1].

For the rest of this thesis, the following terms will be used.

1. *Background Images:* The images captured when there is no movement in the scene.
2. *Foreground Images:* The images captured when an object is being tracked.

3. *Difference Images:* The modulus of the difference between the foreground and background images.

4. *Edged Background and Edged Foreground Images:* The above named images, after the edge detection procedure has been applied.

Computations aimed at deriving edge information and the maintenance of an edge map are described in Sections 4.2.1 and 4.2.2. Section 4.2.3 will describe how difference images are thresholded to provide input for the nearest-neighbour clustering described in Section 4.2.4, respectively. Section 4.2.5 will deal with how estimates of the background can be maintained through time. Backgrounds are used to derive difference images for thresholding. Once the edges and clusters have been extracted they are combined to provide the outlines required by the matching algorithm. Section 4.2.6 describes how this is achieved. The combination of information, in this manner provides more reliable segmentation.

4.2.1 Edge Extraction

Disparity estimation requires edges for matching. These can be extracted from the difference, background and foreground images by differentiation followed by tracking. In the current implementation a lateral differentiation, across the image, is performed on both the background and current images. No initial smoothing is applied to control the edge noise. Thus, false edges are eliminated using a minimum line length, comparison with extracted clusters, and the disparity gradient limit. However the hardware cost of a simple averaging filter is not so great as to preclude its use. It may turnout to be desirable from the results of a larger field trial. One problem, with initial smoothing filters, discussed in the literature review, is that of edge localisation. There is no point in applying an extra mask if it is not needed. After lateral differentiation, a modified non-maximal suppression is applied to the resultant image. However, the process is simplified to one dimension, with pixels either side of peaks being set to zero. Suppression reduces the number of falsely tracked edges.

Tracking is applied using hysteresis thresholding for both edge strength and edge length. For edge strength, if an upper threshold is exceeded, tracking pro-

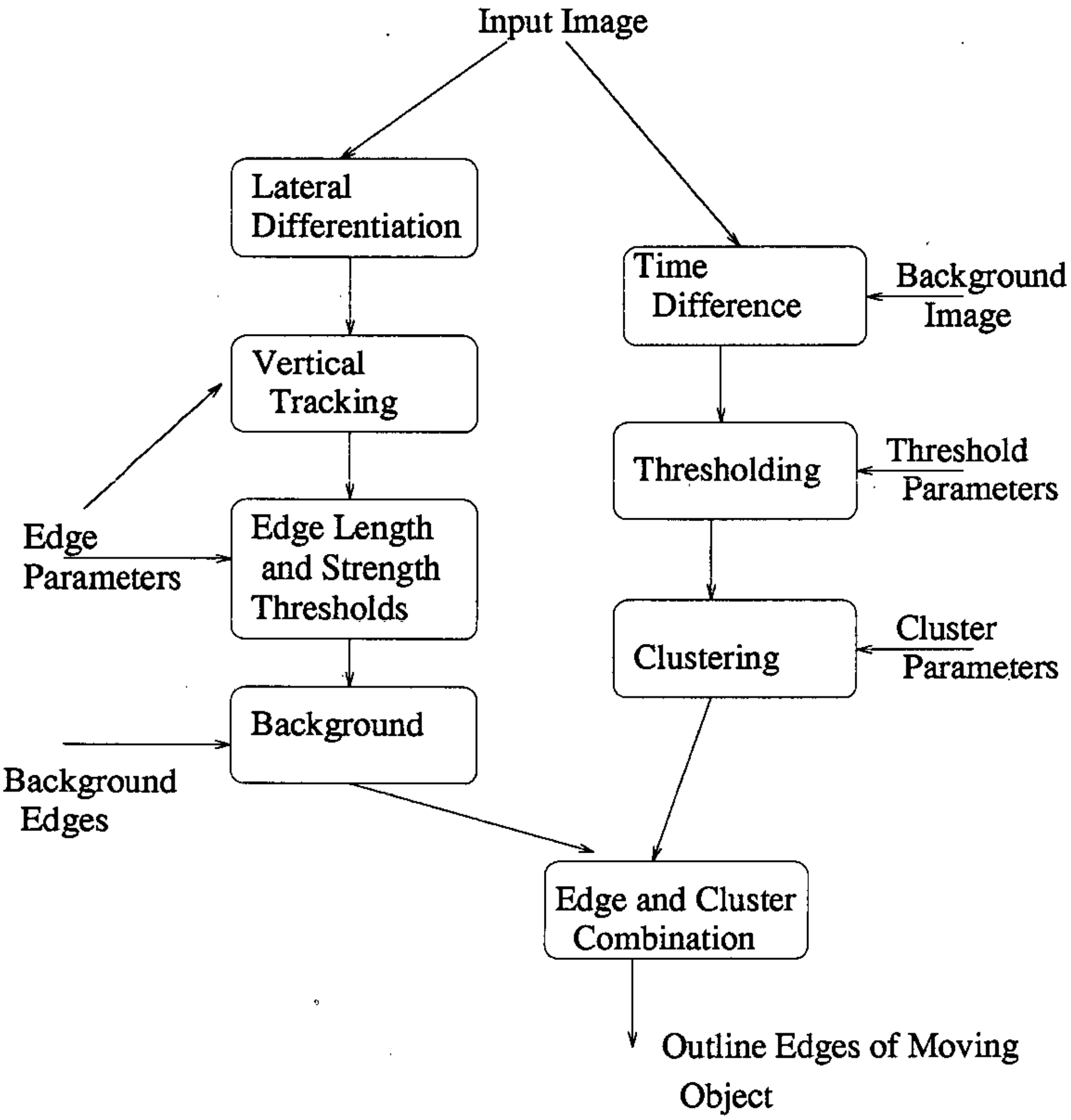


Figure 4–2: Overview of Low Level Segmentation Stages

```
for a = 0 to IMAGESIZE{
  for b = 0 to IMAGESIZE{
    if (latdiffimage[a][b] > startthreshold){
      track = 1
      currentrow = a
      currentcol = b
      while (track)
      {
        - compare lower three pixels
          ie. currentrow+1, currentcol+(-1,0,1)
        - take the largest value
        if (the largest value is greater than contthreshold){
          - currentrow = currentrow + 1
          - currentcol = value of largest pixel
          - if tracklength > minimum track length store in edge structure
          - if tracklength > minimum track length for the firsttime
            transfer edge buffer to edge structure
        else store column value in temporary buffer
        }
        else
          track = 0
      } end of while loop
    } end of starting track if statement
  } end of b loop
} end of a loop
```

Figure 4–3: The Tracking Algorithm

ceeds downwards to the next row, where the three adjacent pixels are considered as edge candidates. Assuming a lower strength threshold is exceeded and one of the three pixels is a peak, the edge is extended. Tracking proceeds until a minimum edge length has been exceeded. During this initial phase the results are stored in a temporary buffer. If the minimum length is exceeded tracking continues until no pixels in the neighbourhood of the next row exceed a lower threshold. Thus candidate edges are selected on two accounts. Firstly, each must exceed a certain minimum length and, secondly, exceed either of the two strength thresholds. The tracking algorithm employed can be summarised in the pseudo-code shown in Figure 4–3. The current DETECT implementation applies edge detection to the raw foreground and background images blindly. As an extension, use could be made of edges extracted from the difference image and then compared with those from foreground and background edge maps. The outline of a new object should have similar edges in both the foreground and difference images, but not the background. The technique is illustrated, for the one dimensional case, in Figure 4–4. A problem with this approach is that overall lighting conditions change from frame to frame. Thus, the edge strengths will be dependent on two images and strength comparisons may fail under these circumstances. However, the main reason for not using this method is that the extracted edges still have to be grouped using the difference clusters, described in Sections 4.2.3 and 4.2.6. In order to extract these clusters we threshold the difference image and tend to extract sections of important outline edges, by default. Parts of the foreground edge will coincide with the cluster boundary. Outlines can then be defined by their proximity to a significant cluster.

Separate extraction of edges from the foreground and background, followed by a comparison with edges from the clustered difference image, is therefore preferable for this work. Its advantage is that the edges are dependent on position and strength relative to its own neighbourhood. Edge thresholds are therefore based on differences which are more likely to be constant over time. Also, the use of length as an indicator of edge significance allows the edge strength threshold to be less critical to the system as a whole. Figure 4–5 shows the edges extracted for a single scene. Although noisy in comparison with other edge detectors, eg. the Canny operator described in Chapter Two, the computational complexity is much reduced. Deficiencies are compensated by the combination of information from the clustering and thresholding modules of the system.

4.2.2 Edge Map Use and Update

Initial detection is based on edges rather than absolute frame to frame differences. The extracted edge map from the foreground image is stored for reference and used to eliminate persistent edges. Decisions must be made as to which edges are stationary and which are not. Several parameters can be employed in estimating

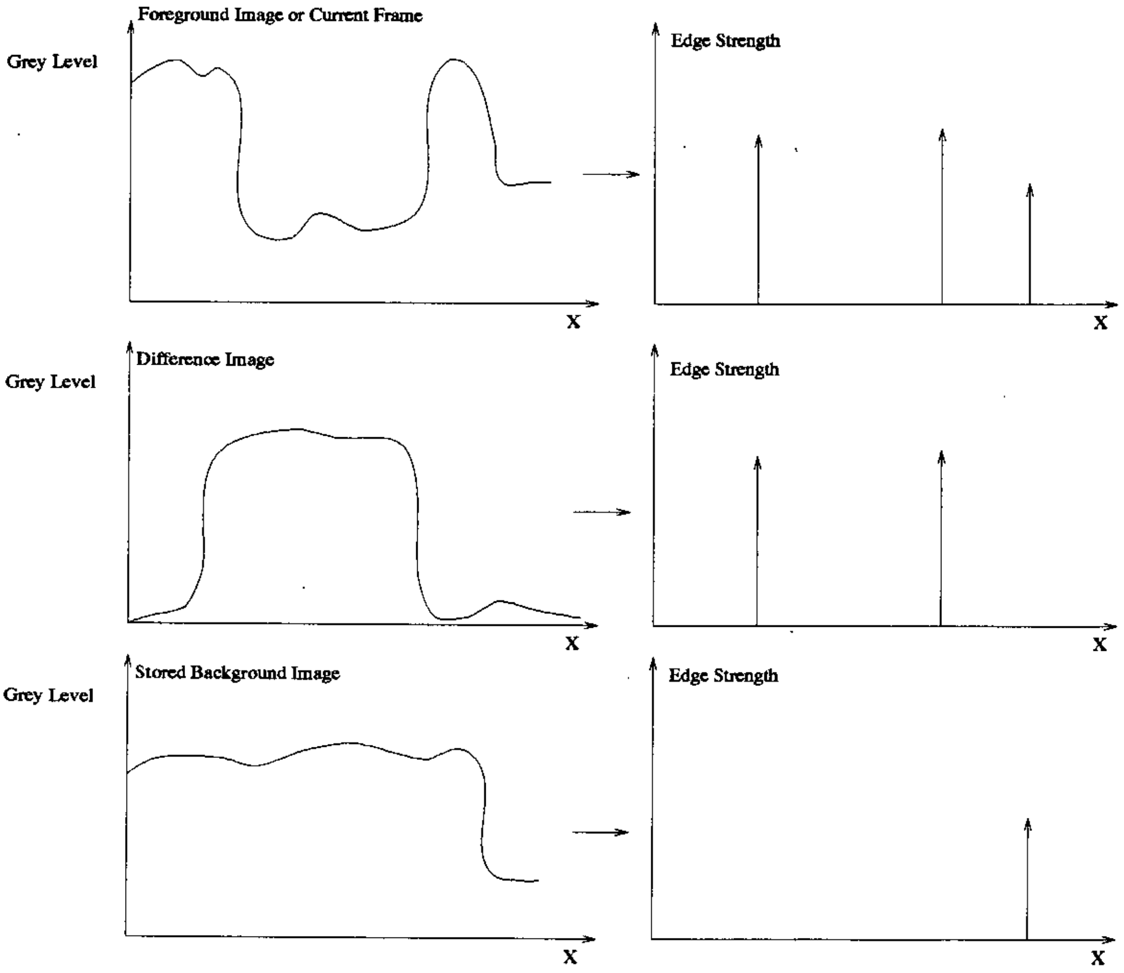


Figure 4–4: Outline Edge Extraction Using Difference Images

the permanence of an edge through time. The following criteria are the most obvious:

1. **Difference in edge strength:** Edges represent changes in spatial luminosity and are relatively independent from temporal light differences. Thus the strength of a stationary edge often remains constant in time.

2. **Position:** Assuming the cameras are not moving, edges should remain constant in position. An edge pixel from a stationary edge in one frame should be an edge in the next frame. Various sources of noise can prevent this occurring. For example, if the true position of an edge is halfway between the centres of two pixels then a slight change in luminosity may cause a change, of one pixel, in the marked edge position.

3. **Orientation:** Clearly, if two edges have the same position, individual edge pixels should have similar orientations. Edges extracted from individual frames are rarely complete and orientation could be employed to extrapolate from known segments, to other more doubtful parts. In this situation, assumptions must then be made about edge curvature. Such predictions are hard to make in the wide range of proposed scenes targeted by this system.

Within the bounds of computability, consideration was given to techniques for building edges maps. Firstly, laterally differenced frames can be continually added and normalised. Here, advantage is taken of the fact that moving edges reduce in significance as time progresses. It is vice versa for stationary features. Using edge strengths in this manner can cause problems when different edges have variable strengths. A particularly strong edge, resulting from a large change in spatial luminosity, may cause problems with scaling and push weaker edges under a strength threshold. These "weaker edges" may be consistently in the same position and valid in each individual frame.

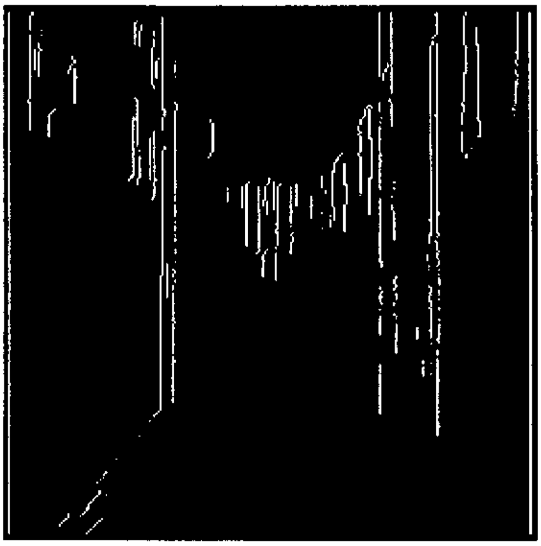


Figure 4–5: A Typical Edge Map

The obvious solution to this problem is to express edges in binary representation. Now, edges are exclusively dependent on their persistence through time. This is the method used in the present system. As an example, Figure 4–5 shows the vertical edge map extracted for a 12 frame sequence. Every time a pixel is designated as an edge, the map for that position is updated. The final image is grey level stretched to display the more significant stationary edges.

Once the stationary edge map has been established, current edges can be compared to previous ones. As a current edge is tracked along its path, a value of positional similarity can then be calculated, dependent on length and edge map strength at that point. Equation 4.1 shows one possible measure.

$$S_{edge} = \frac{\sum_{i=0}^{L} \frac{P_{pix}}{dE_{pix}}}{L} \tag{4.1}$$

S_{edge} is the edge's calculated persistence, L is the length of the current edge, P_{pix} is an individual edge's persistence [1] and dE_{pix} is the difference in edge strength

[1] This is simply the value of the accumulated edge map, through time, for a particular pixel.

through time. Unfortunately, the computation of Equation 4.1 involves division and multiplications which would be unreasonable in a final CMOS sensor system. However, adaptations are possible. Instead of dividing the overall sum by the total length, the edge can be sub-sampled to allow a power-of-two divisor. Also in the current implementation values of dE_{pix} are not used, removing a multiplication. This has been achieved without an appreciable reduction in performance.

4.2.3 Thresholding

Some threshold techniques were reviewed in Section 2.3. Most of the described algorithms can be dismissed on grounds of their arithmetic complexity. Exceptions to this are the P-tile and mode methods [77]. Although not reviewed in Chapter Two, local mean thresholding was also considered for this application.

The P-tile method uses *a priori* knowledge of the size of the object and chooses a threshold to achieve it. Once a moving object is being accurately tracked thresholds can be chosen to maintain a particular size within allowed limits. This has several problems. Objects may suddenly change, in size, when two separate clusters merge. The resultant threshold will wrongly increase to separate the clusters. An even poorer threshold at the next frame will result. This process can cause large jumps in the applied threshold and tracking is likely to fail. Another difficulty arises in establishing, for three dimensional scenes, a correct estimate of object size. Although implemented, the P-tile method is not applied in the current DETECT system. The problem of finding an initial size estimate together with the likely tracking errors are the main reasons. However, with longer test sequences, recorded at faster rates, it might be feasible to make better estimates of the correct cluster size. Such a tracking technique, based on known cluster sizes, is a possible line for future research for this type of low cost system.

A second possibility is the use of local means as adaptive thresholds. Figure 4–6 shows an example where a difference image has been thresholded according to the local mean. However this technique tends to extract poorly connected clusters which fall below relevant size thresholds. The basic problem with mean

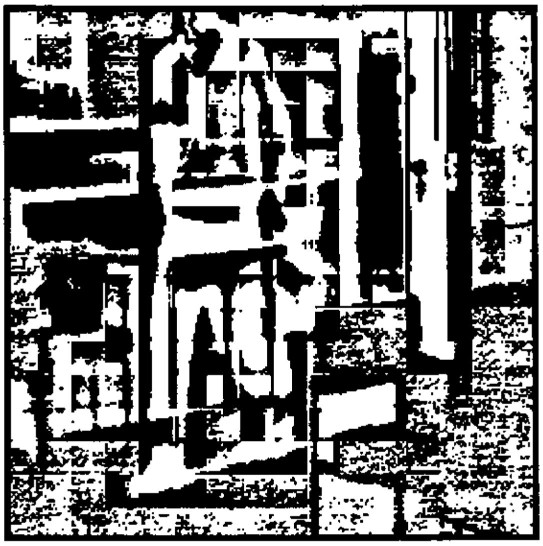


Figure 4–6: Mean Thresholding Applied to a Difference Image

thresholding is that it takes no account of global information and tends to work best if sub-regions can be scaled with a known object size. This situation does not arise in this application where objects move and change size with time.

As an alternative, to the above, one can classify pixels using histogram analysis [2]. The histogram of a difference image should, theoretically, consist of two clear foreground and background peaks. Thresholds are chosen to be at the bottom of the valley between the two peaks. Problems occur with this type of thresholding when there are multiple peaks, multiple valleys and different sized frequency amplitudes.

The current implementation of DETECT uses the fraction of the global mean as an initial threshold. After four frames of tracking, the grey level distribution, of the previous frames main tracked cluster, was extracted. The mean of this distribution was used to calculate a higher bound. A second distribution was calculated by subtracting the previous cluster s histogram from the current frame s overall difference distribution. A mean is again taken from this new distribution and used as a lower bound. A threshold can be chosen somewhere between the

[2] Histogram techniques are often referred to as mode methods

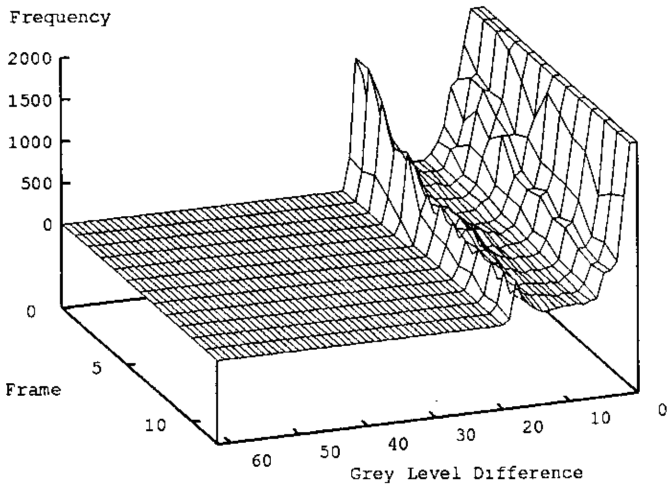


Figure 4–7: Overall Difference Histograms Through Time

two boundaries. The advantages of this technique are several. Firstly, it is based on histograms which can easily be implemented as accumulators in hardware. Computation of means from the histogram is simpler than calculating local information for every neighbourhood. Secondly, account is taken of the difference distribution specific to a tracked cluster. An example thresholded image with its main cluster marked is shown in Figure 4–10. This image is extracted from the Sequence 9, from the test data described in Chapter Six. Difference histograms, for sequence 9, are shown in Figures 4–7, 4–8 and 4–9. Figure 4–7 shows the complete difference histogram through time. Figure 4–8 shows the complete difference histogram, minus the cluster distribution, from the previous frame and Figure 4–9 shows the distribution of the major changing clusters through time. Overall, the calculation of these statistics can be implemented efficiently using hardware accumulators on a VLSI chip. Mean calculations could be performed, on an associated microprocessor, using these histograms as input. It seems clear that custom VLSI is efficient at collating data, into a format suitable for a microprocessor, although not performing the final calculations.

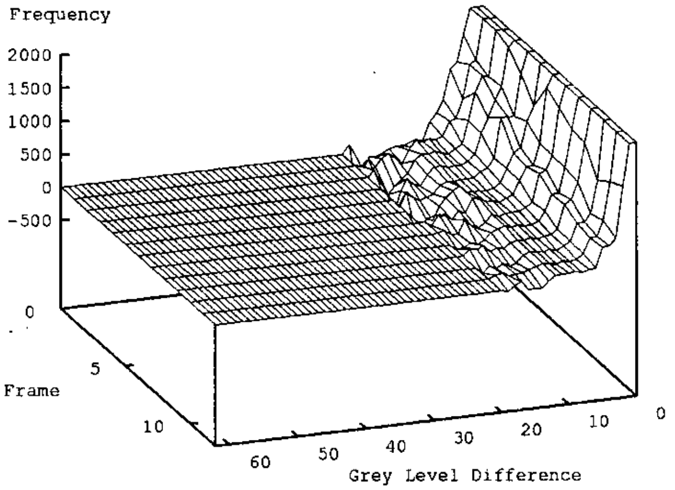


Figure 4–8: Difference Histograms Through Time with Clusters Subtracted

4.2.4 Clustering

There are many definitions and meanings of the word "cluster"[22]. In this work, the term cluster is used to define a connected area of the thresholded difference image. This can be used to connect disparate edge segments into one object and extract a combined outline from both the cluster data and the edge data.

As with thresholding, the cluster algorithm was written to expose hardware implementation problems. Thus, procedures have been written without recursion in order to expose the true storage requirements for a hardware implementation. The routine s basis is a local search which pushes thresholded pixels from each neighbourhood onto a stack. The pixel in the input image is then set to zero. Once done for a locality, another pixel is "popped" off the stack and used as the next search centre. The following pseudo-code, Figure 4–11, illustrates the code used to generate Figure 4–10. Although not mentioned in Section 4.2.3, it is possible, with minor alterations to the above code, to perform thresholding at the same time as the connectivity search. More sophisticated adaptive difference

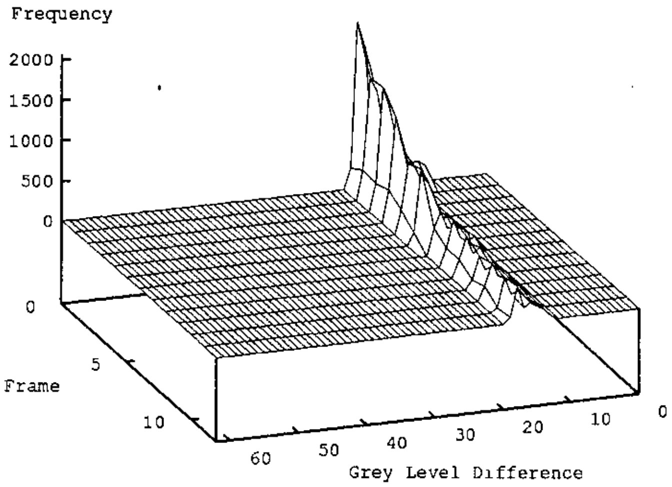


Figure 4–9: Difference Histograms of Clusters Through Time

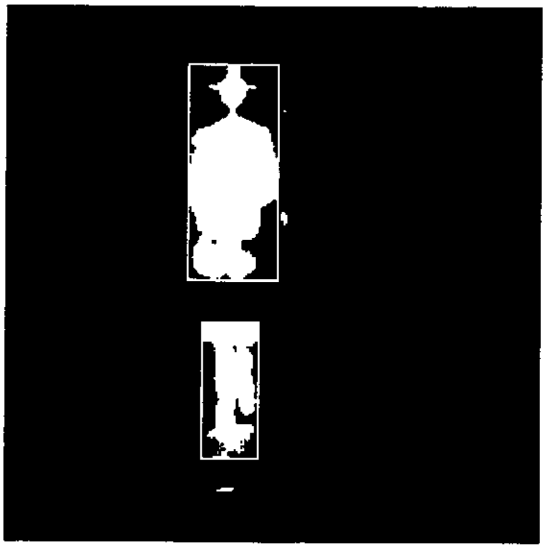


Figure 4–10: Cluster Thresholding Applied to a Difference Image

```
- XOR thresholded image to find cluster edges
for a = 0 to IMAGESIZE{
  for b = 0 to IMAGESIZE{
    if (binaryimage[a][b] is an edge of a cluster){
      currentrow = a
      currentcol = b
      continuecluster = TRUE
      while (continuecluster){
        for c = -1 to 1 {
          for d = -1 to 1 {
            if (binaryimage[currentrow + c][currentcol + d]
                is an edge of a cluster) then
              - push (currentrow + c, currentcol + d).
              - set binaryimage[currentrow+c][currentcol+d] = 0.
              - update cluster max/min left/right boundaries.
        } end of d loop
      } end of c loop
      if stack is not empty pop (currentrow, currentcol)
      else{
        - create new cluster structure.
        - store cluster parameters eg. size, boundaries etc.
        - continuecluster = FALSE
      } end of while loop
    } end of main if condition
  } end of b loop
} end of a loop.
```

Figure 4–11: The Cluster Algorithm

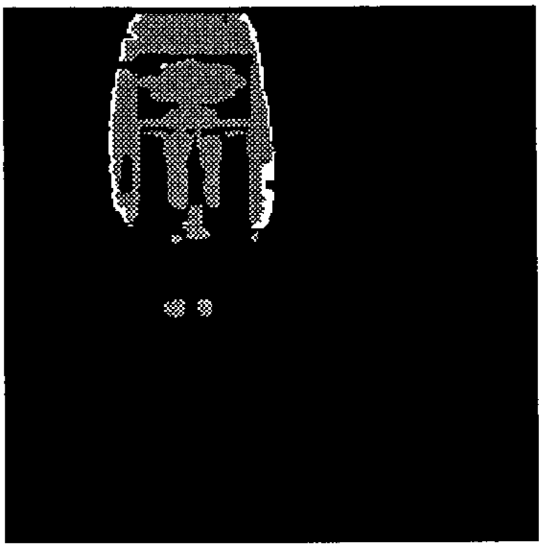


Figure 4–12: Cluster Expansion (White) with Core Cluster (Grey)

thresholds could then be used, using parameters such as cluster size and shape, running totals of grey level differences and the cluster's position in the scene. In this work, a hysteresis technique was implemented. Thus, a cluster search only started if a pixel exceeded an upper threshold and stopped when the current value had no untouched neighbours above a lower threshold. However, after combining thresholding and clustering in this manner, there was little difference in the overall performance of the DETECT system and this technique is not currently used.

Another adaptation attempted was a form of cluster expansion. Currently multiple sources of information are used to decide which edges are outline and which are not. Thus it is better to choose a more severe threshold to ensure that clusters are well separated. Once the outlines are known they can be used as seed points in a lateral expansion, at a lower threshold, or until an edge is met. Figure 4–12 shows a core cluster with its lateral expansion marked in white. Although a slight improvement in overall segmentation was achieved, the current implementation does not employ this routine. As is discussed in Section 4.2.6 cluster outlines are used to extract relevant edges. If a individual edge coincides with the cluster outline it is used for matching. Such a lateral expansion tended to "connect" with noise edges and reduce the number of correct stereo matches.

Overall, accurate extraction of clusters is not a crucial part of the system as the results are combined with edge information to provide a more complete segmentation.

4.2.5 Background Update and Extraction

The thresholding/clustering algorithms just described, assume a previously stored and continuously updated background image. We wish to store all areas of the stationary background which are not covered by the tracked object. This task is similar to many problems in the computation of motion flow and image coding[30][26][36], where one wishes to quantise those blocks of the image which have changed.

As an intruder in an alarm system is likely to be an unusual event, it should be possible to completely update the background during the normal operation of the system. Attaining this goal requires clear discrimination between an object's presence in the scene and a general change in luminosity. Changes in luminosity can be caused by slow effects such as clouds and day light, or higher frequency effects such as sine waves, in time, produced by mains strip lighting. These effects must be eliminated or prevented from affecting the tracking and detection algorithms.

Discrimination between *global* and *local* luminosity changes can be based on edge information. They measure *relative* and not *absolute* changes in luminosity. Edges will remain constant in position and, often, in strength. Thus, the sudden appearance of significant moving edges provides a better indicator of an intruder's presence than frame to frame luminosity differences.

Figure 4–13, shows the ratio of the number of *changed* edges to the number of background edges through time. This sequence was created artificially by merging two half sequences, such that a person suddenly appears in Frame 6. Clearly the ratio of new edges to old increases as a person appears in the scene. A more realistic example of how the number of edges change, as an intruder enters the

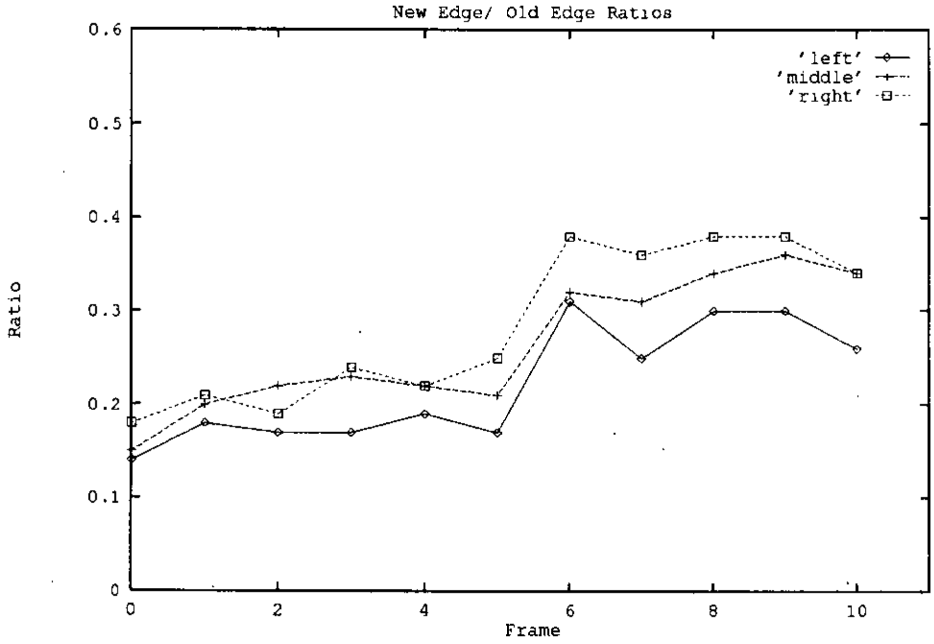


Figure 4–13: Graph Showing the Ratio of New Edges to Background Edges Against Time for Artificial Sequence

scene, is given sequence 10 [3]. In all three sequences an intruder enters gradually through a door. A graph showing the new to old edge ratio is shown in Figure 4–14. The ratio increases as the person moves towards the camera.

Such a new to old edge ratio parameter is largely independent of absolute light changes and is therefore a good measure of physical changes in the scene. In a final implementation experiments would have to be conducted to establish the optimal ratio at which complete backgrounds would be captured.

Once detection has been achieved, background extraction and update can be restricted to areas outside the main regions of interest determined by the clustering/thresholding routines. In DETECT, the background is renewed everywhere, every frame, except within the main clusters boundaries. This is performed on a simple pixel for pixel substitution without smoothing or neighbourhood averaging and provides a satisfactory low cost solution for the current data.

[3] The details of Sequence 10 can be found in Chapter Six.

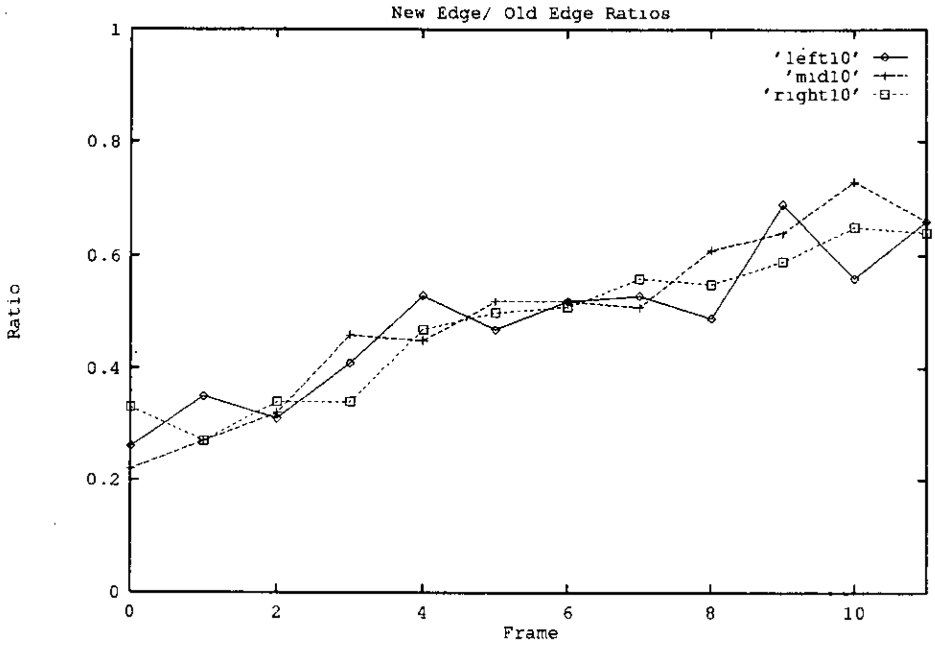


Figure 4–14: Graph Showing the Ratio of New Edges to Background Edges Against Time for Sequence 10

Errors in update will inevitably occur, due to shadows and reflections. These are dealt with in several simultaneous ways. Most local lighting changes are restricted to small areas and can be eliminated using a minimum cluster size. Thus if the cluster is insignificant the background, in this area, will update.

4.2.6 Edge and Cluster Combination

At this stage in the system we now have two sets of data representing the scene. Firstly the thresholded and clustered regions of the image and, secondly, the list of vertically oriented edges. As shown in Figure 4–2, these two sets of data are combined to extract the relevant edges for stereo matching.

The problem is one of establishing which of the disconnected edges are from the same object. Section 2.4 discussed the different approaches. Maximal cliques can be used to find strengths of connection between different edges. Such strengths could be based on edge orientation, edge strength and proximity. The computation involved rules this avenue out of bounds.

For this work, a modified "connection strength" technique was attempted. Edges from the foreground image were compared with the difference image. Lateral scanning started from each edge pixel in a direction determined by the surrounding pixels in the difference image. If the difference pixels on the left were larger, then the scan would proceed to the left and vice versa. The scan continues across the difference image, following the pixels of largest value, until a connection is established with an edge of the opposite difference sign.

Edges were grouped on the number of connections and grey level difference strength between edges. There are several fundamental problems associated with this approach. The maxima, in the difference image, are based on foreground to background differences and not on the connectivity of the object. Additionally, there is no explicit way to separate outline edges from any other type of edge. Due to these deficiencies this technique was not pursued.

The technique used in DETECT, eliminates edges which are not attached to the outline of the tracked cluster. Each cluster outline is represented by the number of the cluster from which it is derived. If an edge pixel is on a particular cluster outline then the reference number is used, as an index, to increment the appropriate bin of a histogram. The edge is assumed to be part of the cluster with the largest accumulator value. The following pseudo-code, Figure 4–15, illustrates the method. The above technique simply attaches each proposed edge to the outline with the most similar path. Also, the number of pixels which correspond to each overlap between the edge and the cluster must exceed a threshold. This will eliminate all the edges which are not attached to any cluster and those which only just touch. An advantage of this technique is that it allows, at a reasonable cost, both cluster and foreground edges to be incomplete and partial, without the entire procedure failing. Additionally, the combination of two separate sources of data can control the errors of an imperfect segmentation. For example, thresholding is unlikely to be perfect and some extracted clusters will exceed the significance threshold. However it is improbable that a significant *moving* edge will also be attached to that cluster. A further defence against false edges being used for matching is implicit in the use of multiple cameras. If a false edge has attached to

```
while (notendofedgelist)
{
  - set cluster accumulators to zero
  while (notendofpixellist)
  {
    - extract the cluster number from a reference image of significant clusters.
    - increment a cluster accumulator using the reference value as an index.
  } end pixel tracking loop
  - search accumulator array and find the maximum valued index.
  - search cluster list for the cluster with equivalent index value.
  - add a pointer from that cluster to this particular edge.
} end of edge loop
```

Figure 4–15: Extraction of Outline Edges

a false cluster then the same error must occur in either two or three cameras for a correct disparity match. These two "filters" reduce the number of false matches to a minimum.

4.3 The Software Environment and Data Representation

A more detailed explanation of the DETECT software can be found in Appendix B. The DETECT system was entirely written in C, operating under UNIX, on SUN 3 and 4 Work Stations. Demonstration and debugging software was written using *sunview* libraries, allowing results to be displayed in both the SUNTOOLS and OPENWINDOWS colour graphics environments. All the initial operations of segmentation are performed on 256x256 raster scan images. As described in Section 6.2, these were captured using custom framegrabber hardware linked to a PC. Each pixel was digitised to 256 grey levels requiring 64k for each image. The lower levels of processing required the most storage due to the requirement of a background image and intermediate images. In a hardware system only the background would have to be stored completely and intermediate storage could be compressed to smaller on-chip buffers.

Data storage can also be reduced at the higher levels when using edges, thresholded images and clusters. These features can all be given a binary representation. Edge storage can be further compressed if it is known that there will not be more than one column between rows. In this case, the largest possible edge, from top to bottom, would require only 64 bytes (512 bits) for the column data plus 2 bytes for the starting offsets. Also, as will become clear in the next chapter, disparity histograms are required for each non-background edge. The range of possible disparities is unlikely to exceed 30 requiring a similar number of 8 bit accumulation registers. Each histogram could be stored with the edge as a suffix.

Cluster data structures are also fairly compact, only requiring boundaries to be stored with pointers to attached edges. Again limited storage has to be reserved for the clusters own histograms for grey level distribution and disparity. However it is unlikely that more than ten significant cluster structures will have to be stored for each frame.

Overall, the storage requirements for the above data structures are small when compared to that necessary for the background image, of which parts will have to be updated every frame. In addition, a record must be kept of stationary edges for comparison with those of the current frame. In the current software this is done by updating a list of edges, but in a hardware implementation the stationary edge map is likely to be maintained as a one bit array. This is simpler to access and has a reduced storage overhead.

4.4 Discussion of a Hardware Implementation and Storage Requirements

Section 2.5 considered some of the memory architectures possible in an image processing system such as this. A reasonable observation was that most digital chips had memory close to the processing elements to assemble regions of interest.

It seems clear that storage is a prime consideration for this type of vision system. Brief consideration is now given to possible architectural implementations of the algorithms discussed in Section 4.2. For a final system a more detailed analysis between functionality and architecture would have to be performed. Such low level algorithms are the most likely to be implemented in custom hardware, as most operations are based on neighbourhoods. Thus, caches can be used to cycle through the image, reducing access time and memory storage.

A point, which is of relevance for a hardware implementation, is the simplicity of the downwards tracking technique. If tracking is conducted in a square neighbourhood then problems arise when edges cross each other and go round in circles. Mechanisms must be employed to ensure that no endless loops and re-tracking occur. Obviously this can be done by setting tracked pixels to zero. However this would require a separate memory to store the original image being tracked. Setting tracked pixels to zero would also have to be done after the minimum edge length had been achieved, otherwise a considerable number of other edges would be wrongly broken. Also, with a downward technique there is a well defined limit to edge length: the height of the image. Thus edge storage can be reduced to a simple starting row and column, followed by a stream of binary data defining the position of the current pixel in relation to the previous. This sort of structure could be stored and accessed in off-chip FIFO buffers. Control and extraction logic could be implemented on the CMOS sensor.

In contrast to the above, the current implementation of clustering is a random process and the theoretical maximum stack possible is limited only by the size of the image. In an efficient implementation one would want to restrict the possible stack size to avoid off-chip interfaces. To this end, the clustering algorithm has been altered to track around, rather than through, a cluster. This allows the stack to be restricted to two pixel positions, one for each end of the current line. Cluster significance is now dependent on perimeter length rather than absolute size. The trials, described in Chapter Six, indicate that this does not appear to seriously increase the number of false clusters.

We still require storage of a complete thresholded image in memory. An im-

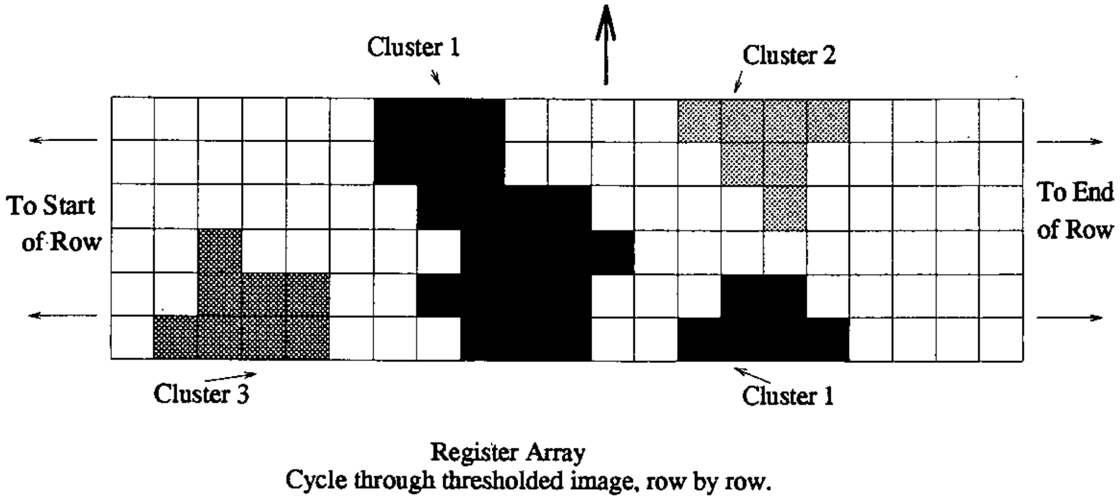


Figure 4–16: Possible Implementation of Shift Memory for Clustering

provement would allow the processing to be performed on several complete raster scans or bands as described in Figure 4–16. Every cluster will be assigned a number, which together with the boundary, will be recorded for all "live" threshold regions. Raster scans can be loaded and shifted upwards as they are produced by the sensor/ADC/thresholding circuitry. The register bank can be searched for connected regions and the boundaries extracted. An advantage of this technique is that, depending on the number of rows stored, many of the smaller insignificant clusters could be eliminated without having to store its details. A problem is that areas, of the same cluster, which are connected, on a lower row will be given different reference numbers. The simple solution is to produce a list of reference numbers indicating which clusters have become connected further down the image.

The last part of the DETECT system described in this chapter was the grouping of extracted edges with significant clusters. Three possible implementations could be:

1. **Clusters and edges held as raster scan data:** Processing would be performed on two images, one of which contained the edges, stored as reference numbers, and the other, the clusters, again stored using references. The relevant pixel values could be used as indexes to a numbered array of accumulators. Unfortunately the array sizes are unpredictable and dependent on edge thresholds, difference thresholds

and the scene. However in the current system there are usually about ten clusters and, up to, 300 significant edge segments.

2. **One data source held as raster scan and the other in list format:** This has the advantages of reducing the required storage for either the cluster or edge lists. The processor would store one in raster format and the other as a list. The lists would then be fed in and compared to the raster scan. Which was stored as which would depend on the architectural experiments together with expected operating thresholds.

3. **Both clusters and edges stored as lists:** Of the possible three combinations this technique requires the least storage. However, as usual, the price for reduced memory is an increase in processing. For each edge pixel the entire cluster list, with associated outline edges, has to be searched. Again, reference numbers would be used to index an array of accumulators.

4.4.1 Conclusion

This chapter has described the early segmentation stages of the DETECT system. These are aimed at providing a low cost solution allowing extraction of outline edges in stationary scenes. These cost considerations are particularly important at the pixel levels of computation where the largest number of calculations are required. At this level the DETECT system employs no multiplications, divisions or floating point calculations, eliminating the need for expensive silicon implementations of these functions. Consideration has also been given to possible architectures. As mentioned in Chapter Two, most image processors have local memory close to the processing elements to allow neighbourhoods to be assembled. It seems clear that the problems of memory organisation would need to be considered in any architectural study.

Further, and implicit to the idea of simple hardware, is the combination of different sources of data, to provide the outline of the clusters for matching. This is important as, often, machine vision researchers apply different processing modules, blindly, without considering possible savings inherent in the overall problem. From a practical engineering point of view systems should be specialised to an application's requirements.

The next chapter will describe how the edges and cluster outlines are used to provide estimates of interocular disparity. Although discussed, in a separate chapter, the matching process is essentially a continuation of the segmentation stages described here. Matching should be considered in terms of the outlines produced by the low level segmentations described in this chapter.

Chapter 5

Stereo Matching and Time Domain Filtering

5.1 Introduction

This chapter will describe the stereo matching and disparity estimation techniques used in the DETECT software. As described in Chapter Three, there are many different ways to solve the stereo correspondence problem. Two types appear to be evident; those based on correlation searches and those on feature matching. For reasons of complexity reduction a feature based matching system has been implemented. It is now intended to describe one such system which makes use of the fact that under certain camera configurations and object dimensions an image overlap occurs when local coordinate systems of the two cameras are aligned. Thus, if the outline edges of the object can be separated from the surrounding features then the matching problem is reduced to a computationally simple one way search along the epipolar raster scans. There is no requirement to solve the stereo correspondence problem for all types of edge. In this manner, the matching algorithms described are interlinked with the segmentation techniques discussed in Chapter Four. In particular, segmentation was designed to extract only the outline edges of a moving object.

Following on from the techniques used to match left and right features, Section 5.3 describes how a sub-pixel measure of disparity is extracted from moving objects in large scenes. Such scenes require the use of wide angle lenses resulting in significant pixel quantisation. Techniques are presented to reduce and control this problem. Brief consideration will also be given to the problems of calibration although, as suggested in Chapter Seven, this is an area for future research. As in previous chapters, hardware constraints have been taken into consideration. Here, the problems are likely to be less severe, given the reduced data rates required once segmentation has been performed. Thus, in the calculation of disparities and in filtering operations, more complex arithmetic operations are feasible. The speeds required for this type of calculation are not excessive, at frame rates of 25Hz. The chapter will end with some conclusions.

5.2 The Matching Process

Stereo matching is the process of finding *corresponding* features from two or more views of the same scene. It has many similarities with the problem of tracking features through time except that stereo disparities are caused by a spatial separation of the cameras whereas time disparities are caused by movement of the object or camera. In addition stereo disparities have predictable directions parallel to the camera displacements. Some stereo vision algorithms[54][53][76] were described in Chapter Three. However, many are computationally intensive and have been developed to imitate the human vision system. The constraints imposed by possible commercial VLSI and hardware implementations suggest that a reduction in processing would be advantageous. In effect, control of the search space for stereo matches is required.

Chapter Three also covered the various constraints which can be used to help solve the stereo correspondence problem. The two major, published, search constraints of use in this work are:

1. **The Epipolar Constraint:** If cameras are separated by a lateral translation then a feature from one camera will have its correspondent on the same raster in the other camera. Clearly, this constraint can reduce the problem of finding corresponding features to one dimension. In reality, raster scans from two spatially separated cameras will never be perfectly aligned. Some form of calibration is required to allow equivalent rasters to be compared.
2. **The Disparity Gradient Limit:** The disparity gradient limit has been shown to be an effective technique in eliminating incorrect matches[84][66]. Figure 3–4, in Chapter Three, illustrates this constraint. As the edge is tracked, disparity is calculated by simply scanning across from one edge to the other. If the disparity from one match to the next exceeds a threshold then that match can be rejected.

```
/* Stage One */
while (ǹotendofsignificantrightclusterlist)
{
  - set reference image to zero
  while (notendofattachededgestocurrentcluster)
  {
    if edge is the left cluster edge then code = 1
    if edge is the right cluster edge then code = 2
    if edge is a left edge segment then code = 3
    if edge is a right edge segment then code = 4
    while(notendofpixellist)
    {
      - calculate the calibrated row and column values
      for this particular pixel. In the current system
      this simply involves adding a translational offset.
      - set reference image pixel, using the above row and
        column to the correct code.
    } end of pixel list
  } end of attached edge loop
} end of right cluster list
```

Figure 5–1: Matching Algorithm: Stage 1

In addition to the above two general stereo matching constraints,,assistance is also extracted from what is called in this work, the *overlap constraint.* This is described in Section 5.2.1. This allows the entire object to be taken into consideration when matching is performed.

Figure 5–3 shows a block diagram of the matching procedure described in Figures 5–1 and 5–2.

Matching begins with the transfer of all the edges extracted from the main right hand clusters into a reference image. Edges are recorded in the reference image under four different codes, to represent the left and right outlines coming from the separate cluster and edge detection algorithms. It is also important to note that the edges are transferred on a cluster by cluster basis. Thus only the largest clusters with attached edges are used for matching. Following the definition of the edge types, calibration information can be added when setting the reference

```
/* Stage Two */
while (notendofsignificantleftclusterlist)
{
  while (notendofattachededgestocurrentcluster)
  {
    - set disparity histogram, for this edge, to zero
    while(notendofpixellist)
    {
      - row = row of this pixel
      - col = column of this pixel
      - startcol = column of this pixel
      while (code of the current pixel is not equal
        to the code of the reference image pixel and col ¡ IMAGESIZE)
      {
        - increment col
      }
      - disparity = col - startcol
      - update this edges histogram using ”disparity” as an index.
      - disparitygradient = lastdisparity - disparity
      - lastdisparity = disparity
      if (disparitygradient > DISPARITYGRADIENTLIMIT)
      {
        - break edge at this point
        - create new edge from the remainder
        - add new edge to list end
        - notendofpixellist = TRUE
      }
    } end of pixel list
  } end of attached edge loop
} end of left cluster list
```

Figure 5–2: Matching Algorithm: Stage 2

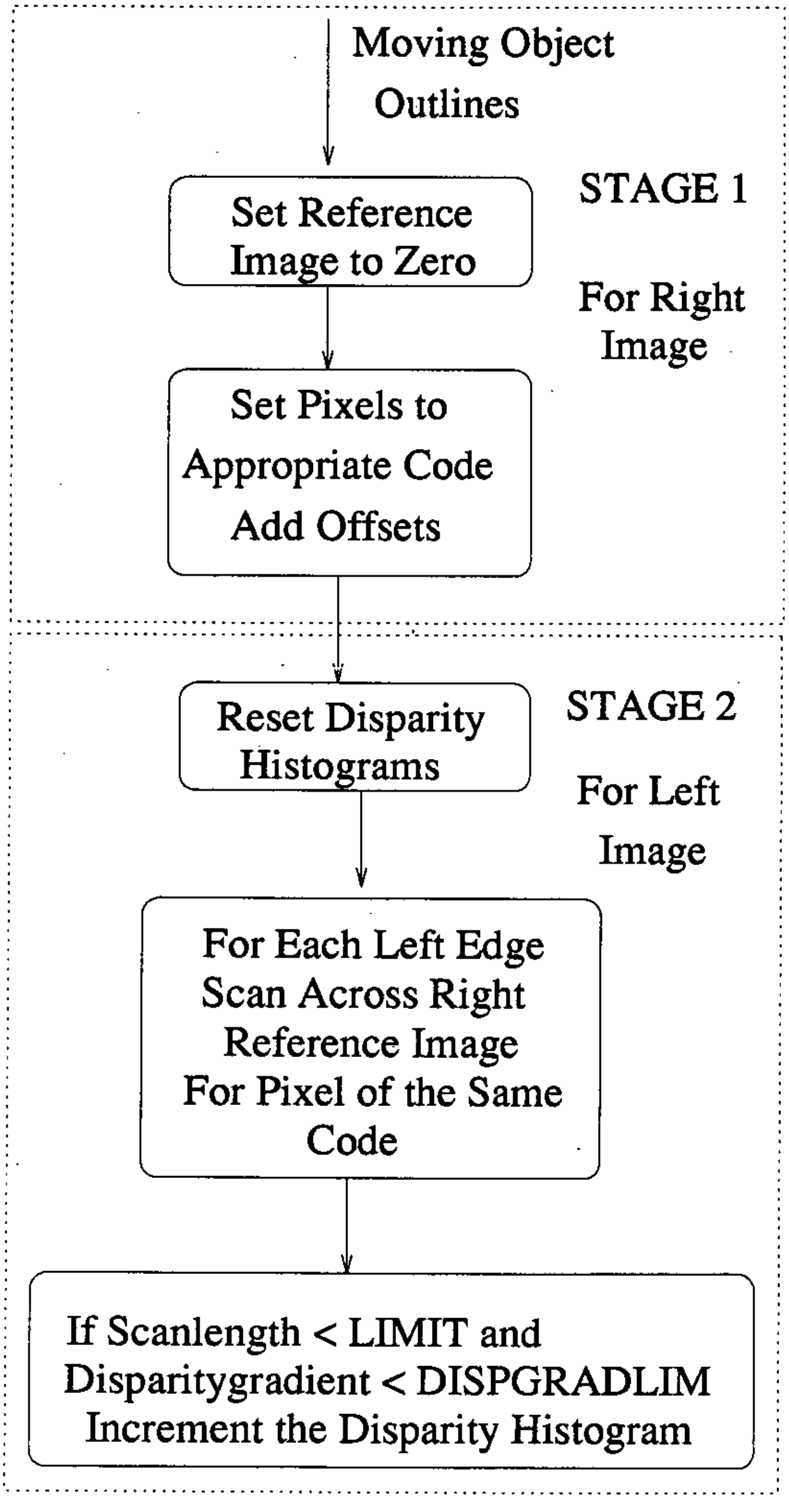

Moving Object Outlines
STAGE 1
For Right Image
Set Reference Image to Zero
Set Pixels to Appropriate Code Add Offsets
STAGE 2
For Left Image
Reset Disparity Histograms
For Each Left Edge Scan Across Right Reference Image For Pixel of the Same Code
If Scanlength < LIMIT and Disparitygradient < DISPGRADLIM Increment the Disparity Histogram

Figure 5–3: The Matching Process

image. In the current system this involves a simple (x,y) translation added to the edge coordinates. Calculating such an offset will be done by the calibration stage when the system is installed. For the applications being considered in this work, it should be possible to calculate these offsets using the trial and error approach, discussed in Section 5.6.

The second stage in the matching process involves scanning, in the x direction, from a known start point, and then finding the first pixel with the same code as that edge. The length of the scan is a measure of the disparity. As the edge is being tracked downwards, the disparity gradient limit is applied. If this exceeds a limit, usually one pixel, then the edge is re-segmented and a new edge formed from the remainder. It is unlikely that false matches will track with a low disparity gradient and a re-segmentation allows a later separation of good matches from false. Such a "goodness" calculation would complicate custom hardware due to the likely divisions involved. It has therefore been kept separate from the main matching algorithm and would be implemented on a microprocessor.

Several techniques, inherent in the above two stages, reduce the possibilities of false matches. The use of codes ensures that edges only match their own kind: an edge attached to the left hand side of a cluster will only match with another edge attached to the left hand side of a cluster. In addition, some edges are directly extracted from the cluster itself. The significance of these in the final disparity calculation can vary depending on the segmentation for a particular frame. However they add an extra layer of safety to the final results. In summary, once the outline edges have been extracted, matching can be performed by a simple scan from a left image edge to a right image. Thus the correspondence problem has been transferred to the segmentation stages of processing and explicit matching searches have been avoided.

Attention is now turned to the geometry which allows this type of matching. In particular the geometry required to ensure that an object s images in the left and right cameras overlap when their local origins are aligned.

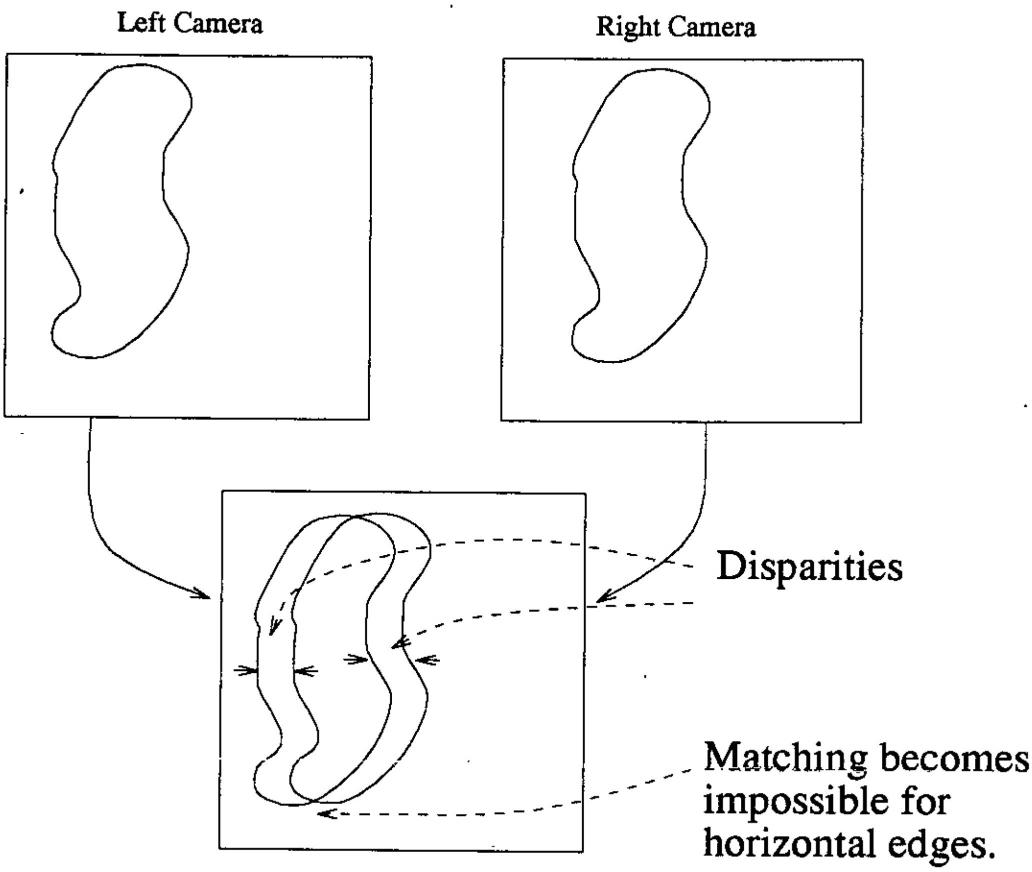


Figure 5–4: Overlap Matching

5.2.1 The Overlap Constraint

The above technique, which has not appeared in the surveyed literature, employs the fact that an object's image in two spatially separated cameras will overlap when their local origins are aligned. Figure 5–4 shows two simplified images where a blob is projected onto both cameras. The cameras are assumed, unrealistically, to be perfectly calibrated and spatially separated. Thus a simple translation will allow the local origins to be aligned. Such a matching algorithm requires that corresponding edges from two cameras be positioned, in the reference image, as close to one another as possible. The chances of other false edges "getting in the way" obviously increases as edges are separated either by translational offsets or by genuine disparity. In this respect it is important to be able to calculate the geometric conditions when an overlap will not occur and ensure that the camera is set up correctly.

The following analysis derives a general formula to calculate the disparity-overlap ratio for two idealised images of the same object. Figure 3–1, in Chapter Three, shows a *general* two camera stereo arrangement. For simplicity we will assume that there is no rotation around the x or z axis and reduce the problem to that in Figure 5–5. The problem is now two dimensional and the left coordinate system can be projected onto the right using Equation 5.1. θ represents the angle of rotation between the two coordinate systems and (x_T, z_T) the translation between the two origins.

$$\begin{pmatrix} x' \\ z' \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} + \begin{pmatrix} x_T \\ z_T \end{pmatrix} \tag{5.1}$$

An object of width W is now placed in the scene and its end points, L and R are projected onto the two camera image planes through the focal points LF and RF. L has coordinates (X_{LL}, Z_{LL}) in the left hand camera coordinate system and (X_{RL}, Z_{RL}) in the right hand coordinate system. R has coordinates (X_{LR}, Z_{LR}) in the left hand camera coordinate system and (X_{RR}, Z_{RR}) in the right hand coordinate system. The projected points are LR (x_{LR}, z_{LR}), LL (x_{LL}, z_{LL}), RR (x_{RR}, z_{RR}) and RL (x_{RL}, z_{RL}). The following equations represent these projections, through focal lengths of F, onto the image plane.

$$x_{LR} = \frac{FX_{LR}}{F - Z_{LR}} \quad x_{LL} = \frac{FX_{LL}}{F - Z_{LL}} \quad x_{RR} = \frac{FX_{RR}}{F - Z_{RR}} \quad x_{RL} = \frac{FX_{RL}}{F - Z_{RL}} \tag{5.2}$$

Before the parameters, which define disparity are used, it is necessary to translate the right hand camera coordinate system onto the left using Equation 5.1. Thus,

$$x'_{RR} = \frac{F(X_{LR}cos\theta + Z_{LR}sin\theta + x_T)}{F + X_{LR}sin\theta - Z_{LR}cos\theta - z_T} \tag{5.3}$$

$$x'_{RL} = \frac{F(X_{LL}cos\theta + Z_{LL}sin\theta + x_T)}{F + X_{LL}sin\theta - Z_{LL}cos\theta - z_T} \tag{5.4}$$

Figure 5–6 shows the position, in a combined image, of the above parameters, x_{LR}, x_{LL}, x'_{RR}, x'_{RL}. It allows a definition of both overlap, O_{lap}, and disparities, δ_1 and δ_2, as in Equations 5.5 and 5.6.

$$\delta_1 = x'_{RR} - x_{LR}, \quad \delta_2 = x'_{RL} - x_{LL} \tag{5.5}$$

$$O_{lap} = x_{LR} - x'_{RL} \tag{5.6}$$

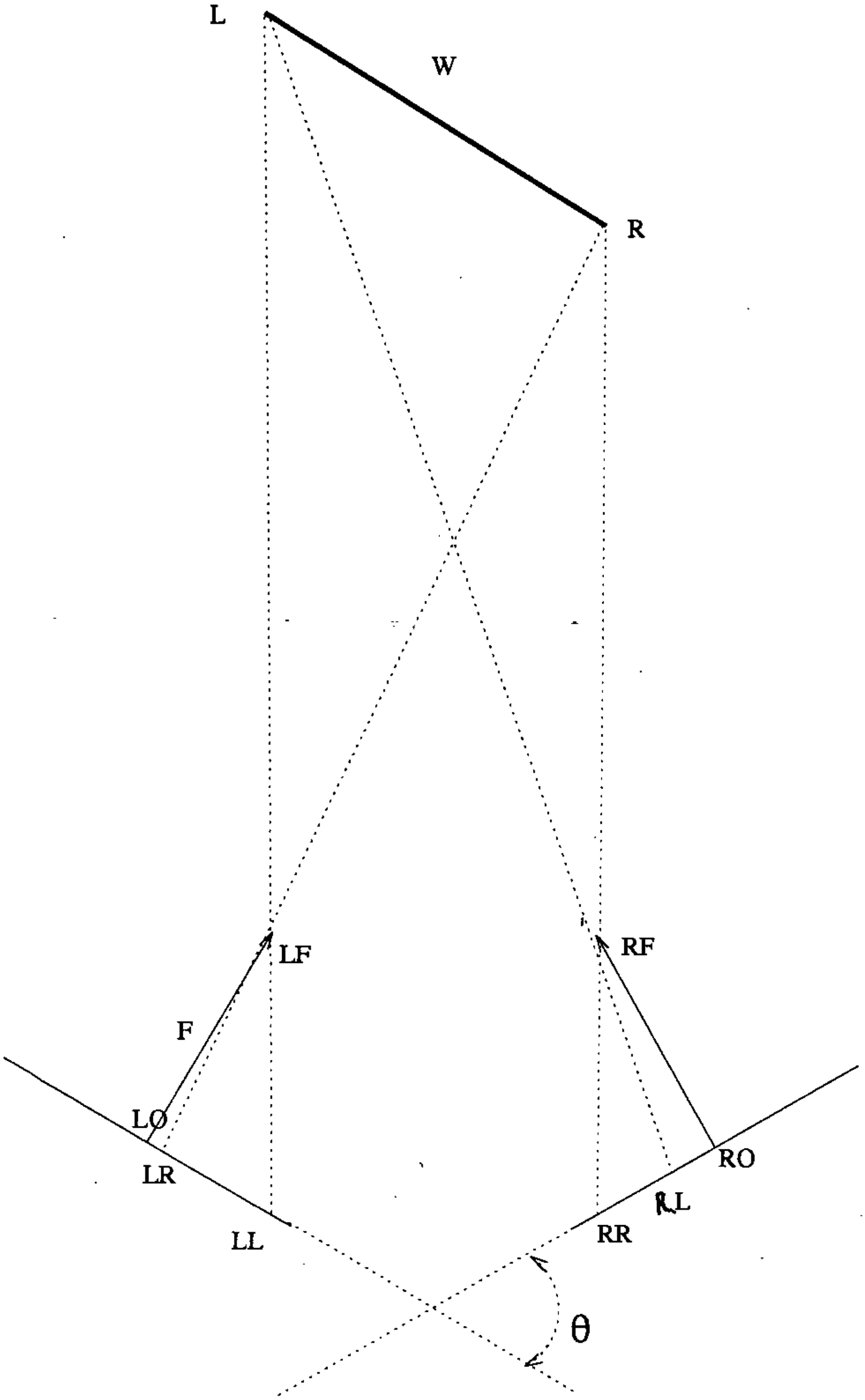


Figure 5–5: Stereo arrangement with no roll nor tilt

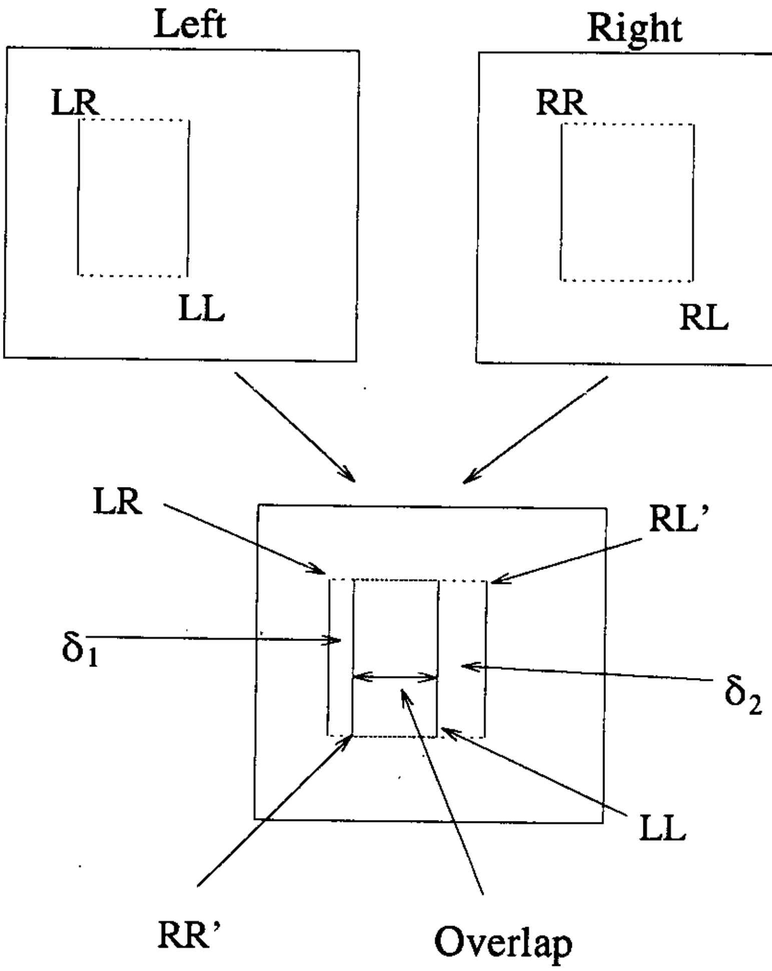


Figure 5–6: Overlap and Disparity Definition

When the object is thin, in relation to the range, then the two values of disparity, δ_1 and δ_2, may be assumed identical. This assumption is true for the scenes and equipment employed in this work where an alarms requirements will likely lead to short focal lengths and longer ranges. We can now say that $\delta_1 = \delta_2 = \delta$.

The geometry can again be simplified by assuming that $\theta = 0$ and $z_T = 0$ as shown in Figure 5-7. For mechanical reasons this is the camera geometry used in the trials, described in the next chapter. Equations 5.3 and 5.4 now reduce to Equations 5.7 and 5.8.

$$x'_{RR} = \frac{F(X_{LR} + x_T)}{F - Z_{LR}} \tag{5.7}$$

and

$$x'_{RL} = \frac{F(X_{LL} + x_T)}{F - Z_{LL}} \tag{5.8}$$

If the angle, ϕ, which the object makes with the imaging plane, is zero, and $W = X_{LR} - X_{LL}$, then $Z_{LL} = Z_{LR} = Z$ and Equation 5.6 can be replaced by Equation 5.9.

$$O_{lap} = \frac{F(W + x_T)}{F - Z} \tag{5.9}$$

Also, disparity can be represented by Equation 5.10.

$$\delta = \frac{F x_T}{F - Z} \tag{5.10}$$

From Equation 5.9 it can be seen that an overlap will always occur if the interocular distance, x_T, is less than the width of the object, W [1]. Further, Equations 5.9 and 5.10 can be combined to calculate the overlap/disparity ratio, R.

$$R = \frac{O_{lap}}{\delta} = \frac{W + x_T}{x_T} \tag{5.11}$$

The important point to note is that R is constant throughout the entire scene. It is dependent entirely on the geometry of the camera and object and is independent of position in the scene.

One must now consider what happens to an overlap, and therefore the matching algorithm, when an object rotates in the scene. If the object width parallel

[1] O_{lap} becomes negative at this point.

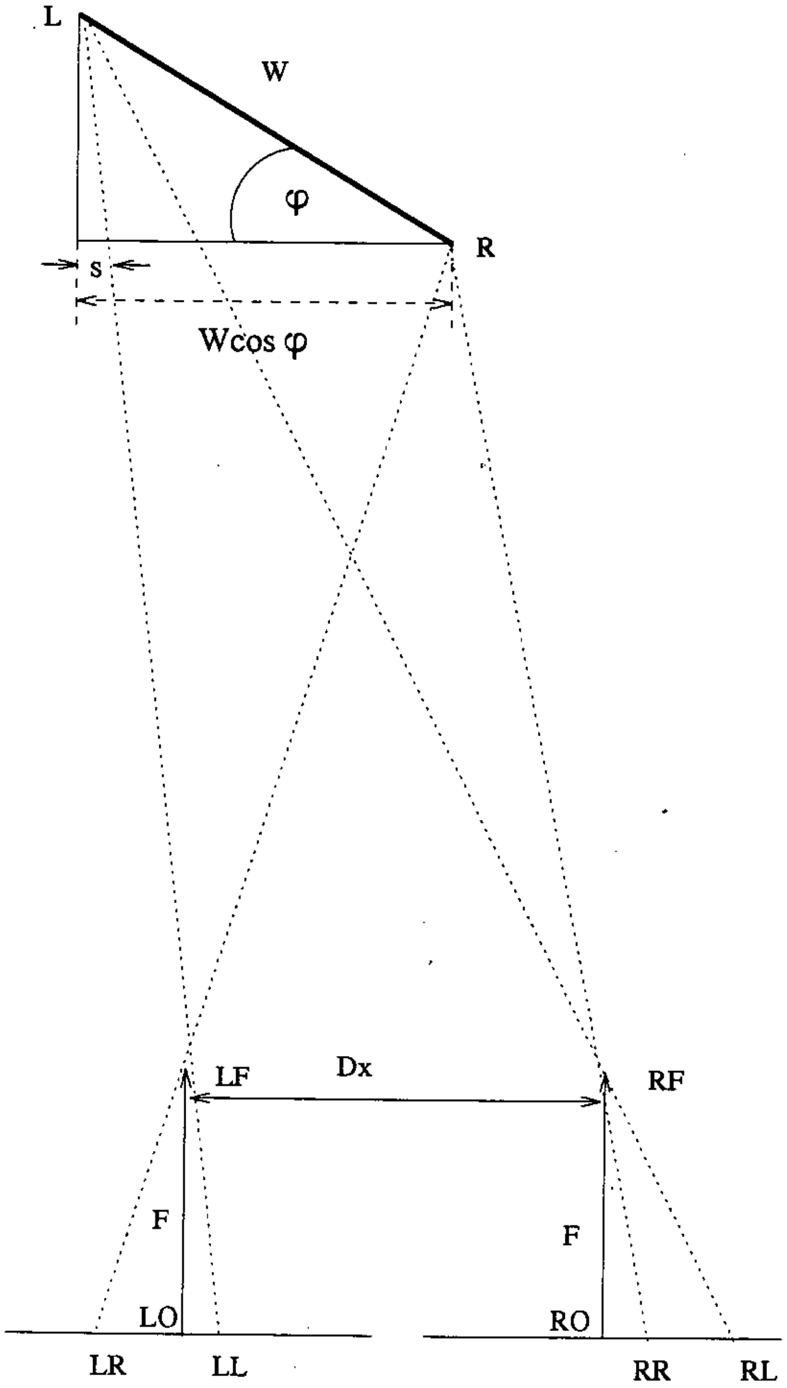


Figure 5–7: Simplified Camera Geometry, Showing Limiting Condition for Overlap ($x_{RR} = x_{LL}$)

to the imaging plane is greater than the translation between the cameras, the interocular distance, then overlap will always occur. However, if an object rotates, by some angle ϕ, the object will eventually separate in the two images. Equation 5.12 is the limiting condition for overlap to occur when an object rotates. This is extracted from the geometry shown in Figure 5–7. Thus D_x is the interocular distance where there is no overlap and $x_{LL} = x_{RR}$. D_x is constructed by subtracting s from the distance $Wcos\phi$, where s is found using similar triangles, such that $s = \frac{X_{LL}Wsin\phi}{F}$. The main point to note is that, due to the sine term, overlap is now position dependent.

$$D_x = W\left(\cos\phi - \frac{X_{LL}}{F}\sin\phi\right) \tag{5.12}$$

The general conclusion from the above analysis is that, for this algorithm, it will thus be impossible to match objects for which the width parallel to the imaging plane is narrower than the distance between the cameras. In the experiments described in Chapter Six the camera rig was set up with this in mind. Such a restriction is not a serious problem in this application and is, in fact, an advantage. It prevents the extraction of disparities for spurious objects which fall below a physical size threshold. The extraction of such disparities is now described.

5.3 Disparity Extraction

The previous section dealt with the mechanics of finding correspondences for outline edges between two different images. Explanations of the disparity gradient limit and overlap were also provided. The result of the above matching process is a collection of clusters with attached edges and associated disparity histograms. These histograms are now used to provide estimates, to sub-pixel accuracy, of the current main cluster's overall disparity. Section 3.5.2 discussed some of the accuracy issues related to the extraction of disparity. Pixel quantisation noise and mis-matches will cause the disparity to spread over several values of the histogram. As we are using short focal lengths, these errors become more significant, especially, when tracking longer range objects.

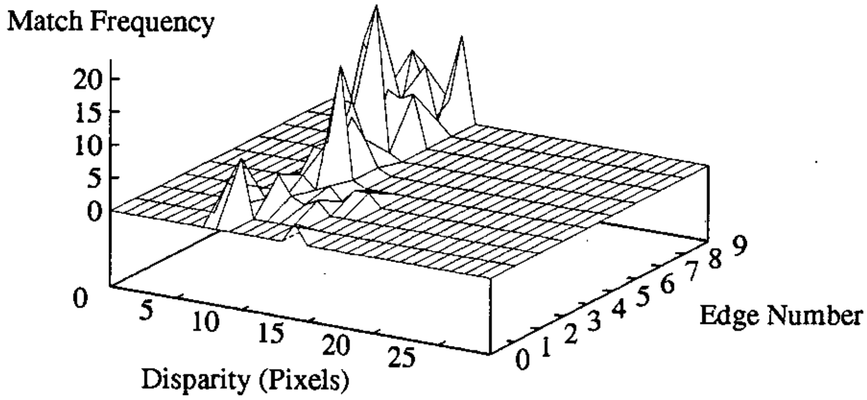


Figure 5–8: Edge Disparity Histograms for One Cluster

Figure 5–8 shows the disparity distributions for the three edges from an extracted cluster. To improve the cluster's disparity distribution, measures of confidence can be assigned to each individual edge.

The current implementation of DETECT uses the mean and variance of relevant parts of the disparity histogram as measurements and associated confidence. Once a cluster has been detected in the scene and is being tracked, disparities from previous frames are used to restrict those allowed in the current frame. Thus the means are calculated on a specific band of the histogram around the major peak. Provided enough pixels are matched and the usual Gaussian assumptions applicable, the mean provides a sub-pixel disparity measurement for the entire object. Figure 5–9 describes how this is done for the edge clusters. Example disparity histograms, plotted through time for each of the three measurements, are shown in Section 6.3.2.

Consideration was also given to weighting match frequencies according to their distance from the previous frames disparity and in relation to other edges in that cluster. A problem associated with this type of weighting is that, while reducing

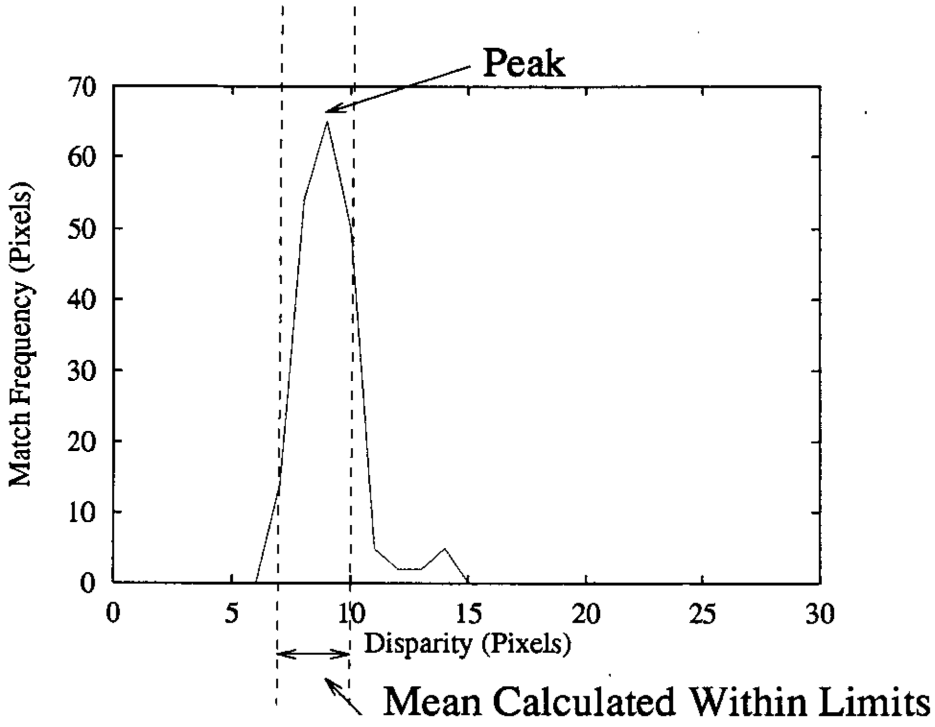


Figure 5–9: Disparity Extraction from Matching Histograms.

the contribution of x disparities,It distorts the distribution for a particular edge. Thus it is not currently used.

An important feature of this work is that a measure of the error is inherently provided by the calculation of the variance of the disparity. This not only takes into consideration the errors caused by quantisation but also those caused by inaccurate feature matching. Such combined measures, as discussed in the next section, can be utilised in the tracking filters described in Section 5.5.

5.4 Disparity Histogram Error Analysis

With the three camera rig suggested, by Figure 3–15 in Chapter Three, three disparities can be extracted: two from the inner pairings and one from the outer cameras. The error PDF's shown in Figure 3–15 cover the entire scene. However as described in Section 3.5.2 errors vary with distance from the camera. Thus different points in the scene will have different PDF's. Probably, in any final system, it would be necessary to calculate PDF's for separate range bands in the scene. This would allow confidence envelopes to be calculated for particular disparities.

In view of the error considerations, described in Chapter Three, the DETECT system has employed three cameras to estimate an overall disparity. This reduces the combined effects of pixel quantisation noise and the matching errors of an individual point in the scene.

Consideration is now given to estimation of combined errors for the three stereo measurements from a triple camera rig. The disparities from each possible measurement from three cameras are *not* independent. This is clear from the fact that a poorly extracted edge from the left camera will cause inaccuracies in two out of the three measurements. In this application we assume that the errors in feature *extraction* are independent and calculate our error covariance matrix for feature *matching* on this assumption. The advantage of this approach is that it provides a *combined* variance for quantisation errors and feature matching errors.

The three possible disparity measurements are represented by

$$\delta_1 = x_1 + \eta_1 - x_2 - \eta_2 \qquad \delta_2 = x_2 + \eta_2 - x_3 - \eta_3 \qquad \delta_3 = x_3 + \eta_3 - x_1 - \eta_1 \tag{5.13}$$

where x_i is the edge position with respect to the local coordinates and η_i is noise. δ_1 is the disparity between the left and middle cameras, δ_2 is the disparity between the middle and right cameras and δ_3 is the disparity between the two outer cameras. A false match is considered part of the noise. Thus the errors in disparity, Δx_i, can be summarised as

$$\Delta x_1 = \eta_1 - \eta_2 \quad \Delta x_2 = \eta_2 - \eta_3 \quad \Delta x_3 = \eta_3 - \eta_1 \tag{5.14}$$

and considered as combinations of independent noise sources η_i. From this an error covariance matrix can be derived based on the experimentally calculated values of Δx_i. The error covariance matrix can be represented by

$$Cov(\epsilon) = E[\mathbf{\Delta x \Delta^t x}] \tag{5.15}$$

where the main diagonal elements, t_{ii}, are $E[\Delta^2 x_i]$. The other elements in the matrix are

$$t_{12} = t_{21} = \frac{t_{33} - t_{11} - t_{22}}{2} \tag{5.16}$$

$$t_{23} = t_{32} = \frac{t_{11} - t_{22} - t_{33}}{2} \tag{5.17}$$

$$t_{13} = t_{31} = \frac{t_{22} - t_{33} - t_{11}}{2} \tag{5.18}$$

It is important to note that the values of t_{ii} are simply the variances of the extracted disparities. They can be extracted from the disparity histograms and then used to calculate the other elements of the matrix. We have used the $Cov(\epsilon)$ matrix in the Kalman formulation where the disparity velocity is modelled as signal noise.

5.5 Filtering Techniques

Figure 5–10 shows the disparity output curves from a 16 frame sequence. In the main the results are fine. However there are problems with sudden large spikes and errors in particular frames. This section is about the control of such errors and the combination of the three disparities such that these spikes are eliminated in the final averaged disparity trace. Additionally, it would be useful if a confidence measure could be provided to allow the calculation of alarm thresholds. Adaptive filtering techniques can be used to combine current information with that from previous frames and from other measurements.

Several common techniques have been developed to smooth such time dependent data series. The two considered here are:

1. Least Mean Squares
2. Kalman Filtering

The Least Means Squares, (LMS), algorithm has been applied generally in many areas of signal processing[31]. The following equations represent the basic

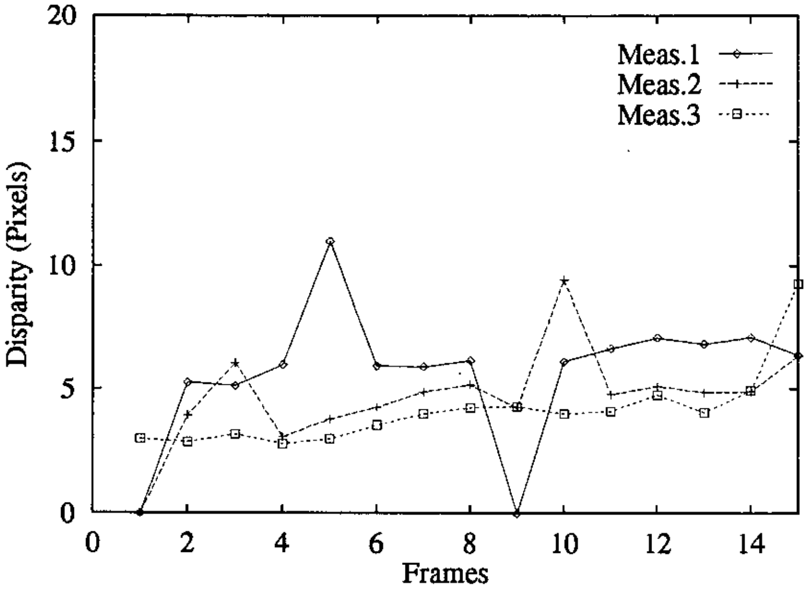


Figure 5–10: Extracted Raw Disparities for Sixteen Frame Sequence

structure of the algorithm:

Filter output,

$$y(n) = \hat{\mathbf{w}}^{\mathbf{T}}(n)\mathbf{u}(n) \tag{5.19}$$

Estimation Error,

$$e(n) = d(n) - y(n) \tag{5.20}$$

Tap-Weight Adaptation,

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)\mathbf{e}(n) \tag{5.21}$$

where $y(n)$ is the estimated output signal for time step n, $\hat{\mathbf{w}}(n)$ is the vector of weights to produce $y(n)$. $\mathbf{u}(n)$ is the input vector of measurements. The estimation error, $e(n)$, is calculated from the difference between the training signal, $d(n)$, and the current estimate. It is then used to calculate the next set of weights with respect to a convergence parameter, μ.

In terms of the current application of stereo tracking, problems arise with the LMS formulation. The first and most significant problem is that a training signal is required to calculate the initial values of the filter weights. In any practical installation, it will be very difficult to calculate true disparities for the current object position. Secondly, the calculated weights will only be valid for a single series of disparities, ie. the ones that were used as the training signal. In wide angle systems it is likely, that the object in the scene will be free to move anywhere, and in any direction. One set of weights will not be sufficient for the many possible tracks through an individual scene and no account is taken of the current velocity or acceleration. The third problem associated with the LMS, and similar algorithms, is that it is difficult to combine information from different measurements. It therefore does not utilise all the available information.

Kalman filtering provides an alternative to the LMS algorithm which can explicitly model the object's motion in a scene. It can also be formulated to take account of interdependencies between the three extracted disparities. It is a recursive technique which uses *a priori* knowledge about the uncertainties associated with particular measurements of the state vector. In this work the state vector could be a measurement of the three possible disparities, or depths, from a three camera stereo rig. Each of these measurements has a variance associated with it. An estimate of the current state, based on the measurement error covariance matrix is maintained, with the system error matrix which models the likelihood of changes in acceleration. The formulation applied in this project is based upon examples and theory presented by Haykin [31], Matthies [55], Hwang [12] and Bozic [11]. The equations for a vector implementation are stated below:

Estimate

$$\hat{\mathbf{x}}(k) = \mathbf{A}\hat{\mathbf{x}}(k-1) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{C}\mathbf{A}\hat{\mathbf{x}}(k-1)] \tag{5.22}$$

Filter Gain

$$\mathbf{K}(k) = \mathbf{P_1}(k)\mathbf{C^T}[\mathbf{C}\mathbf{P_1}\mathbf{C^T} + \mathbf{R}(k)]^{-1} \tag{5.23}$$

$$\mathbf{P_1}(k) = \mathbf{A}\mathbf{P}(k-1)\mathbf{A^T} + \mathbf{Q}(k-1) \tag{5.24}$$

Error Covariance Matrix

$$\mathbf{P}(k) = \mathbf{P}_1(k) - \mathbf{K}(k)\mathbf{C}(k)\mathbf{P}_1(k) \tag{5.25}$$

The $\hat{x}$ matrix is the state vector being estimated at a time step represented by k. In this application it consists of the three possible disparities extractable from a three camera rig and is therefore a 3x1 vector. y is the measurement vector and K the Kalman gain matrix, which is calculated from 5.23 and 5.24. In reference to Equations 5.22 to 5.25, we can approximate the change in disparity, between frames, using the following vector equation.

$$\boxed{\mathbf{x}(k) = \mathbf{A}x(k-1) + \mathbf{w}(k-1)} \tag{5.26}$$

$\mathbf{A}$ is the state transition or signal dynamics matrix. The $\mathbf{w}$ vector is a 3x1 vector containing the expected disparity changes between frames. This is dependent on the speed and depth of the moving object and can theoretically be altered as the system is running. However sudden jumps in the system error are liable to make the filter unstable. Thus in the current implementation the values of $\mathbf{w}$ are kept constant. These could be extracted, from raw disparities, during equipment installation with a person walking backwards and forwards in the scene. Equation 5.27 re-expresses Equation 5.26 in it's matrix form.

$$\begin{pmatrix} \delta_1(k+1) \\ \delta_2(k+1) \\ \delta_3(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_1(k) \\ \delta_2(k) \\ \delta_3(k) \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \tag{5.27}$$

The second part of the Kalman formulation is the measurement equation. This can be defined as shown in Equation 5.28 and expanded to the matrix system shown in Equation 5.29.

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \tag{5.28}$$

$$\begin{pmatrix} \hat{\delta}_1(k) \\ \hat{\delta}_2(k) \\ \hat{\delta}_3(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_1(k) \\ \delta_2(k) \\ \delta_3(k) \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{5.29}$$

The values of $\mathbf{v}$ are the variances of the disparity measurements and represent the expected noise in the value of $\mathbf{y}$. Again, these can be extracted from the raw disparity histograms by averaging disparity variances over a series of measurements.

Using the matrices described in the Equations 5.27 and 5.29 a Kalman filter was programmed to estimate the three possible disparities for which the results are described in the next chapter and Appendix A.

Apart from the above, where the system matrices are defined with velocities included as noise, two other formulations were attempted on the basis that they model reality more accurately. The first included a single common velocity as part of the state vector $\mathbf{x}$. Acceleration, $\Delta\delta$, of this velocity was then defined as the system noise. Equation 5.30 describes the situation where w is an expected change in acceleration with zero mean.

$$\begin{pmatrix} \delta_1(k+1) \\ \delta_2(k+1) \\ \delta_3(k+1) \\ \Delta\delta(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_1(k) \\ \delta_2(k) \\ \delta_3(k) \\ \Delta\delta(k) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ w \end{pmatrix} \tag{5.30}$$

The second system attempted used a different velocity state for each estimated disparity. Both these systems were programmed and tested on the trial data. However the results were not satisfactory and tended to be unstable. Better results may be achievable for longer sequences of images.

At this point it is worth considering the non-stationary nature of the stereo system. Measurement variances will change through time as an object crosses different backgrounds and mis-matches come and go. Such variances could, in theory, be used to constantly update the error covariance matrix. Unfortunately, variances will tend to stay constant and then suddenly jump as different backgrounds are crossed. These are likely to cause the Kalman Gain Matrix to become unstable. The integration of non-stationary time series into the Kalman and other filter formulations is a possible future area of research.

5.6 Calibration

The problems of accurately calibrating spatially separated cameras and automatically establishing the true translations and rotations were discussed in Section 3.6. Apart from the algorithm's unreliability, it is clear that a low cost VLSI implementation of the techniques described would be impractical, due to the required arithmetic. One possibility would be to provide an interface to a portable computer which would perform the necessary arithmetic and then download the appropriate calibration offsets. However, for this application accurate camera calibration and explicit extraction of metric distances is not required and processing can be done in disparity space.

This still leaves the problem of calibrating translational and, possibly, rotational offsets necessary before matching can be started. Dealing first with rotation, it is the experience of the experiments in the next chapter that rotation was less of an inhibition to matching than at first thought. In all cases the matching algorithm managed to provide a disparity trace which correlated with the person's movement through the scene. The main reason for this is that we are extracting an *average* disparity for the entire object and not for any particular edge. Compared to the accuracy in alignment possible when using a PCB with three mounted sensors the equipment used in this work was relatively crude. Thus, in a final implementation it is likely that rotation will not be a significant source of error.

In contrast, translational offsets were required in the trials described in Chapter Six[2]. Depending on the architecture and mechanical set-up they may or may not be required in a final system. One possible technique, which could be an area for future research, would be to repeatedly try different offsets on a sequence of

[2] Translational offsets can, sometimes, be used to increase the chances of an accurate match.

images where the object motion was known. This could be done until a consistent series of disparities were extracted from the sequence. Using a more accurate camera rig, ie., PCBs, the number of possible offsets could be limited, reducing the search.

An extension to the above, which is described in Section 6.4.3, is to run the algorithm or installation with objects of known position. This could be performed for several points in the scene and used to build up a rough disparity map of the scene. Section 6.4.3 described experiments where this was done.

5.7 Hardware Implementation

This section will discuss a possible hardware implementation of the above matching algorithm and attendant time domain analysis. It follows on from Section 4.4 in the last chapter, where implementation in VLSI was considered for the lower level processing. To complete the picture, Figure 5–11 shows a possible architecture using a microprocessor and three ASIS processors. Not included on this diagram are the DAC's and ADC's required. These are shown with the correct relationships, to other processing elements, in Figure 5–12. The basic idea behind such an architecture is that all the low level processing up to stereo matching would be performed on specialised hardware. This has been made possible by the fact that there are no multiplications, divisions or floating point calculations during these processing stages. Further efficiencies can be obtained by implementing some functions, which act exclusively on local areas of the image, in analogue. The shaded areas in Figure 5–12 show this for edge detection, differencing and difference thresholding. The algorithms described in Chapter Four have been adapted to allow this. For example, edge detection has been reduced to a lateral scan which is ideal for processing raster scan data. Other functions, such as tracking and clustering, are less likely to be developed in analogue due to the unpredictable neighbourhoods upon which this type of processing has to be carried out. However, experiments with regions of interest and addressing random parts of the imaging array may allow an analogue implementation of these functions as

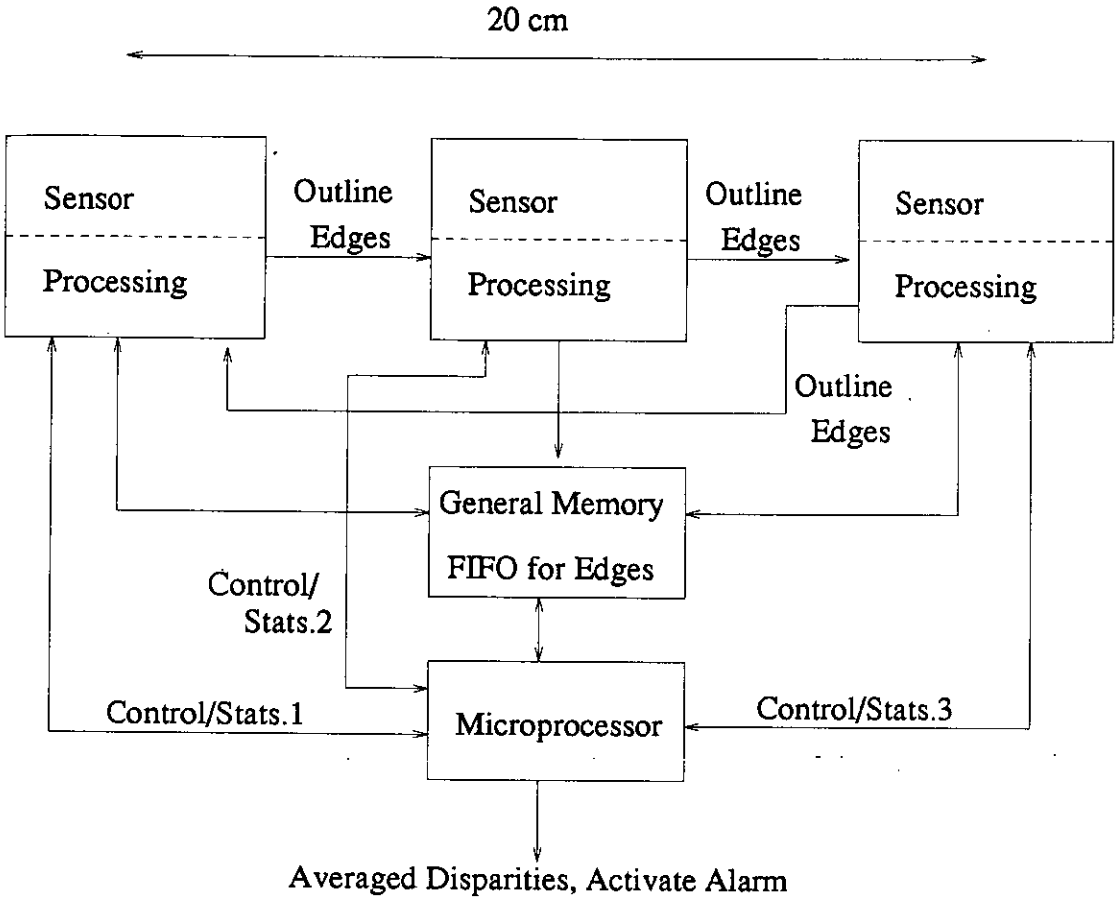


Figure 5–11: Hardware Overview of a Possible Detect Implementation

well. Locational information could be fed back from the higher levels of processing directly to the imager array. There appears to be little in the hardware literature describing such an approach and it could be a fruitful area of future research.

A crucial area in such an architecture will be the required memory and how it is organised. By implementing the lower functions in analogue, the system will avoid costly storage and processing of grey level pixels. The processing of binary information instead of grey levels may also allow the use of an on-chip memory for outline extraction and stereo matching. An interesting trade-off would be between the techniques used to store edges and the required on-chip memory. A 256x256 array is indicated in Figure 5–12. However this could be reduced if some form of spatial organisation was imposed on the edge storage. In the DETECT software, edges are stored as lists of displacements from an initial row and column. The

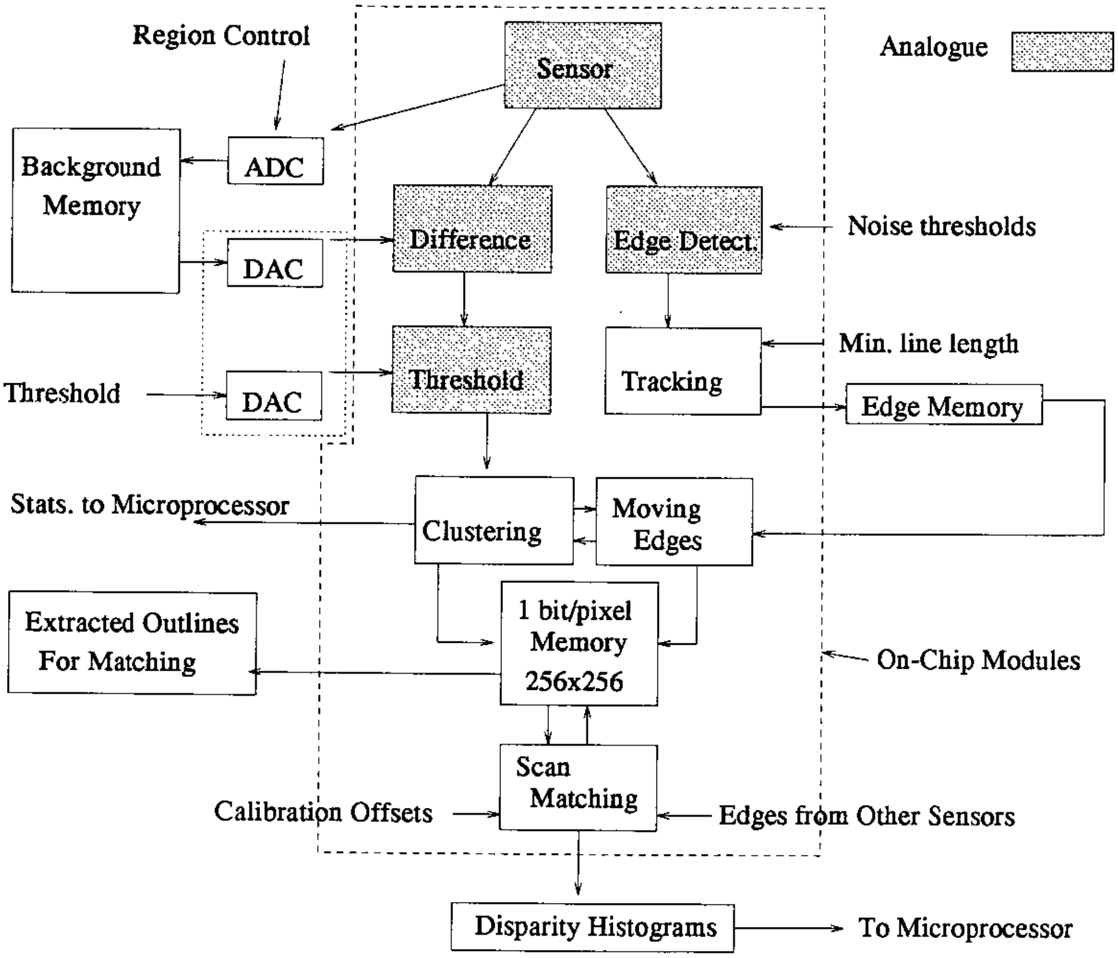


Figure 5–12: Possible Architecture for ASIS Implementation

obvious way to store these lists is in a FIFO. In an implementation these edges would be loaded into the on-chip array for outline extraction with clusters and stereo scan matching. To reduce the size of this on-chip buffer, edges from the four quarters of the image could be stored and processed separately. In this situation the on-chip memory could be reduced from 8Kb to 2Kb. Clearly, more reductions could be made, at the likely expense of functionality, by dividing the edge information further.

Use of the same on-chip memory could be made by the scan matching part of the algorithm. In this case, edges from other cameras will have to be loaded with those from the current sensor. Unfortunately the matching algorithm works better if different types of edge, ie., left and right are matched separately. The current DETECT software uses codes to differentiate between types of edge. Due to the one bit nature of the above memory it is likely that this option will not be available. Different types of edge will have to be loaded into the array separately. Also, at this stage, calibration offsets require to be added to each edge location before being recorded in the array.

As said above, Figure 5–11 shows the overall architecture with connections between microprocessors, background, edge and disparity memories and ASIS processors. To allow all processors to perform scan matching it is necessary to pass edges between the different cameras. Some form of communication link will be required to accept and transmit edges. Also, shown in both figures are interfaces to the controlling microprocessor. This will be used to implement the threshold calculations based on integer statistics and histograms from the sensors themselves. The arithmetic required should not be too severe at a maximum of 25 frames per second, if the statistics from the processing are presented in a compact manner. It may also be possible to perform the Kalman filter matrix operations using the same microprocessor, again due the relatively low calculation speeds of 25Hz.

In conclusion to this section, several general points are worth stating. Firstly, it seems very important that local processing of grey levels is performed in analogue avoiding the space costs of large busses and temporary registers. Secondly, there is a trade-off between the compactness of the data storage for edges and the use of on-chip buffers for temporary storage. As the on-chip memory is likely to cost more than the standard off-chip edge store, it would make sense to group edges in memory according to their position in the overall image. The third feature involves the techniques required to calculate thresholds, image statistics and perform time domain filtering. It will not be worthwhile implementing this type of calculation on chip. However relatively simple processing can be used to accumulate the statistics required by a microprocessor into compact histogram forms. The above discussion gives a good idea of what sections of the algorithm are implementable on the sensor and what would be best left to more general purpose microprocessors. Although, in a final implementation, the designer would have go into more detail, it is hoped that this description will give some idea of general possibilities of the DETECT algorithms.

5.8 Conclusions

An original stereo matching algorithm has been developed with the aim of being implementable in VLSI hardware. The matching process is interlinked with the segmentation stages of the lower level processing described in Chapter Four. Only outline edges are used as matching primitives. The matching algorithm takes advantage of the fact that for cameras on the same imaging plane objects overlap when there local origins are aligned. With this camera geometry, an overlap will occur provided the object is wider than the lateral translation separating the cameras. Further to the overlap constraint, the disparity gradient is applied as edges are being matched, pixel by pixel. Edges are re-segmented at points which break a disparity gradient limit. This, together with limits on the allowed disparities should remove the vast majority of false matches.

Following the matching of the extracted outlines, disparity histograms are calculated for edges. These can allow the elimination of poorly matched edges, on the basis of disparity variance and amplitude. False and unmatched edges created by the application of the disparity gradient limit, can also be eliminated using the same variance and amplitude thresholds. The final part of the current DETECT system uses the confidences and error estimates available from the disparity measurements in a Kalman filter. Estimates are extracted experimentally from the trial data and combine both matching and localisation errors.

Later sections of this chapter discussed calibration and hardware issues relating to the DETECT system. Compared to other stereo applications, calibration is less of an issue, as accurate metric estimates of distance are not required. It appears from the results in the next chapter, that simple translational offsets are adequate. Finally, hardware was dealt with in Section 5.7 with a general description of those parts of the system which would be implemented on an ASIS sensor and those, for which, it would be more sensible to use a microprocessor.

Chapter 6

Results

6.1 Introduction

Whereas the last two chapters have dealt with algorithmic concepts, this chapter will consider the system's operation. In particular, a series of trials will be described, showing DETECT's ability to track and estimate disparity for moving objects.

The chapter will start with a description of the hardware and software used to capture the images. A significant part of this work involved the design and construction of equipment capable of simultaneously capturing stereo pictures from CMOS cameras. CMOS cameras, mentioned in Chapter One, allow application specific processing on the same substrate as the sensor. Pixel aspect ratios can also be altered according to an implementation's requirements. For example, if the expected edges are vertically oriented it might be sensible to have a similarly oriented pixel. One other advantage of the ASIS architecture, is the on-chip generation of signals, such as "pixel valid" and "frame start", useful in image digitisation. Most frame grabbers have to estimate when a pixel should be sampled using video line signals. The pixel valid signal, from the ASIS sensor, emanates directly from the chips own clock and is automatically synchronised with the video waveform. Sampling errors can therefore be reduced at a minimal cost in frame grabber hardware.

When capturing stereo pictures, synchronisation of separate video sources and framegrabbers is also necessary. The majority of low cost commercially available framegrabbers allow, incoming video signals to be multiplexed into the same grabber on a frame by frame basis. However, parts of human hands and legs can move significantly in one frame. Stereo mismatches and disparity errors will inevitably occur between frames captured at different time intervals. As an alternative, several boards can be run together on the same computer with synchronisation being performed by the computer. Such a technique is limited by the fact that each framegrabber board will have to be addressed separately. Synchronisation is limited by the speed of the computer and operating system. Another major restriction was that reasonably costed commercial boards are limited by the memory available to store sequences of image, usually around 4 frames per card.

Video signals must also be synchronous. Most commercial, and non-expensive, CCD video cameras do not allow the input of a synchronisation signal. The ASIS range of CMOS cameras have been designed to allow both the input and generation of a SYNC pulse. In the current system, a fourth camera generates the SYNC pulse which is then fed, in parallel, to the other three cameras. The above considerations led to the design and construction of custom image capture hardware for the ASIS camera series.

Following the hardware description, Section 6.4 will present a worked example and summary of results obtained from applying the DETECT algorithms to sequences obtained from CMOS cameras and framegrabbers. The full set of results, for the twelve sequences tested, can be found in Appendix A. Appendix A includes graphs of the raw and averaged disparities extracted through time, Kalman estimates of disparity over time, disparity histograms and measures of confidence. Also included in Section 6.4 are some general statistics extracted from the data in Appendix A. The last section of this chapter will discuss these results and provide conclusions.

6.2 Equipment Description and Operation

Figure 6–1 shows a diagrammatic representation of the capture apparatus. CMOS cameras surrounded by test-jig PCBs were attached to aluminium lens mountings. Plastic screws were used to adjust the sensor's focal length and distance from the lens. The cameras were then mounted on a track and adjusted as described in Section 6.2.3. Video signals were synchronised using a fourth camera in SYNC generation mode. Also shown in Figure 6–1 are the power supply, exposure control, video and digitisation connections to the PC frame grabbers. These connections, and the fact that the cameras are powered directly from the PC make the system easier to use and more portable. Exposure can be explicitly defined in the

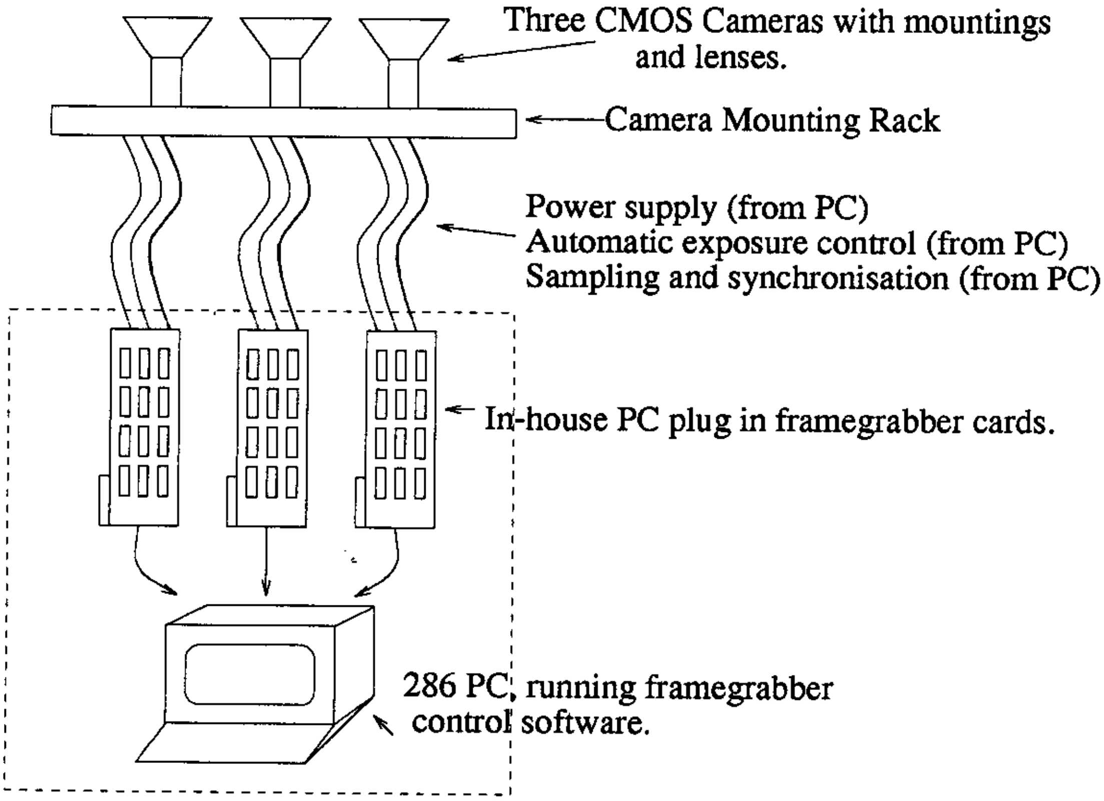

Three CMOS Cameras with mountings and lenses.
Camera Mounting Rack
Power supply (from PC)
Automatic exposure control (from PC)
Sampling and synchronisation (from PC)
In-house PC plug in framegrabber cards.
286 PC, running framegrabber control software.

Figure 6–1: The Image Capture Equipment

framegrabber software. Further, signals such as "pixel valid" and "frame start" allowed a simplification of the frame grabber hardware avoiding noisy synchronisation circuits.

6.2.1 The ASIS Camera

The ASIS cameras were developed after several years research in the Department of Electrical Engineering at Edinburgh University culminating in commercial applications. Over the last three years sensors have been designed for different applications. For example, a finger print recognition system has been implemented on the same chip as the video sensor [1]. For the tests performed in this work the ASIS1010 sensor was used.

ASIS1010 was a prototype 256x256 pixel array and was designed to be the first video camera with sensor and video generation circuitry on the same substrate[17]. The video circuitry also included both automatic and manual exposure control circuitry. The manual control of exposure allowed sensitivity mismatches, between different cameras, to be corrected; a feature which is useful when sensors respond differently to the same scene. Since the initial prototypes several cameras have been produced with increasing performance. Problems such as fixed pattern noise and blooming have been to a large extent resolved and current ASIS versions are now comparable, in performance, to existing CCD sensors [17].

6.2.2 Framegrabber Design

Development of the custom framegrabber started with a fourth year project by Ramsay [71]. The author expanded the design to include a PC interface and also to power the cameras directly from the computer[23]. Additions were also made to the analogue amplifier to improve the signal to noise ratio and extra memory was added, allowing sequences of up to sixteen frames to be captured per board. Signals to control exposure and a flash are also produced by the frame grabber. After developing the above circuit on prototype cards the design was implemented as a plug in PCB for the PC/AT bus. Ten frame grabbers in all were assembled. In

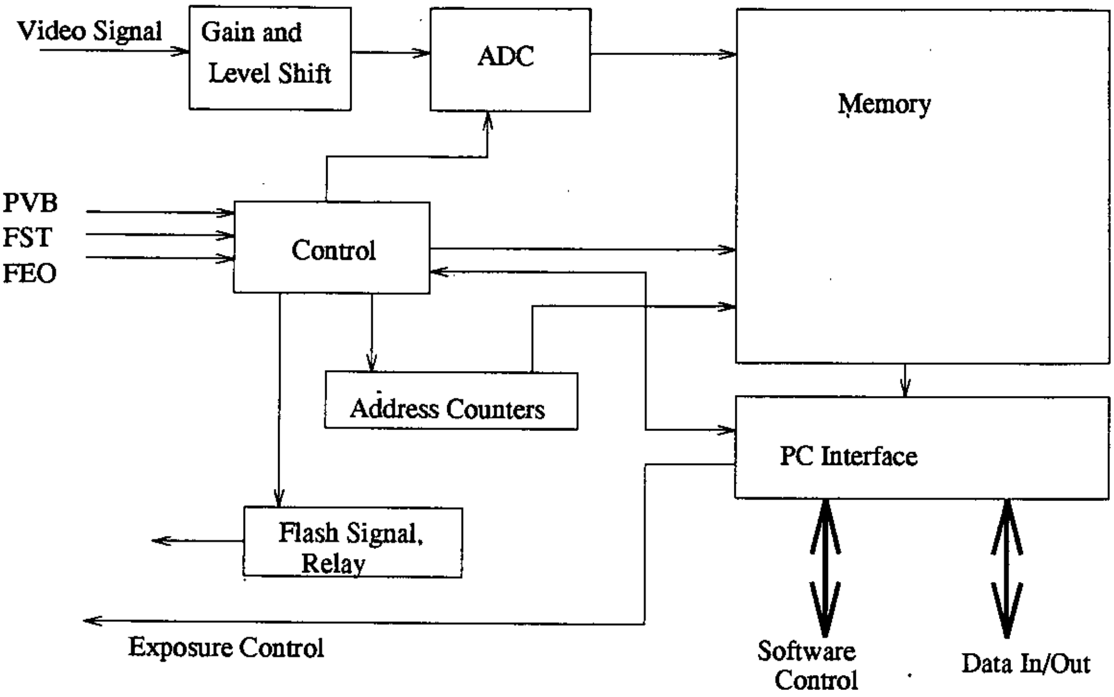


Figure 6–2: Frame Grabber Design

the trial apparatus three cards were used in parallel. These could be synchronised using a single write command from the computer, but read independently. Each board had a common address for writing, but an individual address for reading.

A block diagram of the system is shown in Figure 6–2. The analogue circuitry consisted of amplification and level shift on the incoming video signal before being fed into the video ADC. Potentiometers were positioned at the rear of the board allowing access through the PC's back. These allowed the gain and DC voltage level to be controlled, allowing variation of the contrast and absolute luminosity. As indicated above, the cameras provide several useful signals. The two most important are PVB, indicating when a pixel is valid, and FST, indicating when the frame is about to start. PVB and FST have allowed the elimination of "lock-on" circuitry to estimate when a video signal should be sampled. In particular, a delayed PVB is used, indirectly, as the sampling signal to the ADC. The delay has been made variable to allow sampling on the least noisy part of the incoming video signal.

In parallel with the above hardware development, menu driven PC software was written. The software set up the appropriate board logic to grab images into the frame grabber's memory, save the captured images to disk, display those pictures on the screen and allow calibration of analogue offsets using grey level histograms. Further to this, separate routines were developed to capture, read and align three individual cameras. Alignment is discussed in the next section.

6.2.3 Camera Setup

Once the three frame grabbers were installed in a PC, the stereo cameras could be adjusted to minimise misalignment. This was done using a white cross on a black background and differencing between pairs of cameras. Adjustment was done by repositioning the sensors behind the fixed lens. Once a pair were correctly aligned the difference image would have no horizontal white bands. The rest of the image should be black except for vertical white columns representing the disparity between the cameras. In practice, the cameras could never be perfectly aligned and corrections had to be made using software offsets. Calibration and translational offset techniques are considered in Sections 3.6, 5.6 and 6.4.3. The rig was then mounted on a trolley, together with a PC, and taken to various scenes for sequence capture. Capturing triple sequences of sixteen images presents problems for disk storage with a single sequence require 3 Mbytes.

All sequences were digitised at 5 frames/second. A higher frame rate was not chosen, as a final implementation would be more expensive at 25 frames/second. Further, each frame grabber can only capture 16 frames. The three seconds of capture time, provided at 5 frames/second allow a person to walk a sensible distance between frames. For a person walking at 2m/s this ensures 40cm, frame to frame. It is a good test of the system to measure disparity differences to a resolution of 40cm.

Scene	Length	Width
1	12m	4m
2	17m	1.5m
3	6m	7m

Table 6–1: Scene Dimensions

6.3 System Simulation

The DETECT algorithms, described in Chapters Four and Five, were implemented in software, the details of which are described in Section 4.3 and Appendix B. This section will describe the scenes used to test these algorithms and provide a worked example.

6.3.1 Description of Scenes

Three scenes were chosen for comparison. In all, twelve sequences were captured. The backgrounds are shown in Figures 6–3, 6–4 and 6–5 with their dimensions in Table 6–1.

Figure 6–3 shows an entry hall scene with a bright outdoor background. The outdoor light is dominant compared to the indoor strip lighting. The camera points directly at the light source. Figure 6–4 shows a long 17 metre corridor bathed in artificial light and a generally dark background. This background provides a contrast to scene 1. A problem encountered with this scene was camera blooming caused by strip lighting. To reduce this problem the camera contrast has been considerably reduced. Such a contrast reduction provides a stiffer test for the segmentation algorithms. Figure 6–5 shows the third background of a scene lit with outdoor lighting coming from behind the camera.

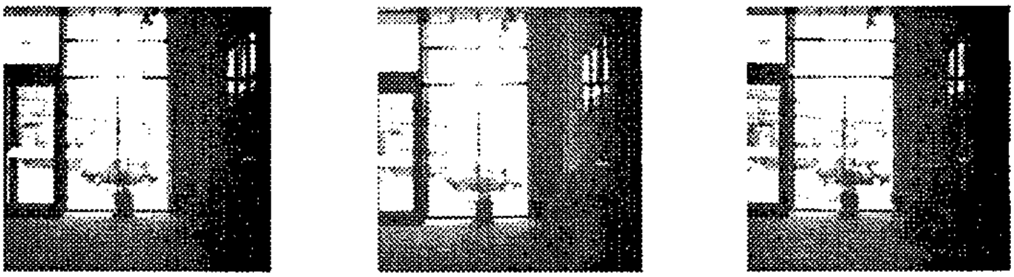


Figure 6–3: Scene 1: Entrance Hall with Bright Outdoor Background.

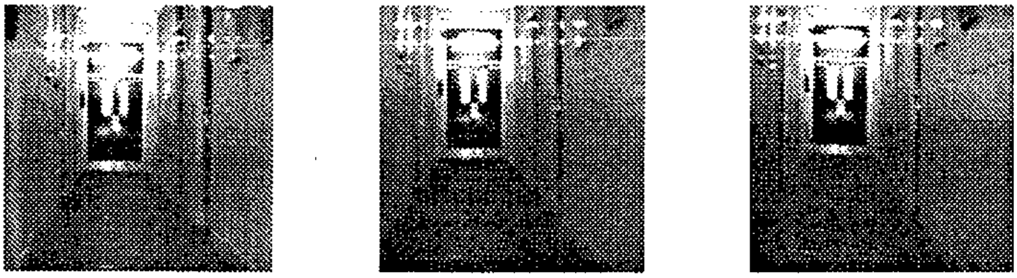


Figure 6–4: Scene 2: Indoor Corridor with Artificial Light

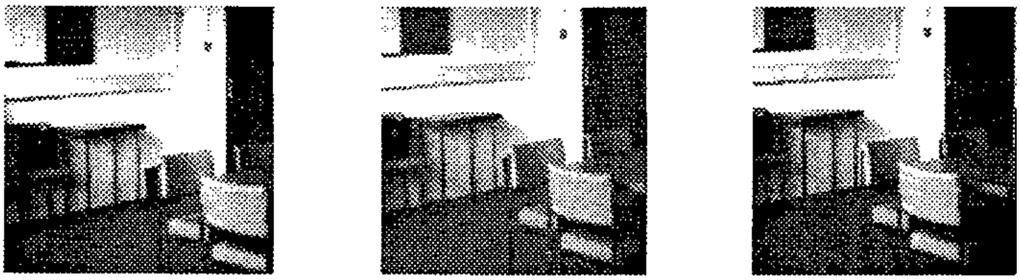


Figure 6–5: Scene 3: Room with Table and Lit with Outdoor Light.

6.3.2 Worked Example of the DETECT System in Operation

In order to provide a fuller explanation of the DETECT system a worked example is now provided. Sequence 7 is typical with the intruder walking away from the camera. Sequence 7 was taken in Scene 2, Figure 6–4 where the problems of blooming and attendant contrast reduction were at their worst.

The first stage of the algorithm is the determination and storage of a background image and establishing a lack of movement in the scene. Change in the number of detected edges from frame to frame can be used to indicate when there is no intruder. The techniques to indicate presence, with example sequences, are described in Section 4.2.5 as they are interrelated with the problem of background update.

Once an initial background is established it can be updated when current object positions are known. In the current version of DETECT, this is done for regions greater than ten pixels in the x-direction from a significant cluster. No updating was done above or below the main cluster as legs and heads are particularly vulnerable to fragmentation and separation. Using the background we can calculate the thresholded difference and combine with the tracked edges to provide an estimate of an object's outline. Overall this provides segmentation results which are not dependent on a single source of data. Thus if the edge detection is poor in a particular part of the image, cluster outlines can be used instead and vice versa.

To illustrate, Figure 6–6 shows the eighth frame of sequence 7. Edges were extracted according to their length, connectivity and whether they exceeded a minimum noise threshold. Thresholding was performed globally on the difference image using an experimentally estimated fraction of the mean. In many systems such values are critical and changes in histogram distribution cause segmentation failures. The advantage, of using different sources of segmentation information, is that individual thresholds are less critical. Figures 6–7 and 6–8 show the resultant edge detected and thresholded images.

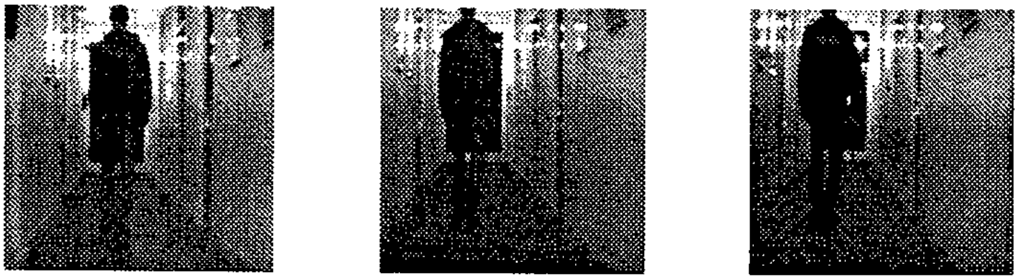


Figure 6–6: Sequence 7: Example Left, Middle and Right Foreground Images

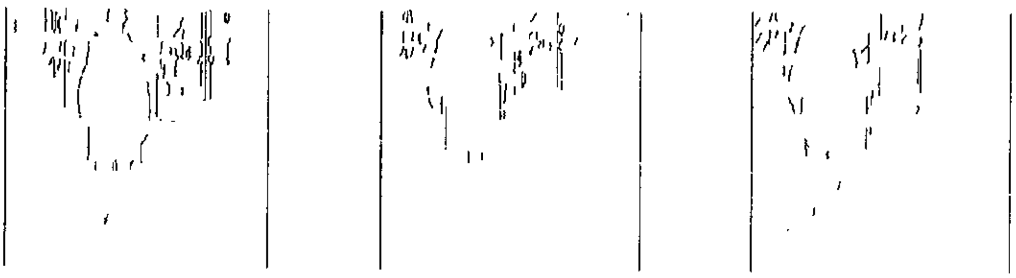


Figure 6–7: Sequence 7: Edge Detected Images Before Background Edge Removal

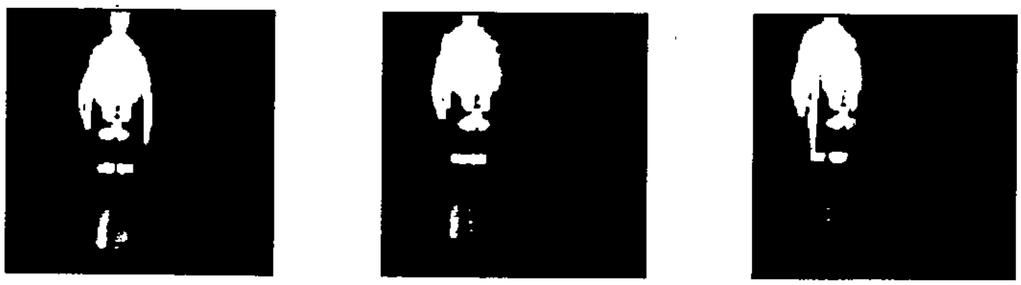


Figure 6–8: Sequence 7: Thresholded Difference Image Showing Significant Clusters

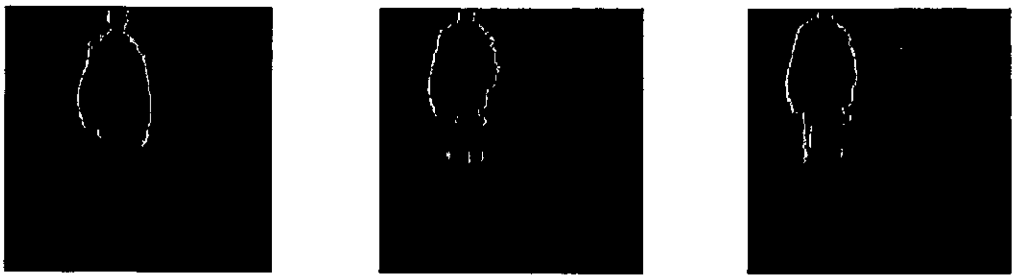


Figure 6–9: Sequence 7: Outline Edges Extracted by Comparison with Previous Edges and Thresholded Clusters

Explicitly extracted edges are compared to cluster outlines and if they correlate are used in the matching process. The resultant edges, before matching is attempted, are shown in Figure 6–9. In this particular frame the outline edges come mostly from the cluster segmentation whereas in other frames and sequences, from the trial, the reverse is true. In general, it appears that cluster outlines are better for matching at longer ranges and directly extracted edges better nearer the camera or when the object is large.

We now discuss DETECT's computationally simple solution to the correspondence problem. There are three possible pairings using three cameras: left and middle (left image), middle and right (middle image) and left and right (right image). After correction with translational offsets the extracted outlines from the two images are overlayed as in Figure 6–10. Possible matches are shown as scan lines between the overlayed left and right camera edges. Continuous bands, of similar width, represent correct matches. At this point the disparity gradient and overlap constraints apply. If the width of a band suddenly jumps, ie. exceeds the disparity gradient limit, it is likely that there is a false match at that point and one of the matches, either side of that point, will be wrong. The application of such a gradient limit, in this way, requires that the entire object be assumed at the same depth. As we are working with wide angle systems, at longer ranges, this assumption is valid. The disparity gradient and the particular segmentation of outline edges eliminate the majority of false matches. An additional concern, in reducing the number of false matches, are the translational offsets used to align

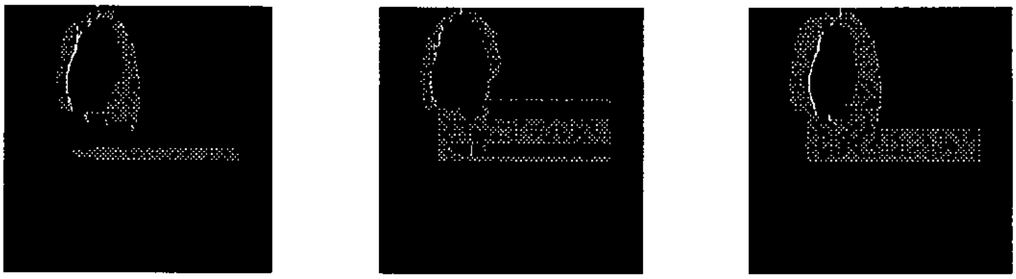


Figure 6–10: Sequence 7: Three Matching Measurements Possible from a Triple Camera Rig

the left and right images. These offsets are significant due to the likely clutter of "in-between" edges. Thus, if the spacing, between the true matches, is too large then more matches will be erroneous. Alternatively, if the edges are too close together then there is no room for any disparity variation. In this situation lens distortion and other types of calibration error will cause some parts of overlayed edges to cross one another and invert. A reduction in the number of correct disparities will again result. In view of such error considerations, translational offsets can be varied to achieve an optimum for a particular camera set up. This could be done automatically using objects moving in a known direction and the measures of disparity confidence described in Section 5.4. Also important in the control of false matches is the maximum allowed disparity between two edges. Long lines across the images in Figure 6–10 show matches eliminated on the basis of an absolute disparity limit.

Having estimated the correspondences for a particular cluster, edges can now be weighted according to "goodness" factors. The overall variation of disparity as the edge is tracked can be used as can the amplitude of the disparity histogram for a particular edge. Using the weighted disparities each match is entered in a cluster histogram. Normalised histograms, for this example frame, are shown in Figures 6–11, 6–12 and 6–13 and for the entire sequence, through time, in Figures 6–14, Figure 6–15 and Figure 6–16.

After some histogram smoothing the main peak is found and the mean around

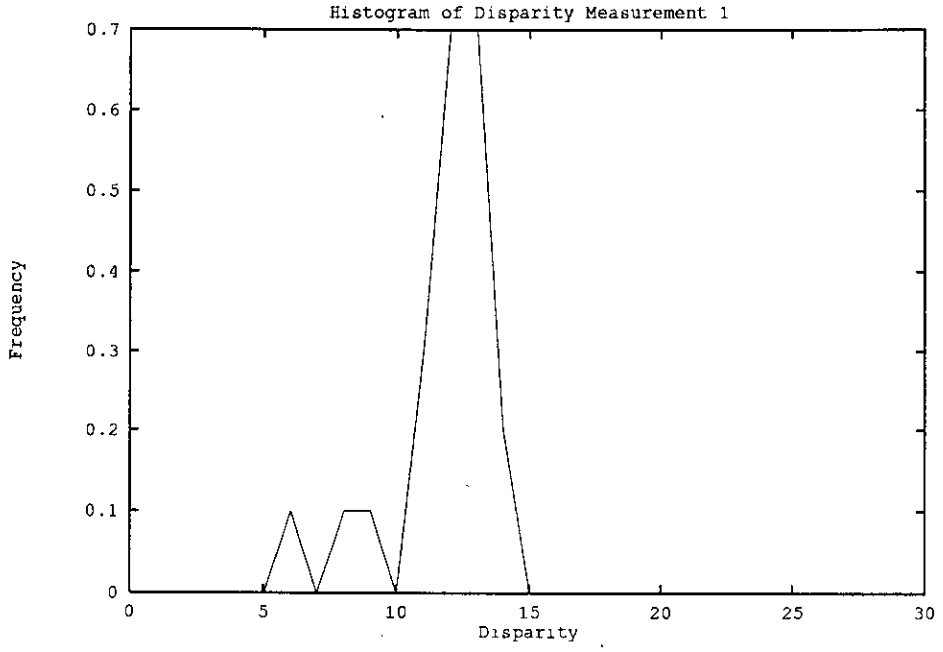


Figure 6–11: Sequence 7: Disparity Histograms Extracted from the Matching Frames Shown in Figure 6–10, Measurement 1

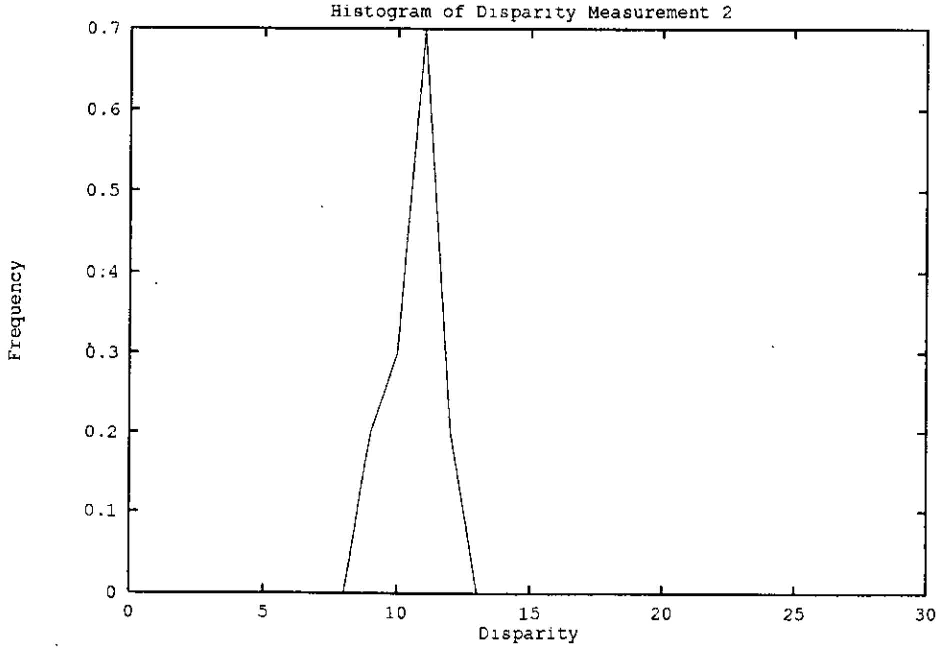


Figure 6–12: Sequence 7: Disparity Histograms Extracted from the Matching Frames Shown in Figure 6–10, Measurement 2

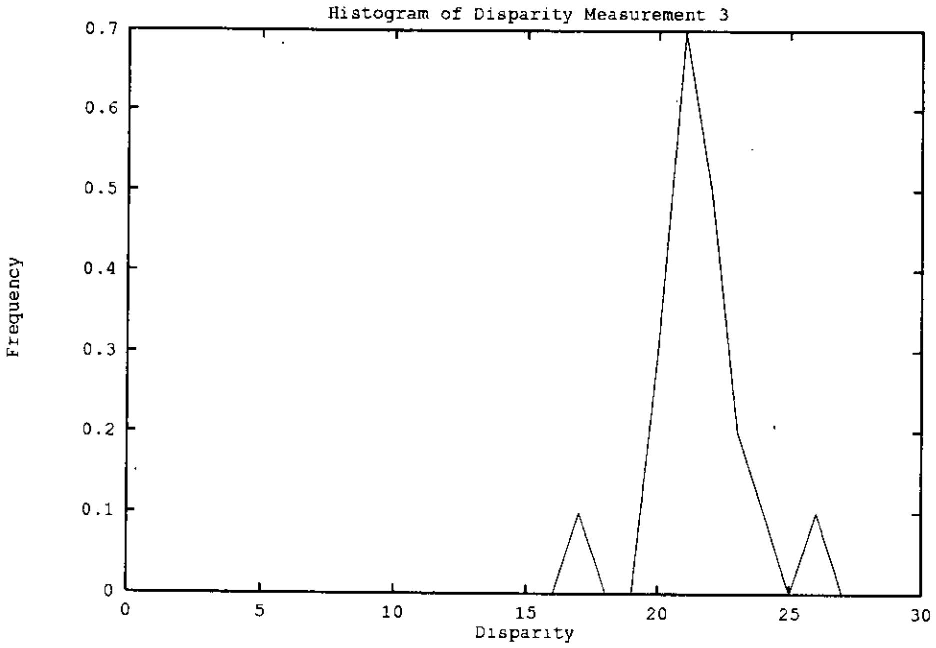


Figure 6–13: Sequence 7: Disparity Histograms Extracted from the Matching Frames Shown in Figure 6–10, Measurement 3

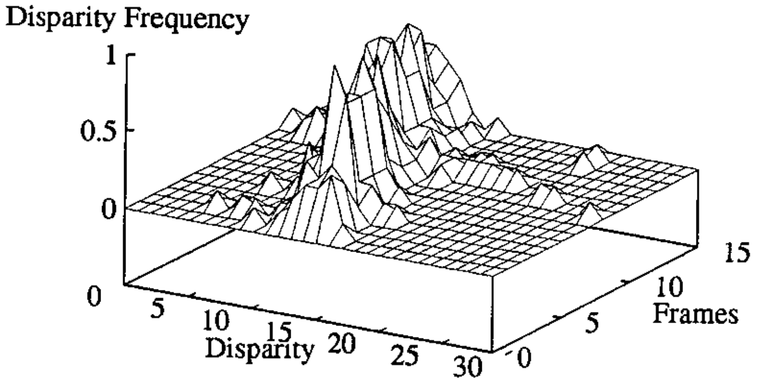


Figure 6–14: Sequence 7: Disparity Histograms Shown Through Time, Measurement 1

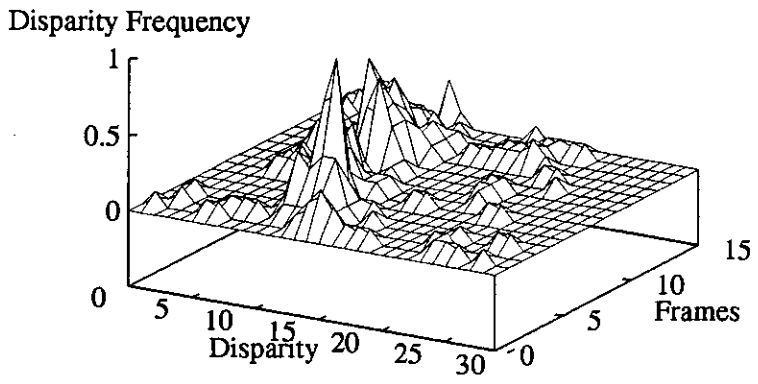


Figure 6–15: Sequence 7: Disparity Histograms Shown Through Time, Measurement 2

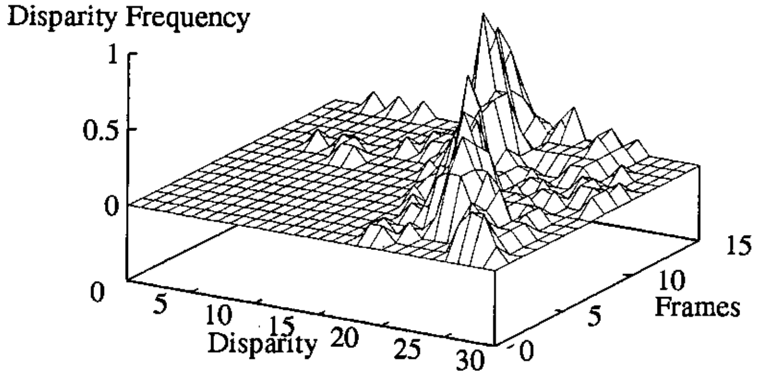


Figure 6–16: Sequence 7: Disparity Histograms Shown Through Time, Measurement 3

that peak calculated. The proportions of disparities, either side of the peak, give an estimate to sub-pixel precision. The calculation of peaks is carried out on the entire 16 frame sequence. Thus, the outlying values, away from the main peaks, are not included in the calculation of disparity. The final disparity estimates are done through time using the Kalman filter, described in Section 5.5, followed by confidence weighted averaging. The calculations are based on the assumption that the disparity distribution for a particular measurement is Gaussian.

6.3.3 Sequence 7: Results

The results shown in this section are purely for Sequence 7, Scene 2. Figure 6–17 shows the disparities calculated as described above, and before any further Kalman calculations have been performed[1]. They show a clear reduction in disparity as the intruder walks away from the camera. In addition the third disparity, from the outside cameras, has roughly twice the gradient of the two inner pairings.. Absolute values of disparity cannot be compared due to translational offsets. However the trend is clearly correct and the gradients would be sufficient to reliably activate an alarm for some disparity threshold.

The raw data was then used as input data to a Kalman filter, where the initial error covariance matrices are calculated from raw error variances for all twelve sequences. The Kalman filter output is shown in Figure 6–18. In this graph the third measurement has been halved before being input to the Kalman filter to allow a proper comparison. The final graph, Figure 6–19, shows the weighted average of the three Kalman estimates, where the weights are calculated using the disparity frequency of the peak.

[1] It should be noted that the disparity scale has been increased to 30 to show the whole graph. This differs from the graphs shown in Appendix A, which have a reduced scale allowing time disparity gradients to be shown more clearly.

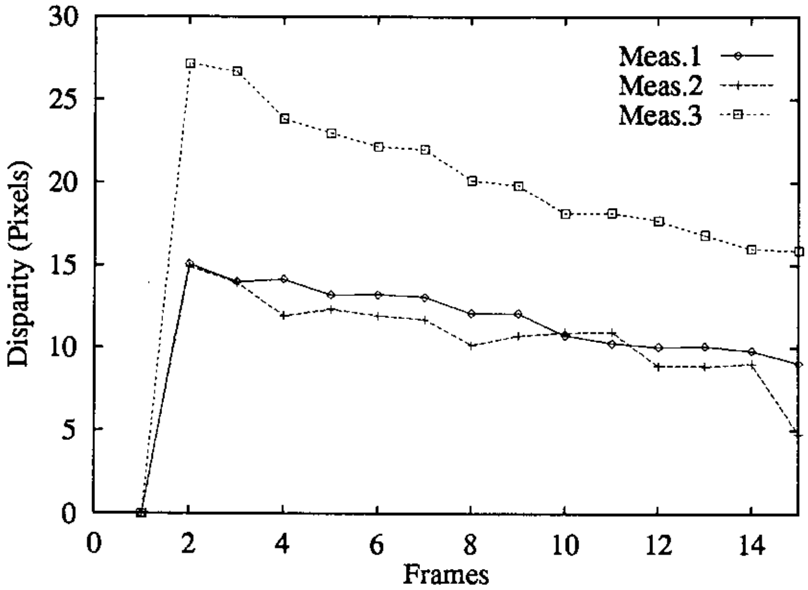


Figure 6–17: Raw Disparities

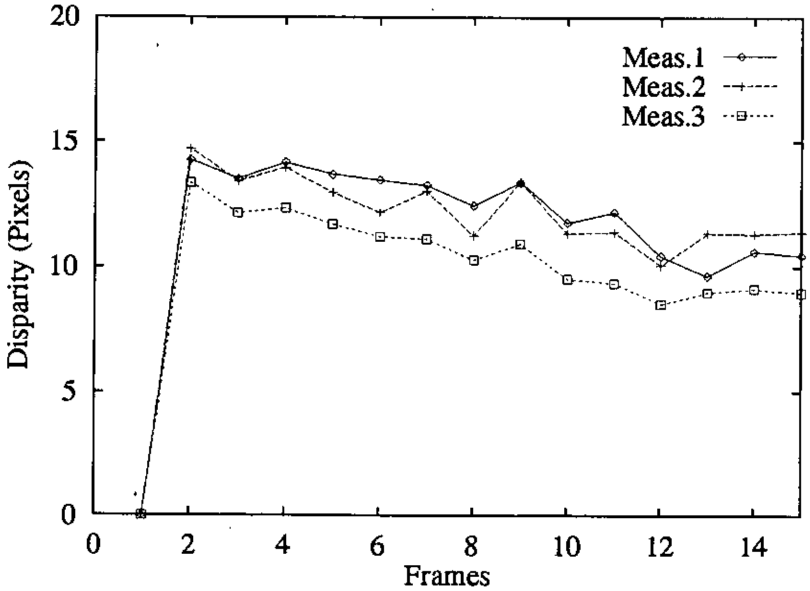


Figure 6–18: Disparities Output from the Kalman Filter

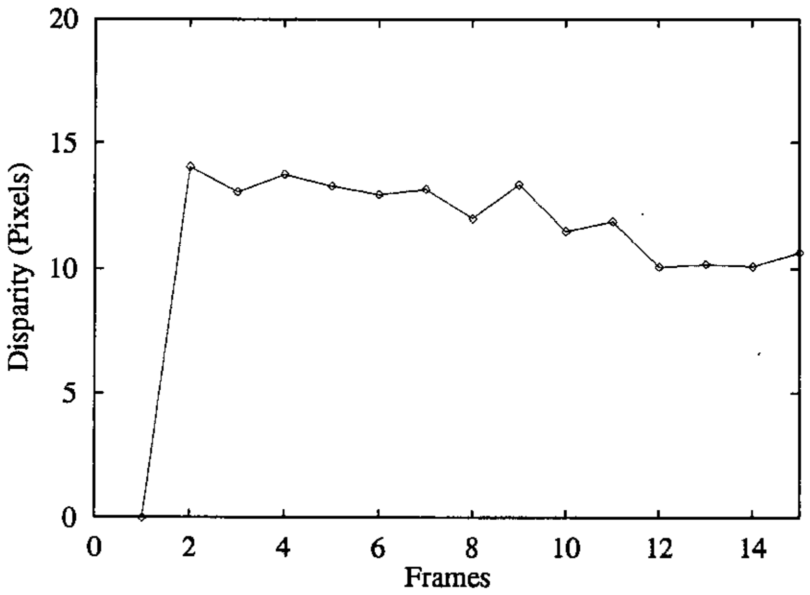


Figure 6–19: Averaged Kalman Disparity

6.4 General Results

Appendix A shows, for each sequence, the raw, Kalman and averaged Kalman disparity traces. Following this are graphs of the measurement strength and variance. For completeness the disparity histograms are also presented.

In all twelve sequences, a human was detected and tracked with disparity trends in the correct direction, ie. inversely to the range. Theoretically, the outside disparity should be twice the two inside values. However this is not always the case due to translational offset variations. The important point to note is that the *time gradient*, not the absolute value, of the disparity through time, for the outside cameras, should be twice that of the internal pairings.

6.4.1 Time Gradient Comparison

Table 6–2 shows the gradients of linear regression best fit lines for each of the twelve raw disparity traces. Also shown are the intruder starting ranges from the camera and the direction in which he is moving. Calculation of gradients, for these traces, presents problems, in that an intruder will not always be walking at the same speed. Often the person accelerated as the sequence progressed. Therefore a straight best fit line will not follow the data exactly and results for a particular sequence should not be taken in isolation. However, the best fit line does provide overall measures of movement through the scene without being solely dependent on the first and last disparities.

As expected, there are clear differences in best fit gradients between those sequences where the "walker" starts near the camera, at 4m, and those where the "walker" starts at 17m. Time disparity gradients are less for the sequences where the person is further away. If the number of frames, while a particular disparity threshold is breached is used as an alarm threshold, it might be sensible to increase this number of frames in line with the boundary depth. The only exception to the gradient variations described above is sequence three, where a person walks, parallel to the camera, but outside the window of scene 1. In this case the disparity gradient should be small anyway.

In the case of sequences 2,5,7,9 and 11 the third measurement gradient is roughly twice that of the inner pairing. For the other sequences there is less of a difference. There are three reasons for this discrepancy. Firstly the errors associated with the longer range sequences are bound to be larger. Secondly, in sequences 3 and 4, the individual is not moving, in depth, very much at all. Thirdly, the measurements extracted from the best fit analysis, in Table 6–2, only provide a rough guide to the general direction of motion. For individual traces, a more accurate picture can be found in Appendix A.

Sequence	Time Disparity Gradient (Pixels)		
	1	2	3
1 (11m - T)	0.26	0.10	0.29
2 (11m - T)	0.14	0.11	0.21
3 (14m - 14m)	0.00	0.03	0.00
4 (17m - T)	0.03	0.06	0.08
5 (4m - A)	-0.25	-0.23	-0.47
6 (17m - T)	0.06	-0.11	0.14
7 (4m - A)	-0.20	-0.13	-0.31
8 (17m - T)	0.03	0.04	0.06
9 (4m - A)	-0.32	-0.20	-0.51
10 (8m - T)	0.05	0.09	0.10
11 (8m - T)	0.16	0.13	0.25
12 (8m - T)	-0.00	-0.00	0.04
Ave. (Mod)	0.12	0.10	0.20

Table 6–2: Best Fit Disparity Gradients, (Through Time), of Image Sequences, T = towards the camera, A = away from the camera

Sequence	Meas. 1	Meas. 2	Meas. 3
1	0.46	0.45	0.33
2	0.51	0.36	0.37
.3	0.67	0.39	0.38
4	0.43	0.52	0.40
5	0.31	0.40	0.24
6	0.45	0.68	0.53
7	0.33	0.45	0.27
8	0.40	0.49	0.41
9	0.33	0.39	0.28
10	0.45	0.42	0.48
11	0.41	0.65	0.57
12	0.39	0.38	0.32
Ave.	0.43	0.46	0.38

Table 6–3: Averaged Disparity Measurement Variances for the Twelve Sequences

6.4.2 Amplitude and Variance Comparison

This section discusses some of the noise measurements obtained from the 12 sequences used in the trial data. Section 5.5 described how a Kalman filter could be applied to make estimates of disparity. Also, Section 5.4 described the calculation of an initial covariance matrix based on variances from the disparity histogram. In effect, only the central diagonal values of the 3x3 matrix need be calculated experimentally. Others elements can be derived from these three initial values. As was stated in Section 5.4, such variances combine both the matching errors and locational errors for the entire system and can be used in the initialisation of the Kalman measurement error matrix. Table 6–3 shows the averaged variances for each of the twelve sequences and their three measurements. Measurement 1 is the variance from the left and middle cameras, Measurement 2 is from the middle and right cameras and Measurement 3 is from the left and right camera.

Also mentioned in Section 5.4 was the independence of the errors, or confidence measures, between the three disparity measurements. In a final application of the DETECT algorithms, for an alarm application, such error measures, and their relative independence, would be important in a decision to alert the attention of a controller. The two most obvious measures of confidence are amplitude and variance. Amplitude suffers from the problem that its value will vary with distance. Clearly, objects nearer the camera are more likely to produce larger matched edges and a higher amplitude in the disparity histogram. However, a larger amplitude does mean that there are more matches and better sub-pixel accuracy. It is worthwhile comparing the independence of the disparity histogram amplitudes with the independence of the disparity histogram variances. This can be done using the correlation coefficient between two data series $\mathbf{x}$ and $\mathbf{y}$. This is defined in Equation 6.1.

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{6.1}$$

The independence of the errors between measurements 1 and 2, measurements 2 and 3 and between 1 and 3 are shown in Tables 6–4 and 6–5. Using the definition of the correlation coefficient, described above, values of r can vary between -1 and 1. 0 represents no correlation, neither positive or negative. The averaged values for r for both amplitude and variance are all positive indicating the level of dependence between the measurements. However they are still low enough to allow large accuracy improvements to be made by combining the results from the three measurements.

6.4.3 Calibration Data

Calibration was discussed in Sections 3.6 and 5.6. For the applications being considered in this thesis, ie., intruder detection, accurate calibration is not required. Indeed for computational reasons it is undesirable. The presence or absence of a person within a set boundary only requires a disparity threshold to be defined which, if crossed, will activate the alarm. Thus the calibration of an alarm system could be reduced even further, by walking around at the required boundary and extracting disparities only at that distance.

Sequence	Correlation Coefficient		
	Meas. 1	Meas. 2	Meas. 3
1	-0.10	0.27	-0.09
2	0.47	-0.08	0.18
3	-0.00	0.41	0.13
4	0.26	0.29	0.38
5	-0.00	0.34	0.20
6	-0.43	0.89	-0.25
7	0.11	0.43	0.45
8	0.00	0.62	0.00
9	0.22	0.36	0.77
10	0.82	0.68	0.79
11	0.37	0.03	-0.10
12	0.31	0.37	0.96
Ave.	0.24	0.38	0.28

Table 6–4: Correlation Between Measurement Variances

Sequence	Correlation Coefficient		
	Meas. 1	Meas. 2	Meas. 3
1	0.07	0.54	0.26
2	0.89	0.82	0.91
3	0.35	0.61	0.22
4	0.84	0.80	0.83
5	0.06	0.52	0.40
6	0.47	0.63	0.55
7	0.30	0.37	0.32
8	-0.04	0.83	-0.12
9	0.03	0.02	0.43
10	0.77	0.69	0.50
11	0.92	0.74	0.79
12	0.77	0.69	0.80
Ave.	0.45	0.60	0.49

Table 6–5: Correlation Between Measurement Amplitudes

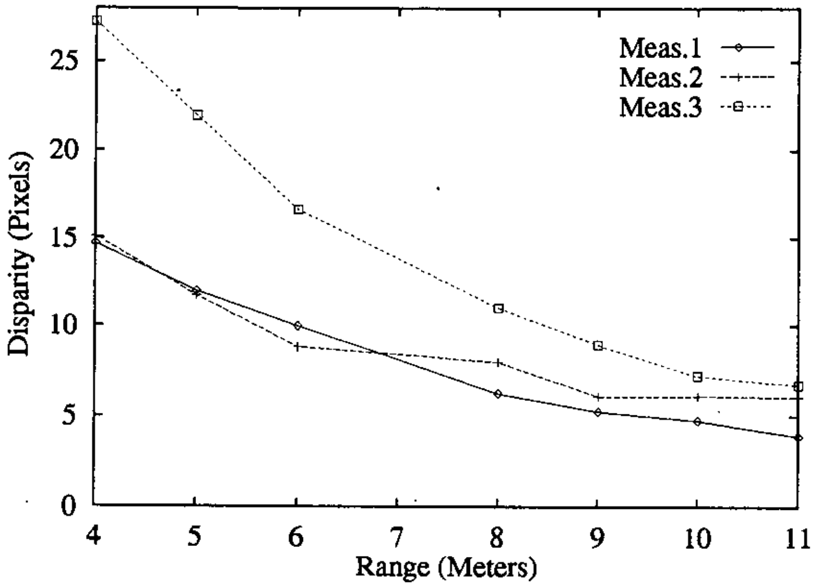


Figure 6–20: Disparity (Pixels) versus Measured Distance (Meters)

This section presents a simple technique which avoids the problems of numerical analysis and its associated computation. Disparities were extracted for known ranges marked in the scene. If the camera rig is stable these can be used in a look-up-table to estimate absolute distances for each disparity. Figures 6–20 and 6–21 show plots of estimated disparity against measured distance. Figure 6–20 is taken from known distances in Scene 1 whereas Figure 6–21 is taken from known distances in Scene 2. They show a clear inverse relationship between distance and disparity and the expected difference in gradient between measurements 1/2 and 3. The same measurements were not taken in scene 3 due to furniture.

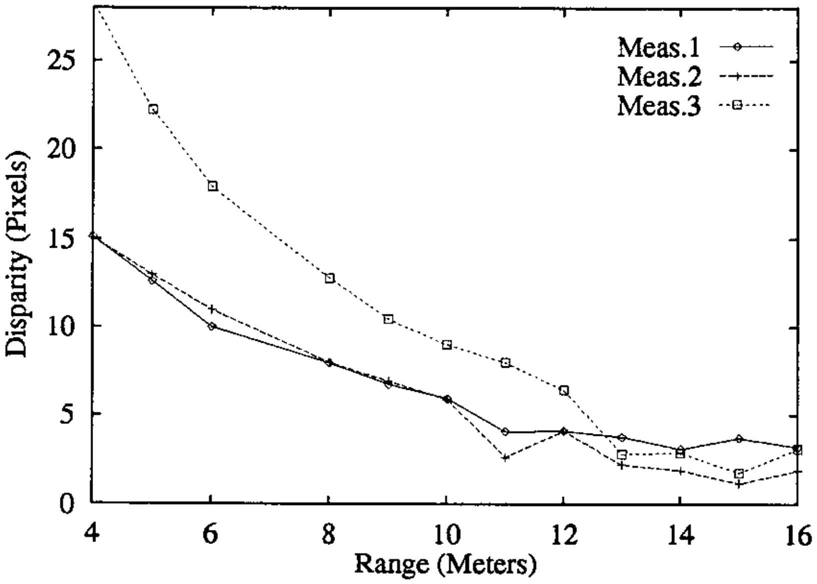


Figure 6–21: Disparity (Pixels) versus Measured Distance (Meters)

6.5 Discussions and Conclusions

This chapter started with a review of the equipment used in this research. A frame grabber was designed to allow the simultaneous capture of video pictures from the ASIS cameras into a PC. The techniques used to set up the cameras were also described. For these experiments a simple black and white cross was used and the left and right cameras were adjusted accordingly. Software was written to perform this and other camera calibration functions. Twelve sequences were captured from three scenes and used to simulate the system as described in Section 6.3. For explanation of the algorithms described in Chapters Four and Five a worked example is provided in Section 6.3.2. This goes through the separate stages of the DETECT system with intermediate images.

The final section of this chapter summarises the results data shown in Appendix A. Tables were extracted for best fit time disparity gradients, variances and amplitudes, and the correlations between the three possible measurements from three cameras. The variances were extracted and averaged for each sequence's disparity measurement. They were used as the starting measurement error covariance matrix in the Kalman formulation described in the last chapter. Variances, together with amplitude, can be used as measures of confidence in a final alarm system. As such it is important to know the correlation between the variances of the three measurements. These can be found in Tables 6–4 and 6–5. As expected, and stated in Section 5.4, amplitude and variance are not independent. Clearly, lighting conditions are likely to cause similar matching problems in each of the three disparity measures. However the average correlation for both amplitude and variance is still low enough to allow the combination of the information from the three sources using a weighted average. Again, such information would be of use in a final installation.

A few final comments are worthwhile on the subject of computation. As said in Section 4.3, the algorithms were written on Sparc 1 work stations. The current software takes about 5 minutes to process a 16 image trinocular sequence. This time includes all the processing from loading images, through early segmentation and stereo matching, to the final stages of higher level processing. As the software was developed over a period of time, considerable improvements could be expected after a rewrite. Due to the restrictions placed on the allowed arithmetic, there is no reason why low cost commercially viable hardware could not be developed to operate at standard frame rates.

Chapter 7

Conclusions

7.1 Introduction

Stereo and machine vision problems have been studied extensively over the last twenty years, as attempts have been made to imitate the human system. Despite this, relatively few commercial vision applications have been developed. There are several reasons for this, the most obvious being that general machine vision problems are difficult. As a result, researchers have divided the problems into separate vision functions and attacked each independently. This approach has had some success. For example, the tradeoffs involved in edge detection and multi-scale feature extraction are now generally agreed. However the interactions between these modules in larger systems, are not well understood. There are few rules and most artificial intelligence systems use functions such as edge detection and stereo matching algorithms in a "black box" manner. It seems likely that efficiency savings can be made if different vision modules can be considered together.

The work described in this thesis provides an example of how stereo correspondence has been simplified by segmenting only certain types of edges. We are also aiming at a specific type of implementation using the recently developed low cost ASIS sensor/processors, described in the introduction. This differs from the approach taken by Hakkarainen [28] who describes part of a general stereo vision algorithm developed in more expensive CCD technology. The main problem here is the integration of other image processing functions onto a single substrate. They would all have to be developed in analogue CCD. CMOS sensors with appropriately adapted algorithms, provide an alternative implementation architecture.

In view of the above, we have considered alarm systems as a possible application of vision algorithms, together with CMOS technology. The question asked here, and answered, for this application, is: Can machine vision algorithms be developed, for low cost implementations, without a problematic loss of function? Chapters Four and Five describe such a system which employs no floating point calculations, multiplications or divisions at the lower levels of processing. Due to the data rates required by pixel based operations, a microprocessor implementation is impossible and parallel floating point processor arrays are prohibitively expensive.

Having defined the initial aim, a review of current image processing and machine vision techniques was undertaken. Chapter Two surveyed low level techniques. Edge detection, thresholding and segmentation were described in the context of recognition techniques such as graph matching and the generalised Hough transform. Chapter Three considered the current theories of stereo vision and what constraints are normally used to solve the correspondence problem, in the light of the likely errors inherent in the geometry of the system. The important problem of calibration was discussed with a description of current solutions. The calibration techniques, surveyed, were considered to be too computationally complex for an alarm installation where accurate metrics might not be necessary. Here, a disparity threshold could be used as an invisible boundary and the system offsets and thresholds arranged to activate the alarm when an intruder crosses the boundary for some number of frames.

The second half of the thesis described the DETECT algorithms and their possible hardware implementations in detail. Chapter Four described the segmentation stages used to extract only outline edges for stereo matching. Other savings have been made by acknowledging that stereo matching does not work well with horizontal edges, and that a moving human intruder is likely to consist of, mainly, vertical edges. Edge detection can then be reduced to a lateral differentiation followed by a vertical track along peaks. Apart from the relationship between extracted edges and the stereo matching algorithms, emphasis was placed on the interdependencies between the different processing modules. For example, outline edges are found by combining the cluster extraction with vertical edge detection. Finally, for the segmentation stages, it is also important to note that effort was directed at extracting the relevant image information from the raw images as quickly as possible. Apart from storage, there are considerable computational advantages to be gained by acting, only, on relevant parts of binarised and segmented images.

Following on from Chapter Four, Chapter Five described the stereo matching algorithms. In particular, an original low cost correspondence algorithm was developed. Notice was taken of the fact that, for laterally symmetric stereo camera geometries, with imaging sensors on the same plane, objects will always overlap when their images are overlayed. Overlap will occur no matter where the object is positioned in the scene. Scan matching was followed by the creation of disparity histograms for each significant edge and cluster. In an ASIS implementation it is likely that disparity histograms would be the final output of the chip. Due to it's compact nature further statistical processing could be developed using a microprocessor to perform floating point calculations to provide the required accuracy in disparity measurements. The final part of the DETECT system calculates the disparity for the three possible measurements available from a three camera rig. Confidences can be extracted from the disparity histograms and used in a Kalman formulation to correct for obvious errors and integrate the three measurements in time and with each other. In this application Kalman measurement and system error covariance matrices are kept constant to prevent the gain becoming unstable. However, as can be seen in Appendix A, the reality is different and variances change in time and between different scenes. As mentioned in the next section, non-stationary filtering is a possible area for future research in the stereo field.

7.2 Future Research

The work described in this thesis has been aimed at evaluating possibilities of low cost stereo systems with wider than normal angles of vision. Having developed a system capable of detecting and tracking a moving intruder, three specific areas of future research are now open.

Firstly, a trial could be conducted on a larger number of sequences from a variety of different scenes. Also, in order to test alarm disparity thresholds and Kalman filter convergence it would be desirable to analyse longer sequence lengths. The main problem with a larger trial would be storage and capture of the required number of longer image sequences. This is especially so with stereo vision, where multiple sequences of matched images require to be processed. It is clear from the literature that the capture and storage of such large amounts of data is a major restriction on current image processing work. Most of the published work appears to describe results from only one or two sequences of trial data. As was shown by the finger print work described by Anderson et. al. [1] such large trials are necessary before a final implementation is qualified for acceptance.

The second direction for future research would be a hardware implementation of the algorithms using the CMOS sensors. As discussed in Section 5.7 consideration would have to be given to what functions were most suitable for integration on the sensor substrate. The DETECT algorithms have been developed to make this task easier. For example, as edge detection is basically a lateral differentiation, it could be performed on raster scans without any local storage. Also, investigations could be conducted into combining thresholding and ADC conversion and more consideration would have to be given as to the best techniques for storing and processing edges.

A third problem area is calibration. In this work the issue has been avoided due to the application under consideration. Calibration can be simplified to a person moving around at the threshold depth at the time of camera installation. If a more complicated calibration were required it might be possible to perform the necessary calculations on a portable computer attached to the installation. In this case the standard numerical techniques could be applied. However, there is considerable doubt about the accuracy of these techniques, over individual frames. Thus, several researchers have developed algorithms which integrate calibration parameters over a sequence of images. There is scope for improvement here, including using the three measurements possible from a triple rig to further constrain calibration. Indeed, it should be possible to vary rotation and translation parameters, in a sequence, until a consistent series of disparities was achieved from the three measurements.

7.3 Concluding Remarks

A stereo vision image processing system has been developed which can allow the detection and distance tracking of large vertical moving intruders in a scene. An original stereo matching algorithm has been developed which maps into low cost integrated sensing and processing hardware, linked to a microprocessor. Results have been presented showing the system working and confidence statistics extracted for use in a Kalman tracking filter. It is the author's view that the work presented in this thesis would provide a sound basis for further development of hardware efficient vision systems.

References

[1] S Anderson, W H Bruce, P B Denyer, et al. A single chip sensor and image processor for fingerprint verification. In *IEEE Custom Integrated Circuits Conference*, 1991.

[2] M. Annaratone, E. Arnould, T. Gross, et al. Warp architecture and implementation. In *Proc. 13th IEEE/ACM Annual Int. Symposium on Computer Architectures and Implementation*, pages 346–356, June 1986.

[3] M. Annaratone, E. Arnould, T. Gross, et al. The Warp computer architecture, implementation and performance. *IEEE Transactions on Computers C-36(12)*, pages 1523–38, 1987.

[4] D. H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice-Hall, 1982.

[5] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1966.

[6] S. T. Barnard and M. A. Fischler. Computational stereo. *Computing Surveys*, 14(4), 1982.

[7] R. J. Beattie. *Edge Detection for Semantically Based Early Visual Processing.* PhD thesis, University of Edinburgh, 1984.

[8] S. D. Blostein and T.S. Huang. Error analysis in stereo determination of 3D point positions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:752–765, 1987.

[9] G. Borgefors. Distance transforms in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[10] N.A. Borghese and G. Ferrigno. An algorithm for 3D automatic movement detection by means of standard TV cameras. *IEEE Trans. on Biomedical Engineering*, 37:1221–1225, 1990.

[11] S.M. Bozic. *Digital and Kalman Filtering.* Arnold, 1979.

[12] R. G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons., 1983.

[13] P.J. Burt. Smart sensing within a pyramid vision system. *Proceedings of the IEEE*, 76:1006–1015, 1988.

[14] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, COM-31:532–540, 1983.

[15] F. W. C. Cambell and J. Robson. Application of Fourier analysis to the visibility of gratings. *J. Phsysiology*, 197:551–566, 1968.

[16] J Canny. Finding edges and lines in images. Technical Report MA. Rep. AI-TR-720, M.I.T., 1983.

[17] P. B. Denyer, W. Gouyu, L. M. Ying, and S. Anderson. On-chip cmos sensors for VLSI imaging systems. In *VLSI91*, 1991.

[18] P.M. Dew, R.A. Earnshaw, and T.R. Heywood, editors. *An Approach to Automatic Generation of Linear Systolic Array Programs*, pages 3–16. Addison Wesley Publishing Company, 1989.

[19] S. P. DeWeerth and C. A. Mead. A two dimensional visual tracking array. In *Proceedings of the 1988 MIT Conference on Very Large Scale Integration*, pages 259–275, 1988.

[20] U. R. Dhond and J. K. Aggarwal. Structure from stereo - a review. *IEEE Trans. Systems, Man and Cybernetics*, 19:1489–1510, 1989.

[21] J. A. Elliot, J.M. Beaumont, and P. M. Grant. Techniques for motion video processing on transputer based reconfigurable multicomputers. In *IEE Colloquium on Parallel Architectures for Image Processing Applications*, April 1991.

[22] B. Everitt. *Cluster Analysis.* Heinemann Educational Books, London, 1980.

[23] K.W.J. Findlay. A PC framegrabber for the ASIS range of cameras. Internal report, Edinburgh University, Dept. of Elect., 1991.

[24] J. R. Fram and E. S. Deutch. On the quantative evaluation of edge detection schemes and their comparison with human performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, C-24, no. 6:616–628, 1975.

[25] A. Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1985.

[26] R. M. Grey. Vector quantisation. *IEEE ASSP Mag.*, pages 4–20, April 1984.

[27] W.E.L. Grimson. A computer implementation of a theory of human stereo vision. Technical Report AI Memo 565, MIT AI Laboratory, 1980.

[28] J. M. Hakkarainen. *A Real-Time Stereo Vision System in CCD/CMOS Technology.* PhD thesis, Massachusetts Institute of Technology, May 1992.

[29] L. G. C. Hamey, J.A. Webb, and I.C. Wu. An architecture independent programming language for low level vision. *Computer Vision, Graphics and Image Processing*, 48:246–264, 1989.

[30] H. Harasaki, M. Yano, and T. Nishitani. Background separation/filtering for videophone applications. In *IEEE Proceedings of ICASSP'89*, pages 1981–1984, 1990.

[31] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 1991.

[32] W. Hoff and N. Ahu ja. Surfaces from stereo: Integrating feature matching disparity estimation, and contour detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(2), February 1989.

[33] David. C. Hogg. *Interpreting Objects of a Known Moving Object.* PhD thesis, University of Sussex, January 1984.

[34] B. K. P. Horn. The Binford-Horn line finder. Technical Report AI Memo 285, M.I.T. Artificial Intelligence Lab, 1971.

[35] B. K. P. Horn. Parallel networks for machine vision. Technical Report AI Memo 1071, MIT AI Laboratory, 1988.

[36] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *Int. Journal of Computer Vision*, 2:51–76, 1988.

[37] P.V.C. Hough. Method and means for recognising complex patterns. Technical Report US Patent 3069654, US Patent Office, 1962

[38] R. J. Howlett. Multiprocessor techniques for use in low-cost vision systems. In *IEE Colloquium on Parallel Architectures for Image Processing Applications*, April 1991.

[39] D. P. Huttenlocher. *Three Dimensional Recognition of Solid Objects from a Two Dimensional Image.* PhD thesis, MIT Artificial Intelligence Laboratory, 1988.

[40] P.. K. Sahoo J. N. Kapur and A. K. C. Wong. A new method for grey-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics and Image Processing*, 29:273–285, 1985.

[41] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1983:122–139, 1983.

[42] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2), March 1983.

[43] M. R. M. Jenkin, A.D. Jepson, and J. Tsotsos. Techniques for disparity measurements. *Computer Vision, Graphics and Image Processing*, 53:14–30, 1991.

[44] A. D. Jepson and D. J. Fleet. Fast computation of disparity from phase differences. In *Proceedings IEEE Conf. Computer Vision and Pattern Recognition*, pages 398–403, 1989.

[45] G. Johannsen and J. Bille. A threshold selection method using information measures. In *Proceedings 6th International Conference on Pattern Recognition, Munich, Germany.*, pages 140–143, 1982.

[46] C. L. Keast. *An Integrated Image Acquisition, Smoothing and segmentation Focal Plane Processor.* PhD thesis, Massachusetts Institute of Technology, 1992.

[47] H. T. Kung. Why systolic architectures. *IEEE Computer*, pages 37–45, 1982.

[48] H C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, September 1981.

[49] A. Lucas, J. Vliet, and I. T. Young. A nonlinear Laplace operator as edge detector in noisy images. *Computer Graphics and Image Processing.*, pages 167–192, 1989.

[50] M. A. Mahowald and T. Delbruck. An analog VLSI implementation of the marr-poggio stereo correspondence algorithm. In *Abstracts of the First Annual INNS Meeting*, volume 1, page 392, 1988.

[51] D Marr. *Vision.* Freeman, 1982.

[52] D. Marr and E. Hildreth. Theory of edge detection. *Proc. of the Royal Society of London.*, pages 187–217, 1980.

[53] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[54] D. Marr and T. Poggio. On parallel stereo. In *IEEE Proc. of the Image Understanding Workshop*, 1986.

[55] L. Matthies and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[56] R. Mohan, G. Medioni, and R. Nevatia. Stereo error detection, correction, and evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:113–120, 1989.

[57] R. Mohr and E. Arbogast. It can be done without camera calibration. Technical report, LIFIA- IMAG, Grenoble, France, 1990.

[58] H. Moravec. *Robot Rover Visual Navigation.* MI: UMI Research Press, 1981.

[59] P.J. Morrow and R.H. Perrot. *The Design and Implementation of Low Level Image Processing Algorithms on a Transputer Network, Ed. Ian Page*, pages 243–261. Oxford Science Publications, March 1987.

[60] N. M. Nasrabadi. Hopfield network for stereo vision correspondence. *IEEE Trans. on Neural Networks*, 3:5–12, 1992.

[61] N.M. Nasrabadi and Y. Liu. Stereo vision correspondence using a multichannel graph matching technique. *Image and Vision Computing*, 7:237–245, november 1989.

[62] H. K. Nishihara. Prism: A practical real-time imaging stereo matcher. No. 780, MIT AI Memo, 1984.

[63] N. Ostu. A threshold selection method from grey-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, SMC-8:62–66, 1978.

[64] A. J. Parker and J. M. Harris. Human stereoscopic performance, April 1991. Technical Meeting of the Institute of Physics on Computer Vision. No published proceedings. For further details contact Dr. A.J. Parker, Dept. of Physiology, Oxford University, U.K.

[65] P.Burt and B. Julesz. Modifications of the classical notion of panum's fusional area. *Perception*, 9:671–682, 1980.

[66] S. B. Pollard, J. Porrill, J. E. W. Mayhew, and J. P. Frisby. Disparity gradient, Lipschitz continuity, and computing binocular correspondences. In *Robotics Research: The Third International Symposium*, pages 19–26, 1986.

[67] J. Porrill, S. Pollard, T. P. Pridmore, et al. Tina: A 3D vision system for pick and place. *Image and Vision Computing*, 6:91–99, 1988.

[68] W. H. Press, B. P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipies in C: The Art of Scientific Computing*, pages 305–309. Cambridge University Press, 1989.

[69] T. Pun. A new method for grey level picture thresholding using the entropy of the histogram. *Signal Processing*, 2:223–237, 1980.

[70] L. H. Quam. Hierarchical warp stereo. Technical note no. 402, SRI International, 1986.

[71] C. Ramsay. A framegrabber design for the ASIS range of cameras. Internal report, Edinburgh University, Dept. of Elect., 1989.

[72] D. Renshaw, P. B. Denyer, G. Wang, and M. Lu. Asic vision. In *IEEE Custom Integrated Circuits Conference*, 1990. 7.3, Massachusetts.

[73] C. Weems, A. Hanson, E. Riseman and A. Rosenfeld , An integrated image understanding benchmark: recognition of a 2 $\frac{1}{2}$ D mobile, in *Int. Conf. on Computer Vision and Pattern Recognition*, Ann Arbor, MI June 1988.

[74] B. Rudberg, M.N. Chong, S. Marshall, and J.J. Soraghan. Transputer topologies for image sequence transmission. In *IEE Colloquium on Parallel Architectures for Image Processing Applications*, April 1991.

[75] B. Ruff. *A Pipelined Architecture for a Video Rate Canny Operator Used as the Initial Stage of a Stereo Image Analysis System*, pages 171–187. Oxford Science Publications, March 1987.

[76] J E W Mayhew S B Pollard and J P Frisby. A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[77] P K Sahoo, S Soltani, A K C Wong, and Y C Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[78] K Skifstad and R Jain. Range estimation from intensity gradient analysis. Technical Report CSE-TR-02-88, University of Michigan, 1988.

[79] N. A. Thacker and P. Courtney. Statistical analysis of a stereo matching algorithm. In *Proceedings British Machine Vision Conference*, pages 316–326, 1992.

[80] N. A. Thacker and J.E.W Mayhew. Optimal combination of stereo camera calibrations from arbitrary stereo images. In *Proceedings British Machine Vision Conference*, pages 25–30, 1990.

[81] V Torre and T A Poggio. On edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 147–163, March 1986.

[82] R. W. S. Tregidgo and A. C. Downton. Generalised parallelism for embedded vision applications. In *IEE Colloquium on Parallel Architectures for Image Processing Applications*, April 1991.

[83] H. P. Trivedi. Estimation of stereo and motion parameters using a variational principle. *Image and Vision Computing*, 5:181–183, 1987.

[84] H. P. Trivedi and S. A. Lloyd. The role of disparity gradient in stereo vision. *Perception*, 14:685–690, 1985.

[85] R Y Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. Technical Report Rep. R-921, University of Illinois, August 1981.

[86] R Y Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 13-27, Vol.6,1984

[87] J.R. Ullmann. An algorithm for subgraph isomorphism. In *J. ACM Vol. 23 No. 1*, pages 31–42, 1976.

[88] J. S. Weska and A. Rosenfeld. Histogram modification for threshold selection. *IEEE Trans. Systems, Man and Cybernetics*, SMC-9:38–51, 1979.

[89] R. N. Wolfe. A dynamic thresholding scheme for quantisation of a scanned image. In *Proc. Automatic Pattern Recognition*, pages 143–162, 1969.

[90] A. Y. Wu and A. Rosenfeld. Threshold selection using quadtrees. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-4:90–94, 1982.

[91] Y. Yasumoto and G. Medioni. Experiments in estimation of 3D motion parameters from a sequence of image frames. In *Proceedings IEEE Conf. Computer Vision*, pages 89–95, 1985.

[92] M.J.A. Zemerly, M. Holden, and J.P. Muller. Parallel stereo matching of spot satellite images. In *IEE Colloquium on Parallel Architectures for Image Processing Applications*, April 1991.

Appendix A

Trial Results

This appendix described the results when the DETECT algorithms were applied to twelve triple stereo image sequences. These results should be viewed in conjunction with the discussions presented in Chapter Six. The capture and storage of the sixteen frame sequences was described in Section 6.2 and the dimensions and details of the direction in which the person was moving can be found in Tables 6–1 and 6–2. Backgrounds for each scene can be found in Figures 6–3, 6–4 and 6–5. A worked example is described in Section 6.3.2.

There are two pages of results for each sequence. The first page shows the raw disparities, Kalman filtered disparities and the weighted averaged Kalman disparities. Also shown are the variances and the amplitudes of the disparity measurements. In all graphs time, in frames, is along the x axis.

The second page of results shows the disparity histograms for each sequence through time. It is from these graphs that the sub-pixel disparity measurements, shown in the first page, are extracted. Confidences can also be extracted from the disparity histograms.

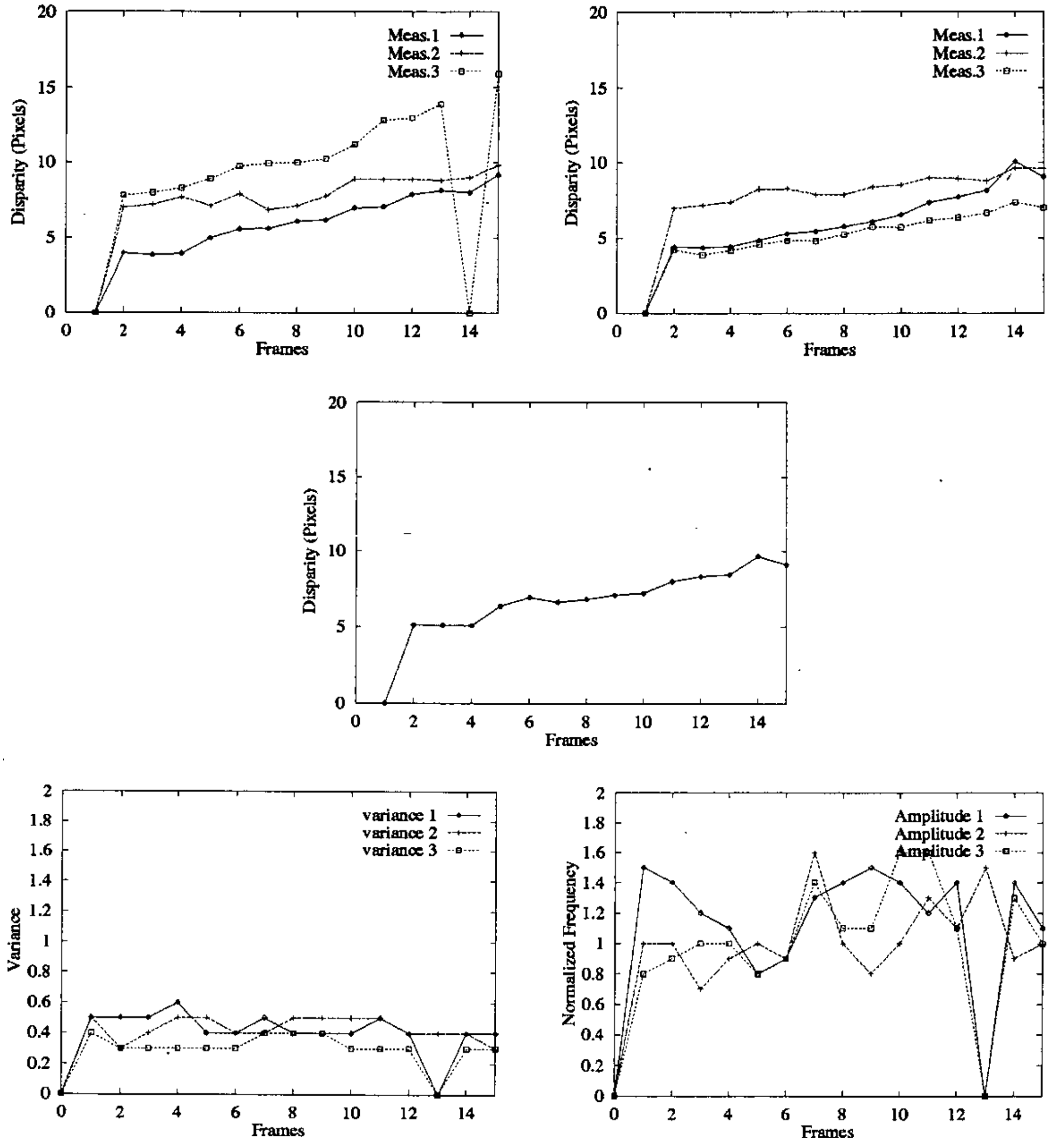


Figure A–1: Sequence 1: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

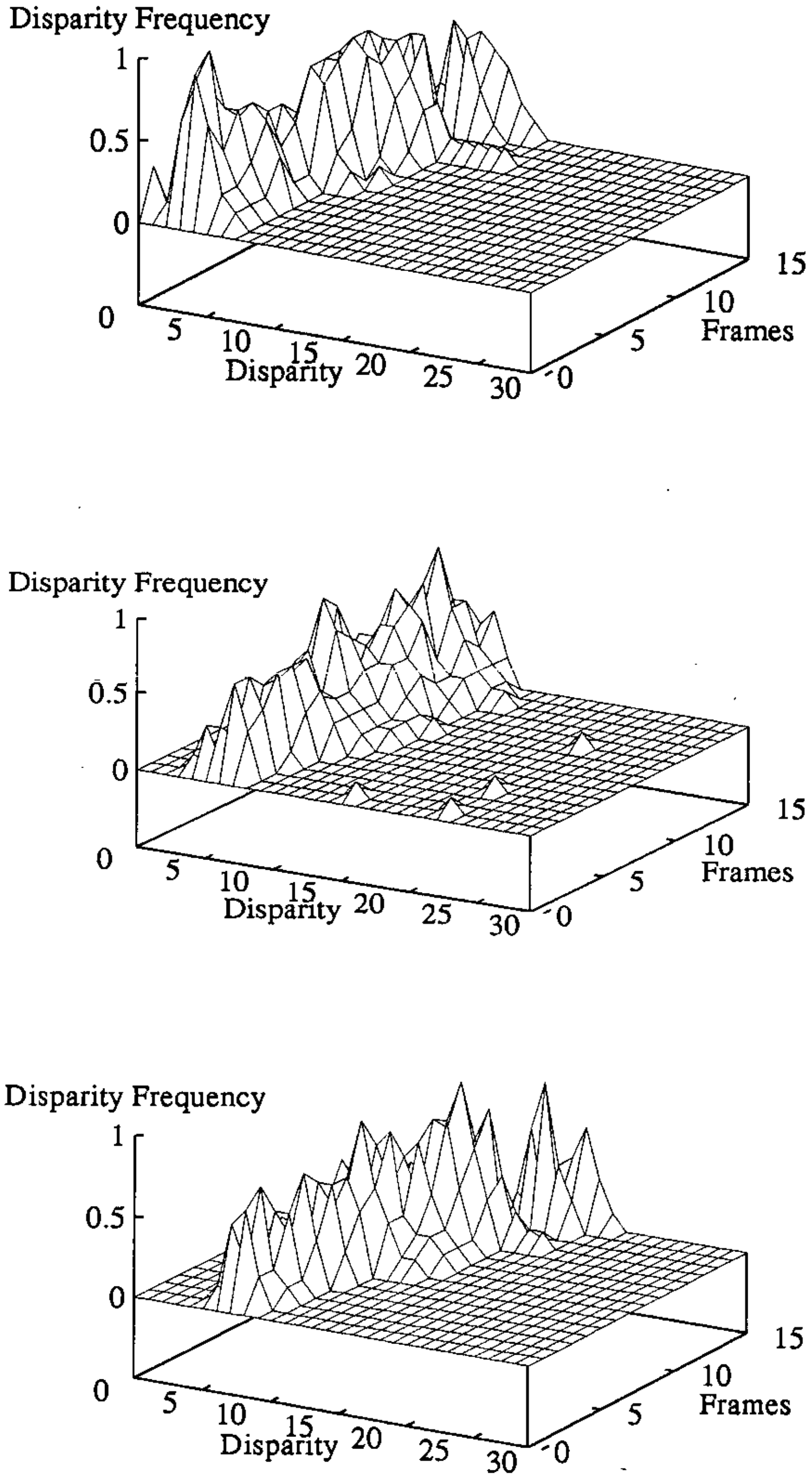


Figure A–2: Sequence 1: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

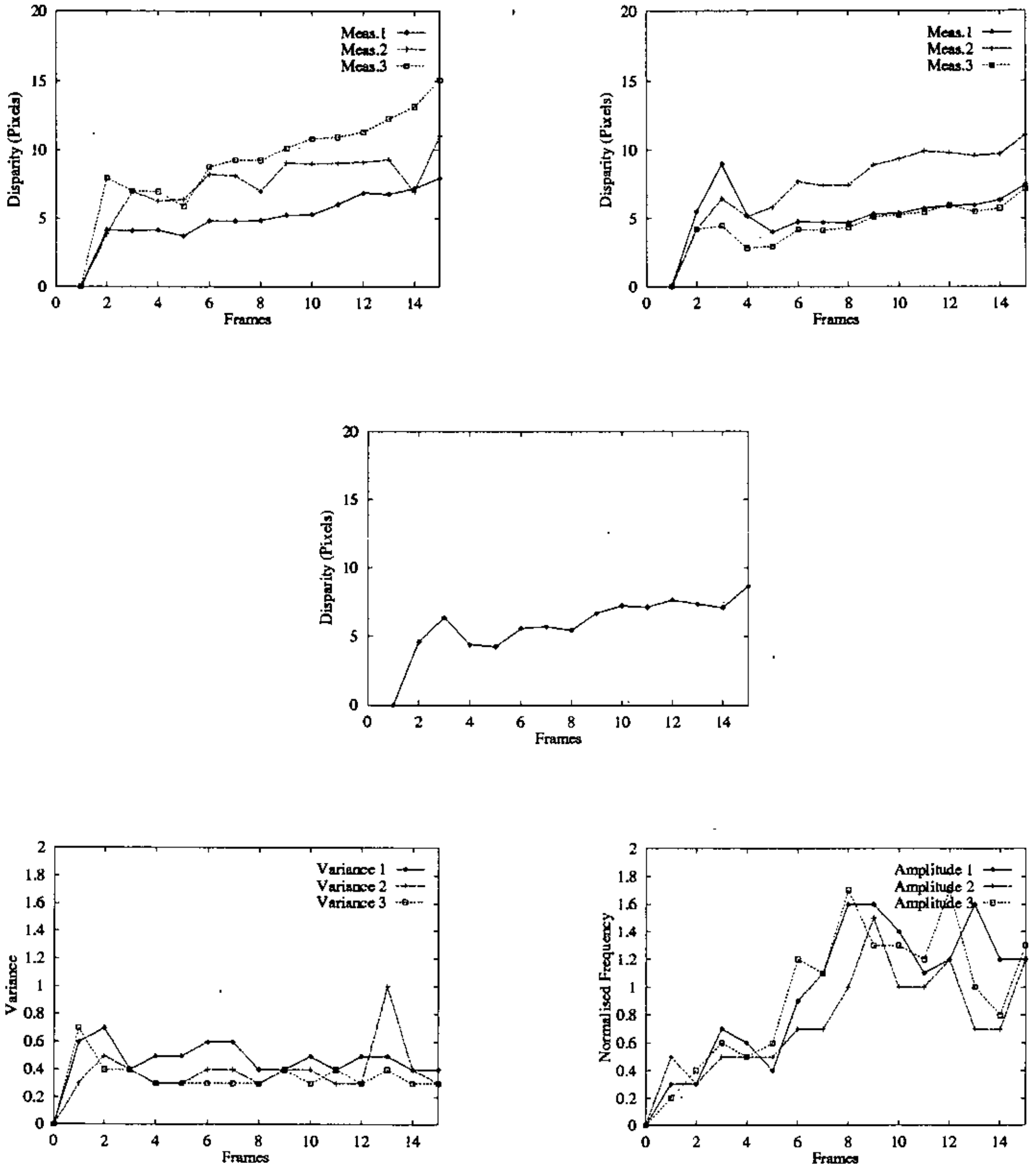


Figure A–3: Sequence 2: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

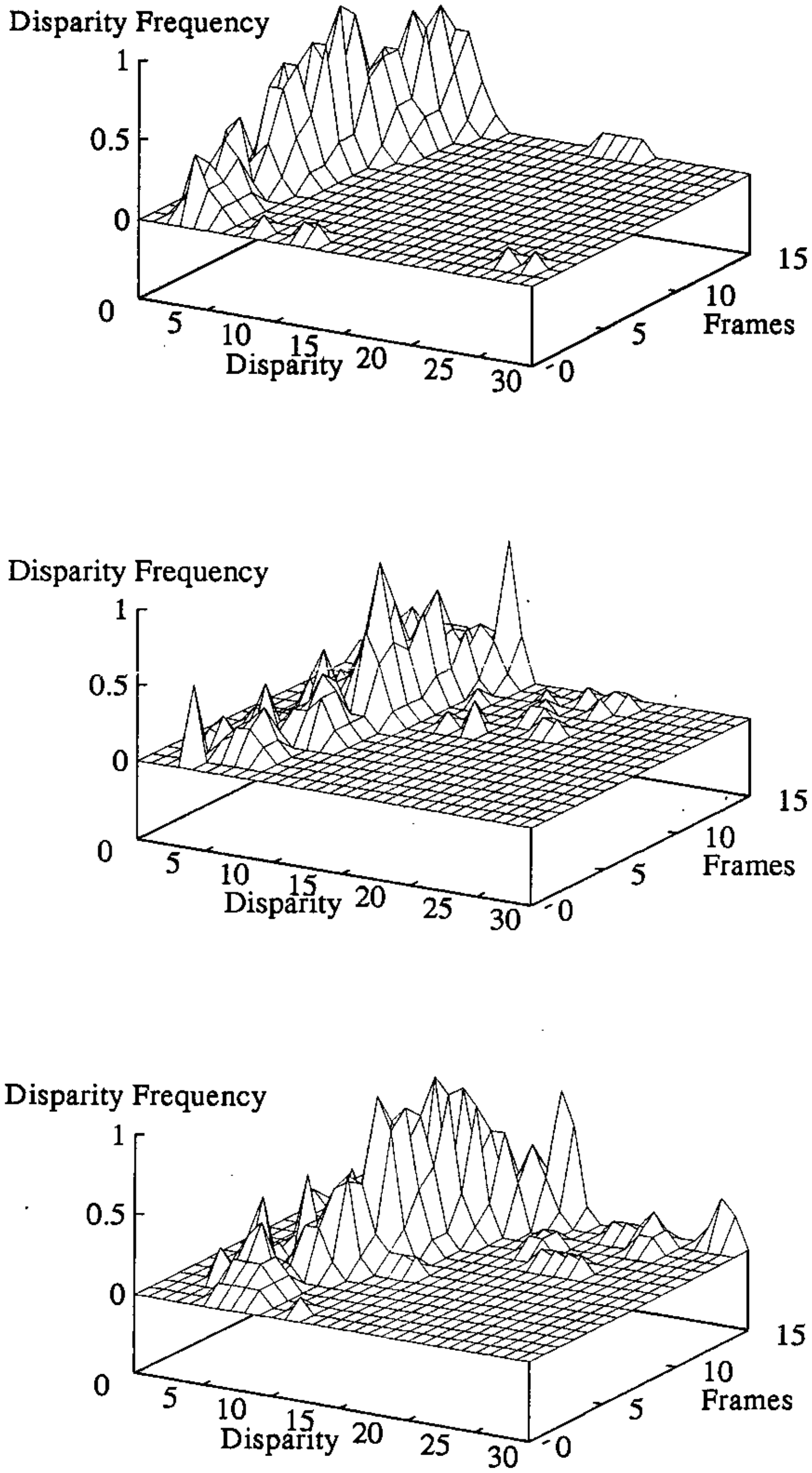


Figure A–4: Sequence 2: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

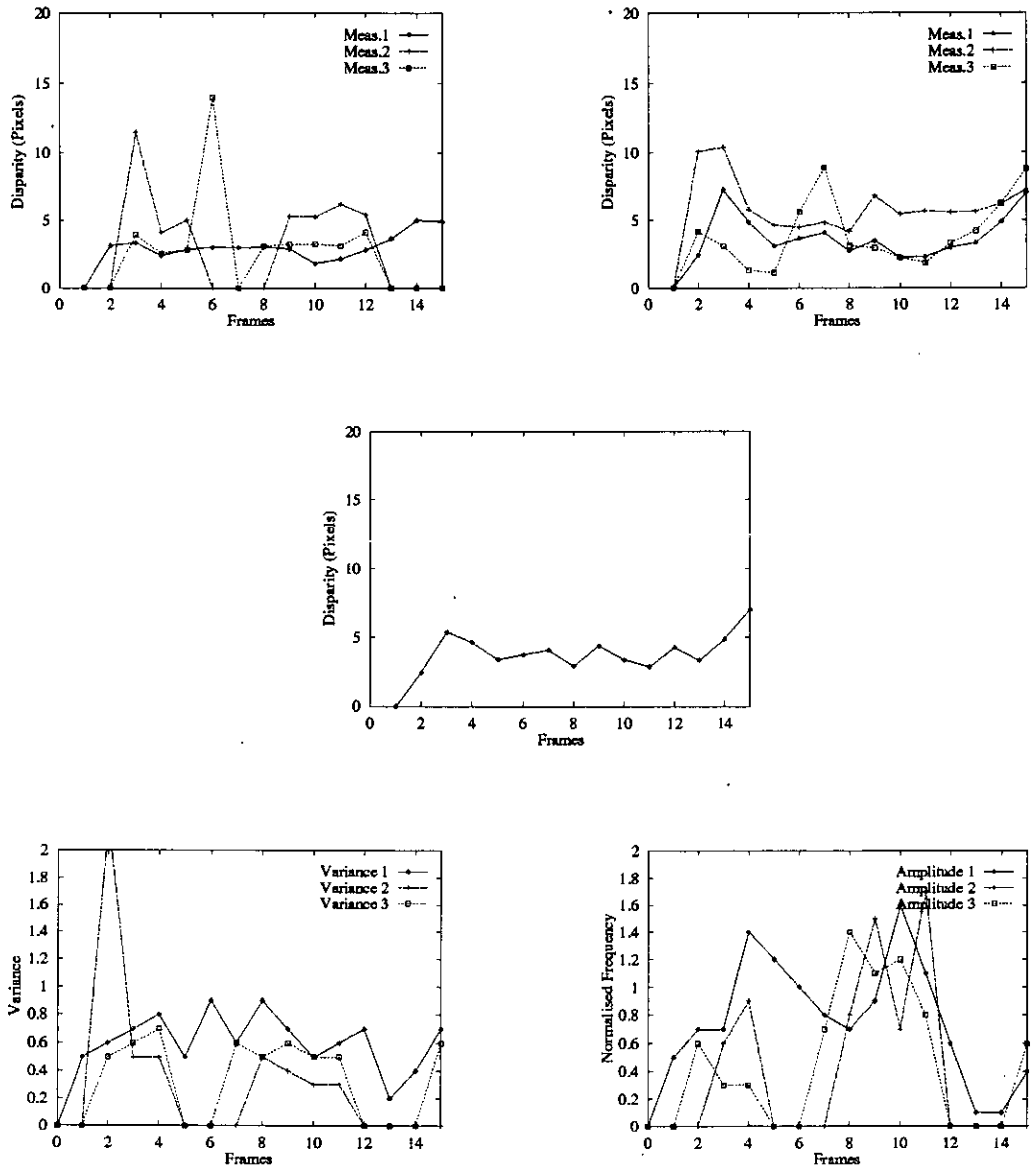
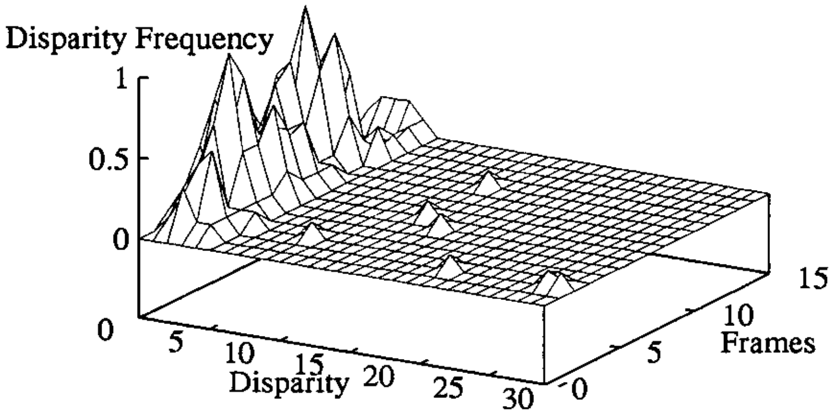
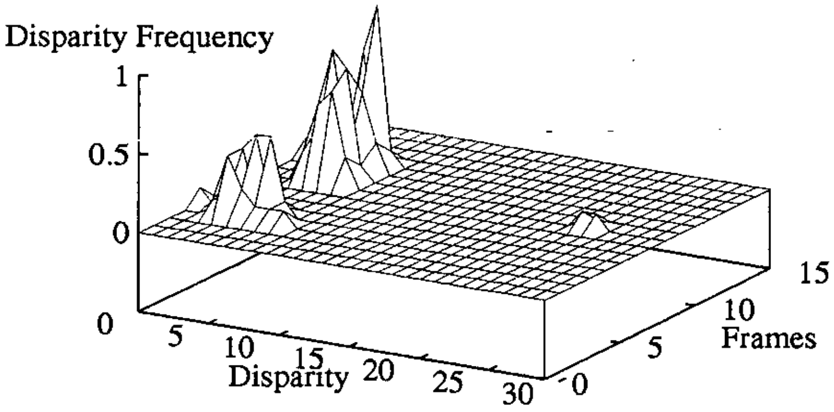


Figure A–5: Sequence 3: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)





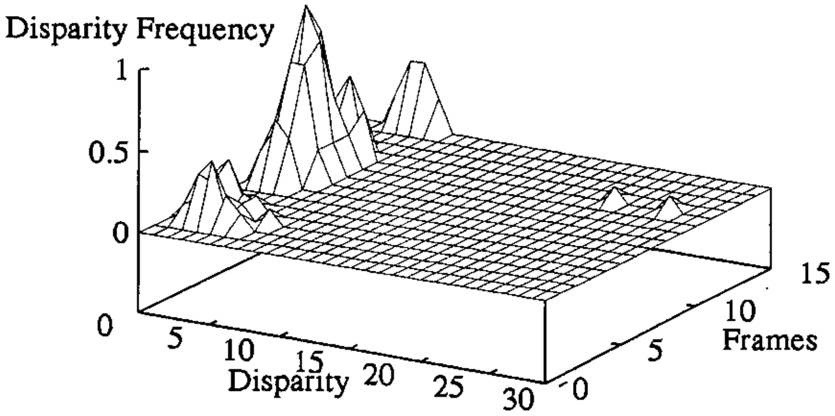


Figure A–6: Sequence 3: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

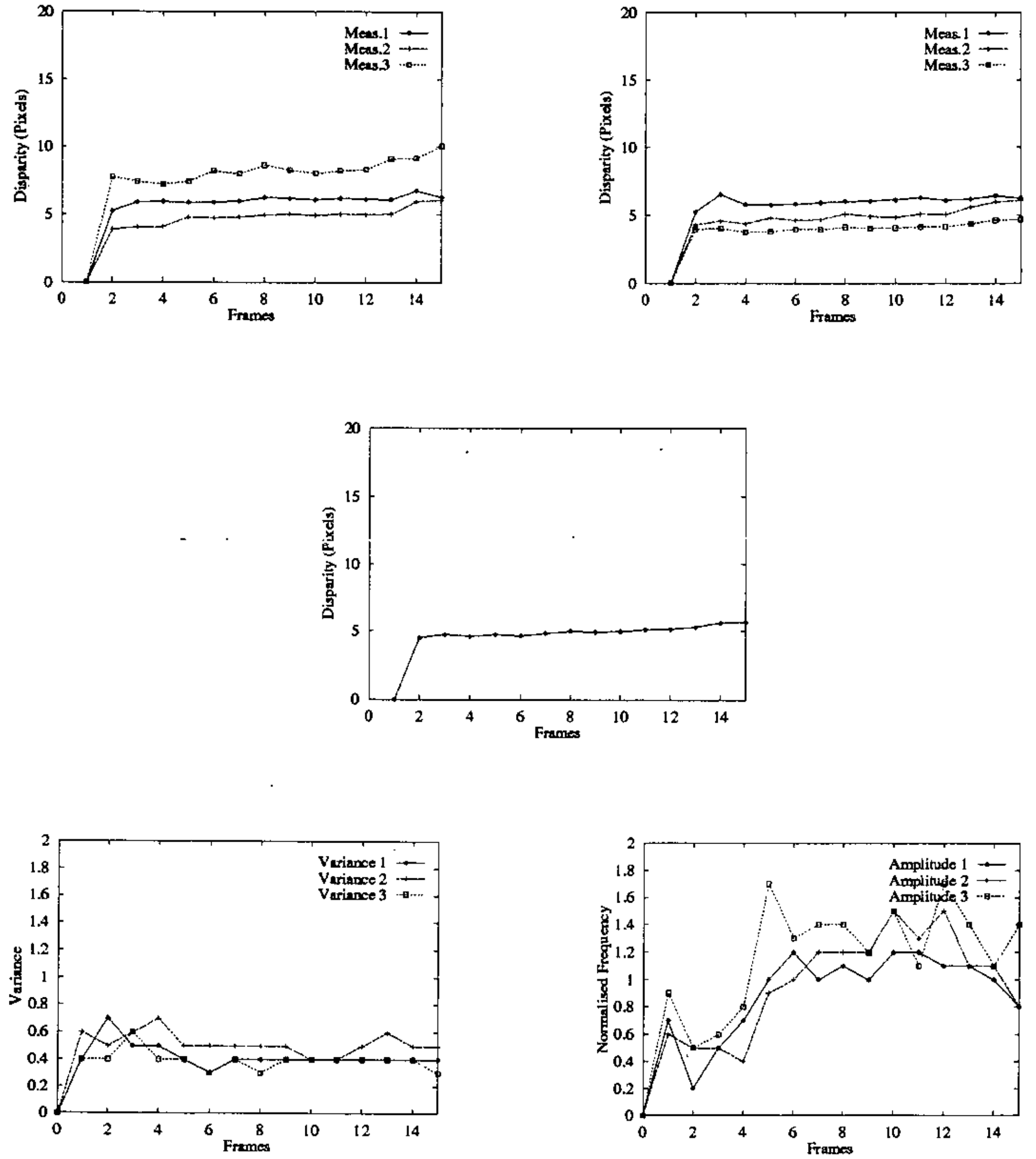
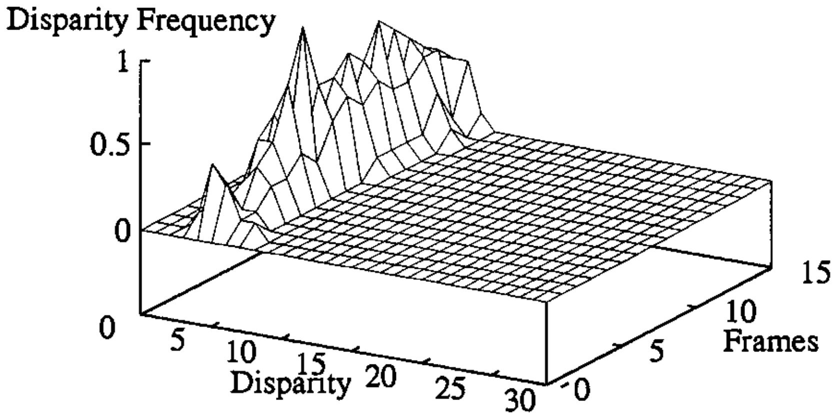
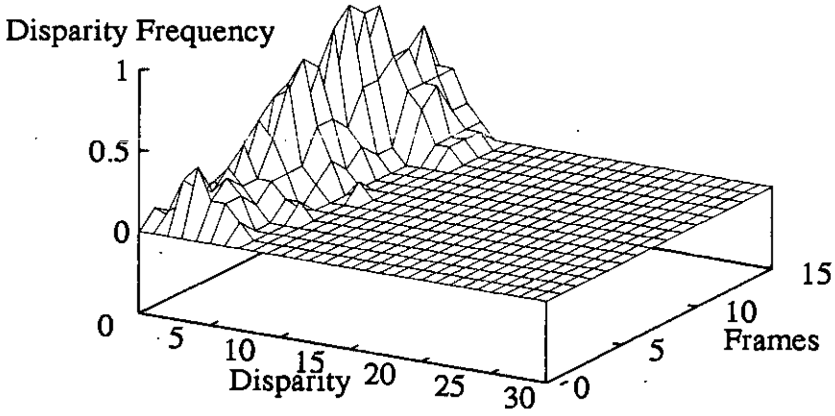


Figure A–7: Sequence 4: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)





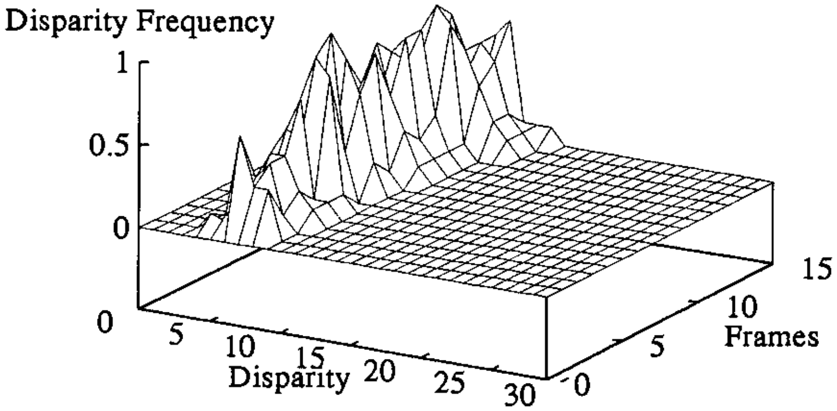


Figure A–8: Sequence 4: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

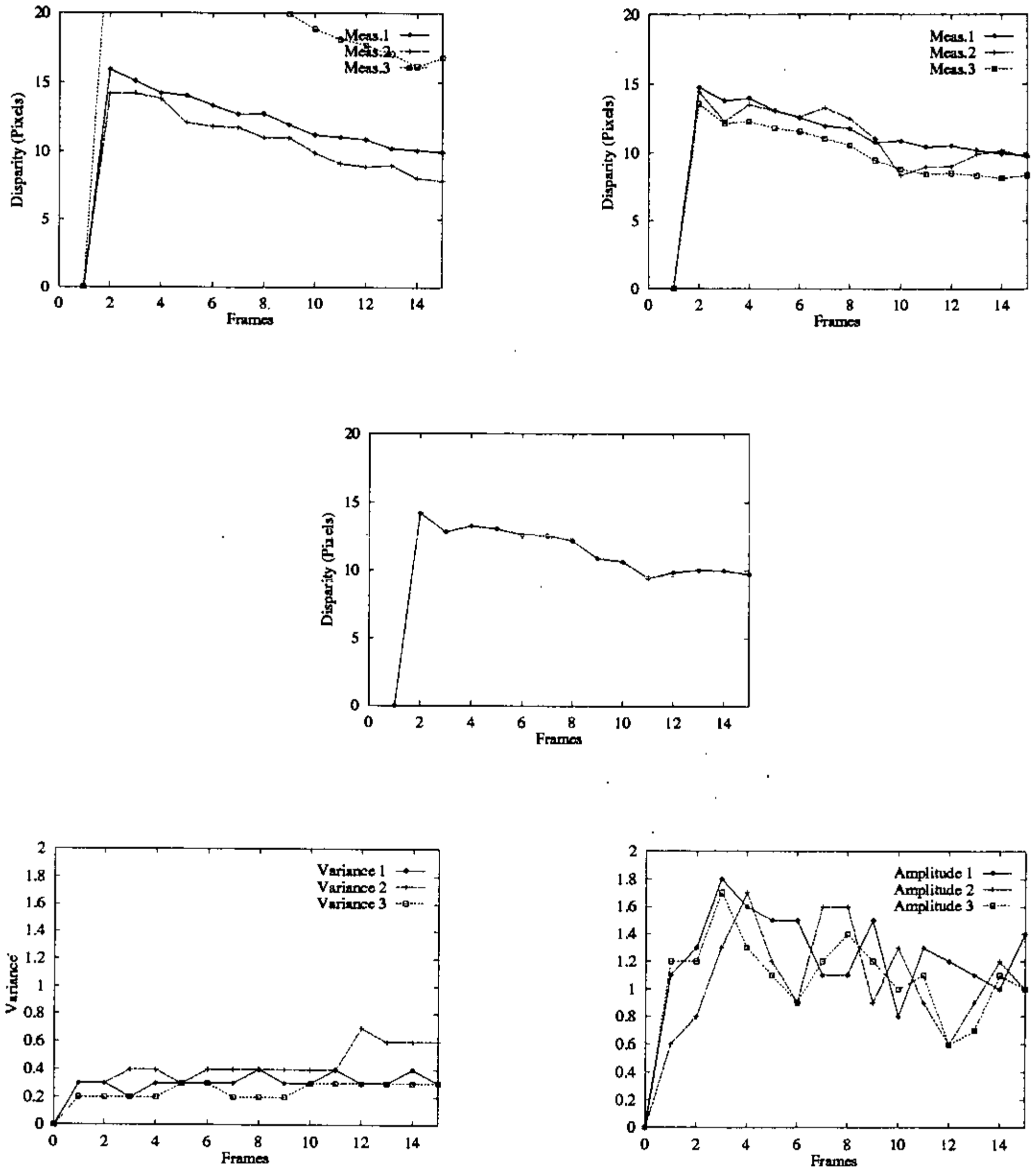


Figure A–9: Sequence 5: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

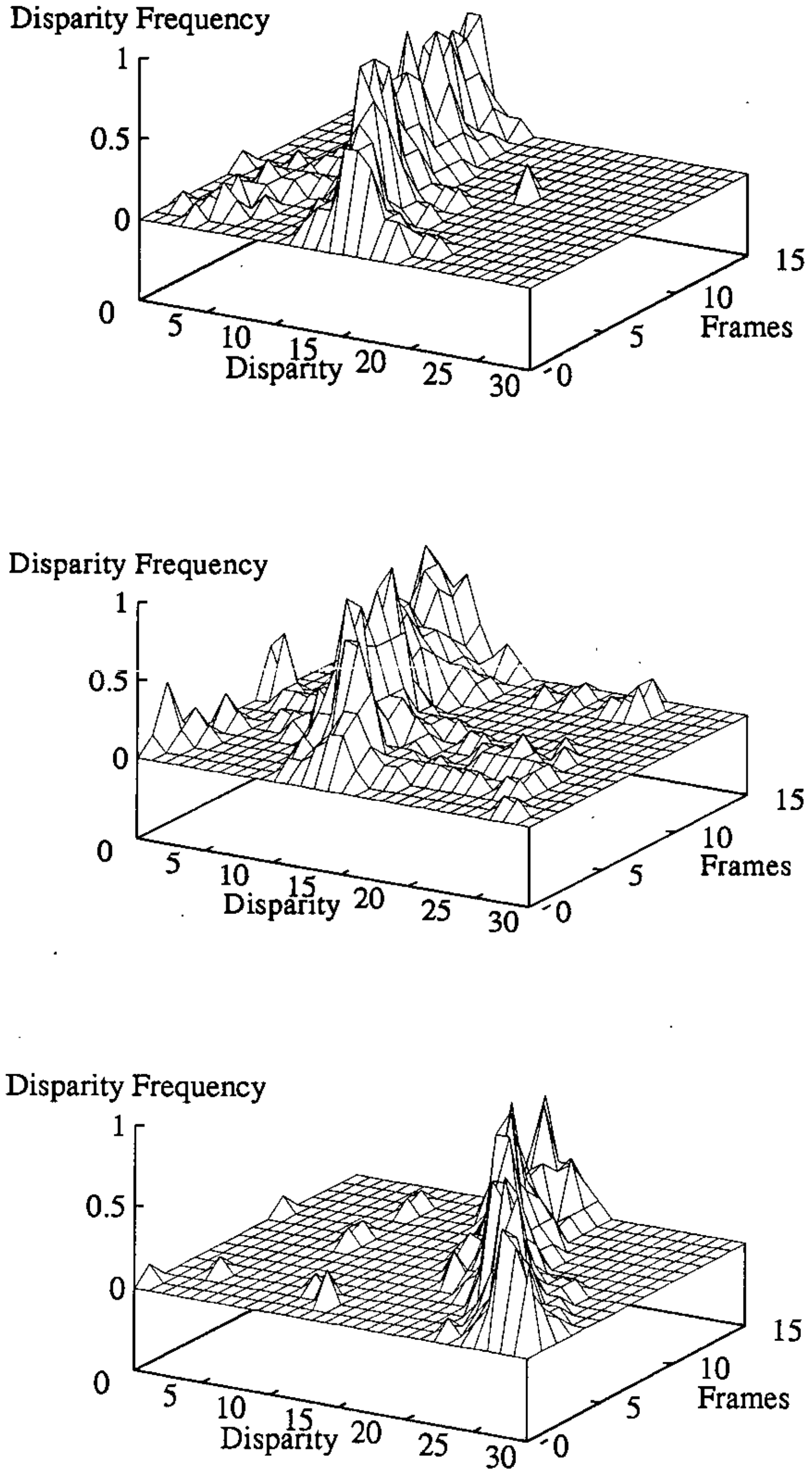


Figure A–10: Sequence 5: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

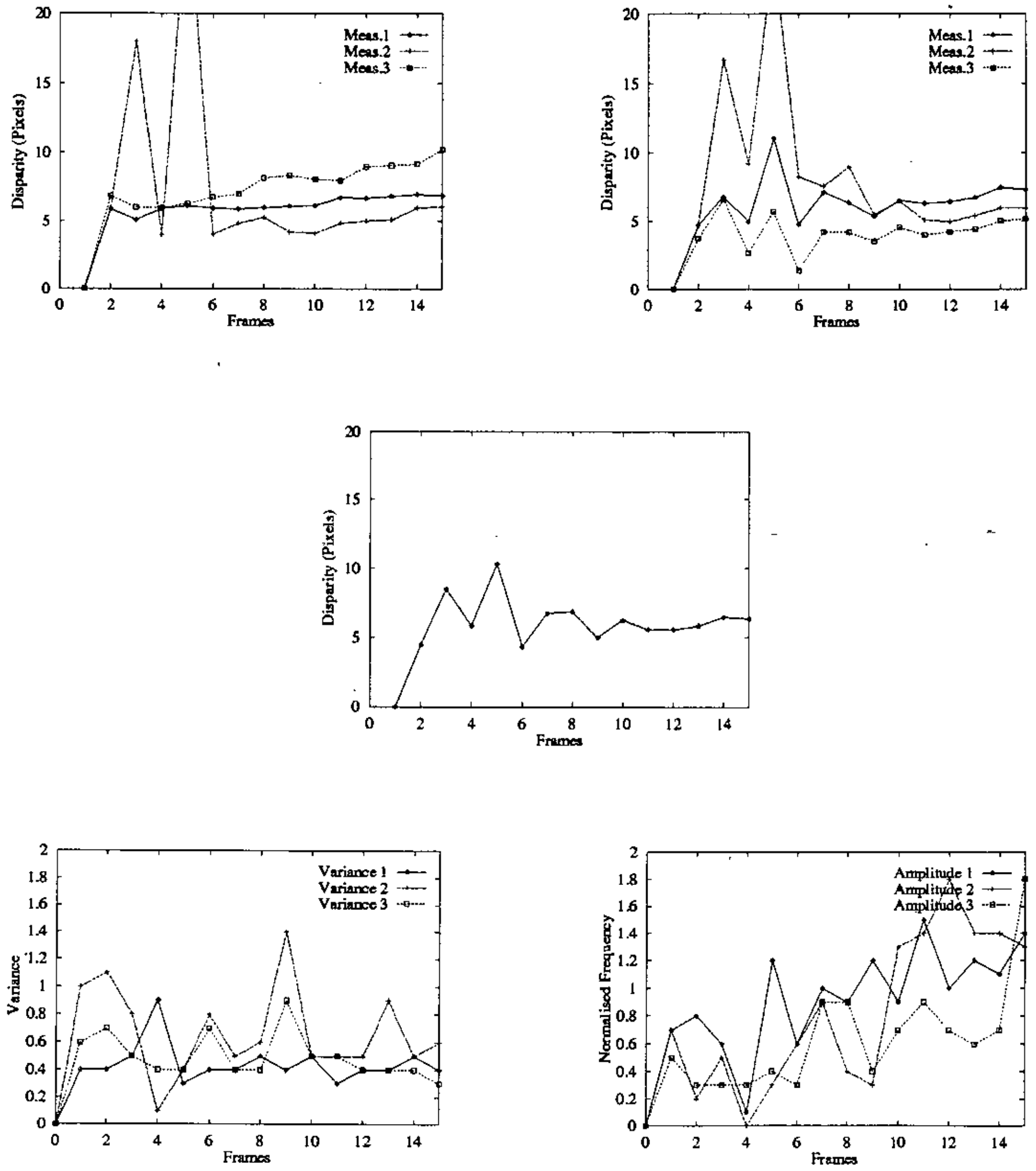


Figure A–11: Sequence 6: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

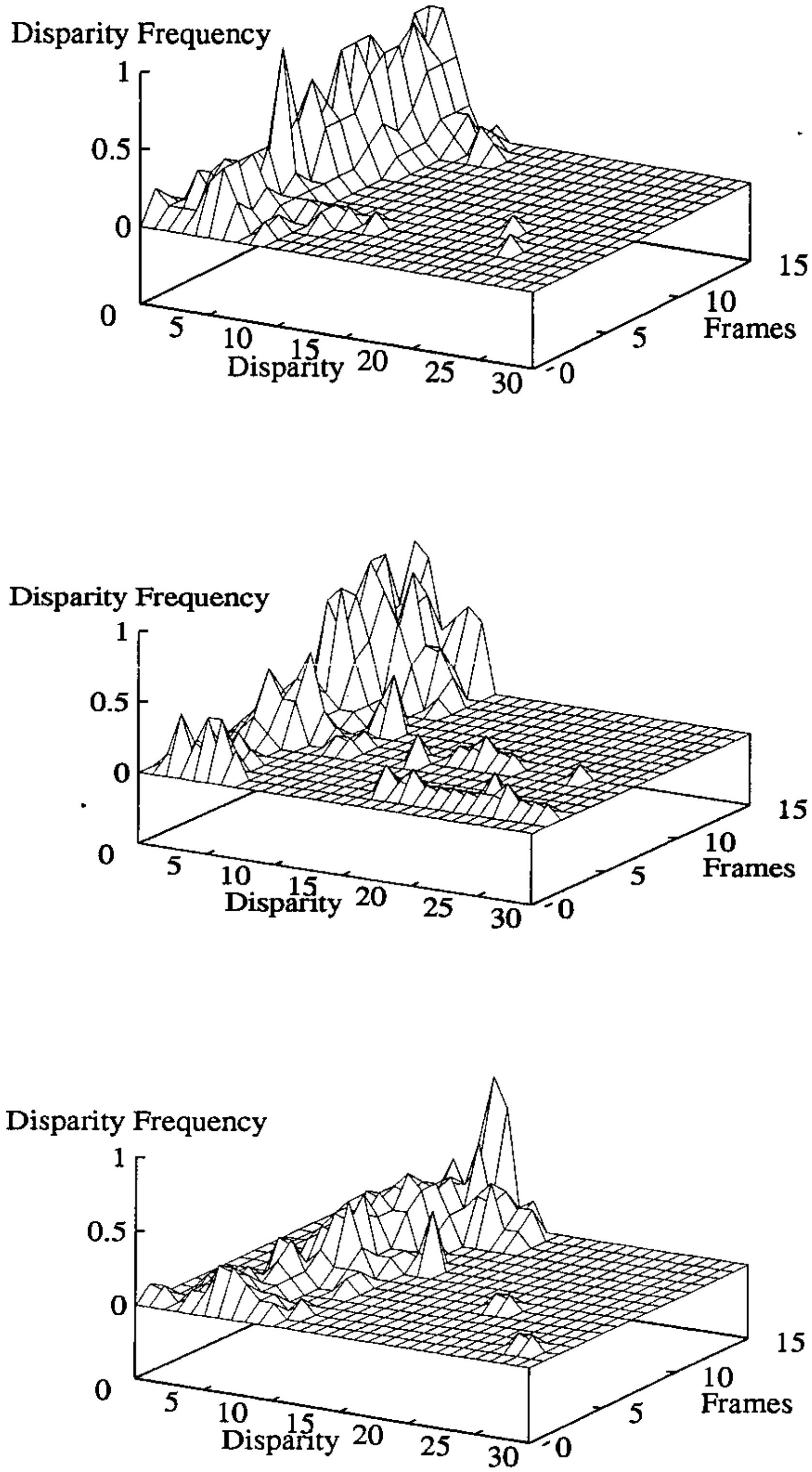


Figure A–12: Sequence 6: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

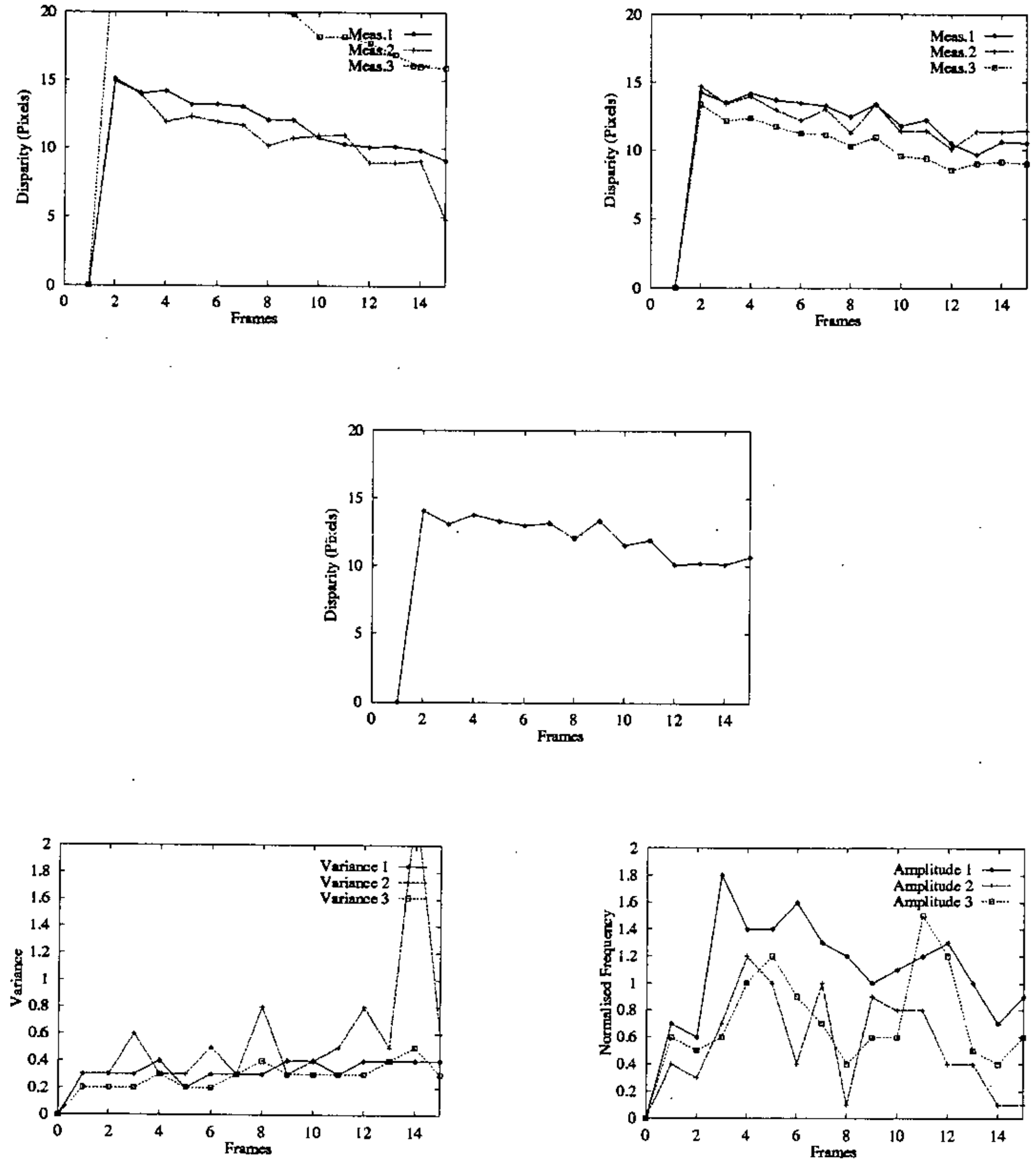
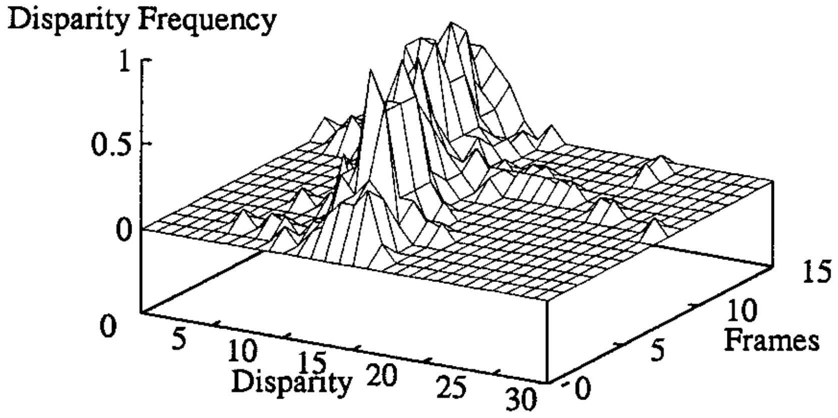
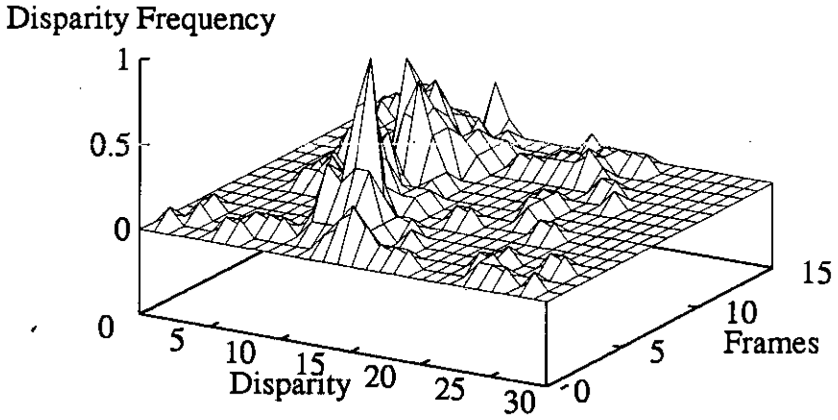


Figure A–13: Sequence 7: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)





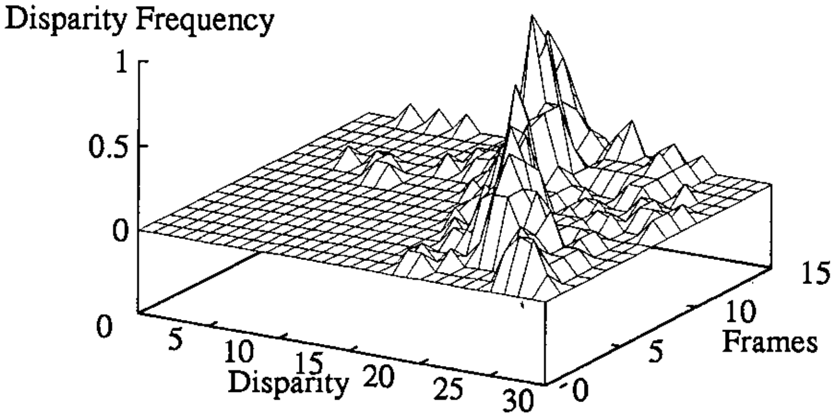


Figure A–14: Sequence 7: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

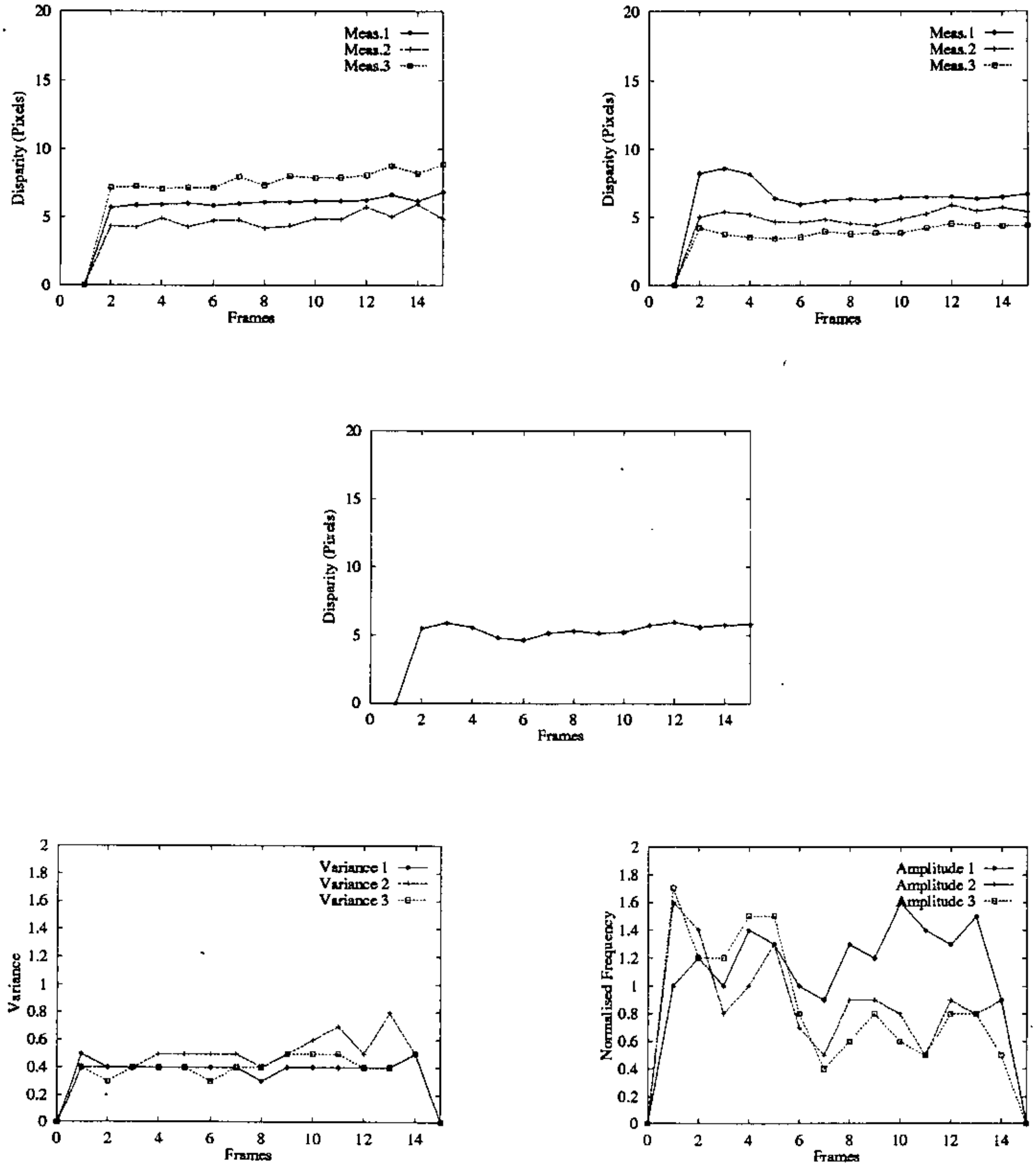


Figure A–15: Sequence 8: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

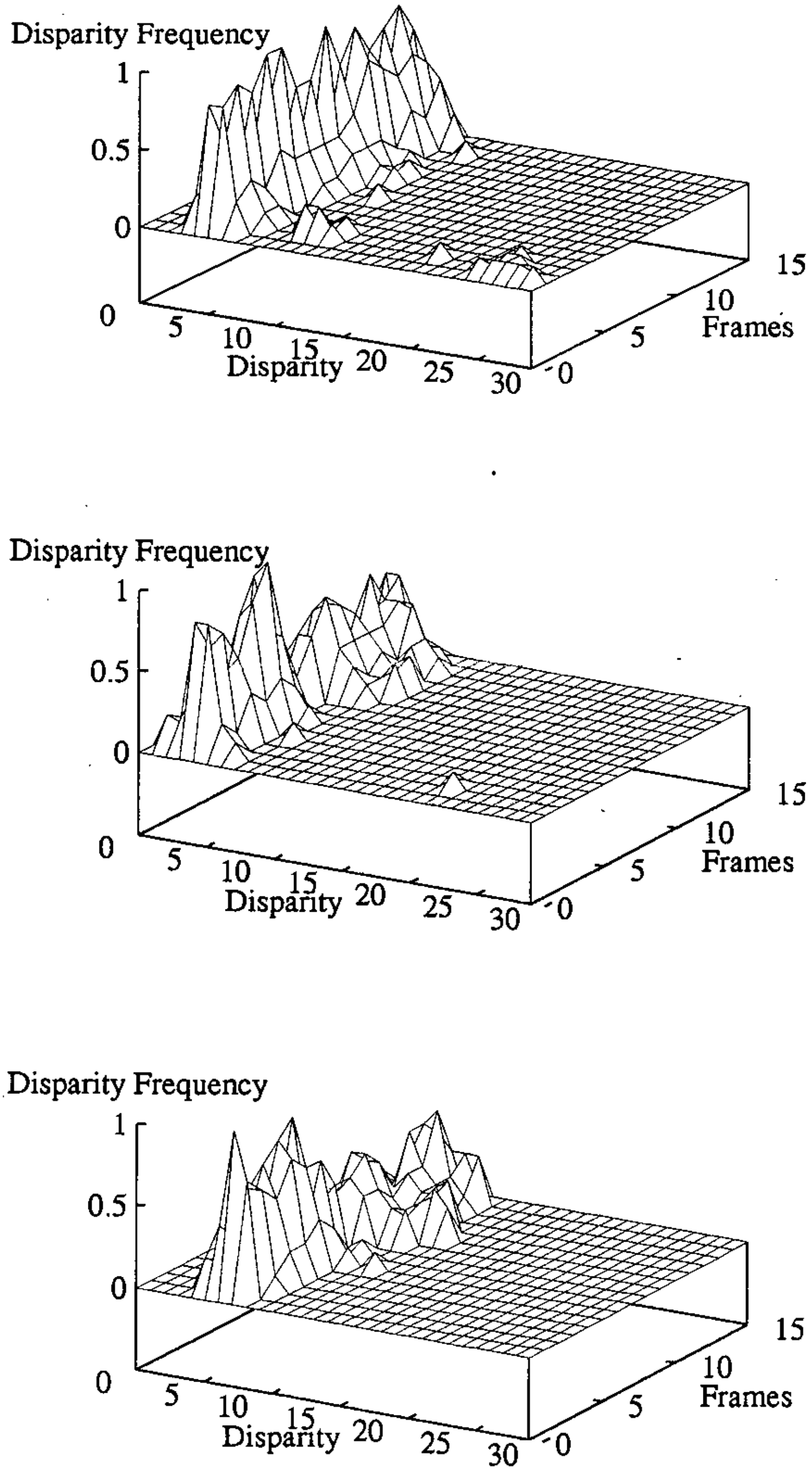


Figure A–16: Sequence 8: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

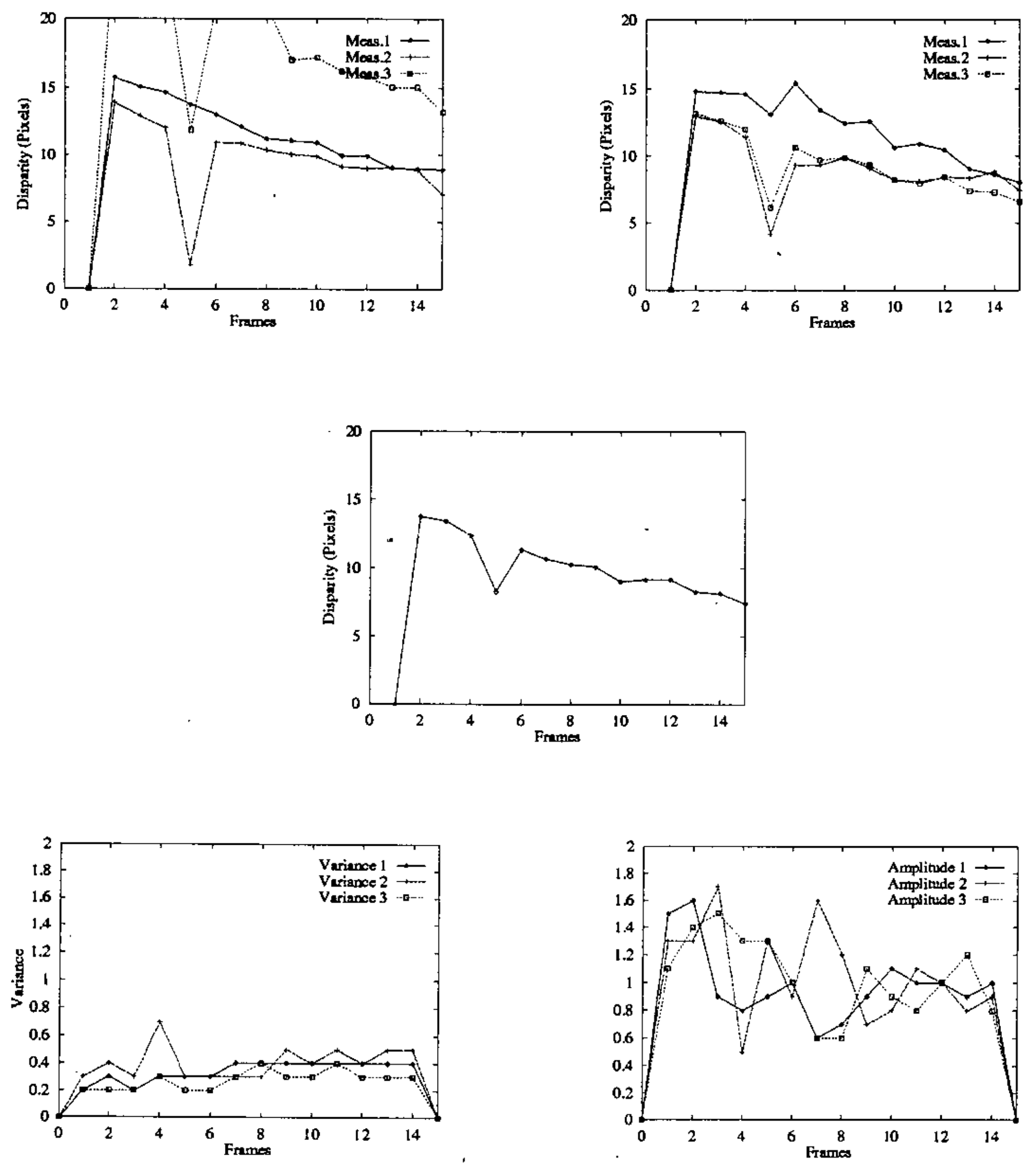


Figure A–17: Sequence 9: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

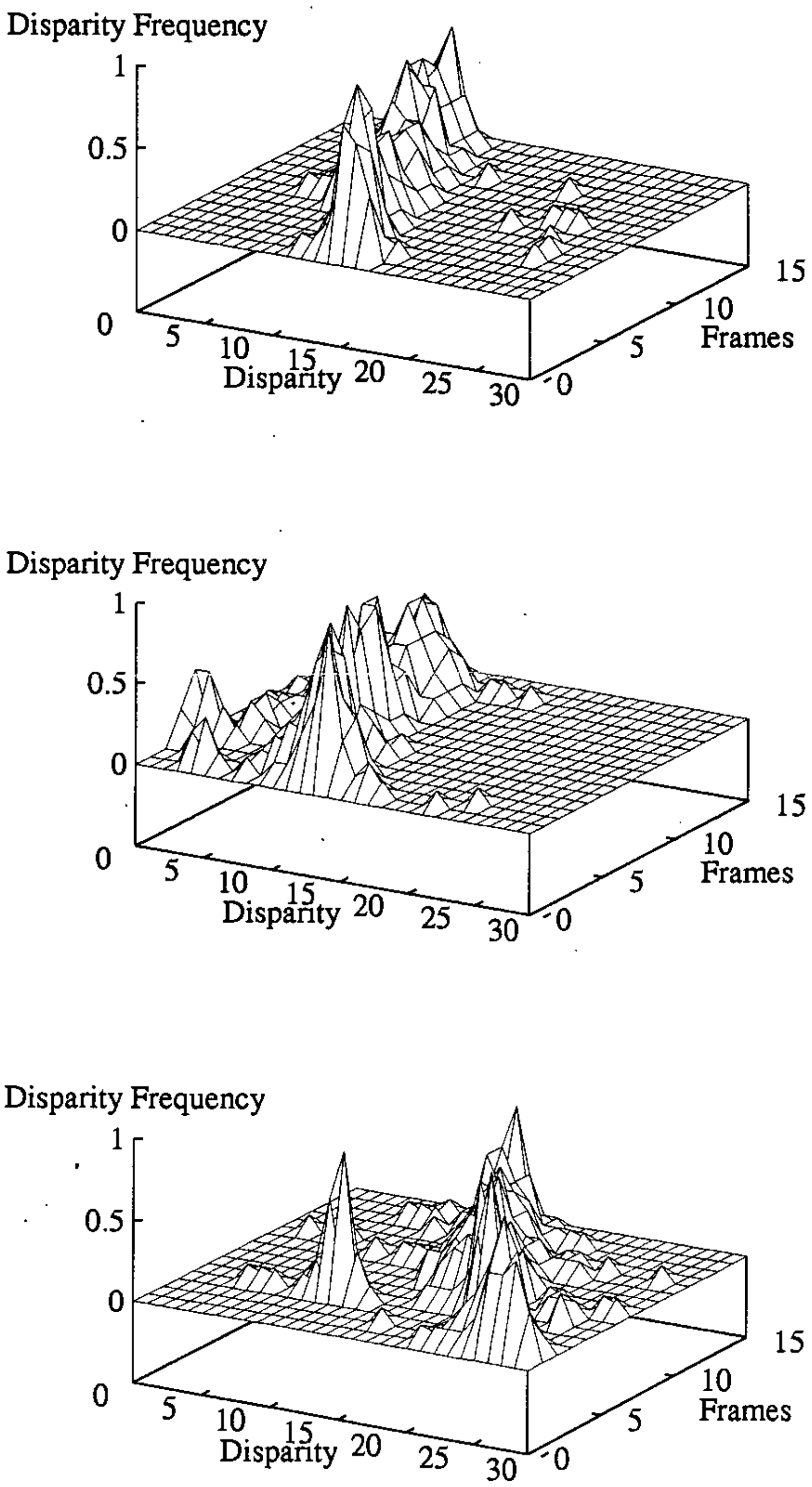


Figure A–18: Sequence 9: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

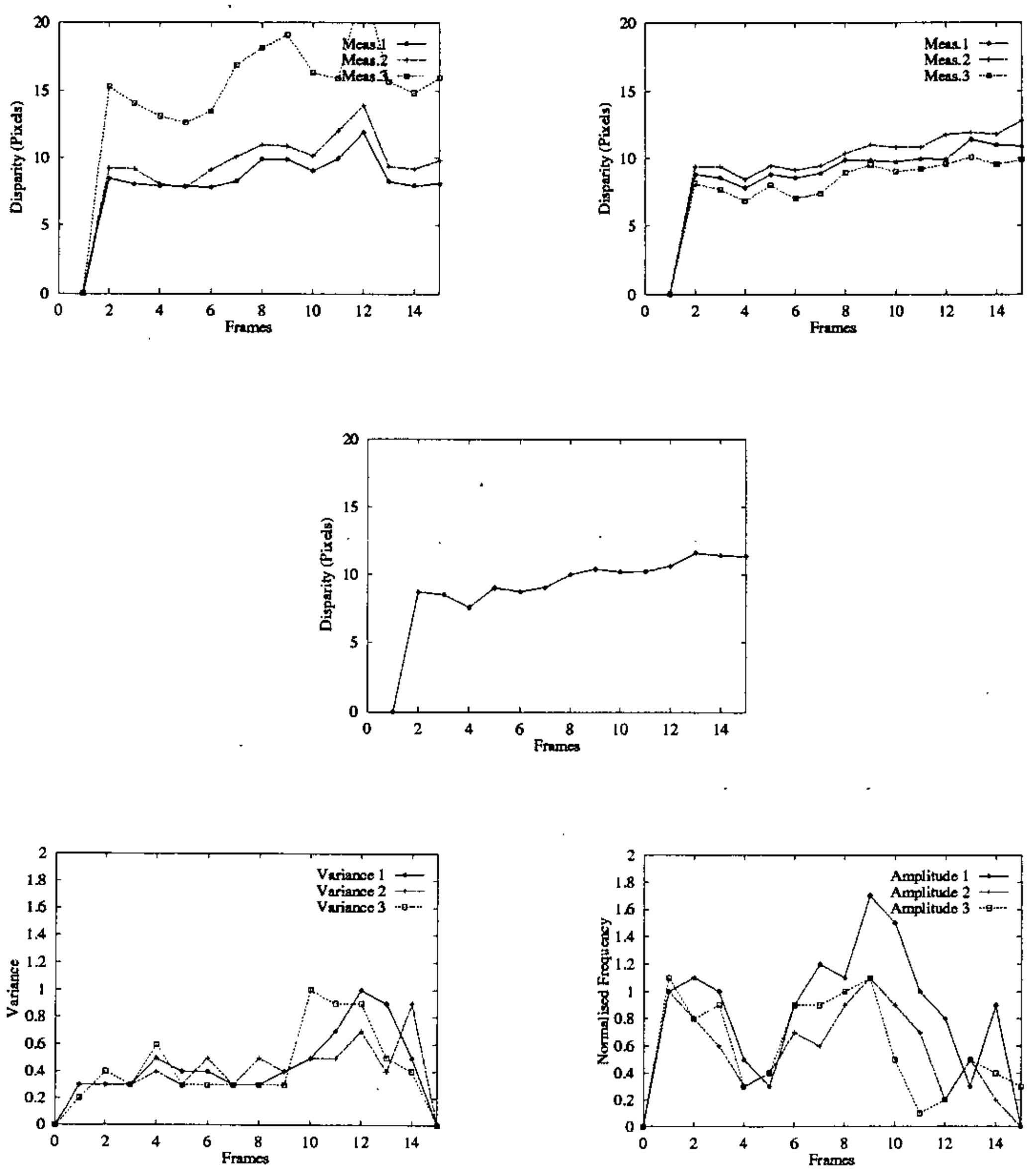


Figure A–19: Sequence 10: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

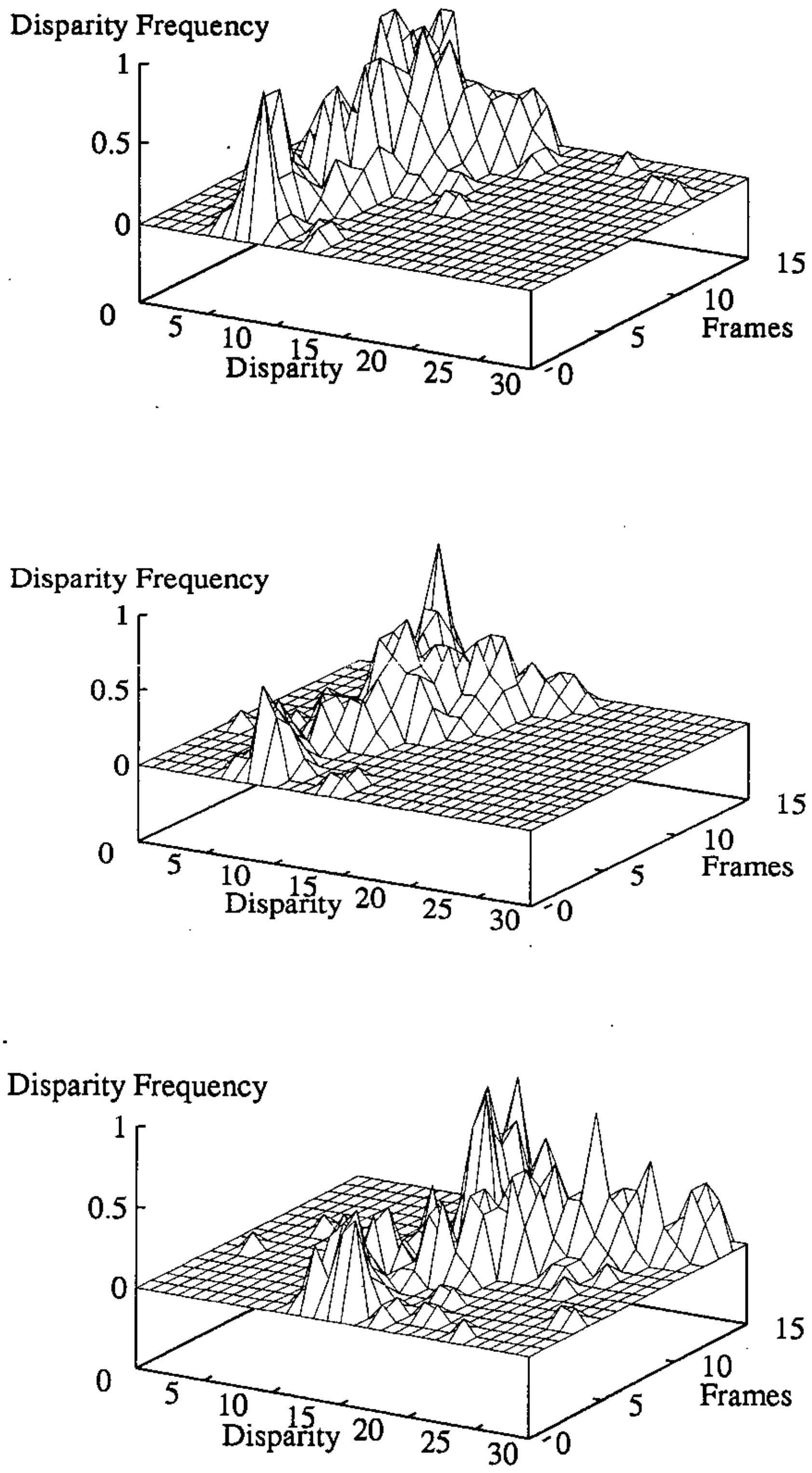


Figure A–20: Sequence 10: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

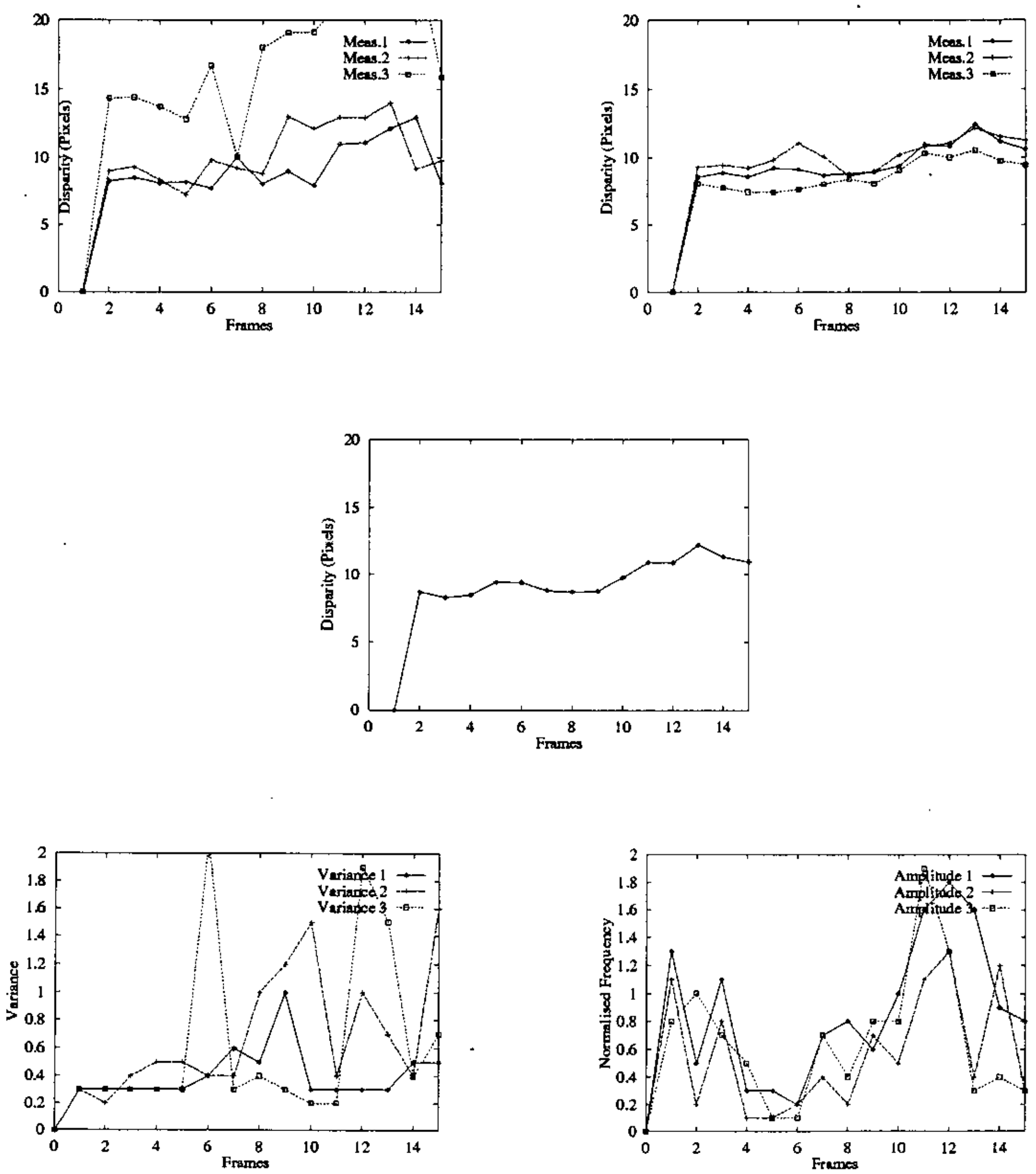
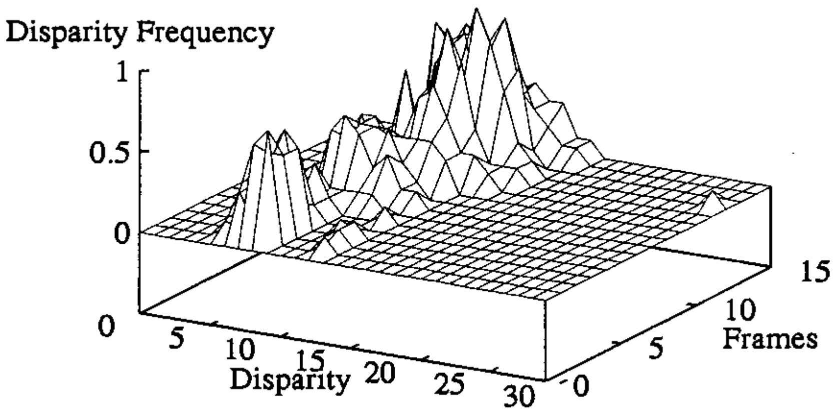
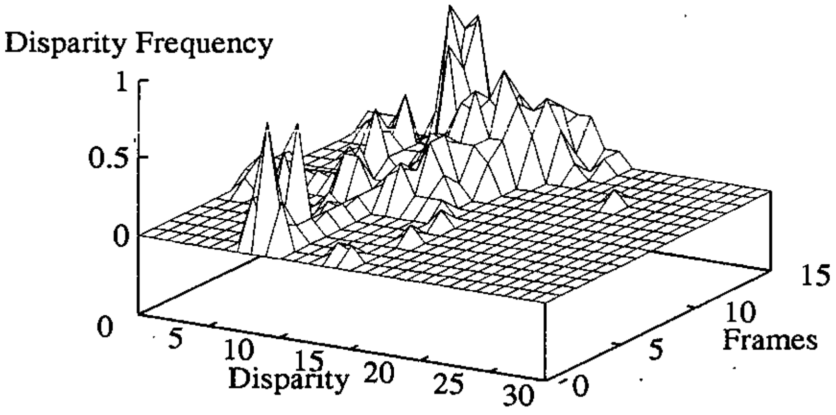


Figure A–21: Sequence 11: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)





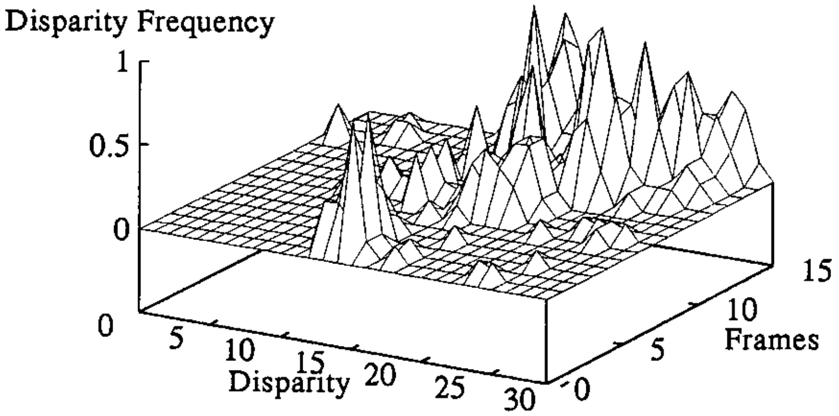


Figure A–22: Sequence 11: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

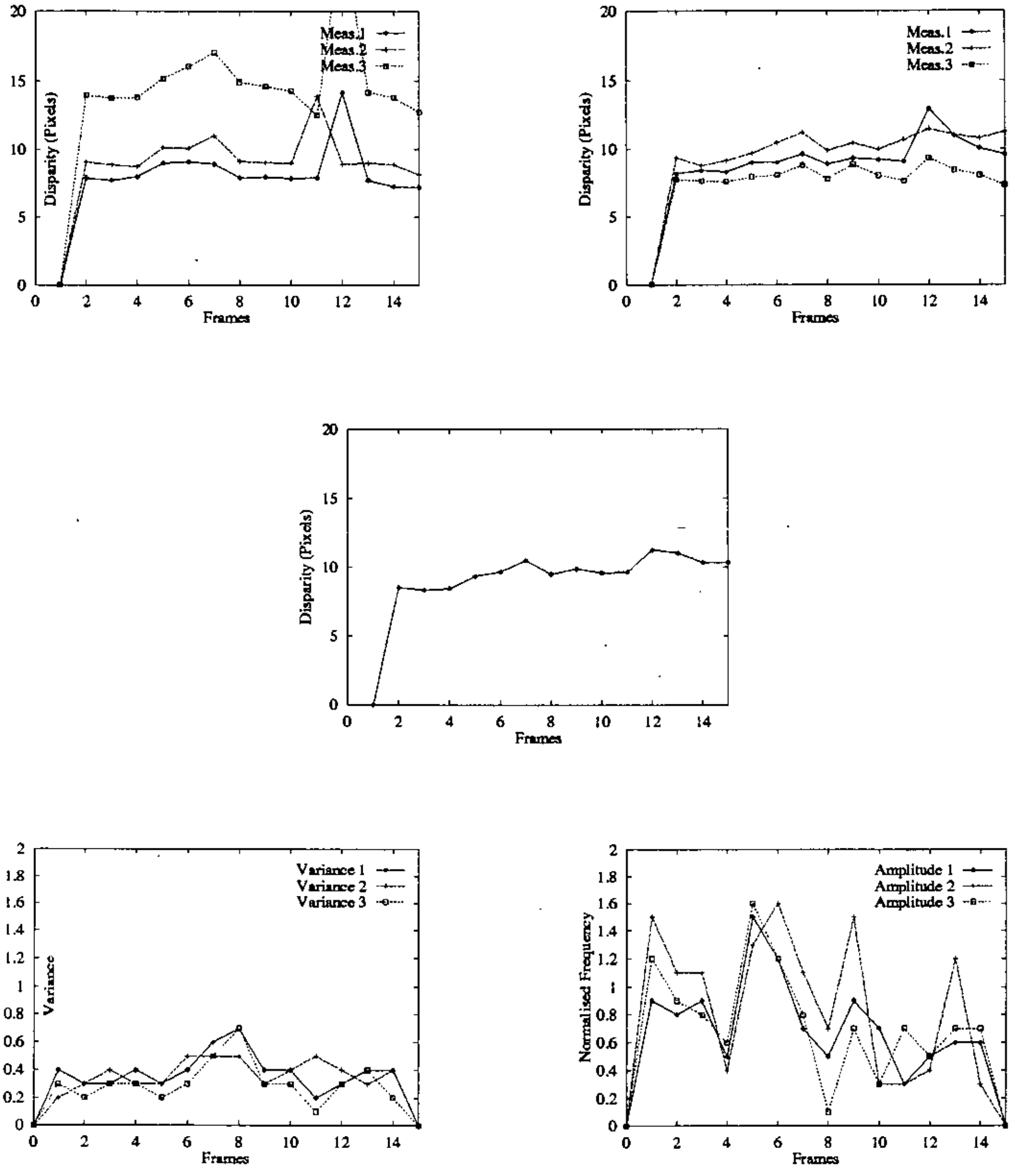


Figure A–23: Sequence 12: Raw Disparities (Top Left), Kalman Output (Top Right), Average Output (Middle), Variances (Bottom Left), Amplitudes (Bottom Right)

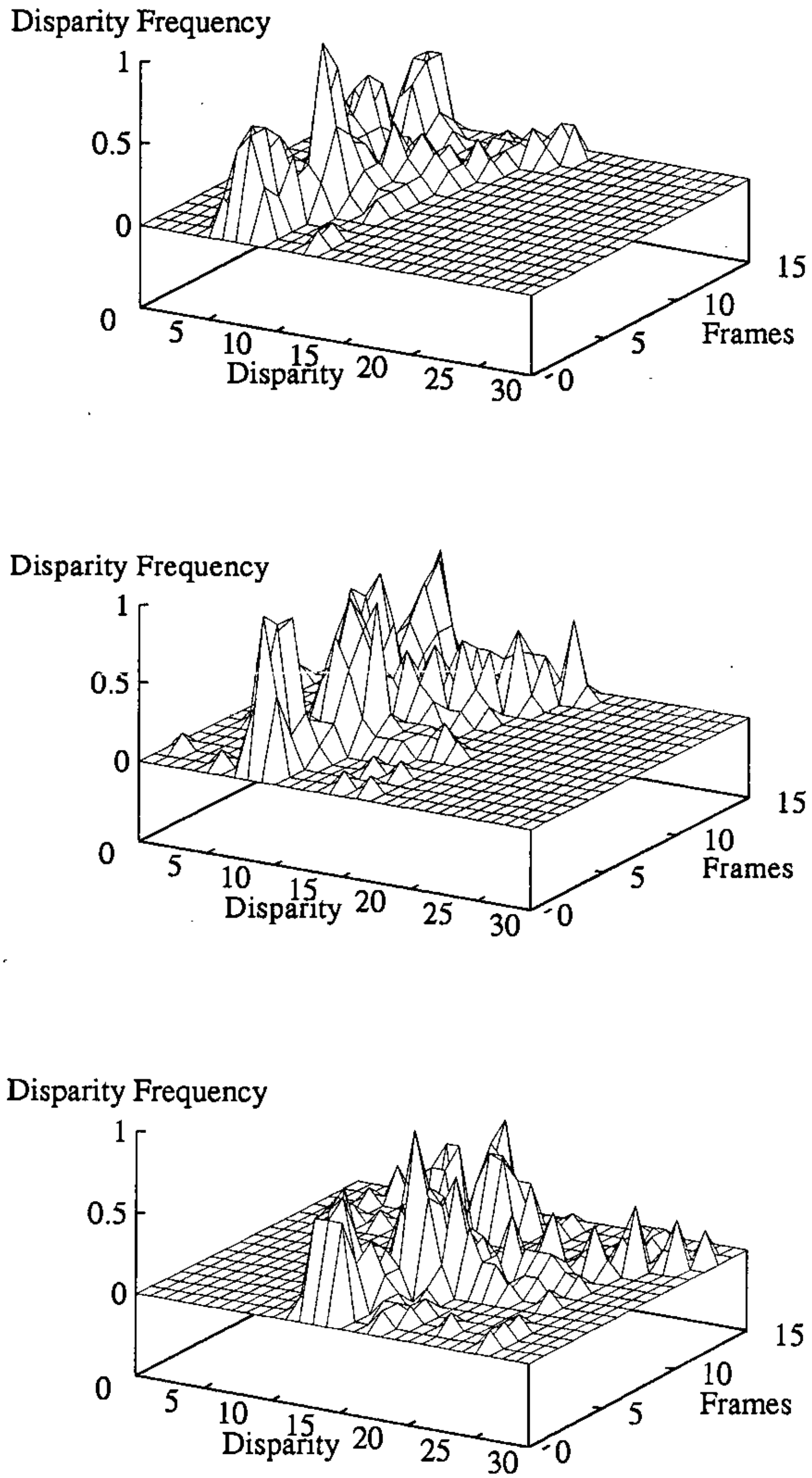


Figure A–24: Sequence 12: Disparity Histograms: Measurement 1 (Top), Measurement 2 (Middle), Measurement 3 (Bottom)

Appendix B

The DETECT System Software

This appendix describes the function of the main procedures used to test the DETECT algorithms on the trial sequences. The *main* procedure is contained in *detect.c* which handles all file loading and saving operations. Each time step consists of three 256x256 images for the left, middle and right cameras requiring considerable disk storage for the 12 trial sequences. Due to the storage requirements, the image archives are stored over two disk systems.

Separate files contain routines which implement different aspects of the algorithm:-

1. **cluster.c:** *cluster.c:* contains the routines, *clusters, mergeclusters:* and *expandclusters* which, respectively, extract connectivity from the input images, combine smaller regions into larger ones and expand cluster outlines. The task of these routines is to find significant connected areas of the difference image. The boundaries of the connected region are output, together with its size and grey level histogram, as a size sorted linked list. Each cluster structure is defined to allow direct connections to individual items in the edge list. The code is written without recursion to make the memory and resources required explicit for implementation in hardware.

2. **edgediff.c:** Routines to laterally differentiate, *latedge*, and calculate the difference between current background and foregrounds, *diff.* Lateral non-maximal suppression is also applied in the *latedge* routine.

3. **group.c:** A routine which combines cluster information with the edges provided by *tracking.c.* Direct connections, dependent on proximity, are established between the cluster list and the, normally, longer edge list.

4. **kalman.c:** Provides all the necessary matrix arithmetic and procedures to maintain Kalman estimates of the current disparity. The initial covariance matrices are also defined in this file.

5. **matching.c:** This file contains several routines to perform stereo matching, disparity histogram extraction and time domain disparity analysis. These are contained, respectively, in *matchingclusters:*, *disparityprocessing* and *postprocessing.* Also implemented, but not currently used, is a routine to match edges through time, *timematching.*

6. **tracking.c:** A downward tracking algorithm using hysteresis thresholding, *trackpeaks*, is defined in this file. Criteria for an edge's existence such as strength and length are also defined in *trackpeaks* and associated sub-routines, *edgestart*, *edgecont* and *edgecont1.*

7. **defs.h:** Definitions of data structures such as edges, lists of pixels and clusters.

8. **gendefs.h:** The various thresholds used throughout the system are defined in this file.

The segmentation stages produce candidate edges and possible clusters, both stored as linked lists. The main features included in edge list elements are:

1. Reference Number.

2. Starting Row and Column.

3. List of pixels in this edge.

4. Length.

5. Cluster with which this edge is grouped.

6. Histogram to compile disparity frequencies as edge is matched.

7. Total changes in disparity as the edge is tracked.

The main components of the cluster list are:

1. Cluster number.
2. Size.
3. Cluster boundaries.
4. List of outline pixels.
5. List of attached edges.
6. Combined disparity histograms of attached edges.
7. Histogram of grey level difference values

Although a list of background edges has to be retained at each frame, the storage requirements, in this software, for the above data structures are small when compared to that necessary for the background and intermediate images.

Appendix C

Publications

This appendix includes examples of work published during the course of this research.

1. K.W.J Findlay, W.J.C. Alexander and A.J. Walton, "CORSIM: A 2-D Simulator for Contacts with Arbitrary Geometry", Alvey Club Meeting, Process and Device Modelling, September, 1988.
2. K.W.J Findlay, W.J.C. Alexander and A.J. Walton, "The Effect of Contact Geometry on the Value of Contact Resistivity Extracted from Kelvin Structures", Proc. IEEE 1989 Int. Conference on Microelectronic Test Structures, Vol. 2, No. 1, March 1989.
3. K.W.J Findlay and D. Renshaw, "Stereo Algorithm to Reduce Quantisation Noise Effects in Alarm Systems", SPIE International Symposium on Optical Applied Science Engineering, San Diego, July, 1992.
4. K.W.J. Findlay, D Renshaw and P. B. Denyer, "An Intelligent Alarm System", IEE International Conference on Intelligent Systems, August, 1992.
5. K.W.J. Findlay, D Renshaw and P. B. Denyer, "A Low Cost Stereo Alarm System for VLSI", IEEE International Conference on Image Processing, Singapore, September, 1992.
6. K.W.J. Findlay, D Renshaw and P. B. Denyer, "A Stereo Algorithm to Reduce Quantisation Noise", IEEE International Conference on Au-

tomation, Robotics and Computer Vision, Vol. 2, Singapore, September, 1992.

CORSIM: A Two Dimensional Simulator for Contacts with Arbitrary Geometries

K.W.J. Findlay, W.J.C. Alexander and A.J. Walton

Edinburgh Microfabrication Facility
Department of Electrical Engineering
Edinburgh University
King's Buildings
Edinburgh
EH9 3JL.

ABSTRACT

A program has been written which can simulate the contact resistance of arbitrarily shaped contacts between different resistivity materials. The simulator uses triangular finite elements and overcomes the restrictions of previous software. The effects of geometry, contact window misalignment, sheet resistivity and specific contact resistance have been examined for circular contacts and compared to previous results for square ones. Finally the accuracy of the simulator under various extremes was investigated.

1. INTRODUCTION

As geometries used in integrated circuits have reduced the ohmic contact resistance has become more important and starts to limit circuit performance. This paper describes the development of a program that can simulate contacts that become circular due to fringing effects which occur during optical lithography. This is effect is illustrated in figure 1 as contact geometries are reduced.

The 2-D contact simulator FECORS [1] can only model square contacts and has been used to examine the limitations of the Kelvin structure shown in figure 2. It uses two mesh planes connected by a set of vertical resistors which model the contact resistivity. The square element employed is ideally suited to modelling rectangular contact windows and collars. The simulator CORSIM is based on a similar concept except that it uses triangular elements for the two conductor levels. This gives it the ability to simulate contacts with curved boundaries.

2. CORSIM: A GENERALISED CONTACT RESISTANCE SIMULATOR

2.1 Generation

The mesh generator used was adapted from a program called GRID [12, 13] which defines the region to be simulated using eight noded super elements. The contact to be simulated is divided up into several super elements as shown in figure 3. The program uses a curvilinear coordinate system which is capable of representing the curved boundaries of the contact region. GRID proceeds through each region in turn and generates the individual triangular elements. An example of a low density mesh generated from the super elements in figure 3 is illustrated in figure 4.

2.2 The Calculation of the Interface Resistance

Once the mesh for the two conducting layers has been generated the next step is to calculate the values of the resistors used to model the interfacial contact resistance. The interface resistance (R_c) can be calculated using

$$\rho_c = R_c \, A \tag{1}$$

where ρ_c is the specific contact resistivity in Ω/cm^2 and A is the area in cm^2 [9].

When the element mesh within the contact region is a regular series of squares the calculation of the interconnect resistors for each node is simple since there is only a set number of conditions. However, with triangular elements there are an infinite number of possible variations. This requires a more complicated calculation which takes into consideration the area adjacent to each node. Every node can be considered to relate to a surrounding area bounded by the perpendicular bisectors of the midpoints of each element side as illustrated in figure 5(a). Figure 5(b) shows an example of the area associated with a node and once this area has been calculated, the contact resistor associated with that node can be evaluated. The only restriction is that obtuse triangles are not permitted within the contact region because, in this case, the point of intersection between the bisectors lies outside the element. The value for each resistor is given by

$$R_{cnode} = \frac{\rho_c}{A_{node}} \tag{2}$$

where A_{node} is the area associated with an individual node. This approach gives good agreement when compared with FECORS.

2.2 Stiffness Matrix Solution

After the calculation of the interfacial contact resistors CORSIM generates the stiffness matrix and the solution is performed using the frontal method [14]. There are several issues which arise relating to the accuracy of the final output resistances.

2.3 Accuracy

It was noted during the initial testing that the convergence of simulations with metal (0.05 Ω/□) to diffusion (27.0 Ω/□) contacts resulted in a degree of instability as the number of elements were increased. This did not occur with the polysilicon (30.0 Ω/□) diffusion contacts which indicates that resistivity differences between the contact resistors and sheet resistance was causing rounding errors. The use of the double precision variables overcame this problem provided a suitable mesh was chosen.

With metal - diffusion contacts there is a negligible voltage drop in the metal which is shown in the field plot of figure 6. In contrast the voltage drop in the diffusion layer can be observed along with the two dimensional current flow. This two dimensional current flow leads to inaccuracies in the extracted value of contact resistance and is one of the problems associated with the Kelvin structure. Another one can be seen in figure 7 which compares the extracted contact resistivity with the true value of interface resistivity. This shows how, at high values of ρ_c, the specific contact resistance can be extracted. However, at lower values of ρ_c the sheet resistivity of the conducting layers dominates the measurement and the extracted resistivity ρ_{ce} becomes independent of the specific contact resistance. This agrees with results obtained using FECORS. One of the advantages of using triangular elements is that the density of elements may be varied in an appropriate manner for the voltage gradients which are present in the structure. This is a very useful feature especially when current flow in 90° contacts is being considered.

3 COMPARISONS OF CIRCULAR AND SQUARE CONTACTS

3.1 The Effects of Shape and Size on Lateral Current Crowding

The Kelvin structure is widely used for the measurement of contact resistance but the value extracted assumes uniform current flow. There have been a number of papers dealing with the inaccuracies introduced by the fringing fields [1,2,4,5]. Correction factors for 2-D effects have been proposed but these have all assumed square contacts. It is also worth considering what proportion of the correction factor is due to the change in current flow as opposed to contact area reduction as the contacts move from a square to circular geometry.

Figure 8 shows a comparison between square [1,5] and circular contact windows for various mesh densities. This shows that the measured value of R_c for a circular contact is higher than that for the equivalent square when the contact's diameter is the same as the square's dimensions. However, the extracted values of ρ_c which takes into account the difference in area are much closer in value as shown in Table 1. As the collar size is reduced, contact geometry has a larger influence on current flow. Table 2 shows the extracted values of ρ_c for different collar sizes when both layers have a sheet resistance of 30 Ω/□. It can be observed that the relative values of ρ_c for square and circular contacts change as the collar size varies. The extracted contact resistivity for circular contacts is the

Contact Type	○ or □	R_c (Ω)	Area (μm^2)	ρ_c (Ωcm^{-1})
Metal to Diffusion	○	8.84	19.6	1.74×10^{-6}
	□	7.06	25.0	1.77×10^{-6}
Poly to Diffusion	○	12.13	19.6	2.38×10^{-6}
	□	9.66	25.0	2.42×10^{-6}

Table 1. Comparison of square and circular contact parameters extracted from a Kelvin structure with 5μm contacts and a 5μm collar. The specific contact resistivity is 1×10^{-6} Ωcm^{-1}.

Collar Size	○ or □	R_c (Ω)	Area (μm^2)	ρ_c (Ωcm^{-1})
5μm	○	24.7	19.6	1.70×10^{-6}
	□	29.3	25.0	1.83×10^{-6}
1μm	○	17.8	19.6	1.25×10^{-6}
	□	16.5	25.0	1.19×10^{-6}
0.25μm	○	17.1	19.6	1.17×10^{-6}
	□	9.66	25.0	1.07×10^{-6}

Table 2. Comparison of square and circular contact parameters extracted from a Kelvin structure with a 3μm contacts and a variable size collars. The specific contact resistivity is 1×10^{-6} Ωcm^{-1}.

smaller of the two for large collar sizes but the situation is reversed as the contact collar reduces below 1 μm. This is because the circular contact still distorts the current flow even when the collar width is zero whereas for the square contact the current flow would be totally one dimensional. current flow. as the collar size increases.

For circular and square contacts with the same area and a collar size of 5μm (see figure 9) the simulated values of R_c were 10.08 Ω and 9.88 Ω for the circle and square respectively. For a circular contact with the diameter the same size as the dimensions of the square contact R_c was 12.1 Ω. It can be concluded that for structures with large collars the shape of the contact has little effect on the measured values. The important parameter is area.

The second comparison examines upon how variations between collar and window size influence the measured value of R_c. This is illustrated in figure 10 and in all cases the resistance increases with both collar and window size. As the window size reduces the difference between the two types of contact (polysilicon - diffusion and metal - diffusion) increases. The value of R_c for polysilicon - diffusion is always greater due to the voltage drop, which in this case, occurs on both layers. As expected the circular contacts always have a higher value of R_c associated with them because of their smaller contact areas.

3.2 Misalignment Comparison

The comparison of misaligned polysilicon - diffusion contacts shown in figure 11 indicates that the same trends apply to both square and circular contacts for these dimensions. The only difference is that the values of R_{co} will be greater due to the size of the contact window and larger collar area.

3.3 The Effects of Sheet Resistance Variation Directly Under the Contact

It has been stated [10] that the sheet resistance directly under the contact window may vary from that in the surrounding area and reference [1] simulated the effect of changing the value of R_c under the contact. The simulations performed by CORSIM for circular windows give similar results with higher values of R_c than those for square contacts as shown in figure 12. As the gradient is linear and the same for both the square and circular contacts the error in any Kelvin measurements will be the same for both shapes.

4. CONCLUSIONS

A finite element program that can model arbitrarily shaped contacts has been developed. It can be used to evaluate the effect that changes in geometry, specific contact resistivity, sheet resistivity and the modification of sheet resistivity under the contact have upon contact systems. The Kelvin test structure has been used to illustrate some of its capabilities. It is intended to use this software to develop correction curves for Kelvin structures that do not have rectangular contacts.

5. REFERENCES

[1] ALEXANDER W.J.C., WALTON A.J., "Sources of Error in Extracting the Specific Contact Resistance from Kelvin Device Measurements", IEEE Proc. on Microelectronic Test Structures, Vol.1, No.1, Feb.1988.

[2] LOHM W.M., SWIRHUN S.E., SCHREYER T.A., SWANSON R.M., SARASWAT K.C., "Modeling and Measurement of Contact Resistances", IEEE Transactions on Electronic Devices, Vol.ED-34, No.3, March 1987.

[3] WALTON A.J., HOLWILL R.J., ROBERTSON J.M., "Numerical Simulation of Resistive Interconnects for Integrated Circuits", IEEE Journal of Solid State Circuits, Vol.SC-20, No.6, December 1985.

[4] GILLENWATER R.L., HAFICH M.J., ROBINSON G.Y, "The Effect of Lateral Current Crowding on the Specific Contact Resistivity in D-Resistor Kelvin Devices", IEEE Transactions on Electron Devices, Vol.ED-34, No3, March 1987.

[5] SCORZONI A., FINETTI M., GRAHN K., SUNI I., CAPPELLETTI P., "Current Crowding and Misalignment Effects as Sources of Error in Contact Resistivity Measurements Part 1: Computer Simulation of Conventional CER and CKR Structures", IEEE Transactions on Electron Devices, Vol. ED-34, No.3 March 1987.

[6] WALTON A.J., HOLWILL R.J., ROBERTSON J.M., "Contact Resistance of Silicon-Polysilicon Interconnection for Different Current Flow Geometries", Electronic Letters 1985, vol.21

[7] PROCTOR S.J., LINHOLM L.W., "A Direct Measurement of Interfacial Contact Resistance", IEEE Electron Device Letters Vol. ED3 no.10, 1982

[8] PROCTOR S.J., LINDHOLM L.W., MAZER J.A., "Direct Measurement of Interfacial Contact Resistance, Contact End Resistance and Interfacial Layer Uniformity", IEEE Trans. Electron Devices, Vol. ED-30, No.11, November, 1983, PP1535-1542.

[9] BERGER H.H., "Contact Resistance and Contact Resistivity", J.Electrochem Soc., Solid State Science and Technology.

[10] BERGER H.H., "Models for Contacts to Planar Devices", Solid State Electronics, Vol.15, 1972 (pp145-158).

[11] REEVES G.K., HARRISON H.B., "Determination of Contact Parameters of Interconnecting Layers in VLSI Circuits", IEEE Solid State Circuits, Special Issue, May 1985.

[12] STEINMUELLER G., "Restrictions on the Application of Automatic Mesh Generation Schemes By Isoparametric Co-ordinates": International Journal for Numerical Methods in Engineering, Vol.8, pp 289-294.

[13] SEGERLIND L.J., "Applied Finite Element Analysis", Published by John Wiley and Sons, Inc., 1976.

[14] HINTON E., OWEN D.R.J., "Finite Element Programming", Academic Press, 1977.

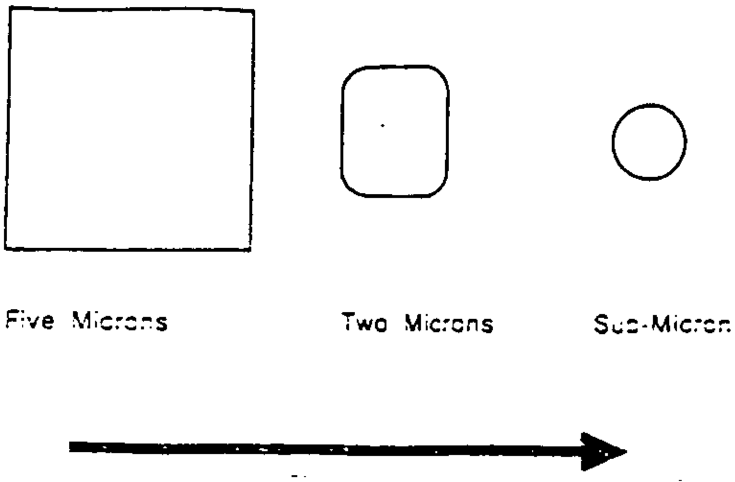


Figure 1. The fringing effect which occurs during optical lithography. The arrow indicates decreasing size.

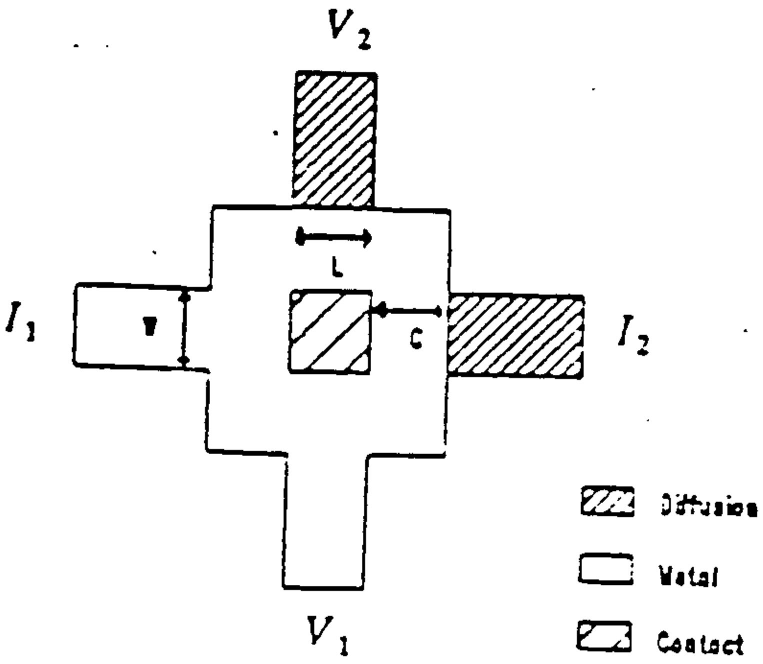


Figure 2. D-Resistor Kelvin device. Current is forced from I_1 to I_2 and the Kelvin potential is measured at V_2 with respect to V_1.

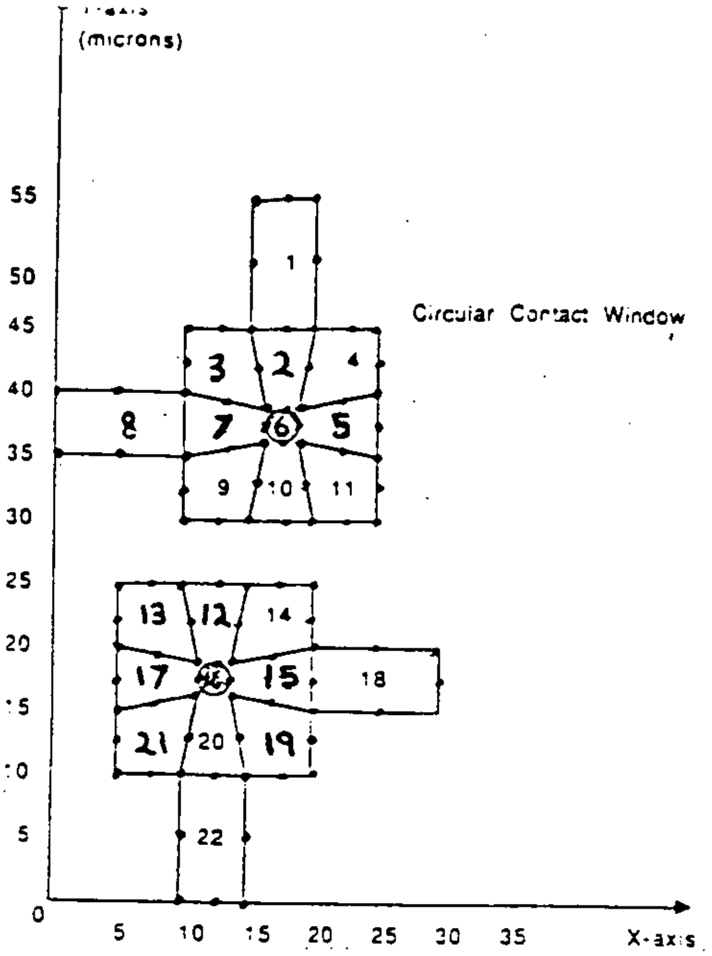


Figure 3. The division of the Kelvin device into super elements before mesh generation (L=3μm and C=5μm).

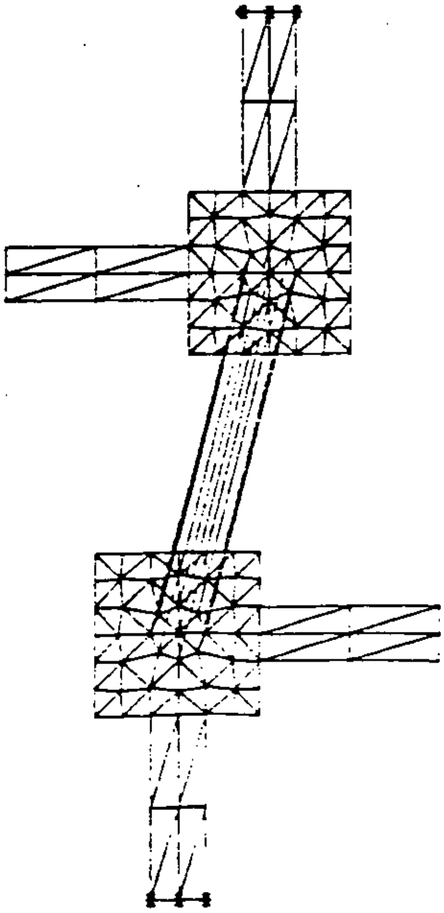


Figure 4. A low density mesh generated using GRID. The 'star' nodes imply prescribed voltages.

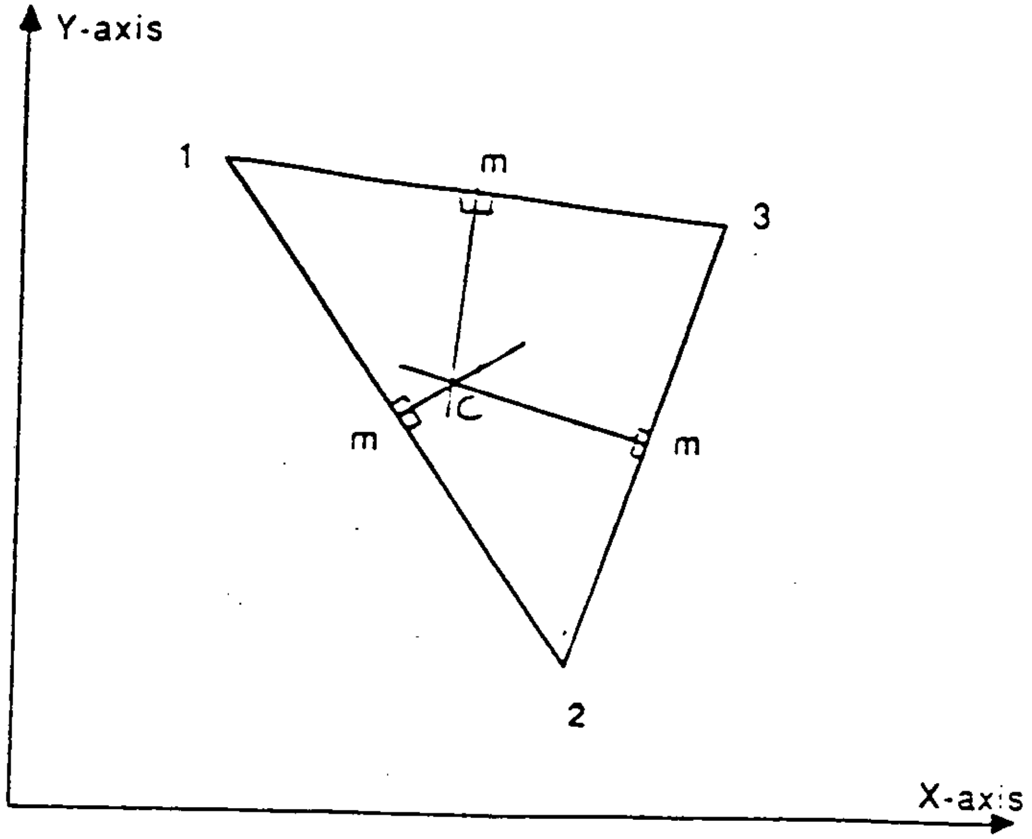


Figure 5(a). The subdivision of each element according to the intersection of the perpendicular bisectors of each side.

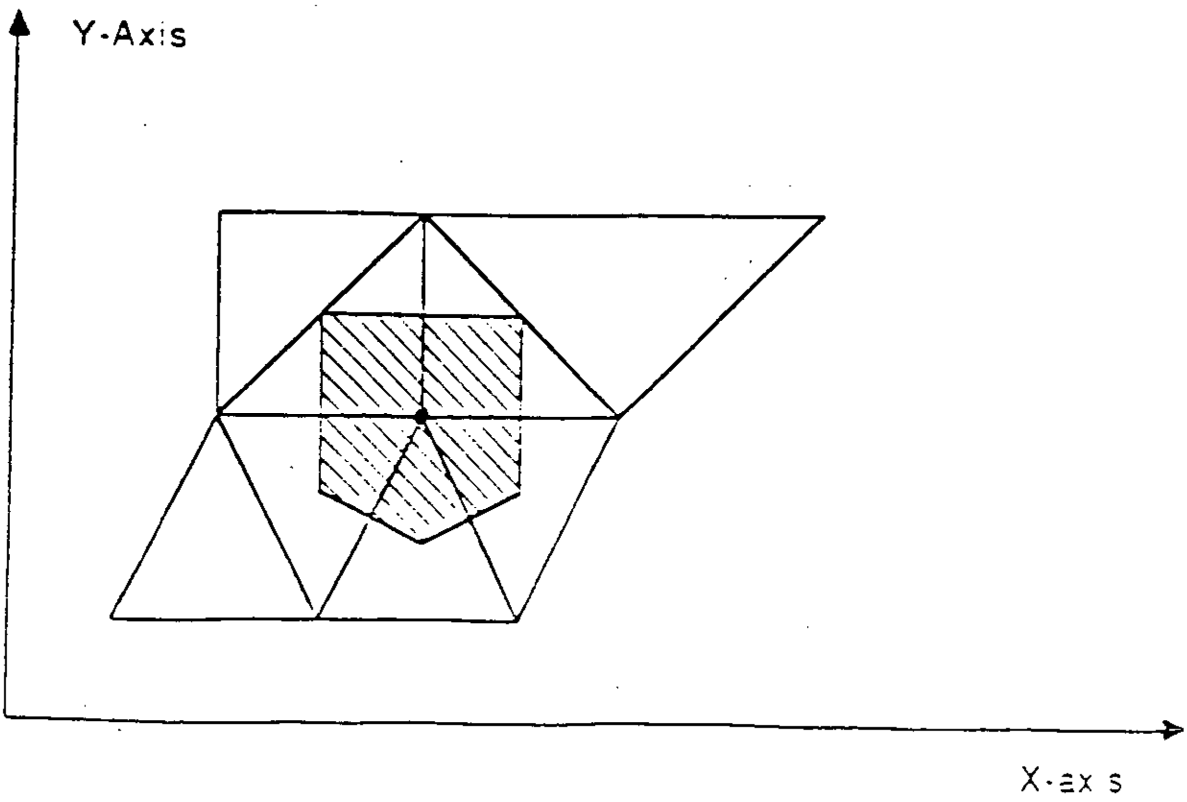


Figure 5(b). The area associated with each contact node and used in the calculation of the contact resistors.

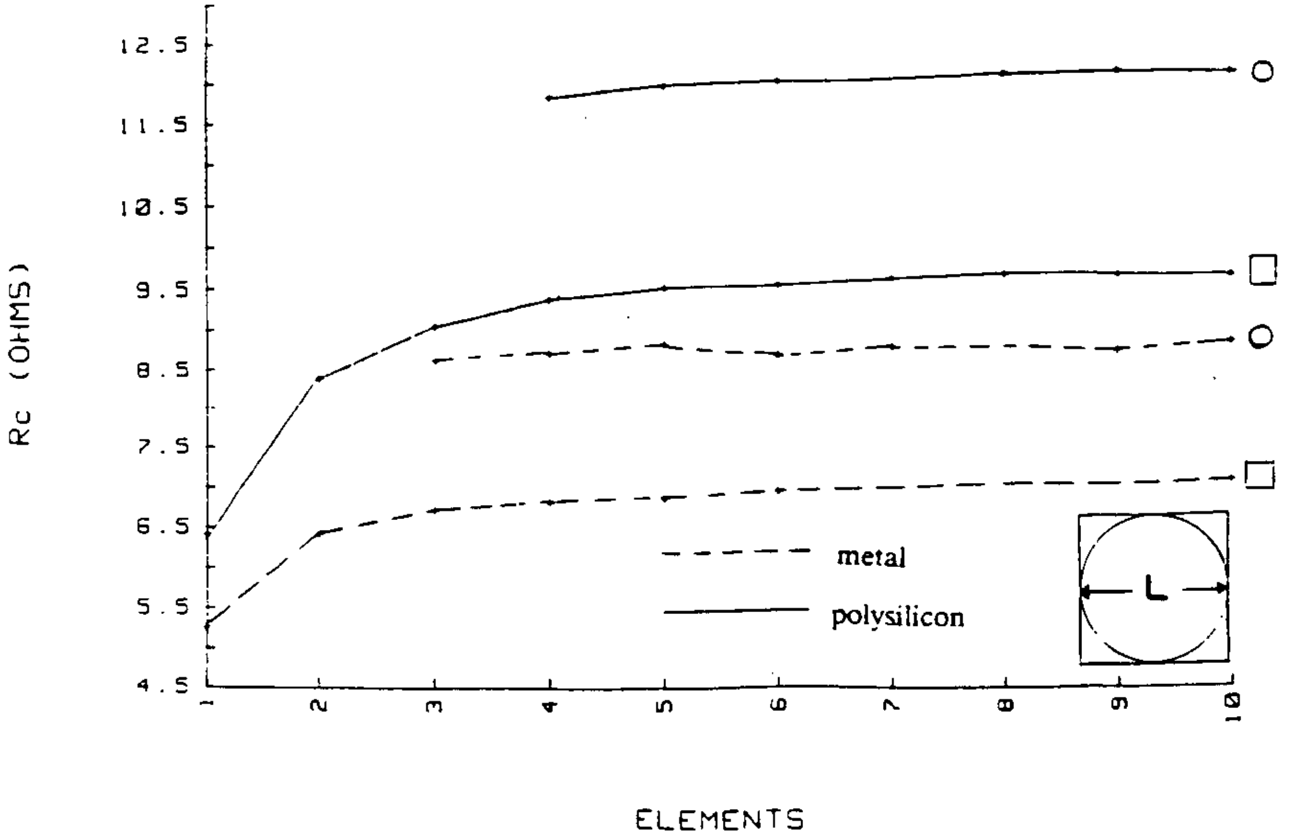


Figure 8. The accuracy of the simulation solution for L = 5μm and C = 5μm, as a function of the number of finite elements modelling the contact region, $\rho_c = 1 \times 10^{-6}$. Both the square and circular contacts had the same diameters.

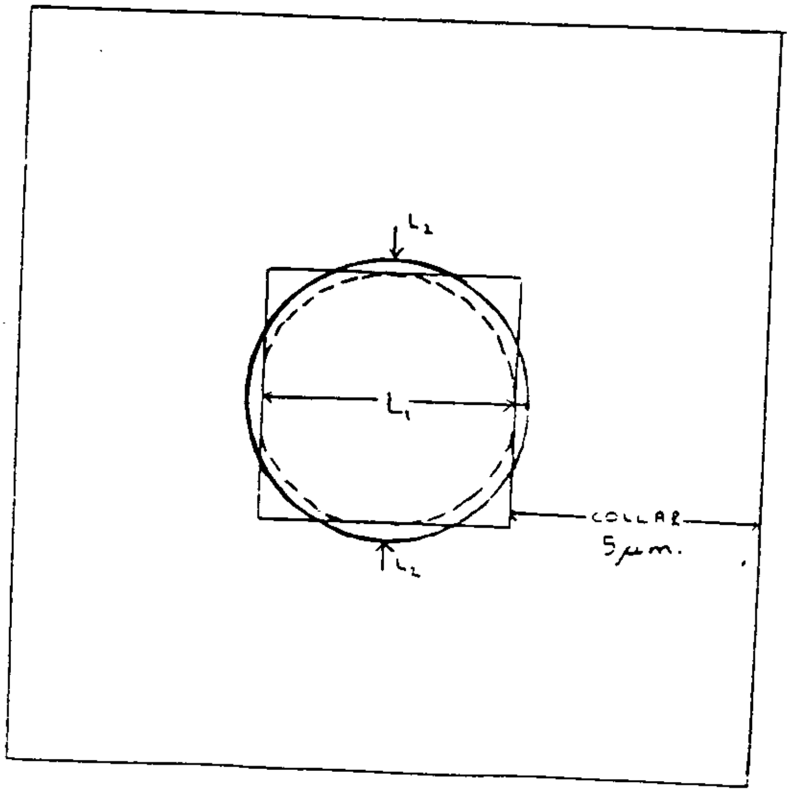


Figure 9. The three contact areas. L_2 is the diameter of a circle with the same area as the square and L_1 is the diameter of the circular contact with the same dimension as the square.

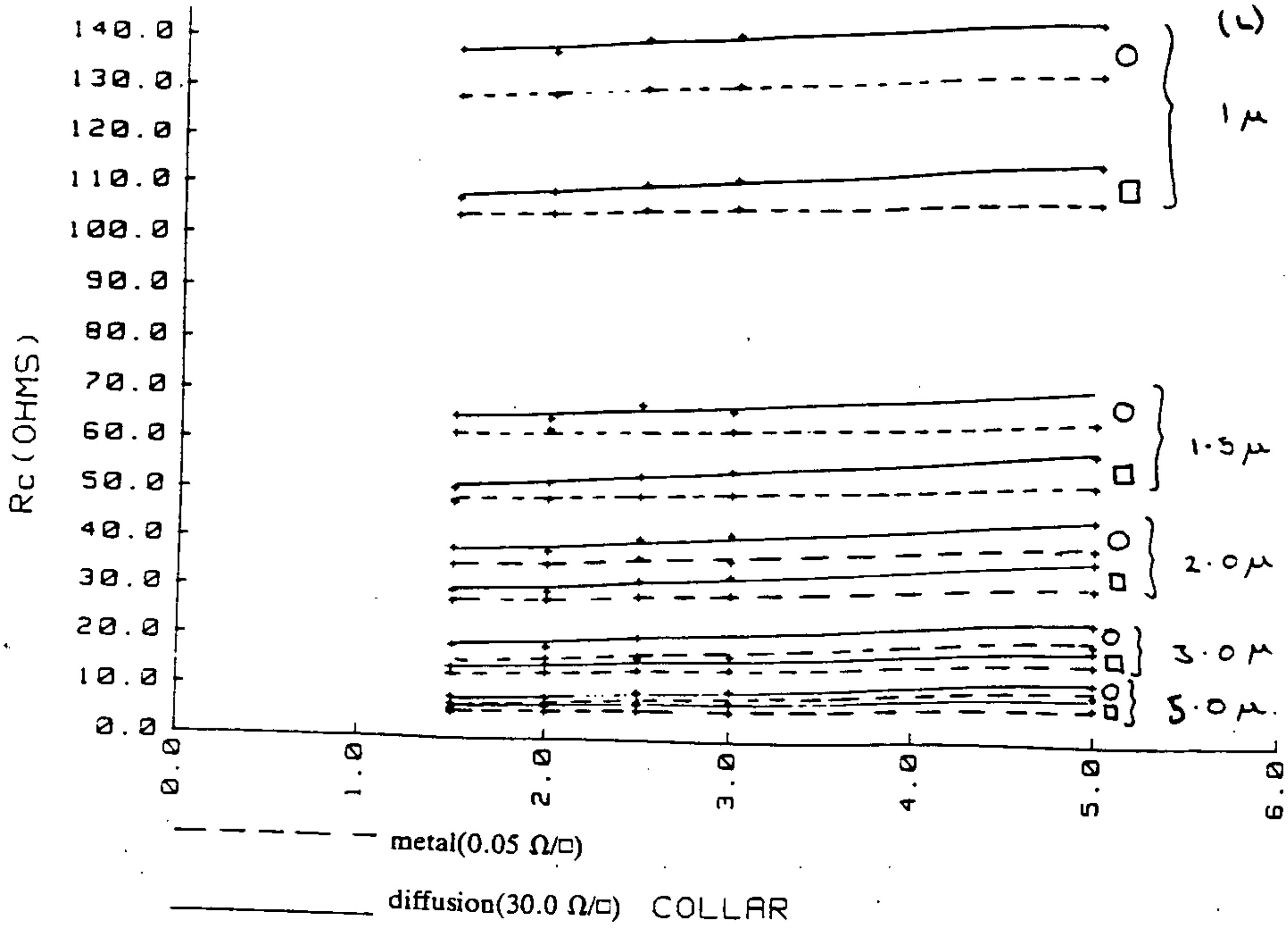
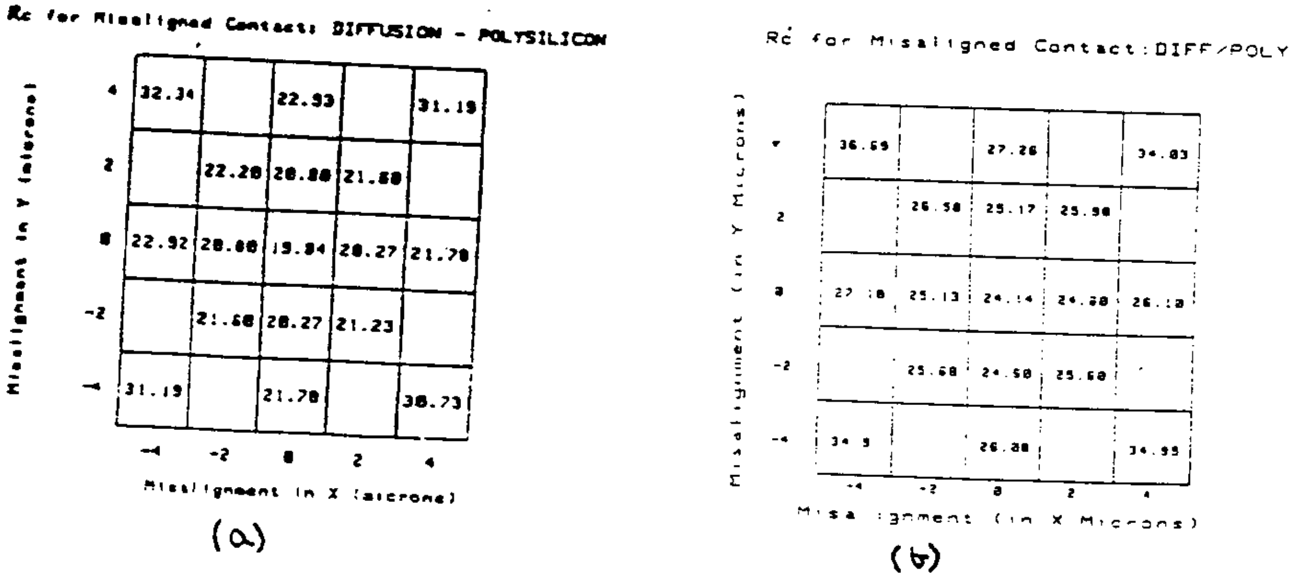


Figure 10. Extracted contact resistance, R_c for a range of collar and window sizes. Both the circular and square contacts are shown, ($\rho_c = 1 \times 10^{-6}$)

$\rho_c = 1 \chi 10^{-6}$.


Rc for Misaligned Contact: DIFFUSION - POLYSILICON
Misalignment in Y (microns)
Misalignment in X (microns)
(a)
Rc for Misaligned Contact:DIFF/POLY
Misalignment (in Y Microns)
Misalignment (in X Microns)
(b)

(a)

	-4	-2	0	2	4
4	32.34		22.93		31.19
2		22.20	20.00	21.60	
0	22.92	20.00	19.84	20.27	21.70
-2		21.60	20.27	21.23	
-4	31.19		21.70		30.73

(b)

	-4	-2	0	2	4
4	36.69		27.26		34.03
2		26.50	25.17	25.90	
0	27.10	25.13	24.14	24.00	26.10
-2		25.60	24.50	25.60	
-4	34.9		26.00		34.95

Figure 11. Extracted resistances, R_c, for various misaligned contacts where L=3μm and C=5μm; (a) is for a square contact and (b) for a circular contact of the same width. The results are for polysilicon (30.0Ω/■) to diffusion (27.0Ω/■) with $\rho_c = 1 \times 10^{-6}$.

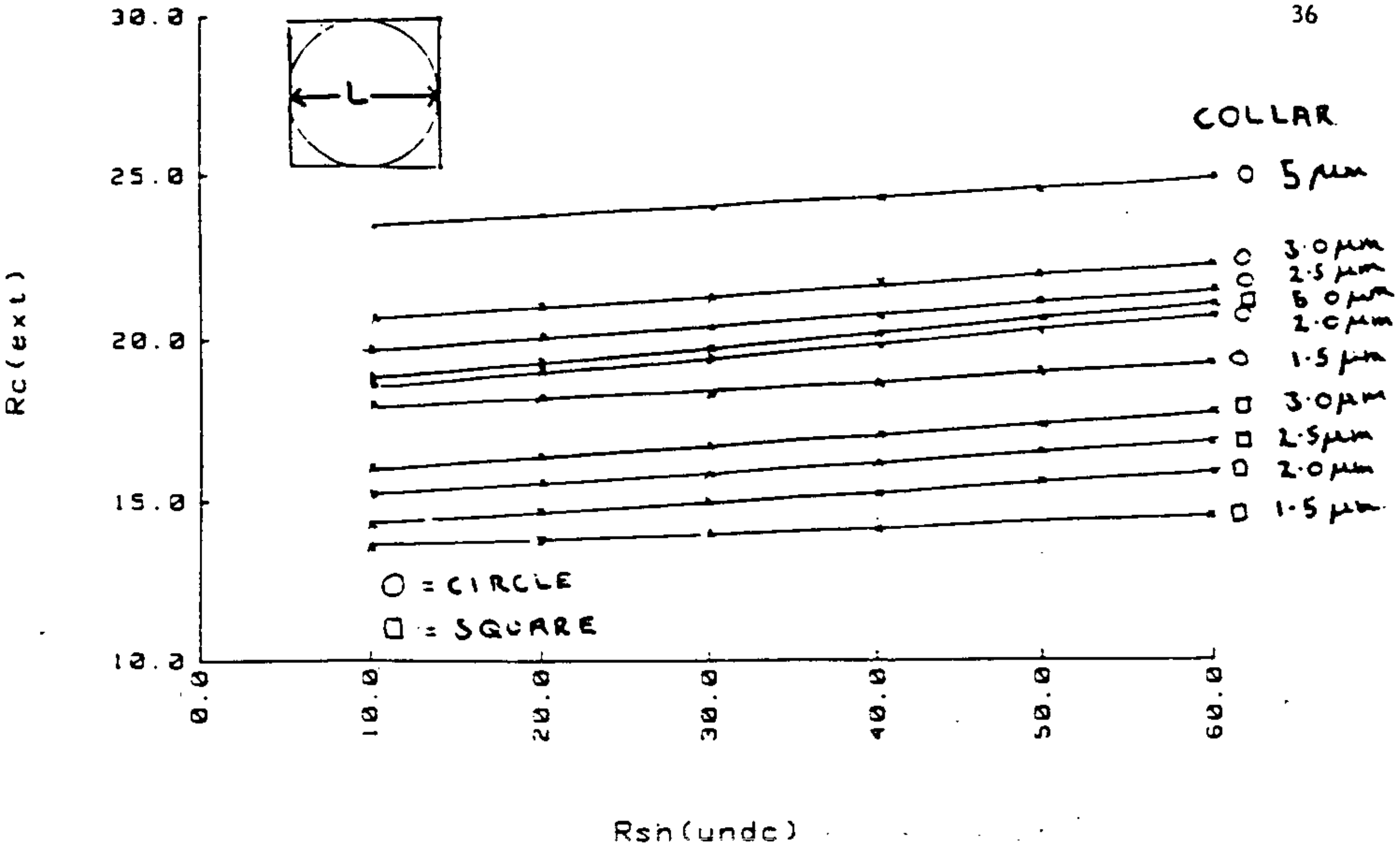


Figure 12. Extracted contact resistance for a $3\mu m$ polysilicon to diffusion contact over a range of collar sizes as a function of the modified sheet resistance under the contact. Both square and circular contacts are shown and the specific contact resistivity is $\rho_c = 1 \times 10^{-6}$.

THE EFFECT OF CONTACT GEOMETRY ON THE VALUE OF CONTACT RESISTIVITY EXTRACTED FROM KELVIN STRUCTURES

K.W.J. Findlay, W.J.C. Alexander and A.J. Walton

Edinburgh Microfabrication Facility
Department of Electrical Engineering
King's Buildings
University of Edinburgh,
Edinburgh, EH9 3JL, UK.

Abstract: The effects of geometry, contact misalignment and sheet resistivity on the extracted values of specific contact resistance have been simulated for both circular and square contacts. These results have been used to detail the errors involved in extracting contact resistivity from Kelvin structures with rectangular and circular contacts.

1. INTRODUCTION

As the geometries used in integrated circuits reduce, the ohmic contact resistance becomes more important and starts to become a limiting factor in circuit performance. Contacts with these reduced dimensions also become more circular due to the fringing which occurs during optical lithography. This trend is illustrated in figure 1 and any test structure which is used to measure contact resistance for a small geometry process will obviously have rounded contacts. It is consequently important that the effect of contact shape on the extracted value of contact resistance is quantified. This paper describes the development and application of a program that can simulate contacts with non-rectangular shapes.

2. CORSIM: A CONTACT RESISTANCE SIMULATOR

2.1 Introduction

The 2-D contact simulator FECORS [1] has been previously used to to examine the limitations of the Kelvin structure shown in figure 2. It uses two resistor mesh planes which are connected by a set of resistors which model the contact resistivity. The square element employed is ideally suited to modelling structures with rectangular contacts and collars as are the other contact simulators reported elsewhere [2-7]. The simulator CORSIM which is detailed in this work, is based on a similar concept except that it uses triangular elements for the two conductor levels. This gives it the ability to simulate contacts with curved boundaries and the element size can easily be graded in regions of high current density.

2.1 Element Generation

To reduce the amount of data input required by CORSIM a mesh generator has been implemented. This was adapted from a program called GRID [8,9] which uses eight noded super elements to define the region to be simulated. The contact system to be modelled is divided up into several super elements as shown in figure 3 and their x,y coordinates provide the input data. The program uses a curvilinear coordinate system which is capable of representing the curved boundaries of the contact region. GRID proceeds through each region in turn and generates the individual triangular elements. An example of a low density mesh generated from the super elements of figure 3 is illustrated in figure 4.

One of the limitations of FECORS, with its square grid, is that element density can not be varied over the structure. An advantage of using triangular elements is that they may be graded in a manner appropriate to the voltage gradients which are present. This is a very useful feature especially when current flow for 90° contacts is being considered.

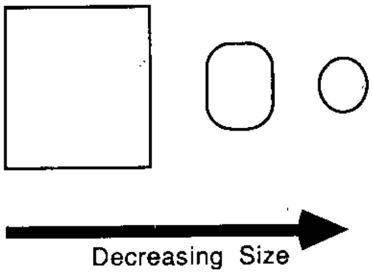


Figure 1. The fringing effect which occurs during optical lithography.

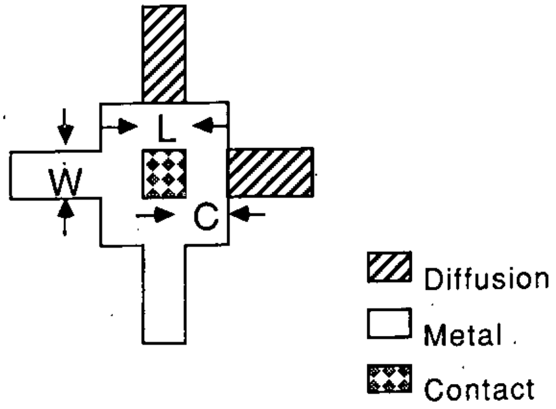


Figure 2. D-Resistor Kelvin device. The four arms have width, W, and a collar, C, which surrounds the square or circular contact, L. Current is forced from I_1 to I_2 and the potential is measured between V_2 and V_1.

CH2693-0/89/0000-133$01.00©1989 IEEE

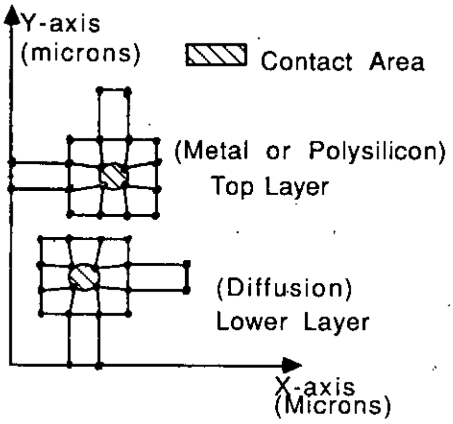


Figure 3. The division of the Kelvin structure into super elements before mesh generation.

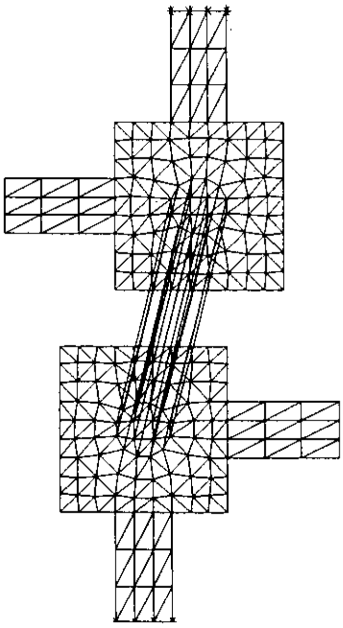


Figure 4. A low density mesh generated using using the super elements illustrated in figure 3.

2.2 The Calculation of the Interface Resistance

Once the mesh for the two conducting layers has been generated the next step is to calculate the values of the resistors used to model the interfacial contact resistance. The interface resistance (R_c) can be calculated using

$$\rho_c = R_c\ A \qquad (1)$$

where ρ_c is the specific contact resistivity in Ωcm^2 and A is the area in cm^2 [10].

When the element mesh within the contact region is a regular series of squares the calculation of the interconnect resistors for each node is simple since there are only a set number of conditions [1]. However, with triangular elements there are an infinite number of possible variations. This requires a more complicated calculation which takes into consideration the area adjacent to each node. Every node can be considered to relate to a surrounding area bounded by the perpendicular bisectors of the midpoints of each element side as illustrated in figure 5(a). Figure 5(b) shows an example of the area associated with a node and having calculated this area, the contact resistor associated with that node can be evaluated. The only restriction is that obtuse triangles are not permitted within the contact region because, in this case, the point of intersection between the bisectors lies outside the element. The value for each resistor is simply given by

$$R_{cnode} = \frac{\rho_c}{A_{node}} \qquad (2)$$

where A_{node} is the area associated with an individual node. This approach gives good agreement when compared with results generated using FECORS.

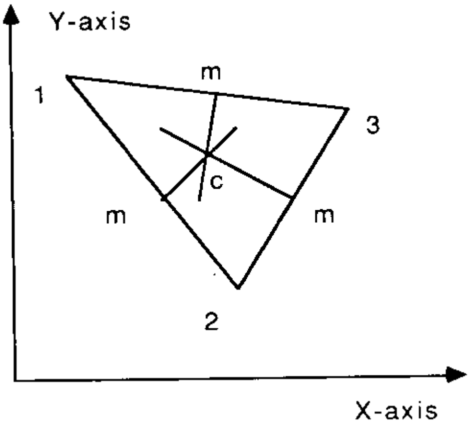


Figure 5(a). The subdivision of each element according to the intersection of the perpendicular bisectors of each side.

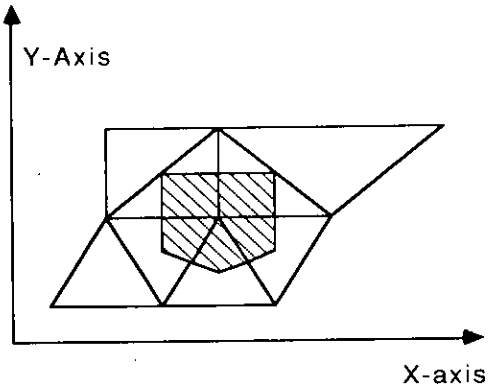


Figure 5(b). The area associated with each contact node and used in the calculation of the contact resistors.

2.3 Solution

After the calculation of the interfacial contact resistors CORSIM generates the admittance matrix and the solution then performed using the frontal method [11]. This gives the node voltages and currents which can then be used to calculate the contact resistance. The solution can also be displayed as a contour plot of the equipotentials to provide further information. Figure 6 and 7 show an example of these types of plots for Kelvin structures with circular and square contacts.

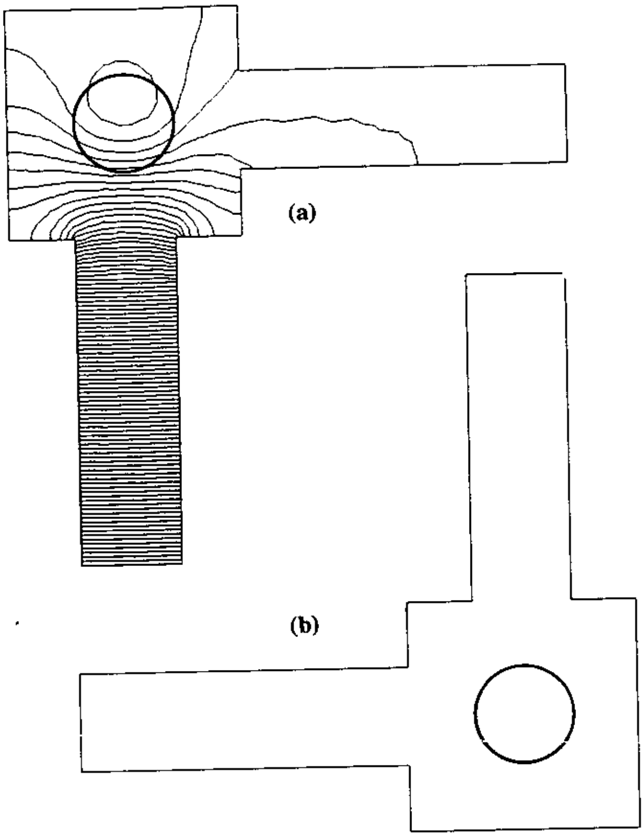


Figure 6. Voltage contour plots for a metal (0.05 Ω/■) to diffusion (30.0 Ω/■) 3μm circular contact with a 2μm collar (ρ_c=10^{-6}Ωcm^2). (a) diffusion. (b) metal.

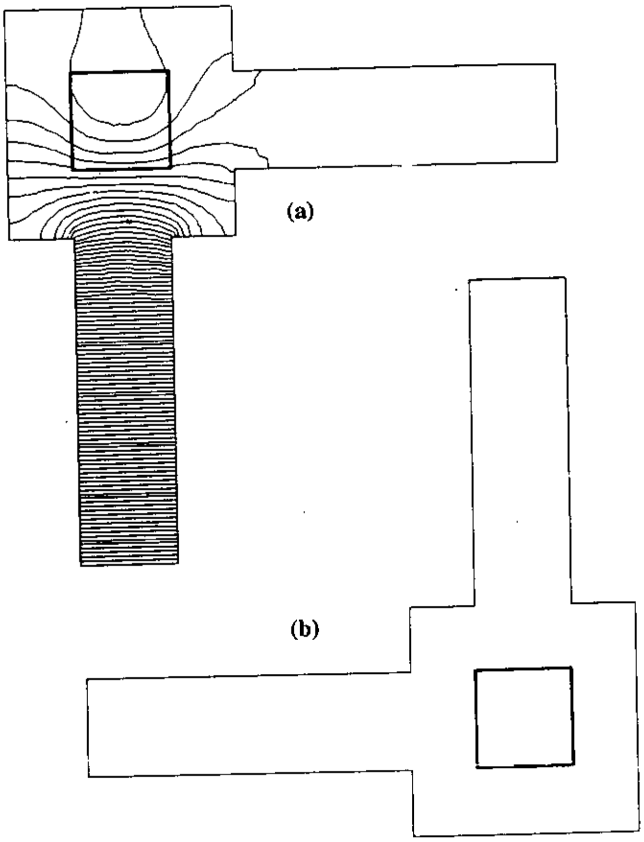


Figure 7. Voltage contour plots for a metal (0.05 Ω/■) to diffusion (30.0 Ω/■) 3μm square contact with a 2μm collar (ρ_c=10^{-6}Ωcm^2). (a) diffusion. (b) metal.

3. COMPARISONS OF CONTACT GEOMETRY

3.1 Introduction

The Kelvin structure [12-13] is widely used for the measurement of contact resistance but the value extracted assumes uniform current flow. The parasitic resistance drops obviously reduce the accuracy of the device and in certain circumstances can totally mask the measured value [1]. It is therefore important that these limitations are understood in order that the structure can be employed to its full potential.

3.2 The Effects of Shape and Size

There have been a number of papers dealing with the inaccuracies introduced by the fringing fields [1-6]. Correction factors for 2-D effects have been proposed but these have all assumed square contacts. When considering the relationship between square and circular contacts there are two options which can be used for comparisons. These are when the diameter of the circle is the same as the dimension of the square and the case when the area of both contacts are equal. These conditions are illustrated in figure 8. In all the following comparisons, the specific contact resistivity in the simulations has been fixed at 10^{-6} Ωcm^2 with the extracted value being calculated from the voltage and current evaluated by CORSIM.

Obviously the measured value of R_c for a circular contact will be higher than that for the equivalent square when the contact's diameter is the same as the square's dimensions. Howẹver the extracted values of ρ_c, which takes into account the difference in area, will be much closer. Table 1 gives a comparison of these results for a 5μm contact with a 5μm collar. Table 2 gives the extracted values of ρ_c for circular and square contacts with the same dimensions while table

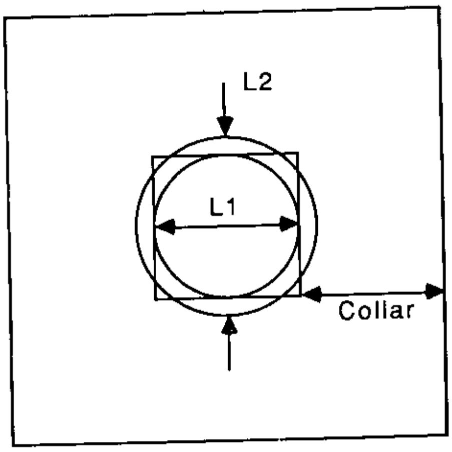


Figure 8. The three contact areas used for comparisons. L_2 is the diameter of a circle with the same area as the square and L_1 is the diameter of the circular contact with the same dimension as the square.

Contact Type	○ or □	R_c (Ω)	Area (μm^2)	ρ_c (Ωcm^2)
Metal to Diffusion	○	8.84	19.6	1.74×10^{-6}
	□	7.06	25.0	1.77×10^{-6}
Polysilicon to Diffusion	○	12.13	19.6	2.38×10^{-6}
	□	9.66	25.0	2.42×10^{-6}

Table 1. Comparison of square and circular contact parameters extracted from a Kelvin structure with 5μm contacts and a 5μm collar. The specific contact resistivity for the simulation was 10^{-6} Ωcm^2.

Dimension of square contact	○ or □	Collar Size			
		5μm	3μm	1μm	0.25μm
5μm	○	2.29	1.88	1.48	1.4
	□	2.4	1.87	1.32	1.18
3μm	○	1.69	1.51	1.25	1.16
	□	1.78	1.5	1.2	1.09
1μm	○	1.12	1.10	1.02	1.02
	□	1.16	1.13	1.06	1.02
0.25μm	○	1.01	1.01	1.01	1.00
	□	1.01	1.01	1.01	1.01

Table 2. Comparison of square and circular contact resistivity (×10^6) extracted from a poly-diffusion Kelvin structure with variable contact and collar sizes. The diameter of the circular contact is the same as the dimension of the square one. The specific contact resistivity for the simulations was 10^{-6} Ωcm^2, $R_{s(poly)}$ = 27Ω/■ and $R_{s(diff)}$ = 30Ω/■.

Dimensions of square contact	○ or □	Collar Size			
		5μm	3μm	1μm	0.25μm
5μm	○	2.46	1.93	1.37	-
	□	2.4	1.87	1.32	1.18
3μm	○	1.80	1.55	1.06	-
	□	1.78	1.5	1.2	1.09
1μm	○	1.16	1.13	1.06	-
	□	1.16	1.13	1.06	1.02
0.25μm	○	1.01	1.01	1.01	-
	□	1.01	1.01	1.01	-

Table 3. Comparison of square and circular contact resistivity (×10^6) extracted from a poly-diffusion Kelvin structure with variable contact and collar sizes. Both contact shapes have equal areas and the specific contact resistivity for the simulations was 10^{-6} Ωcm^2, $R_{s(poly)}$ = 27Ω/■ and $R_{s(diff)}$ = 30Ω/■.

gives the values for circular and square contacts with identical areas. Figure 9 summarises some of the data given in tables 2 and 3. It can be observed that for circular and square contacts with the same dimensions the extracted contact resistivity for circular contacts is smaller when the collar is large. However, for small collar sizes the situation is reversed as the contact collar reduces below 1 μm. This is because the circular contact still distorts the current flow even when the collar 'width' is zero whereas for the square contact the current flow becomes totally one dimensional.

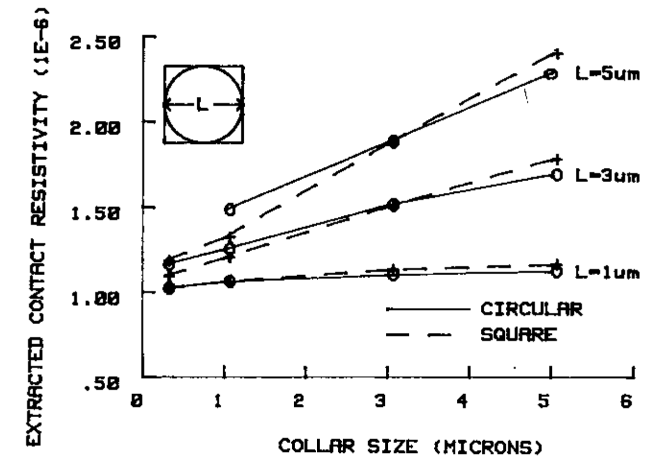


Figure 9. Extracted specific contact resistivity for various collar and window sizes of a polysilicon to diffusion contact. The circular and square contacts both had the same dimensions with different areas and the specific contact resistivity was 10^{-6}Ωcm^2, $R_{sh(poly)}$=27.0Ω/■ and $R_{sh(diff)}$=30.0Ω/■.

From the above results it can be deduced that, for the Kelvin structure, the exact geometry of the contact is not of primary importance when extracting contact resistivity. By far the most important parameter is the area. Kelvin structures with equal area contacts result in very similar values of contact resistivity being extracted.

Figure 10 shows a comparison of the extracted vaiues of R_c as window and collar size vary. In all cases the resistance increases with both collar and window size. As the window size reduces the difference between the two types of contact (polysilicon - diffusion and metal - diffusion) increases. The value of R_c for polysilicon - diffusion is always greater due to the voltage drop, which in this case, occurs on both layers. As expected the circular contacts always have a higher value of R_c associated with them because of their smaller contact areas.

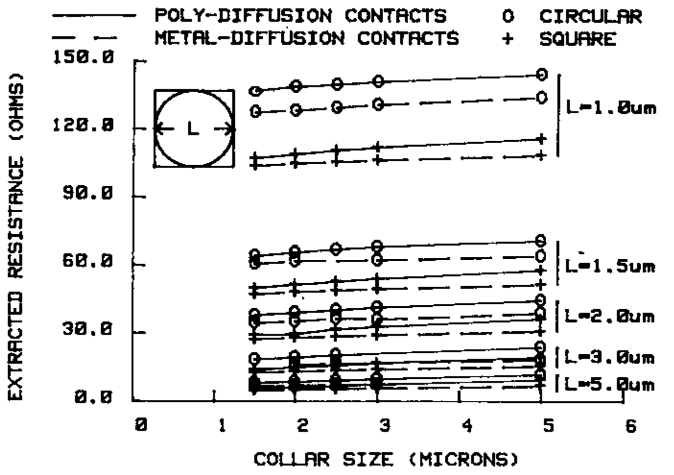


Figure 10. Extracted contact resistance, R_c, for a range of collar and window sizes for both circular and square contacts. The specific contact resistivity was 10^{-6}Ωcm^2, $R_{sh(met)}$=0.05Ω/■, $R_{sh(poly)}$=27.0Ω/ ■ and $R_{sh(diff)}$=30.0Ω/■.

Table 4 shows a comparison of the extracted value of ρ_c for circular and square contacts with the same dimensions. In this case the contacts are metal to diffusion and it can be observed that the effect of the parasitic voltage drops are smaller due to the lower sheet resistance of the metal.

Dimensions of square contact	○ or □	Collar Size 5μm	3μm	1μm	0.25μm
5μm	○	1.73	1.51	1.26	1.21
	□	1.76	1.48	1.17	1.0 7
3μm	○	1.38	1.27	1.13	1.08
	□	1.42	1.28	1.11	1.05
1μm	○	1.07	1.06	1.03	1.01
	□	1.09	1.07	1.03	1.01
0.25μm	○	1.01	1.01	1.00	1.00
	□	1.01	1.01	1.01	1.01

Table 4. Comparison of square and circular contact resistivity ($\times 10^6$) extracted from a metal-diffusion Kelvin structure with variable contact and collar sizes. The diameter of the circular contact is the same as the dimension of the square one. The specific contact resistivity for the simulations was 10^{-6} Ωcm^2, $R_{s(metal)} = 0.05$Ω/■ and $R_{s(diff)} = 30$Ω/■.

3.3 Misalignment Comparison

It is well recognised that misalignment of the contact window within the collar [1,4] is a source of error in a Kelvin measurement. Figure 11 shows the effect of misalignment on the extracted values of resistivity for both circular and square contacts with the same dimensions. The difference in area for the two contact geometries is accounted for by ρ_{ce} and the error in the measurement for both of them can be observed to be very similar. The shape of the contact obviously has a less significant influence on the measurement then the degree of misalignment. It is perhaps of interest to note that for the geometries used in this example the circular contact always results in an extracted contact resistivity closest to the value set in the data that was used as input to CORSIM.

3.4 Sheet Resistance Variation Directly Under the Contact

It has been stated [10] that the sheet resistance directly under the contact window may vary from that in the surrounding area. Reference [1] simulated the effect on the value of R_c when the value of R_s under the contact was varied. The simulations performed by CORSIM for circular windows gives similar results with higher values of R_c than those for square contacts as shown in figure 12. With the gradient being the same for both the square and circular contacts the error in any Kelvin measurements will be the same for both cases.

4. CONCLUSIONS

A finite element program that can model arbitrarily shaped contacts has been developed. It can be used to evaluate the effect that changes in geometry, specific contact

Misalignment in y (μm)	-4.0	-2.0	0.0	2.0	4.0
4.0	2.9		2.06		2.8
2.0		1.99	1.87	1.9	
0.0	2.06	1.87	1.78	1.82	1.96
-2.0		1.9	1.8	1.9	
-4.0	2.8		1.96		2.76

(a)

Misalignment in x (μm)

Misalignment in y (μm)	-4.0	-2.0	0.0	2.0	4.0
4.0	2.59		1.9		2.4
2.0		1.88	1.77	1.8	
0.0	1.92	1.77	1.7	1.74	1.8
-2.0		1.8	1.74	1.8	
-4.0	2.46		1.84		2.74

(b)

Misalignment in x (μm)

Figure 11. Extracted specific contact resistivity for misaligned polysilicon (30.0Ω/■) to diffusion (27.0Ω/■) contacts with $\rho_c=10^{-6}$ and L=3μm and C=5μm. (a) square contact (b) circular contact

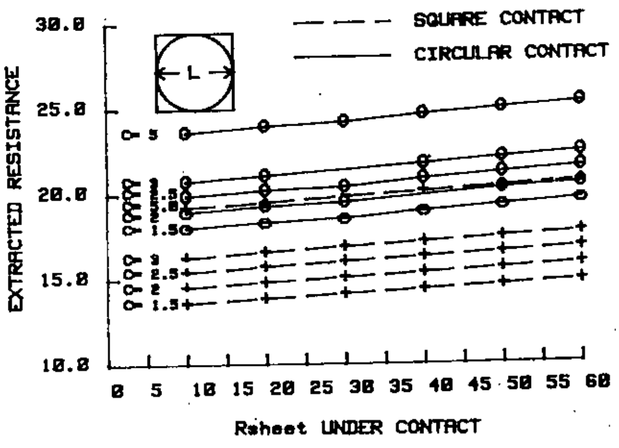


Figure 12. Extracted contact resistance for a 3μm polysilicon to diffusion contact over a range of collar sizes as a function of the modified sheet resistance under the contact. The specific contact resistivity was 10^{-6}, $R_{sh(poly)}=27.0$Ω/■ and $R_{sh(diff)}=30.0$Ω/■.

resistivity, sheet resistivity and the modification of sheet resistivity under the contact have upon contact systems. The Kelvin test structure has been used to illustrate sôme of its capabilities. This work has shown that the contact area of this test structure is of primary importance. The contact shape has little influence on the the extracted value of ρ_c. It is intended to use this software to further examine the effect of contact geometry on a range of different test structures to develop correction curves for non-rectangular contacts.

5. REFERENCES

[1] W.J.C. Alexander, A.J. Walton, "Sources of Error in Extracting the Specific Contact Resistance from Kelvin Device Measurements", IEEE International Conference on Microelectronic Test Structures, Vol.1, no 1, pp 17-22, Feb 1988.

[2] W.M. Lohm, S.E.Swirhun, T.A. Schreyer, R.M. Swanson, K.C. Saraswat, "Modeling and Measurement of Contact Resistances", IEEE Transactions on Electron Devices, Vol. ED-34, no 3, pp 512-523, March 1987.

[3] R.L. Gillenwater, M.J. Hafich, G.Y. Robinson, "The Effect of Lateral Current Crowding on the Specific Contact Resistivity in D-Resistor Kelvin Devices", IEEE Transactions on Electron Devices, Vol. ED-34, no 3, pp 37-543, March 1987.

[4] A. Sconzoni, M. Finetti, K. Grahn, I. Suni P. Cappelletti, "Current Crowding and Misalignment Effects as Sources of Error in Contact Resistivity Measurements Part I: Computer Simulation of Conventional CER and CKR Structures", IEEE Transactions on Electron Devices, Vol. ED-34, no 3, pp 525-531, March 1987.

[5] P. Cappelletti, M.Finetti, A. Sconzoni, I. Suni, N. Circelli, G. Dalla Libera, "Current Crowding and Misalignment Effects as Sources of Error in Contact Resistivity Measurements Part II: Experimental Results and Computer Simulation of Self Aligned Test Structures", IEEE Transactions on Electron Devices, Vol. ED-34, no 3, pp 532-535, March 1987.

[6] T.A. Schreyer, K.C. Saraswat, "A Two-Dimensional Analytical Model of the Cross-Bridge Kelvin Resistor", IEEE Electron Device Letters, vol EDL-7, no 12, pp 661-663, Dec 1986.

[7] A.J. Walton, R.J. Holwill, J.M. Robertson, "Contact Resistance of Silicon-Polysilicon Interconnection for Different Current Flow Geometries", Electronic Letters, vol 21, no 1, pp 13-14, Jan 1985.

[8] G. Steinmueller, "Restrictions on the Application of Automatic Mesh Generation Schemes By Isoparametric Co-ordinates": International Journal for Numerical Methods in Engineering, Vol.8, pp 289-294, 1974.

[9] S.J. Segerlind, "Applied Finite Element Analysis", Published by John Wiley and Sons, 1976.

[10] H.H. Berger, "Contact Resistance and Contact Resistivity", J.Electrochem Soc., vol 119, no 4, pp 507-514, 1972.

[11] E. Hinton, D.R.J. Owen, "Finite Element Programming", Academic Press, 1977.

[12] S.J. Proctor, L.W. Linholm, "A Direct Measurement of Interfacial Contact Resistance", IEEE Electron Device Letters Vol ED-3, no.10, pp 294-296, Oct 1982.

[13] S.J. Proctor, L.W. Linholm, J.A. Mazer, "Direct Measurement of Interfacial Contact Resistance, Contact End Resistance and Interfacial Layer Uniformity", IEEE Trans. Electron Devices, Vol. ED-30, no 11, pp 1535-1542, Nov 1983.

A stereo algorithm to reduce quantisation noise effects in alarm systems

K.W.J. Findlay, D.Renshaw and P.B. Denyer
Department of Electrical Engineering, Edinburgh University,
King's Buildings, Mayfield Road, Edinburgh, U.K.
E-Mail:kevinf@ee.ed.ac.uk, Fax:31 662 4358

ABSTRACT

Over the years a considerable amount of research has been conducted in the area of passive stereo vision. Usually attempts have been made to solve the stereo correspondence problem in its most general sense and build an all purpose stereo module. Possible matches are proposed for all parts or edges of the image.

The above general approach is not always necessary. Indeed there is evidence that the human vision system only attempts to match a small number of possible edges in a particular scene. In this paper we describe a computationally simple algorithm which takes advantage of the nature of the object being tracked. Disparity measurements are made for the entire edge and statistics used to provide subpixel accuracy. This approach reduces the problems caused by quantisation noise when attempts are made to rectify the depth information. We show that stereo algorithms can be used and adapted in an application specific manner to construct viable systems in the areas of alarms and "invisible wall" detection. Results are presented to show the effectiveness of the algorithm in a number of both difficult and simple sequences. In conclusion, we believe our work demonstrates an industrially viable vision system requiring minimal hardware for implementation.

1 INTRODUCTION

Vision algorithms have been developed to solve both general and specific problems. However there are relatively few practical vision systems in use, either in an industrial or consumer environment. Those that have been successful have normally been restricted to recognition tasks on an assembly line or character recognition in places such as post offices. An important reason for this is cost. We aim to design further working vision systems using a minimum of hardware.

In this paper we describe an alarm system which will detect and track a human moving around a scene from stationary cameras. It uses trinocular vision. The system has many applications in situations where "invisible" boundaries are required and could replace or complement systems where light beams and active electronics are currently in use. Two possible applications are automatic door and burglar alarm systems. In the case of doors it is desirable that the position of an approaching object is known. A more sensible decision can then be made as to when the door should be opened.

Stereo vision is a possible solution to the position and size problem and a low cost algorithm has been developed which will utilise constraints particular to this application. It is based on the fact that, provided certain conditions are satisfied, two images of the same object will overlap, [1] if the width of the object is greater than the distance between the cameras. Using this constraint allows stereo matching, without comparing large numbers of features or performing area based correlation.

However a problem with alarm systems is the requirement for wide angle lenses and pixel errors in stereo are inversely proportional to the product of the focal length and the distance between the cameras. Wider angle lenses require shorter focal lengths, increasing the significance of individual pixels. A method using disparity histograms and a basic assumption about the objects nature is described. Using disparity histograms for individual objects and edges allow probabilities to be calculated for each possible disparity. This allows a disparity estimate to be made to sub-pixel precision. Results are presented which employ this technique in tracking a man through a scene.

The final application will use recently developed low cost CMOS cameras [3]. These have all the advantages that fabrication with a standard CMOS process allows.

[1] When their local origins are aligned

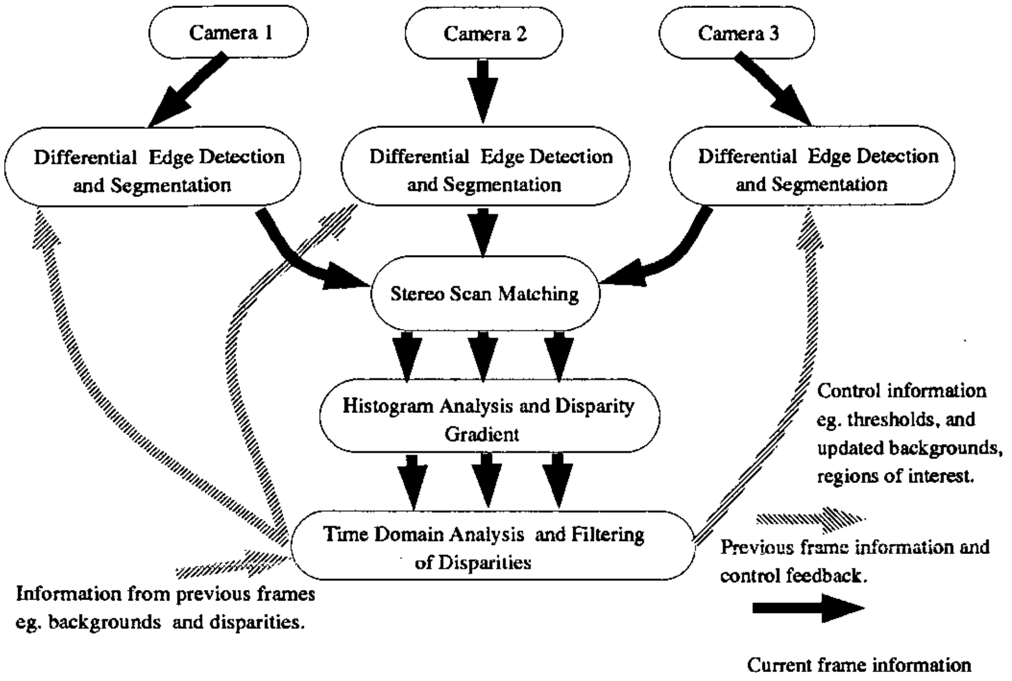

Camera 1
Camera 2
Camera 3
Differential Edge Detection and Segmentation
Differential Edge Detection and Segmentation
Differential Edge Detection and Segmentation
Stereo Scan Matching
Histogram Analysis and Disparity Gradient
Time Domain Analysis and Filtering of Disparities
Information from previous frames eg. backgrounds and disparities.
Control information eg. thresholds, and updated backgrounds, regions of interest.
Previous frame information and control feedback.
Current frame information

Figure 1: Overview of System Blocks

The structure of this paper will be a discussion of the important algorithmic points in section 2. This will be followed by comments on calibration and accuracy in section 3. Section 4 will briefly discuss a possible hardware implementation while section 5 will provide examples of trials performed over fifteen different sequences. Finally, section 6 will provide general conclusions.

2 THE SYSTEM

This section will give a summary of the system from the lower pixel based representation to the higher level edge grouping, stereo matching and false alarm elimination. An overview of the entire algorithm is shown in figure 1. Data flows from the three cameras into the initial segmentation modules before the matching algorithm is applied. It should be noted that these modules are not considered in isolation. Significant computational savings can be made in one by considering them together and in relation to the overall application. After stereo matching between the three cameras we utilise each edge's statistics and apply the disparity gradient limit, Pollard [6]. Decisions about the reliability of a particular match can be made at this stage. The disparity results are then extracted and considered in terms of recent frames and analysed over time. Statistics from this module can be used to alter thresholds at lower levels and provide some indication of the reliability of the current measurement.

2.1 Low Level Segmentation

There is no requirement to build a depth map for the entire scene. Only the outline edges of an object (ie. a human) are desired. The segmentation algorithm can be manipulated to this end. Segmentation is based on combining a roughly thresholded and clustered difference image with the output from a simplified edge detector. Edge detection is also simplified as the multiplications, divisions and floating point calculations, associated with correlation based methods such as Canny [2], are too computationally expensive in terms of hardware. In the case of edge detection one has to pay the price of an increase in noise and false edge generation. However as we only require matching in a small number of edges a disparity gradient limit [6] effectively eliminates most false erroneous edges.

We also take advantage of the fact that only edges with a substantial vertical component need be extracted. There are two reasons for this. Firstly matching becomes difficult for edges parallel to the base line of the stereo camera rig. This unsurprising conclusion has been proven in a more general sense by Skifstad and Jain [7] who show that matching is impossible for surfaces with no luminosity gradient. Secondly by their very nature humans have more significant vertical edges and short horizontal edges. These two facts allow us to restrict edge detection to a horizontal differentiation across the image. Further, a map is maintained of stationary edges. This allows the separation of significant moving edges from the background edges and reduces the effects of noise. Over a sequence of frames we build up an accurate picture of the non-moving vertical background edges which can be used to extract relevant foreground edges. A proposed edge is only accepted for matching if it is attached to a significant cluster edge.

Overall, the above method provides a reasonably robust segmentation and works sufficiently well on our present data. In larger trials alterations may have to be made to take account of possible unforeseen failures. However in this application we are attempting to extract a general trend over a period of time. Failures in any one frame can be compensated for over time and by the use of three cameras.

2.2 Computationally Simple Stereo

There have been many algorithms and constraints developed to solve the correspondence problem in its general sense [6] [5]. However it is not the aim of this work to generate a complete $2\frac{1}{2}D$ sketch for an entire scene. This would unnecessarily complicate the detection algorithm and require more recognition functionality at a higher level. We note from previous work in the form of the PMF stereo algorithm [6] that a disparity gradient limit is effective when attempting to find correct matches. One further constraint which is particularly suited to this application is overlap. We try to avoid explicit searches as much as possible. The system is therefore orientated to extracting only the relevant information from the initial raw image data. In this case, that is the outline edge of a human body. Other information is irrelevant and regarded as noise. The burden of correspondence is thus transferred to earlier stages of processing.

2.2.1 The Overlap Constraint

Use is made of the fact that we only require a single averaged disparity for the entire object. As alarm systems normally use short focal lengths, limiting accuracy, this approach has considerable advantages here. Outline features are assumed to be at a constant depth and statistical techniques used to estimate disparity to sub-pixel accuracy.

The interocular distance is constrained, by the matching algorithm which we employ. In effect, the distance between adjacent cameras cannot be greater than the width [2] of the object, ie. a human, for which we are extracting depth. Thus if the outline edges for an object are known then matching can be performed by aligning the local origins of the images and simply scanning from an edge in one image to the nearest edge in the other image. The standard calibration problem applies here. However we are not attempting to directly extract depth and are only looking for a trend in the disparity. An alarm can be activated if the human crosses a disparity threshold for some number of frames. Also, in the test equipment which we employed, the rotation and lens distortion are not significant enough to prevent a correct match and a trend being extracted. Therefore, with the exception of translational offsets, no calibration is required.

Figure 2 represents the stereo arrangement where two idealised cameras are on the same plane. The above method of scan matching depends on the fact that the two segmented views of the human overlap when their local origins are aligned. Also shown in figure 2 is the limiting condition for overlap to occur. That is when the object in the scene has precisely the same parallel width as the interocular distance. At this point the two objects will lie beside each other and do not overlap. The following equation,

$$D = W\left|\frac{X}{F}\sin\theta + \cos\theta\right| \tag{1}$$

is extracted from the geometry of the situation in figure 2 and represents what happens, to the overlap, when an object rotates by an angle θ in the scene. It is important to note that the disparity/width ratio only remains constant when the cameras have the same focal length and are positioned on the same plane. The ratio is position dependent in these situations.

[2] The width parallel to the camera plane

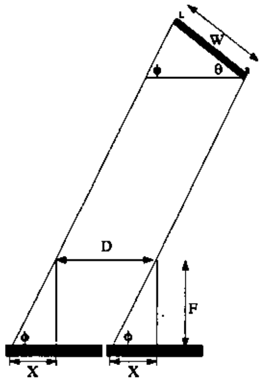


Figure 2: Two Camera Stereo Arrangement

2.3 Disparity Estimation

Quantisation errors in stereo analysis are inversely proportional to the product of the baseline and the focal length [1]. They are also inversely proportional to the range in the scene at which an feature is located. If the pixel size for an imager array is P then the RMS error in position, for a single measurement, is $\frac{P}{2\sqrt{3}}$ and in disparity $\frac{P}{\sqrt{3}}$. As we are using short focal lengths these errors become more significant. However when an edge can be assumed to be at constant depth then its disparity can be estimated more accurately. Edge location will tend to wind around its true location in the image, thus as an edge is tracked and matched we can build up a histogram of disparities. At this stage the edges can also be segmented according to the disparity gradient limit. The mean and variance of relevant parts of the disparity histogram are then used as estimates of disparity and associated confidence. Provided enough pixels are matched and the usual Gaussian assumptions are made a sub-pixel measurement for the entire object can then be calculated. At this level it is quite reasonable to calculate variances and disparities using floating points as the data rates are fairly low, for example, 20 edges per frame.

An important feature of this work is that a measure of the error is inherently provided by the calculation of the variance of the disparity. This not only takes into consideration the errors caused by quantisation but also those caused by inaccurate feature matching. These values can be utilised in any tracking filters which may be employed. This is described in the next section.

2.4 Error Analysis

We have used three cameras in order to estimate comparative disparities. This reduces the combined effects of pixel quantisation noise and the matching errors of a point. Errors which can be particularly significant in alarm systems where wide angle lenses are required. Large distances may also be expected.

The disparities from each possible measurement from three cameras are not independent. This is clear from the fact that a poorly extracted edge from the left camera will cause inaccuracies in two out of the three measurements possible from a triple camera stereo rig. In this application we assume that the errors in feature *extraction* are independent and calculate our error covariance matrix for feature *matching* on this assumption. The advantage of this approach is that it provides a *combined* variance for quantisation errors and feature matching errors.

The three possible disparity measurements,(δ_i), are represented by

$$\delta_1 = x_1 + \eta_1 - x_2 - \eta_2 \qquad \delta_2 = x_2 + \eta_2 - x_3 - \eta_3 \qquad \delta_3 = x_3 + \eta_3 - x_1 - \eta_1 \tag{2}$$

where x_i is the edge position with respect to the local coordinates and η_i is noise. A false match is considered part of the noise. Thus the errors in disparity can be summarised as

$$\Delta x_1 = \eta_1 - \eta_2 \quad \Delta x_2 = \eta_2 - \eta_3 \quad \Delta x_3 = \eta_3 - \eta_1 \tag{3}$$

and considered as combinations of independent noise sources η_i. From this an error covariance matrix can be derived based on the experimentally calculated values of Δ_i. The error covariance matrix can be represented by

$$Cov(\epsilon) = E[\mathbf{\Delta x \Delta^t x}] \tag{4}$$

where the main diagonal elements, t_{ii}, are $E[\Delta_i^2 x]$. The other elements in the matrix are

$$t_{12} = t_{21} = \frac{t_{33} - t_{11} - t_{22}}{2} \quad t_{23} = t_{32} = \frac{t_{11} - t_{22} - t_{33}}{2} \quad t_{13} = t_{31} = \frac{t_{22} - t_{33} - t_{11}}{2} \tag{5}$$

It is important to note that the values of t_{ii} can be extracted from the measurement process and used to calculate the other elements of the matrix. We have used the above measurement error matrix in the Kalman formulation where disparity velocity is modelled as the signal noise.

3 CALIBRATION

As with all stereo systems it is difficult to align cameras with no unwanted translation, rotation or pan. Algorithms have been developed which attempt to correct for these distortions [8] [9] [4]. In this system we use three cameras in order to reduce error and increase our chances of a correct match being found. These cameras are arranged as closely as possible to be on the same imaging plane.

The apparatus was fairly crude and only adjusted as best as possible by hand using a white cross on black background and subtracting the images one from another until there was no fringe around the edges. It appears from our results that rotation was far less of a problem than at first thought and that the translation could be easily corrected for using simple offsets. Also we have taken advantage, in this system, of the fact that only a threshold disparity need be crossed to activate the alarm. If this is consistently breached over a number of frames the alarm is sounded. Full three dimensional rectification is not required. The disparity threshold can be calculated automatically when the system is installed or manufactured.

Using three cameras opens possibilities of being able to correct for translational misalignments automatically. Different offset combinations could be attempted until consistency is obtained over a sequence of frames with a person walking around at the same depth. Alternatively for more accurate distance measurements the installation could be linked to a computer and the calibration parameters calculated using more accurate techniques.

4 HARDWARE

It is the intention that the above algorithms be easily implemented in cost effective hardware. The central feature of the system is the camera. There is little point in developing a commercial piece of processing hardware only to be defeated by the cost of CCD cameras.

Implementation should be possible using a CMOS sensor with some on-chip processing to perform low level segmentation and edge detection. This processing would also be required to maintain histograms and edge maps. The above circuitry could then be interfaced to a general microprocessor to perform the more complicated calculations of thresholds, means and variances. Finally an interface to memory for storage of edge maps and background images will also be necessary.

5 RESULTS

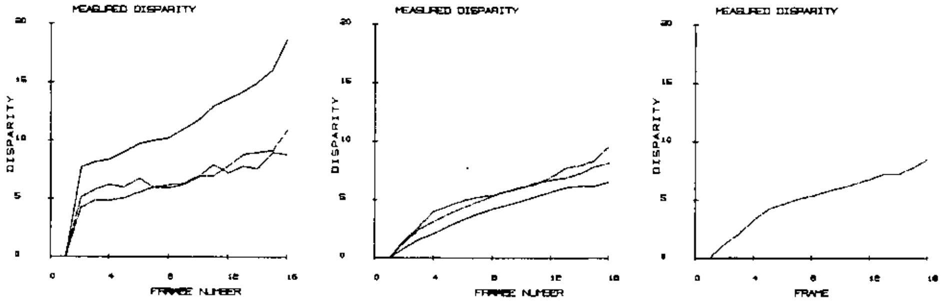
Trials and experiments have been conducted on 15 trinocular image sequences, from scenes of varying difficulty. We present examples of the system working in three scenes with different lighting conditions. Also presented is the output from the Kalman filter and the confidence weighted average of the three Kalman estimates. Absolute values are not significant in this application as we are only interested in a trend for a particular installation. However it should be noted that the disparity from the two outside cameras is halved before being input to the tracking filter.

Each sequence is sixteen images long captured from CMOS cameras [3] of 256x256 pixels. The images are digitised to eight bits at five frames per second using in-house frame grabbers. Figure 3 shows the results extracted from sequence 1, figure 6, as a man walks towards the camera from 12m.

Figure 4 is derived from sequence 2, figure 7 as a man walks towards the camera from 17m. This scene is different from sequence 1 in that background is dark. Figure 5 is derived from sequence 3, figure 8, as a man walks away from the camera. He started at 6m.

In all sequences the human is detected and tracked through the scene. Inevitably there are frames when matching becomes difficult as can be seen in the raw data graphs in figure 4 and figure 5. The large spikes are the result of matching

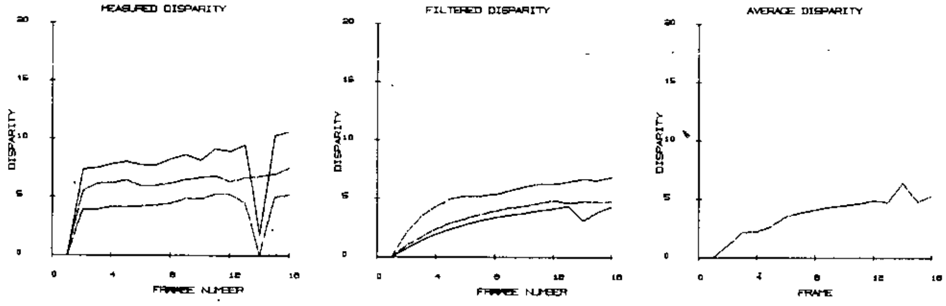
Figure 3: Sequence 1: Disparity against Time (Frames)


Raw

Filtered

Weighted Average

Figure 4: Sequence 2: Disparity against Time (Frames)


Raw

Filtered

Weighted Average

failures and poor extraction. However they are very obvious in relation to the disparities extracted from the previous and next frames. The filter manages to eliminate the worst effects of these spikes.

We intend to expand the trial to include a greater number of sequences. It is only by such an experimental process that the system can be refined and knowledge gained as to when it will fail. This knowledge can be incorporated in an iterative manner by changes in thresholds and small changes in the algorithm. Although tedious this experimental approach has proven its worth in the systems designed by Anderson [3] and Vellacot [10].

6 CONCLUSIONS

We have developed a stereo alarm system which employs the fact that for stereo cameras on the same plane a constant disparity to width ratio exists for the entire scene. This has allowed considerable savings in computational cost which would allow implementation using CMOS cameras with on-chip processing.

A summary of the trials conducted has been described and accuracy examined. Measures of accuracy are extracted directly from the data and used in calculating confidences. This technique, combined with Kalman filtering and three camera stereo, has been used to calculate disparities to sub-pixel accuracy. Finally the system could be of use in automatic alarm systems and door opening and further trials could be performed to this end.

REFERENCES

[1] J. J. Rodriguez J.K. Aggarwal. Stochastic analysis of stereo quantisation error. In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 1990.

[2] J Canny. Finding edges amd lines in images. *MIT AI memo*, 1983.

Figure 5: Sequence 3: Disparity against Time (Frames)

Raw Filtered Weighted Average

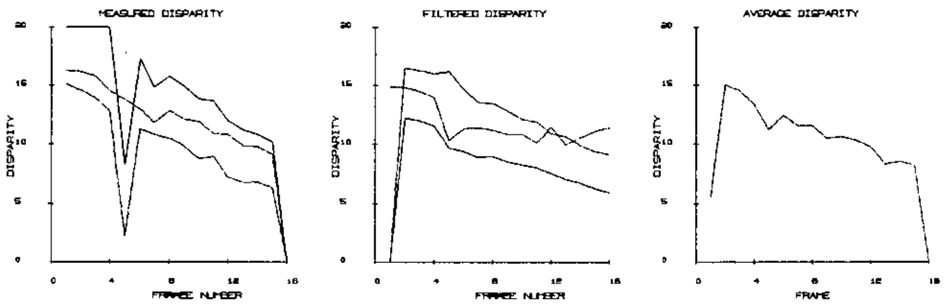


Figure 6: Triple Stereo: Sequence 1
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels

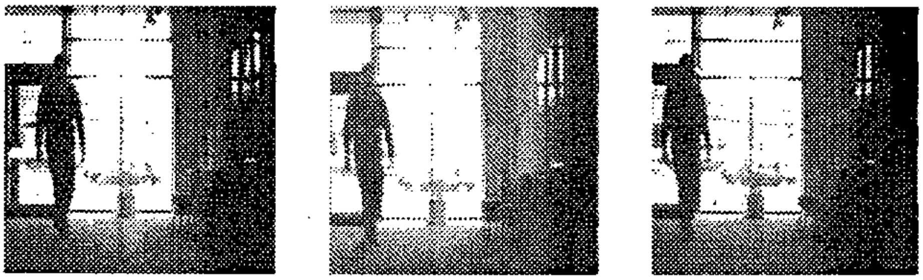


Figure 7: Triple Stereo: Sequence 2
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels

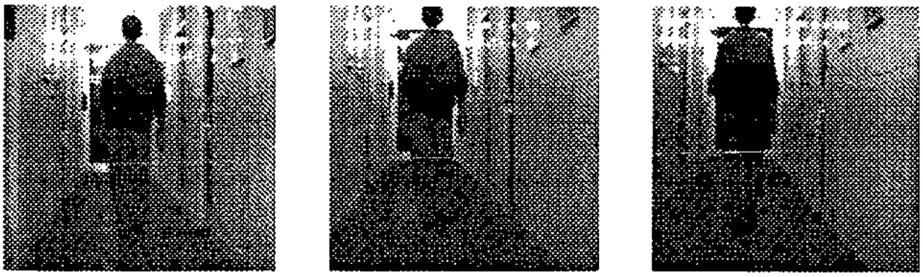
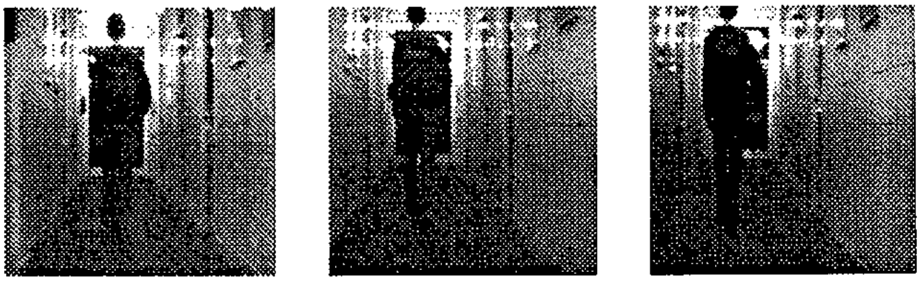


Figure 8: Triple Stereo: Sequence 3
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels



[3] G Wang D Renshaw, P B Denyer and M Lu. Asic vision. In *IEEE Custom Integrated Circuits Conference*, 1990.

[4] H C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, September 1981.

[5] D Marr. *Vision*. Freeman, 1982.

[6] J E W Mayhew S B Pollard and J P Frisby. A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[7] K Skifstad and R Jain. Range estimation from intensity gradient analysis. Technical Report CSE-TR-02-88, University of Michigan, 1988.

[8] R Y Tsai. A versatile camera calibration technique for high accuracy 3d machine metrology using off the shelf tv cameras and lenses. In *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1986.

[9] R Y Tsai. Review of rac-based camera calibration. *Vision*, November 1988.

[10] O. Vellacot. A framework of heirarchy for neural theory, chapter6: Numberplate recognition. In *PhD Thesis, Edinburgh University*, 1991.

AN INTELLIGENT ALARM SYSTEM

K W J Findlay, D Renshaw and P B Denyer

Edinburgh University, Scotland.

1 Introduction

Over the years considerable research effort has been directed towards the theory and practice of AI techniques. Much of this work has been directed at solving vision problems and algorithms have been developed to solve both general and specific problems. However there are relatively few practical vision systems in use, either in an industrial or consumer environment. Those that have been successful are normally restricted to recognition tasks on an assembly line or character recognition in places such as post offices. A main reason for this is cost. We aim to design further working vision systems using minimum hardware at the lower levels of processing. These systems are directed at specific applications.

In this paper we describe an alarm system which will detect and track a human moving around a scene from stationary cameras. The system has many applications in situations where "invisible" boundaries are required and could replace or complement the light beams and active electronics which are currently in use.

Stereo vision is a possible solution to the position and size problem and a low cost algorithm has been developed which will utilise constraints particular to this problem. It is based on the fact that [1] two views of the same object will overlap when their local origins are aligned. Using this constraint it allows us to solve the so called *correspondence problem* [2] without correlation or extensive searching.

The final application will use recently developed low cost CMOS cameras [2]. There is little point in economising in other areas of processing only to be defeated by the current cost of CCD cameras. Another major advantage of this technology is the capability of placing processing on-chip together with the sensor array. Pixel sizes and shapes can also be manipulated. A CMOS camera's use with on-chip processing has been demonstrated by Anderson et. al. [3] in a finger print recognition system.

The structure of this paper will be an explanation of the important points of the algorithm in section 2. This will be followed by discussion about the systems calibration and accuracy, section 3. Also explained, in section 3, are the reasons for using three cameras as opposed to two. Section 4 gives examples of the algorithm applied to trial sequences. Finally section 5 will provide general conclusions.

2 The System

Here we provide a summary of the system from the lower pixel based representation to the higher level edge grouping and stereo matching. An overview of the entire algorithm is shown in figure 1. Data flows from the three cameras into the initial segmentation modules before the stereo matching algorithm is applied. It should be noted that these modules are not treated independently. We have found that significant computational savings can be made in one by considering it in relation to the other. After stereo matching, between the three cameras, we utilise edge statistics and apply the disparity gradient limit [4]. This allows a decision about a particular edges "goodness' to be made. The disparity results are then extracted, considered in terms of recent frames, and analysed over time. The statistics from this module can then be used to alter the thresholds at lower levels of processing.

2.1 Low Level Segmentation

In this work edge detection is based on difference techniques since the multiplications, divisions and floating point calculations, associated with correlation based methods such as Canny [1], are too computationally expensive in terms of hardware. The main problem caused by reducing complexity at the pixel level is false edge generation. These will cause incorrect stereo matches. However it has been shown [4] that a disparity gradient limit is an effective control in determining correct matches between the two

[1] Provided an objects width is greater than the distance between the two cameras.

[2] Solving the correspondence problem involves finding a scene feature in one image and trying to find the same, or corresponding, feature in the other image.

views. Also, inaccurate matches tend to become obvious over a period of time.

Further savings can be made in edge detection by taking account of the nature of the human form. Humans tend to have long vertical edges and short, insignificant, horizontal edges. As we are using laterally displaced stereo cameras we do not attempt matching for horizontal edges. Edge extraction can therefore be confined to lateral differentiation.

The edge information is combined with segmentation information from a clustering/thresholding algorithm which again does not employ the more computationally complex arithmetic described above.

2.2 Computationally Simple Stereo Tracking and Control

The correspondence problem is well known in most vision applications and many constraints have been devised as attempts to solve it in its most general sense. We note from previous work in the form of the PMF stereo algorithm [4] that a disparity gradient limit is an effective constraint when attempting to find correct matches. We also take note of the possibly obvious but nevertheless important fact that depth information cannot be extracted from a scene or area where there is not a luminosity gradient [5] and make our algorithm edge based. Edges have the highest luminosity gradient. There are many other advantages in using edges one of which is their continuing presence when shadows are cast. A further constraint which is particularly suited to this application is what we call the overlap constraint. As said before we are trying to avoid explicit searches or correlation. The system is therefore orientated to extracting only the relevant information from the initial raw image data. In this case that is the outline edge of a human body. Other information is irrelevant and regarded as noise. We thus transfer the burden of correspondence to the segmentation stages of early image processing. Also it is only necessary to find a small part of that outline reliably in order to estimate depth. The segmentation and edge detection modules can therefore be fairly crude. Obviously the more correct edges that are found the better the accuracy of disparity estimation for the whole object.

2.2.1 The Overlap Constraint

Once the candidate outline edges have been extracted it is a simple matter of scanning along a raster until the first edge in the other camera is found. The distance between these two edges is the disparity. Figure 2 represents the stereo arrangement where two cameras are on the same plane. The above method of scan matching depends on the fact that the two segmented views of the human overlap when their local origins are aligned with respect to their calibration offsets. Also shown in figure 2 is the limiting condition for overlap to occur. That is when the object in the scene is precisely the same width as the interocular distance, D. At this point the two objects will lie beside each other and not overlap. The following equation,

$$D = W \left| \frac{X}{F} \sin\theta + \cos\theta \right| \qquad (1)$$

is extracted from the geometry of the situation in figure 2 and represents what happens to the overlap when an object rotates by an angle θ in the scene.

2.3 Error Control

Histogram analysis is used to form an estimate of the disparity of each edge. The advantage of this technique is that it reduces the effect of pixel quantisation, a factor significant in alarm systems due to the effect of wide angle lenses. Also, large distances may be expected, reducing the actual size of the object in the image and increasing the importance of an individual matched pixel. If the correct disparity for an edge is somewhere between two pixel measurements an estimate for the entire edge can be calculated from the proportions of pixels at each disparity. We assume that for all practical situations where this system will be used the entire human is at one depth. Floating points can be used here as an inexpensive microprocessor should be fast enough to calculate a mean for the small number of difference clusters per frame.

In this application we have used three cameras in order to estimate comparative disparities. This also reduces the effects of pixel quantisation noise. Figure 3 shows the normalised error probability distributions for both two and three camera rigs. In the later there are three possible depth measurements which are averaged. The simulations were performed using the camera parameters described in sections 4 and involved rectifying the depth from 100000 randomly generated points in a scene with depth 20 meters. The errors generated were caused by both quantisation noise, dependent on camera geometry, and additional simulated noise, ($\sigma = 0.5$). It should perhaps be noted that the error probability functions vary with distance and that the p.d.f.'s shown in Figure 3 are for the complete simulated area. This factor will have to be taken into account when alarm thresholds are considered.

3 Calibration

As with all stereo systems it is extremely difficult to align cameras with no unwanted translation, rotation or pan. Algorithms have been developed which attempt to correct for these distortions [6]. However, accurate calibration is not required in this system as a simple threshold can be utilised to determine invisible distance boundaries. Translational offsets are sufficient. We have taken advantage of the fact that only a disparity value need be crossed to activate the alarm. If this is consistently breached over a number of frames the alarm is sounded. The disparity threshold could be calculated automatically when the system is installed, or manufactured, by people moving at a known distance.

Also as an alternative to the above, more accurate distance measurements could be extracted if the system is calibrated accurately. The installation could be linked to a computer and the calibration parameters calculated using the more accurate teqniques described in the literature.

4 Results

Trials and experiments have been conducted on fifteen trinocular image sequences of varying difficulty. These have been largely successful in detection and relative depth estimation. In all sequences the moving human has been detected, and then tracked for most of its "walk" through the scene.

We present examples of the system working in three different scenes with varying lighting conditions. The apparatus employed was mechanically fairly crude and only adjusted as best as possible, by hand, using a white cross on black background and subtracting the images one from another until there was no fringe around the edges. It appears that rotation was far less of a problem than at first thought and translation could be easily corrected using simple offsets.

Each sequence is sixteen images long and captured from CMOS cameras [2] of 256x256 pixels. The images are digitised to eight bits at five frames per second using in-house frame grabbers.

Figure 4 shows comparative depths extracted from three sequences. Figure 4(a) is derived from sequence 1, Figure 5, and consists of a man walking towards the camera from 12m. The spike around frame 5 is caused by a sudden change in background. However the system is capable of compensating within 2 frames. Figure 4(b) is derived from sequence 2, Figure 6 and shows a man walking away from the camera from 5m. This scene is different from sequence 1 in that background is dark. Figure 4(c) is derived from sequence 3, Figure 7, and consists of a man walking through a door towards the camera from 8m.

In all cases the traces tend in the correct direction and have sufficient gradient to set off an alarm once a threshold is crossed.

We intend to expand the trial to include a greater number of sequences and refine the system to track these examples. It is only by such an experimental process that the system can be refined and knowledge gained as to when it will fail. This knowledge can be incorporated in an iterative manner by changes in thresholds and small changes in the algorithm. Although tedious this experimental approach has proven its worth in the systems designed by Anderson [2].

5 Conclusions

A low cost vision system which could be utilised in the area of alarm verification and detection has been developed. The stereo system has been designed to provide distance measurements which could be utilised as "invisible" barriers. Considerable savings in complexity have been achieved at the lower levels of processing which would allow a simpler hardware implementation.

References

[1] J Canny. Finding edges amd lines in images. *MIT AI memo*, 1983.

[2] G Wang D Renshaw, P B Denyer and M Lu. Asic vision. In *IEEE Custom Integrated Circuits Conference*, 1990.

[3] P B Denyer D Renshaw S Anderson, W H Bruce and G Wang. A single chip sensor and image processor. In *IEEE Custom Integrated Circuits Conference*, 1991.

[4] J E W Mayhew S B Pollard and J P Frisby. A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[5] K Skifstad and R Jain. Range estimation from intensity gradient analysis. Technical Report CSE-TR-02-88, University of Michigan, 1988.

[6] R Y Tsai. Review of rac-based camera calibration. *Vision*, November 1988.

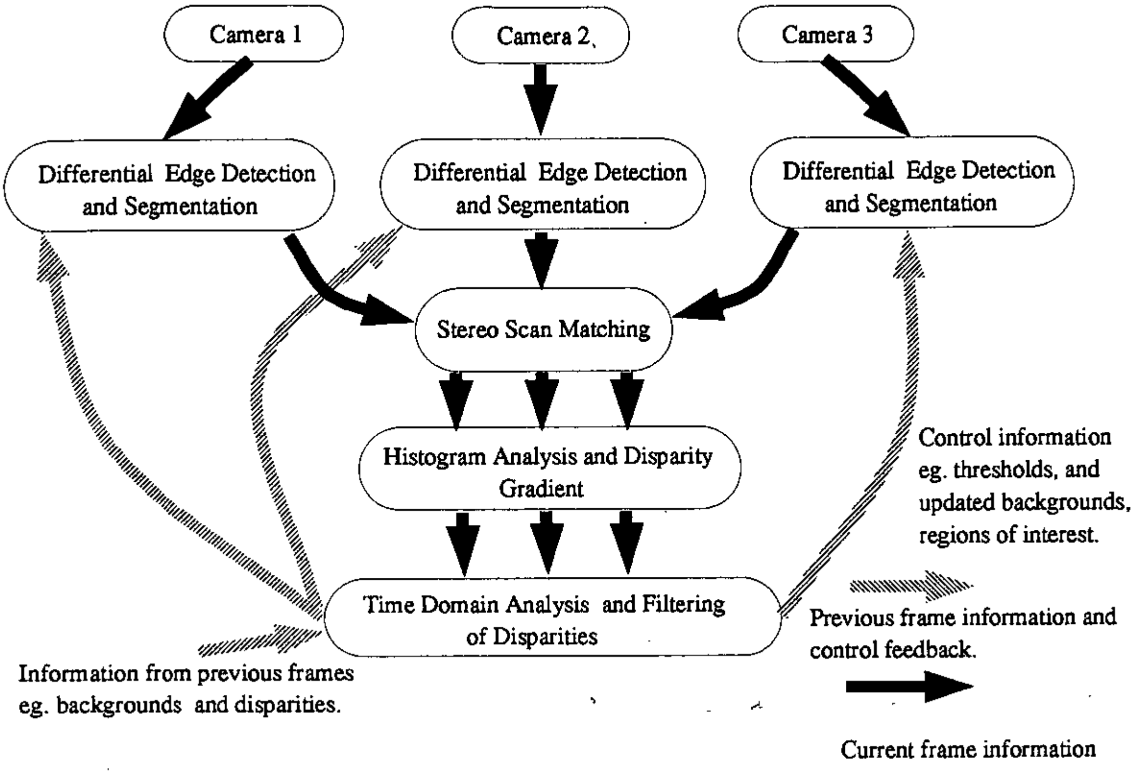

Camera 1
Camera 2
Camera 3
Differential Edge Detection and Segmentation
Differential Edge Detection and Segmentation
Differential Edge Detection and Segmentation
Stereo Scan Matching
Histogram Analysis and Disparity Gradient
Time Domain Analysis and Filtering of Disparities
Control information eg. thresholds, and updated backgrounds, regions of interest.
Previous frame information and control feedback.
Current frame information
Information from previous frames eg. backgrounds and disparities.

Figure 1: Overview of System Blocks

Figure 2: Two Camera Stereo Arrangement

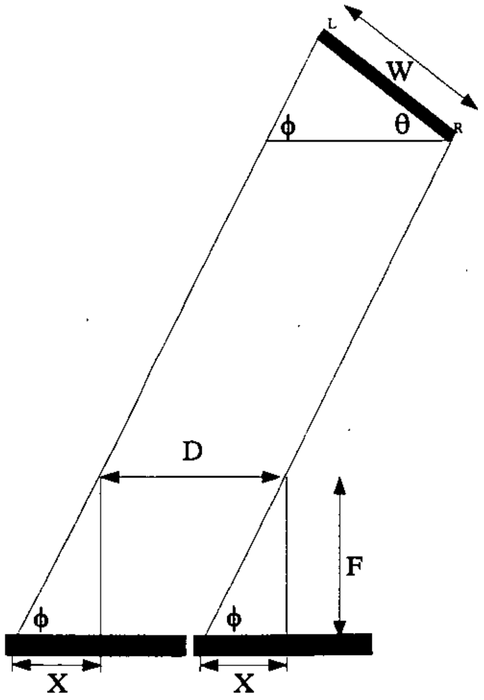


Figure 3: Error Probability Distributions
Three Camera Rigg (top)
Two Camera Rigg (bottom)

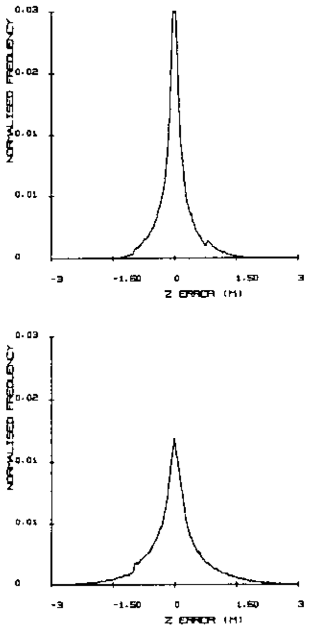
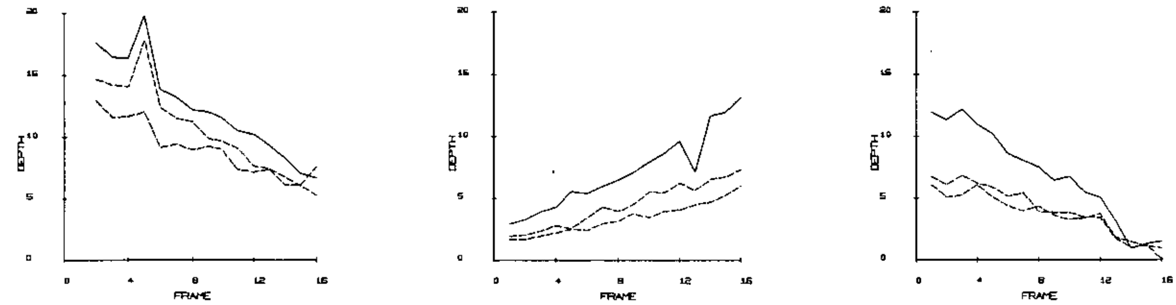


Figure 4: Disparity against Time (Frames) Solid traces are distance extracted from between the two outside cameras; broken lines are distances extracted between the inside camera pairings.



Sequence 1 (a)

Sequence 2 (b)

Sequence 3 (c)

Figure 5: Triple Stereo: Sequence 1
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels

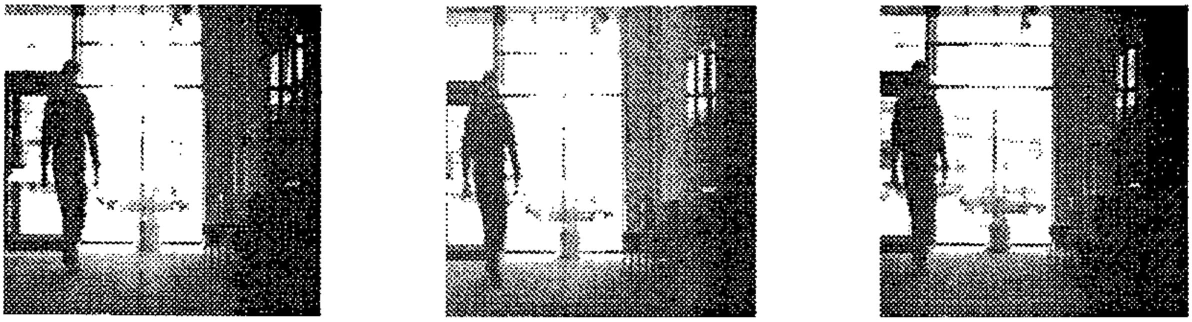


Figure 6: Triple Stereo: Sequence 2
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels

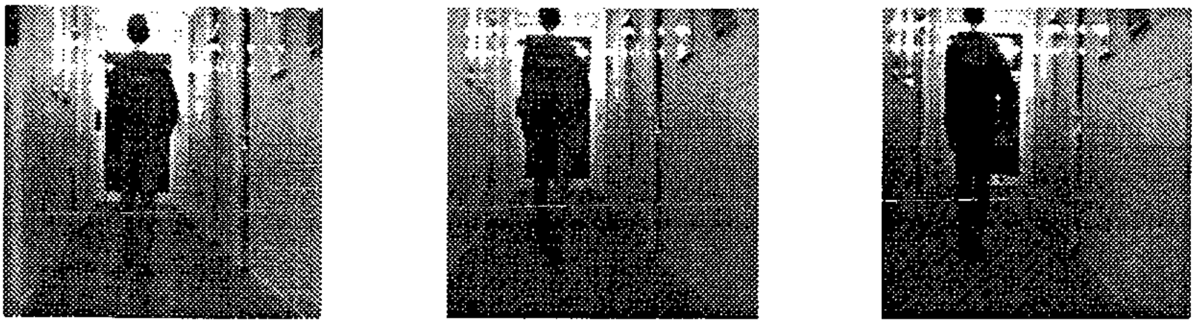
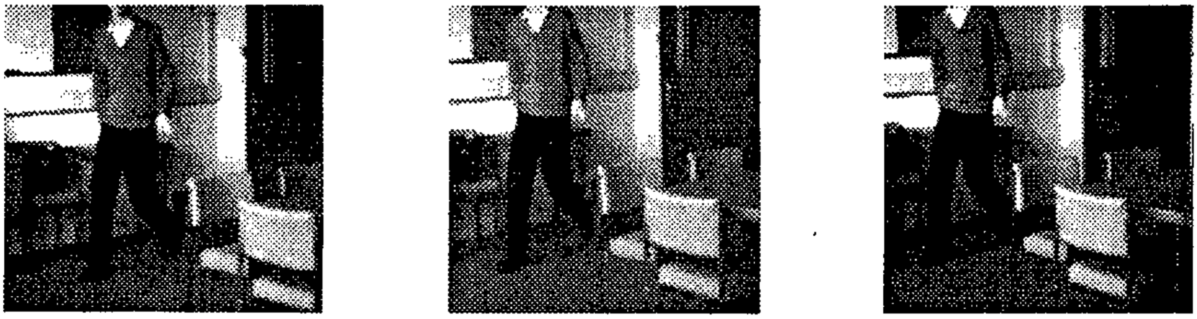


Figure 7: Triple Stereo: Sequence 3
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels



A Low Cost Stereo Alarm System for VLSI

K.W.J. Findlay, D.Renshaw and P.B. Denyer

Integrated Systems Group, Department of Electrical Engineering, Edinburgh University,
E-Mail:kevinf@ee.ed.ac.uk, Fax: 31 622 4358

Abstract

Over the years considerable research has been conducted in the area of passive stereo vision. Most of the algorithms are designed to extract a complete 2 $\frac{1}{2}$D sketch over the entire image. The above approach is not always necessary when a general vision system is not required. This paper describes an original algorithm which utilises *a priori* knowledge about an object's width to reduce the matching search to a simple X-axis scan. In effect it transfers the complexity from the matching process to the segmentation stage which can be performed using traditional difference techniques.

The algorithm is adapted for VLSI implementation using a low cost CMOS image sensor and requires no multiplications, divisions or floating point calculations at the pixel level of the image hierarchy. This will allow a more commercially viable implementation.

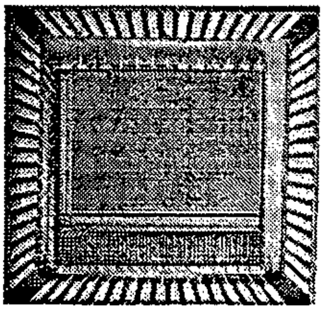


Figure 1: An ASIS Image Sensor With On-Chip Processing

1 Introduction

Much research effort has been directed towards the theory and practice of image processing and vision. However a great many systems require massive processing power in order to operate in real time. This hardware is both expensive and time consuming to design, limiting both the research effort and the range of practical systems which can be implemented. Few vision systems are working today in a commercial environment and one major reason for this is the cost of hardware. It is an aim of our work to demonstrate systems which are commercially viable in terms of the hardware required. This has been done by placing restrictions on the algorithm design to eliminate the more expensive parts. In effect this is an algorithm for a target architecture rather than hardware aimed at computing an algorithm. The architectures aimed for are low cost CMOS VLSI implementations of both image sensor and processing. This technology has been developed by Denyer et.al. [3] [5] and an example, showing both processing and sensor, is given in figure 1.

The above CMOS techniques can be exploited to reduce the final implementation cost, eliminating the need for expensive CCD technology. However, chip space restrictions require that constraints be placed on a particular algorithm's arithmetic. In effect, the more complex arithmetic has to be excluded from the lower levels of processing. An example of this is given by Anderson [3] in a fingerprint recognition system. In this architecture a CMOS sensor was integrated on-chip with all the necessary recognition processing. An important consideration in such a system is the problem of analogue to digital conversion. In this case, the video output from the sensor was thresholded and digitised at the same time. The following stereo algorithm has been developed with the above hardware constraints in mind and there are no multiplications, divisions or floating point calculations at the pixel level of processing.

In view of the above, the work presented in this paper is the design of a low cost alarm system which will detect and track a human moving around a scene viewed from stationary cameras. We have chosen stereo vision as a possible solution to the above problem and have developed an algorithm which will utilise constraints particular to this application. Notice is taken of the fact that for cameras on the same plane an overlap will occur if the width, (parallel to the image plane), of the object is greater than the interocular distance, no matter where the object is positioned in the scene. This constraint is further explained in section 2.2.

The paper will start with a discussion of the important points of the algorithm, section 2, followed by comments on the calibration of the system in section 3. Section 4 will deal with hardware considerations while the last two sections will summarise results and present general conclusions.

2 The System

This section will give a summary of the system from the lower pixel based representation to the higher level edge grouping, stereo matching and false alarm elimination. Data flows from three cameras into the initial segmentation modules before the matching algorithm is applied. It should be noted that vision modules are not considered in isolation. Significant computational savings can be made in one module by considering them together and in relation to the overall application. After stereo matching between the three cameras we utilise the statistics of each edge and apply the disparity gradient limit, Pollard [6]. Decisions about the reliability of a particular match can be made at this stage. The disparity results are then extracted and considered in terms of recent frames and analysed over time. Statistics from this module can be used to alter thresholds at lower levels and provide some indication of the reliability of the current measurement.

2.1 Low Level Segmentation

There is no requirement to build a depth map for the entire scene. Only the outline edges of an object (ie. a human) are desired. The segmentation algorithm can be manipulated to this end. Segmentation is based on combining a roughly thresholded and clustered difference image with the output from a simplified edge detector. Edge detection is also simplified as the multiplications, divisions and floating point calculations, associated with correlation based methods such as Canny [2], are too computationally expensive in terms of hardware. In the case of edge detection, one has to pay the price of an increase in noise and false edge generation. However, as we only require matching in a small number of edges a disparity gradient limit [6] effectively eliminates most false matches.

We also take advantage of the fact that only edges with a substantial vertical component need be extracted. There are two reasons for this. Firstly, matching becomes difficult for edges parallel to the base line of the stereo camera rig. This unsurprising conclusion has been proven in a more general sense by Skifstad and Jain [7] who show that matching is impossible for surfaces with no luminosity gradient. Secondly, by their very nature humans, have more significant vertical edges and short horizontal edges. These two facts allow us to restrict edge detection to a horizontal differentiation across the image. Further, a map is maintained of stationary edges allowing foreground features to be separated and reducing the effects of noise. Over a sequence of frames we build up an accurate picture of the non-moving vertical background edges which can be used to extract relevant foreground. A proposed edge is only accepted for matching if it is attached to a significant cluster. Overall, the above method provides a reasonably robust segmentation and works sufficiently well on our present data.

2.2 Computationally Simple Stereo

There have been many algorithms and constraints developed to solve the correspondence problem in its general sense. However it is not the aim of this work to generate a complete $2\frac{1}{2}D$ sketch for an entire scene. This would unnecessarily complicate the detection algorithm and require more recognition functionality at a higher level. We note from previous work in the form of the PMF stereo algorithm [6] that a disparity gradient limit is effective when attempting to find correct matches. One further constraint which is particularly suited to this application is overlap. We try to avoid explicit searches as much as possible. The system is therefore orientated to extracting only the relevant information from the initial raw image data. In this case, the relevant information is the outline edge of a human body. Other information is irrelevant and regarded as noise. The burden of correspondence is thus transferred to earlier stages of processing.

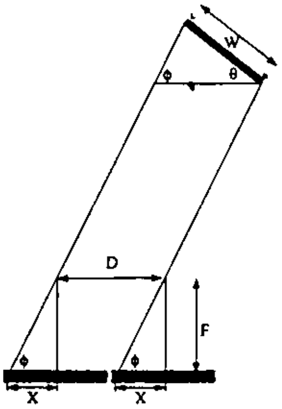


Figure 2: Two Camera Stereo Arrangement

2.2.1 The Overlap Constraint

Use is made of the fact that, for this application, we only require a single averaged disparity for the entire object. As alarm systems normally use short focal lengths, limiting accuracy, this approach has considerable advantages here. Outline features are assumed to be at a constant depth and statistical techniques are to estimate disparity to sub-pixel accuracy.

The interocular distance is constrained, by the matching algorithm which we employ. In effect, the distance between adjacent cameras cannot be greater than the width [1] of the object, ie. a human, for which we are extracting depth. Thus, if the outline edges for an object are known then matching can be performed by aligning the local origins of the images and simply scanning from an edge in one image to the nearest edge in the other image. The standard calibration problem applies here. However, we are not attempting to directly extract depth and are only looking for a trend in the disparity. An alarm can be activated if the human crosses a disparity threshold for some number of frames. Also, in the test equipment which was employed, the rotation and lens distortion are not significant enough to prevent a correct match and a trend being extracted. Therefore, with the exception of translational offsets, no calibration is required.

Figure 2 represents the stereo arrangement where two idealised cameras are on the same plane. The above method of scan

[1] The width parallel to the camera plane

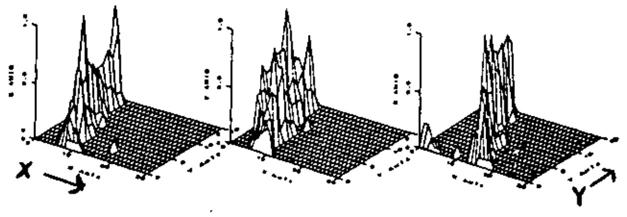


Figure 3: Disparity Histograms, X axis, Represented in Time, Y axis

matching depends on the fact that the two segmented views of the human overlap when their local origins are aligned. Also shown in figure 2 is the limiting condition for overlap to occur. That is when the object in the scene has precisely the same parallel width as the interocular distance. At this point the two objects will lie beside each other and do not overlap. The following equation,

$$D = W\left|\frac{X}{F}\sin\theta + \cos\theta\right| \tag{1}$$

is extracted from the geometry of the situation in figure 2 and represents what happens, to the overlap, when an object rotates by an angle θ in the scene. It is important to note that the disparity/width ratio only remains constant when the cameras have the same focal length and are positioned on the same plane. The ratio is position dependent in these situations.

2.3 Disparity Estimation

Quantisation errors in stereo analysis are inversely proportional to the product of the baseline and the focal length [1]. They are also inversely proportional to the range at which an feature is located. If the pixel size for an imager array is P then the RMS error in position, for a single measurement, is $\frac{P}{2\sqrt{3}}$ and in disparity $\frac{P}{\sqrt{3}}$. As we are using short focal lengths these errors become more significant. However, when an edge can be assumed to be at constant depth then its disparity can be estimated more accurately. Edge location will tend to wind around its true location in the image, thus as an edge is tracked and matched we can build up a histogram of disparities for the entire edge. At this stage the edges can also be segmented according to the disparity gradient limit. The mean and variance of relevant parts of the disparity histogram are then used as estimates of disparity and associated confidence. Provided enough pixels are matched and the usual Gaussian assumptions are made a sub-pixel measurement for the entire object can now be calculated. Example disparity histograms, plotted through time, for each of the three possible measurements, are shown in figure 3. At this level it is quite reasonable to calculate variances and disparities using floating points as the data rates are fairly low, for example, 20 edges per frame.

An important feature of this work is that a measure of the error is inherently provided by the calculation of the variance of the disparity. This not only takes into consideration the errors caused by quantisation but also those caused by inaccurate feature matching. These values can be utilised in any tracking filters which may be employed as described in the next section.

2.4 Error Analysis

We have used three cameras in order to estimate comparative disparities. This reduces the combined effects of pixel quantisation noise and the matching errors of a point. These errors can be particularly significant in alarm systems where wide angle lenses are required. Large distances may also be expected.

The disparities from each possible measurement from three cameras are not independent. This is clear from the fact that a poorly extracted edge from the left camera will cause inaccuracies in two out of the three measurements possible from a triple camera stereo rig. In this application we assume that the errors in feature *extraction* are independent and calculate our error covariance matrix for feature *matching* on this assumption. The advantage of this approach is that it provides a *combined* variance for quantisation and feature matching errors.

The three possible disparity measurements,(δ_i), are represented by

$$\delta_1 = x_1+\eta_1-x_2-\eta_2 \;\; \delta_2 = x_2+\eta_2-x_3-\eta_3 \;\; \delta_3 = x_3+\eta_3-x_1-\eta_1 \tag{2}$$

where x_i is the edge position with respect to the local coordinates and η_i is noise. A false match is considered part of the noise. Thus the errors in disparity can be summarised as

$$\Delta x_1 = \eta_1 - \eta_2 \;\; \Delta x_2 = \eta_2 - \eta_3 \;\; \Delta x_3 = \eta_3 - \eta_1 \tag{3}$$

and considered as combinations of independent noise sources η_i. From this an error covariance matrix can be derived based on the experimentally calculated values of Δ_i. The error covariance matrix can be represented by

$$Cov(\epsilon) = E[\mathbf{\Delta x \Delta^t x}] \tag{4}$$

where the main diagonal elements, t_{ii}, are $E[\Delta_i^2 x]$. The other elements in the matrix are

$$t_{12} = t_{21} = \frac{t_{33} - t_{11} - t_{22}}{2} \tag{5}$$

$$t_{23} = t_{32} = \frac{t_{11} - t_{22} - t_{33}}{2} \tag{6}$$

$$t_{13} = t_{31} = \frac{t_{22} - t_{33} - t_{11}}{2} \tag{7}$$

It is important to note that the values of t_{ii} can be extracted from the measurement process and used to calculate the other elements of the matrix. We have used the above measurement error matrix in a Kalman formulation where the disparity velocity is modelled as the signal noise. Again, in a final implementation the frame rates required will allow these calculations using limited hardware.

3 Calibration

As with all stereo systems it is extremely difficult to align cameras with no unwanted translation, rotation or pan. Algorithms have been developed which attempt to correct for these distortions [8] [9] [4]. In this system we use three cameras in order to reduce the possibility of error and increase our chances of a correct match being found. We have taken advantage of the fact that only a threshold disparity need be crossed to activate the alarm. As a result full three dimensional rectification is not required.

The apparatus used in the experiments was fairly crude and only adjusted as best as possible using a white cross on a black background and subtracting images from one another until there was no unwanted fringe around the edges. With this equipment, rotation was far less of a problem that at first thought and translation could easily be corrected using simple offsets. For more accurate calibration, including corrections for lens distortion, the final equipment could be linked to a computer and parameters calculated using more computational techniques.

4 Hardware

The central feature of the system is the camera. There is little point in developing a commercial piece of processing hardware only to be defeated by the cost of CCD cameras. To this end we intend to utilise a CMOS sensor[3],which can be customised to a particular application, including changing pixel array sizes and altering aspect ratios.

It is the intention of the above algorithms that they be easily implemented in cost-effective hardware. As a result they have been designed to have no multiplications, divisions or floating point calculations at the pixel level. Floating point calculations can be used in deciding thresholds and object disparities, provided that the data is accumulated by the lower level processing and presented in a suitable form to the microprocessor. Standard microprocessors are capable of calculating such arithmetic, at video rates.

5 Results

Trials and experiments have been conducted on fifteen trinocular image sequences, from scenes of varying difficulty. We present examples of the system working in three scenes with different lighting conditions. Also presented is the output from the Kalman filter and the confidence weighted average of the three Kalman estimates. Absolute values are not significant in this application as we are only interested in a trend for a particular installation. However it should be noted that the disparity from the two outside cameras is halved before being input to the tracking filter.

Each sequence is sixteen images long captured from CMOS cameras [3] of 256x256 pixels. The images are digitised to eight bits at five frames per second using in-house frame grabbers. Figure 4 shows the results extracted from sequence 1, figure 7, as a man walks towards the camera from 12m.

Figure 4: Sequence 1: Disparity (y) against Time (x), frames

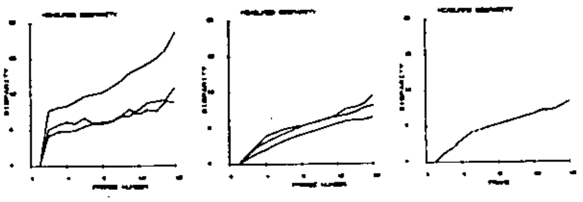

Raw Filtered Weighted Average

Figure 5: Sequence 2: Disparity (y) against Time (x), frames

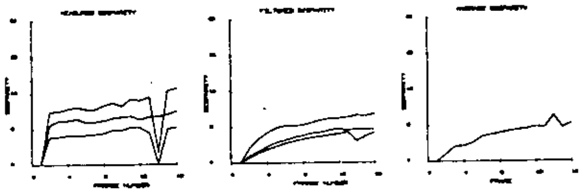
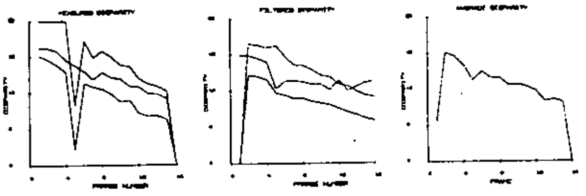

Raw Filtered Weighted Average

Figure 5 is derived from sequence 2, figure 8 as a man walking towards the camera from 17m. This scene is different from sequence 1 in that the background is dark. Figure 6 is derived from sequence 3, figure 9, as a man walks away from the camera. He started at 6m.

In all sequences the human is detected and tracked through the scene. Inevitably there are frames when matching becomes difficult as can be seen in the raw data graphs in figure 5 and figure 6. The large spikes are the result of matching failures and poor extraction. However they are very obvious in relation to the disparities extracted from the previous and next frames. The filter manages to eliminate the worst effects of these spikes.

We intend to expand the trial to include a greater number of sequences. It is only by such an experimental process that the system can be refined and knowledge gained as to when it will fail. This knowledge can be incorporated in an iterative manner by changes in thresholds and small changes in the algorithm. Although tedious this experimental approach has proven its worth in the systems designed by Anderson [3] and Vellacot [10].

Figure 6: Sequence 3: Disparity (y) against Time (x), frames


Raw Filtered Weighted Average

6 Conclusions

A low cost stereo vision alarm system has been developed. To the best of our knowledge the above overlap constraint has not been published explicitly in the stereo vision literature. Its use has allowed considerable savings in complexity and transferred the burden of correspondence, for an object's outside edges, into the segmentation stage.

Results have been presented here which show the effectiveness of the algorithm in different scenes. We have also extracted measures of accuracy directly from the data which can be used in calculating confidences and tracking filters. Finally, hardware implementation would be a viable option in a commercial environment.

References

[1] J. J. Rodriguez J.K. Aggarwal. Stochastic analysis of stereo quantisation error. In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 1990.

[2] J Canny. Finding edges amd lines in images. *MIT AI memo*, 1983.

[3] G Wang D Renshaw, P B Denyer and M Lu. Asic vision. In *IEEE Custom Integrated Circuits Conference*, 1990.

[4] H C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, September 1981.

[5] L M Ying P B Denyer, W Gouyu and S Anderson. On-chip cmos sensors for vlsi imaging systems. In *VLSI91*, 1991.

[6] J E W Mayhew S B Pollard and J P Frisby. A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[7] K Skifstad and R Jain. Range estimation from intensity gradient analysis. Technical Report CSE-TR-02-88, University of Michigan, 1988.

[8] R Y Tsai. A versatile camera calibration technique for high accuracy 3d machine metrology using off the shelf tv cameras and lenses. In *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1986.

[9] R Y Tsai. Review of rac-based camera calibration. *Vision*, November 1988.

[10] O. Vellacot. A framework of heirarchy for neural theory, chapter6: Numberplate recognition. In *PhD Thesis, Edinburgh University*, 1991.

Figure 7: Triple Stereo: Sequence 1
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm.
The focal length was 16mm and the resolution 256x256 pixels

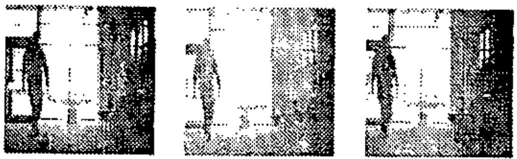


Figure 8: Triple Stereo: Sequence 2
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm.
The focal length was 16mm and the resolution 256x256 pixels

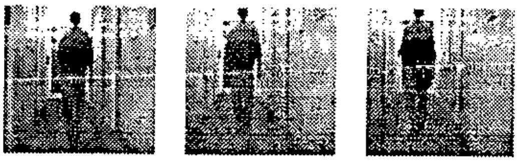
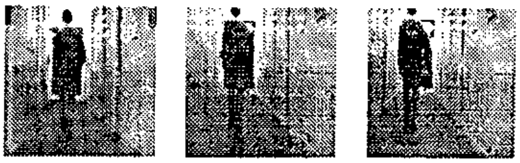


Figure 9: Triple Stereo: Sequence 3
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm.
The focal length was 16mm and the resolution 256x256 pixels



A STEREO ALGORITHM TO REDUCE QUANTISATION NOISE

K.W.J. Findlay, D.Renshaw and P.B. Denyer

Integrated Systems Group, Department of Electrical Engineering, Edinburgh University,
Edinburgh, U.K.
E-Mail:kevinf@ee.ed.ac.uk, Fax: 31 622 4358

Abstract

We aim to track, in three dimensions, humans and other moving objects using wide angle lenses. This has caused problems with quantisation noise which increases as focal lengths reduce. In order to control these errors the assumption, that all the extracted edges from the tracked object are at the same depth, is made.

An original stereo vision matching/segmentation algorithm has been developed which minimises the problems caused by quantisation noise in wide angle stereo ranging systems. It is intended that this algorithm be implemented in cost-effective hardware using recently developed CMOS cameras. It could have many applications in the areas of general tracking systems and passive alarms.

1 Introduction

Usually attempts are made to solve the stereo correspondence problem in its most general sense and build an all purpose stereo module. Possible matches are proposed for all parts or edges of the image. The above general approach is not necessary, in this and other applications.

We aim to track, in three dimensions, humans and other moving objects using wide angle lenses. This has caused problems with quantisation noise which are inversely proportional to the product of the focal length and interocular distance. In order to control these errors the assumption, that all the extracted edges from the difference image are at the same depth, is made. This is justified by the range in which we operate the system and by the fact that we do not require a complete 2 $\frac{1}{2}$-D sketch for the entire scene. Using this constraint a strict disparity gradient limit can be applied and statistics gathered for the "goodness" of a particular match. These are used to provide sub-pixel accuracy and compensate for the problems of quantisation noise. Edges are then grouped and an overall disparity extracted.

In the correspondence stage of the system we take advantage of the overlap which occurs between two views of the same object and match only outline edges. Outline edge detection and early segmentation are based on computationally simple, difference, and clustering techniques, combined with time domain information. This allows the elimination of low level multiplications, divisions and floating point calculations and makes a commercial implementation feasible, using ASIS imaging technology developed by Denyer et. al. [6]. ASIS technology allows implementation of camera sensor and processing, on the same chip, using low-cost CMOS fabrication.

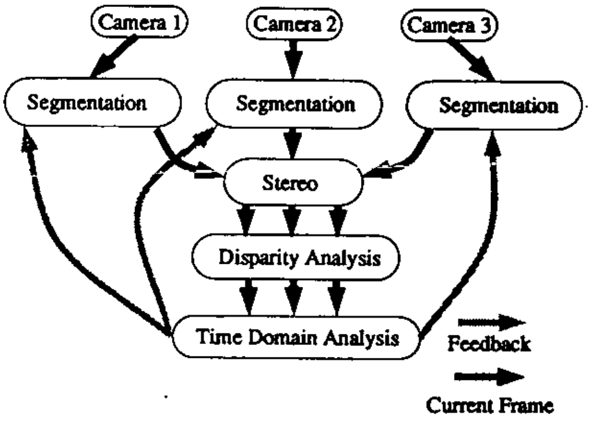


Figure 1: Overview of System Blocks

The structure of this paper will be a discussion of the important algorithmic points in section 2. Section 3 will provide the results of trials performed over different image sequences. Finally, section 4 will draw general conclusions.

2 The System

An overview of the entire algorithm is shown in Figure 1. As said in the introduction one of the aims of this work is an efficient implementation in hardware to allow commercial applications. Restrictions have been, necessarily, placed on the arithmetic allowed at the pixel level of representation. To this end, we have taken advantage of certain application specific features:

1. The vertical nature of a moving human allows edge detection to be restricted to a lateral scan across the image followed by downwards tracking. This simplification is also justified by the lateral separation of the cameras.

2. Only a disparity threshold is required to activate an alarm. Thus accurate camera calibration is not required. If the threshold is crossed, for a number of frames, the alarm can be activated.

Further simplification, of any future hardware implementation, has been achieved by employing only difference techniques at the pixel level. The multiplications, divisions and floating point calculations, associated with correlation based methods such as Canny [1], are too computationally expensive. The main problem, caused by reducing complexity at the lower levels of processing, is false edge generation. These will cause incorrect stereo matches. However it has been shown [5] that a disparity gradient limit is an effective control in determining correct matches between two views. Also inaccurate matches tend to become obvious over a period of time.

Edge information is combined with segmentation information from a clustering/thresholding algorithm which again does not employ the more computationally complex arithmetic described above. Thresholding is performed on differences between background and foreground images and is initially chosen to be some fraction of the mean. As time progresses, tracking confidence will increase, and the threshold can be altered. Grey level difference distributions, of the extracted connected regions, are then used to estimate an appropriate threshold. Thresholded regions are extracted on a nearest neighbour basis and used to group edges into relevant objects. This has the advantage that the overall system becomes system less dependent on one source of data. Also, we need only consider moving edges. Edge maps, based on previous frames, are maintained and used to eliminate stationary features. Overall the above segmentation techniques will almost always provide some part of the outline of a moving object in the scene.

Turning now to stereo matching, this is based on the simple fact that the images of an object in the scene can overlap if their local origins are aligned. This will occur if an object in the scene is wider than the interocular distance and the cameras are on the same imaging plane [2]. Once the candidate outline edges have been extracted it is a simple matter of scanning along a raster until the first edge in the other camera is found. The distance between these two edges is the disparity.

Obviously, the above form of matching requires knowledge of equivalent epipolar lines and rotational translations. Algorithms have been developed which attempt to find these corrections[8] [9] [3] and calibrate the cameras. In the equipment employed in the trials we are conducting rotation is not a significant factor. Thus, translational corrections, and the alarm threshold, can be calculated using a man walking about at a known distance. Different offsets can be attempted until consistency is achieved. We have taken advantage of the fact that only a disparity value need be crossed to activate the alarm. If this is consistently breached over a number of frames the alarm will be sounded.

2.1 Quantisation Error Control

Quantisation errors in stereo are inversely proportional to the product of the baseline and the focal length [7]. They are also inversely proportional to the range at which an object is positioned. The extent to which distance, focal length and interocular distance are significant depend, also, on the pixel size, and the RMS error, in position, for a single measurement is $\frac{P}{2\sqrt{3}}$ and for disparity $\frac{P}{\sqrt{3}}$. As we are using short focal lengths, quantisation errors are large. In addition, we must also expect the system to function at distances of up to 20 meters.

In this work an assumption is made that each matched edge is at one depth. We can use histogram analysis to form an estimate of the disparity of each edge. Edge location will tend to wind around its true location in the image, thus as an edge is tracked and matched we can build up a histogram of disparities. Examples of these are shown, plotted through time, in Figure 2.

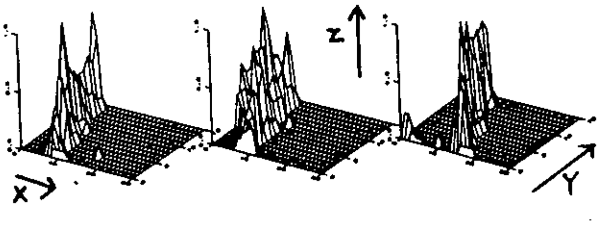
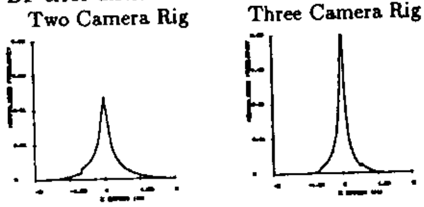


Figure 2: Disparity Histograms(X Disparity, Y Time, Z Frequency)

Figure 3: PDF error distributions

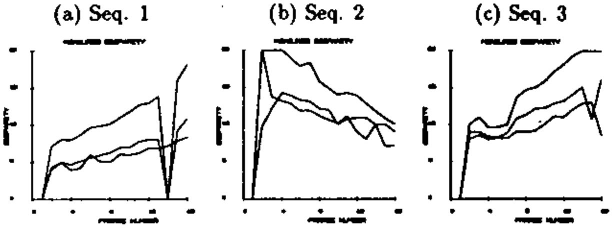


Three histograms are plotted for the three measurements possible from a triple camera stereo rig. The advantage of histogram analysis is that it reduces the effect of quantisation and provides a sub-pixel acuity estimate of the disparity.

At this stage the edges can also be segmented according to the disparity gradient limit [5] eliminating the vast majority of false matches. The mean and variance of relevant parts of the disparity histogram are then used as estimates of disparity and associated confidence. These confidences not only take into account the errors caused by quantisation but also those caused by inaccurate feature matching. This information can be utilised in any tracking filters which may be employed. Also it is, now, possible to calculate variances and disparities using floating point calculations. Data rates, at this level, are fairly low, for example, 20 edges per frame and calculations could be performed using simple microprocessors.

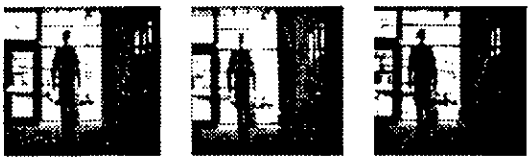
Three cameras have been used as a further attempt to reduce the expected error by averaging. Figure 3 shows the simulated depth error probability distributions for both two and three camera rigs. The three camera rig, where the disparity measurements are averaged before inversion to depth, has a clearly improved PDF. The simulations were generated, using the camera parameters described in section 3, from 100000 randomly generated points in a scene with a depth of 20 meters. It should, perhaps, be noted that the error probability functions vary with distance and that the p.d.f.'s shown in Figure 3 are for the complete simulated area. For example, the PDF between 15m and 20m will be flatter than the PDF between 5 and 10m. This factor will have to be taken into account when alarm thresholds are considered.

Figure 4: Disparity against Time (Frames)



(a) Seq. 1 (b) Seq. 2 (c) Seq. 3

Figure 5: Triple Stereo: Sequence 1
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels



3 Results

Examples are presented from a trial on fifteen image sequences captured in several different scenes. Each sequence was sixteen images long, captured from three CMOS cameras [6] of 256x256 pixels. The images were digitised to eight bits at five frames per second using in-house frame grabbers.

Figure 4(a) shows the results extracted from sequence 1, Figure 5, as a man walks towards the camera from 12m. Figure 4(b) is derived from sequence 2, Figure 6, as a man walks away from the camera. Figure 4(c) is derived from sequence 3, Figure 7, as a man walks initially parallel to the camera's image plane and then towards it. The top trace in all three graphs is the disparity extracted from the outer two cameras. The lower two traces are measurements from the inner pairings. Naturally, the outer measurement has a steeper gradient than that from the inner cameras.

In all sequences the human is detected and tracked through the scene. Inevitably there are frames when matching becomes difficult as the large spikes indicate. However they are very obvious in relation to the disparities extracted from the previous and next frames and would be controlled using tracking filters such as the Kalman formulation [2]. It can also be seen that, overall, disparity through time varies reasonably smoothly and does not jump as pixel quantisation boundaries are crossed. The gradient are steep enough to allow disparity thresholds to activate alarms.

Figure 6: Triple Stereo: Sequence 2
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels

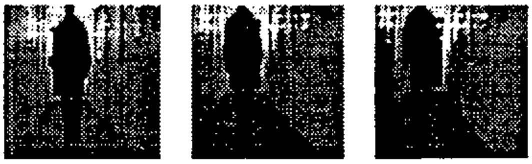
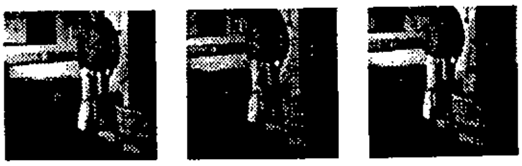


Figure 7: Triple Stereo: Sequence 3
16 Frames at 5 frames/second.
Each camera was separated by an interocular distance of 10cm. The focal length was 16mm and the resolution 256x256 pixels



4 Conclusions

An original stereo vision matching/segmentation algorithm has been developed which minimises the problems caused by quantisation noise in wide angle stereo ranging systems. It is intended that the algorithm be implemented in cost-effective hardware using recently developed CMOS cameras and has been optimised to this end. It could have many applications in the areas of general tracking systems and passive alarms. Further trials would allow the system to be tested in a wider context and allow incremental improvements as problems arise. This method of experimental design has proven its worth in systems described by Anderson [4] and Vellacot [10].

5 Acknowledgements

Thanks are due to the United Kingdom Science and Engineering Research Council for funding this project under Grant GR/F 36538 IED2/1/1159.

References

[1] J Canny. Finding edges and lines in images. *MIT AI memo*, 1983.

[2] K. W. J. Findlay, D. Renshaw, and P.B. Denyer. A low cost stereo alarm system for VLSI. In *Singapore International Conference on Image Processing*, 1992.

[3] H C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, September 1981.

[4] L M Ying P B Denyer, W Gouyu and S Anderson. On-chip cmos sensors for vlsi imaging systems. In *VLSI91*, 1991.

[5] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[6] D. Renshaw, P. B. Denyer, G Wang, and M Lu. ASIC Vision. In *IEEE Custom Integrated Circuits Conference*, 1990.

[7] J. J. Rodriguez and J.K. Aggarwal. Stochastic analysis of stereo quantisation error. In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 1990.

[8] R Y Tsai. A versatile camera calibration technique for high accuracy 3D machine metrology using off the shelf TV cameras and lenses. In *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1986.

[9] R Y Tsai. Review of RAC-based camera calibration. *Vision*, November 1988.

[10] O. Vellacot. A framework of heirarchy for neural theory, chapter6: Numberplate recognition. In *PhD Thesis, Edinburgh University*, 1991.