

**A SMART CMOS CAMERA for
AUTONOMOUS NAVIGATION SYSTEMS**

T.W.J. MOORHEAD

**A thesis submitted in partial fulfilment of the
requirements of Napier University for the
degree of Doctor of Philosophy**

March 2001

Acknowledgement

This thesis was made possible through the assistance of the academic and technical staff of the Napier University School of Engineering. In particular I would like to thank Dr. T.D. Binnie for his encouragement and valuable advice throughout my research work. I had the good fortune to share this research environment with Dr. H. Weller whose enthusiasm was an important source of inspiration. I should also like to thank Dr. H. Weller and Dr. D. Jackson for their corrections to this document. None of this would have been possible without the support and encouragement of my dear wife Ann.

Dedication

In fulfilment of a promise to E.R. Moorhead

Contents Index

Title Page	i
Acknowledgement	ii
Dedication	iii
Contents Index	iv
List of Figures	vii
List of Tables	xii
Abstract	xiii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Thesis Overview	4
Chapter 2 Autonomous Vision Systems Review	6
2.1 Introduction	6
2.2 Autonomous Navigation and Passive Vision	8
2.2.1 Navigation in Controlled Environments	8
2.2.2 Navigation in Uncontrolled Environments	9
2.2.3 Navigation in a Pedestrian Environment	11
2.2.4 Indoor Vision Based Autonomous Navigation	14
2.3 DSP Implementation of Vision Algorithms	17
2.3.1 Low-Level Processing Bottleneck	17
2.3.2 DSP Vision Processors	19
2.4 Neuromorphic Vision Processing	21
2.4.1 Neuromorphic Processors	21
2.4.2 Mead's Silicon Retina	21
2.4.3 Andreou and Boahen Spatial-Temporal Retina	23
2.4.4 CMOS Analogue Array Response Variation	25

2.5	Near Sensor Image Processors	28
2.5.1	Matrix Array Picture Processors	28
2.5.2	Mixed-Signal Array Processor	28
2.5.3	General Purpose Visual Computational Sensor	29
2.6	Conclusion	30
Chapter 3	SLA Algorithm Simulation	33
3.1	Introduction	33
3.2	The SLA Derivative Operators	37
3.2.1	1 st and 2 nd Order Sparse Convolutions	37
3.2.2	Averaged Sparse Convolutions	39
3.2.3	Derivative Sense Retention	40
3.3	Adaptive Thresholds and Edge Assignment	42
3.3.1	Adaptive Thresholds	42
3.3.2	Discrete Derivatives	43
3.3.3	Edge Point Assignment	44
3.3.4	Thresholds for Structural Edges	45
3.4	Post Edge Point Detection Processing	47
3.4.1	Test and Allocate Process	47
3.4.2	Pixel Count Threshold for Noise Removal	49
3.5	SLA Computational Requirements	51
3.6	SLA in Autonomous navigation	53
3.6.1	Integration of SLA into Pose Recovery Algorithm	53
3.6.2	Evaluation of Model Match Errors	55
3.6.3	Smart CMOS Camera Specifications	57
3.7	Conclusion	61
Chapter 4	Analysis of Edge Point Detectors	63
4.1	Introduction	63
4.2	Edge Point Metrics	65
4.2.1	Pratt Metric	65
4.2.2	Kitchen and Rosenfeld Metric	66

4.3	Edge Point Metric	69
4.3.1	Edge Detector Error Classification	69
4.3.2	DGC Algorithm Structure	70
4.3.3	DGC Phase 1	71
4.3.3	DGC Phase 2	72
4.3.3	DGC Phase 3	71
4.3.3	DGC Example Results	78
4.4	EPM Figure of Merit	81
4.4.1	Minimum Quality Specification	81
4.4.2	EPM Scale Factors	81
4.4.3	Figure of Merit Evaluation	83
4.5	Metric Results Edge Detector Comparisons	85
4.5.1	Synthetic Image for Metric Comparison	85
4.5.2	Qualitative Analysis of SLA, SUSAN and Sobel	87
4.5.3	Metric Comparisons	89
4.6	SLA Algorithm for NSIP implementation	92
4.6.1	SLA NSIP Test and Optimisation Synthetic Images	92
4.6.2	SLA Detector Filter Length of 4 Width of 3	95
4.6.3	SLA Detector Filter Length of 2 Width of 3	97
4.7	Conclusion	99
Chapter 5	Smart CMOS Camera Implementation	100
5.1	Introduction	100
5.2	CMOS Phototransduction Diodes	103
5.3	Pixel with Integral Gain	106
5.3.1	BJT Pixel	106
5.3.2	BJT Pixel Gain Analysis	107
5.3.3	BJT Pixel Gain Measurement	110
5.3.4	BJT Pixel Layout and Switching Circuit	112
5.4	Current Mode Processing in the SLA NSIP	117
5.4.1	Sub-Threshold Operation for Spatial Derivatives	117
5.4.2	1 st Order Spatial Derivative	118

5.4.3	Measurement of V_{AW} .	120
5.4.4	Feedback Circuit to Match Complementary Output Conductances	121
5.4.5	Measurement of the 1 st Order Contrast Sensitivity	123
5.4.6	2 nd Order Derivative Circuit	126
5.5	Three Layer SLA NSIP Edge Detector	128
5.6	Smart CMOS Camera Edge Detection Test	131
5.7	Power Consumption	
5.7.1	Vision Based Autonomous Navigation	134
5.7.2	CLPD Implementation of SLA Algorithm	135
5.7.3	Smart CMOS Camera Power Consumption	137
5.7.4	Comparison of Power Consumption Estimates	139
5.8	Conclusion	141
Chapter 6 Conclusion and Future Work		144
6.1	Conclusion	144
6.2	Future Work	148
References		150
Symbolic Terms		153
Glossary		156
Appendix A Example SLA Edge Detection Results		i-vi
Publications		

List of Figures

Figure 1.1	Staged Image Processing Structure	2
Figure 1.2	Smart CMOS Camera, Pixel Array and NSIP	4
Figure 2.1	DSP Implementation of a Vision System	17
Figure 2.2	Autonomous Vision Hierarchical Data Structure	18
Figure 2.3	Meads Silicon Retina.	22
Figure 2.4	Delbruck Mead's Adaptive Photoreceptor	23
Figure 2.5	Andreou and Boahen's Silicon Retina.	24
Figure 3.1	Overview of SLA Algorithm	34
Figure 3.2	SLA Algorithm Processing Structure	35
Figure 3.3	SLA derivative masks	
	(a) horizontal 1 st order convolution,	38
	(b) horizontal 2 nd order convolution,	38
	(c) vertical 1 st order convolution,	38
	(d) vertical 2 nd order convolution	38
Figure 3.4	SLA Derivative Masks Length =2, Width = 3	
	(a) Horizontal 1 st Order Convolution Mask	39
	(b) Horizontal 2 nd Order Convolution Mask	39
	(c) Vertical 1 st Order Convolution Mask	39
	(d) Vertical 2 nd Order Convolution Mask	39
Figure 3.5	Sense Information, SLA Length =2 Width=3	
	(a) Corridor Image	41
	(b) 1 st Order Vertical Derivative	41
Figure 3.6	SLA Directional Edge Sets	
	(a) Vertical Edges,	45
	(b) Horizontal Edges	45
Figure 3.7	Horizontal Edge Set Post Detection Processing	47
Figure 3.8	Test Pixel and Allocate Pixel Spatial Relations	48
Figure 3.9	Combined Faint Outline Horizontal and Vertical Edge Sets	
	(a) No Pixel Count Threshold,	49
	(b) Pixel Count Threshold Set to 40.	49

Figure 3.10	Geometric Model Comparison to Extracted Lines.	
	(a) Wide Angle Corridor View	54
	(b) The Model Estimate and Horizontal Edge Set	54
	(c) Boundary Model Estimate and Selected Lines	54
Figure 4.1	Kitchen-Rosenfeld ' k ' Values in 3x3 Neighbourhood	67
Figure 4.2	DGC Algorithm Decision Hierarchy	70
Figure 4.3	DGC Convolution Kernel	71
Figure 4.4	Kernel for Phase 2 Test 1	72
Figure 4.5	Kernel for Phase 2 Test 2	73
Figure 4.6	Kernel for Phase 2 Test 3	73
Figure 4.7	Kernel for Phase 2 Test 4	74
Figure 4.8	Kernel for Phase 2 Test 5	74
Figure 4.9	Kernel for Phase 2 Test 6	75
Figure 4.10	Kernel for Phase 2 Test 7	76
Figure 4.11	Kernel for Phase 2 Test 8	76
Figure 4.12	Kernel for Phase 2 Test 9	77
Figure 4.13	Kernel for Phase 3	78
Figure 4.14	DGC Algorithm results for three edge detectors.	
	(a) detector with systematic shift,	79
	(b) detector with line broadening,	79
	(c) detector with false returns.	79
Figure 4.15	Vertical Bar Synthetic Test Image.	
	(a) Vertical Bar no added Noise	86
	(b) Vertical Bar Noise $\sigma = 8$	86
	(c) Cross Section Profiles AA and BB	86
Figure 4.16	(a) SUSAN Detector results Noise $\sigma = 8$	87
	(b) SLA Detector results Noise $\sigma = 8$	87
	(c) Sobel Detector results Noise $\sigma = 8$	87
Figure 4.17	SLA Metric Results for Vertical Bar Image	89
Figure 4.18	SLA results Noise $\sigma = 10$ SNR=1.6dB	90
Figure 4.19	SUSAN Metric Results for the Vertical Bar Test Images	90
Figure 4.20	Sobel Metric Results for the Vertical Bar Test Images.	91

Figure 4.21	(a) Double Door Image	92
	(b) Section AA Narrow Groove Feature	93
	(c) Section BB Pair of Spread-Edges	93
Figure 4.22	(a) Narrow Feature Test	94
	(b) Spread Edge Test	94
Figure 4.23	(a) Narrow Feature Cross Section CC.	95
	(b) Spread Edge Cross Section DD	95
Figure 4.24	SLA Detector Filter Length of 4	96
Figure 4.25	SLA Detector Filter Length 2	98
Figure 5.1.	Smart CMOS Camera Block Diagram	100
Figure 5.2	Smart CMOS Camera Timing Diagram	101
Figure 5.3	CMOS diode structures	
	(a) N- well to P- substrate diode	103
	(b) N+ diffusion to P-substrate diode	103
	(c) N- well to P+ diffusion diode	103
Figure 5.4	(a) Cross Section of BJT Pixel	106
	(b) Equivalent Circuit for BJT Pixel	106
Figure 5.5	Simulated Gain for BJT Pixel Emitter $6 \times 6 \mu\text{m}$	109
Figure 5.6	Layout of the $120 \times 160 \mu\text{m}$ Diode Detectors	110
Figure 5.7	BJT Pixel Gain Measurement	111
Figure 5.8	Measured and Simulated Gain of a 120×160 BJT Pixel	113
Figure 5.9	Pixel Readout Switching Circuit	114
Figure 5.10	Pixel Access Timing Diagram	114
Figure 5.11	$80 \times 80 \mu\text{m}$ Pixel Layout	115
Figure 5.12	Current Mirror Implementation of 1 st Order Derivative	119
Figure 5.13	(a) V_{AW} Measurements 2 μA to 500nA	121
Figure 5.13	(b) V_{AW} Measurements 100nA to 10nA	121
Figure 5.14	Line Readout Current Mirror Circuit	123
Figure 5.15	Contrast Sensitivity Evaluation Circuit	124
Figure 5.16	Contrast Sensitivity of Current Difference Circuit	125
Figure 5.17	Frequency Response of 1 st Order Circuit	126
Figure 5.18.	2 nd Order Derivative Connection	127

Figure 5.19	SLA NSIP Edge Point Detector	128
Figure 5.20	Positive Threshold Comparator Circuit	129
Figure 5.21	Layout Overview of the Smart CMOS Camera Test Chip	131
Figure 5.22	Edge Detection Test Instrumentation	132
Figure 5.23	Results from the Edge Detection Tests	132
Figure 5.24	CLPD Implementation of SLA Edge Detector	136

List of Tables

Table 2.1	Percentage Change of I_{ST} as CMOS Parameters Vary by 10%	27
Table 3.1	SLA Algorithm Processing Requirements	52
Table 3.2	Initial Model Estimate Line Pairing Results	56
Table 3.3	Refined Model Estimate Line Pairing Results	55
Table 4.1	Phase1 Heuristic Truth Table	71
Table 4.2	Metric Comparison to Qualitative Results	78
Table 5.1	Currents Generated by CMOS 56x56 μ m Diode Structures	104
Table 5.2	Comparator Switching Thresholds	130
Table 5.3	Truth Table for SLA NSIP Layer 3 Edge Decision Logic	130
Table 5.4	Power Consumption of 100x100 Smart CMOS Camera	139

Abstract

In this thesis I present my research into the implementation of an edge point detection algorithm within a **Smart CMOS Camera**. The research includes the development and implementation of a new edge detection algorithm. The algorithm was designed for implementation in a **Near Sensor Image Processing (NSIP)** structure. This structure was integrated onto a CMOS substrate alongside a random access image-sensing array. The random access array employed pixels with integral gain. Operational specifications for the Smart CMOS Camera were derived from the spatial resolution, the frame rate and edge acuity, necessary to implement corridor autonomous navigation at a walking pace of 1m/s.

The architecture used to implement NSIP structure is referred to as the **Scanned Layer Architecture (SLA)**. This reflects the layered processing adopted to overcome the connection restrictions of the CMOS substrate. The new edge detector was labelled as the SLA detector. This detector was developed from a study of the gradient based edge detection algorithms. Its integration into a mixed signal CMOS processor was facilitated by limiting the spatial derivative convolution coefficients to integer values, and by minimising the number of product terms.

The SLA edge detector was designed to retain edge sense and edge direction information. Two directional edge sets were exported from each processed image. These were a vertical edge set and a horizontal edge set. Within these sets the edge information was encoded in a 3-state format to retain the edge sense information. A new edge point metric was developed for the quantitative assessment of the SLA algorithm results. This allowed the detector to be assessed against the requirements of a vision based navigation algorithm. Simulation results demonstrate the use of the SLA edge data to locate a robot's floor position within a corridor environment.

Chapter 1 Introduction

1.1 Motivation

The earliest fossil records, some 600 million years old, record the existence of flat worms and water invertebrates that employed light sensing spots to assist navigation. The worms still exist today and scientists have trained these worms to navigate on visual stimuli through a maze [1]. Biological evolution has brought us to the state where visual perception is the prime means that most earthly creatures use to find their food and avoid predators. These biological mechanisms have provided the inspiration for the development of artificial vision over the past 30 years.

The desire to create a machine that can perceive the world in a manner equivalent to human perception is the driving force behind this research into vision processing. Approximately 50% of the human higher-level brain functions are devoted to the processing of visual stimuli [2]. Given that the brain has the capacity to perform trillions of synaptic operations per second [3] there is no prospect of this structure being fully replicated in an artificial machine.

In the early 1970's the development of computing systems with memory capacity that was sufficient to store 2-D intensity profiles captured by imaging systems facilitated the first development of machine vision systems [4]. A critical aspect of this research, was the development of data structures that chart the sequence of transforms, needed to convert the data intensive sets generated through image capture, into the succinct scene descriptions. An example of a machine vision processing structure is illustrated in Figure 1.1 [5].

Figure 1.1 illustrates a staged processing structure. At the lowest level of the structure the sampled intensity profile is operated upon by domain independent, low-level processes. These assign edge and region primitives to each image sample, referred to as **picture elements (pixels)**. The primitive data sets occupy address spaces equivalent in size to the sampled intensity profile.

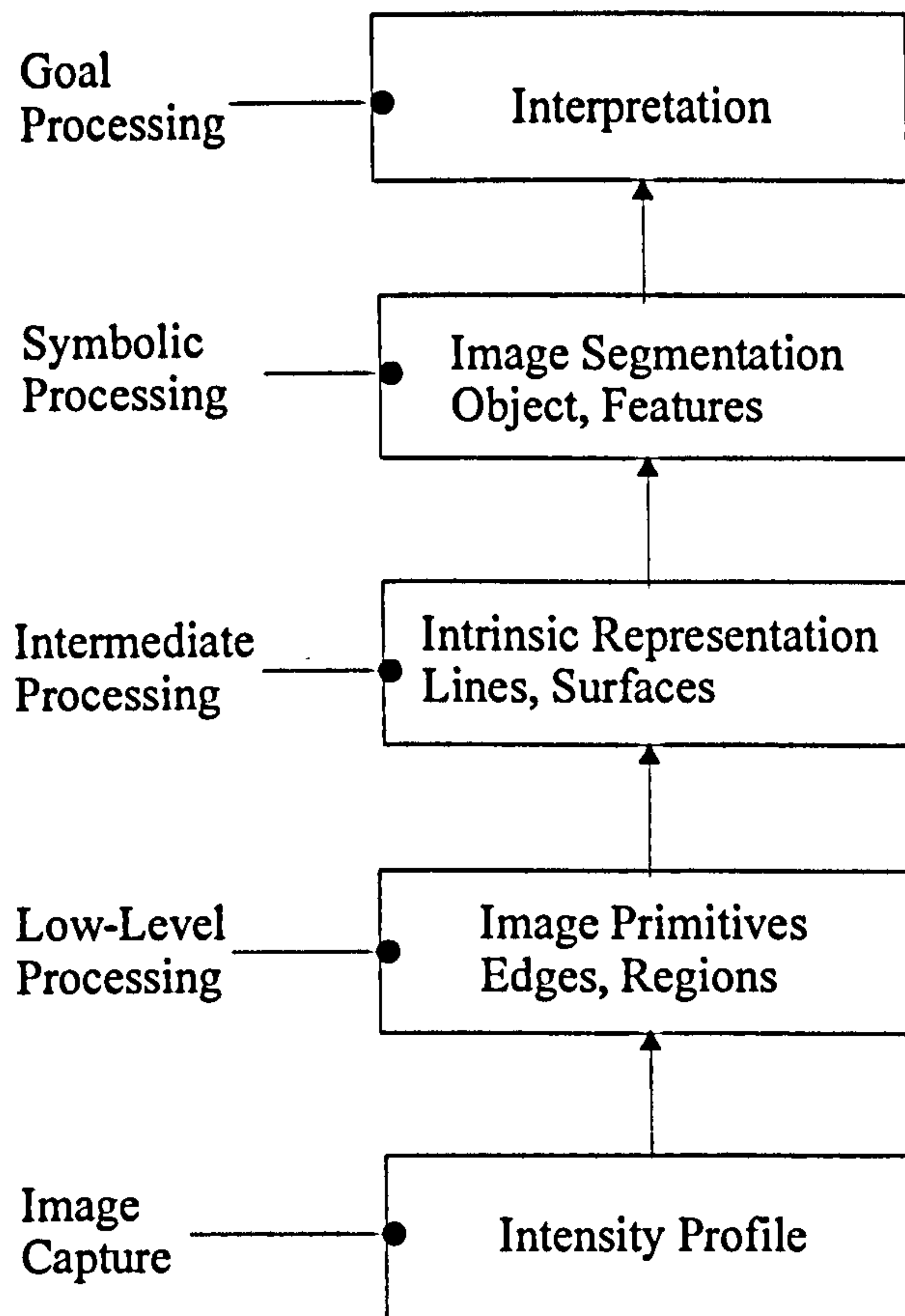


Figure 1.1 Staged Image Processing Structure

In the intermediate-level processing the primitive data sets are processed to give intrinsic representations of the object outlines and surfaces in the sampled intensity profile. In this intermediate stage transforms are applied to convert between data driven pixel assignments and symbolic vector assignments. The output set from the intermediate stage contains vectors that represent lines and surfaces detected in the sampled profile.

The vector sets created by the intermediate-level processes are passed onto the segmentation processes. Here the intermediate vectors are merged to form segmentation vectors that mark whole objects or major features in the image. Finally interpretation processes are applied to the segmentation results to identify objects and generate a scene description.

Current research into the development of vision based autonomous navigation demonstrates that the processing overheads associated with the low-level vision processes limits the practical implementation of autonomous robotic systems. The research reported in this thesis is specifically aimed at resolving the power consumed by the processing needed to implement low-level vision tasks.

1.2 Objectives

If an autonomous system is to navigate freely along a corridor or through a room it needs to carry energy cells with sufficient capacity to supply its locomotion and information processing. The power consumption of the onboard information processing is critical to the operation of autonomous systems. In a review of the state of the autonomous systems carried out in 1996 by Uhlin *et al* [6], it was noted that the means to implement visual perception in an energy efficient way is lacking. Uhlin explained that the causes of this deficiency are centred on a limited understanding of the visual perception models and the lack of energy efficient processing structure to deal with the high data throughput generated by image sensing cameras [7-10].

A review of autonomous systems and the processing resources needed for vision based navigation is given in Chapter 2. This confirms Uhlin's assertion that there is a need for a compact power efficient vision processor that can act as a front-end accelerator for vision systems. The research work detailed in Chapters 3 to 5 describes the algorithmic developments, and the **Complementary Metal Oxide Silicon (CMOS)** circuit designs needed to realise a front-end vision accelerator. The vision accelerator developed was called a **Smart CMOS Camera**. A partial overview of the layout for the Smart CMOS Camera is illustrated Figure 1.2.

The Smart CMOS Camera implements edge point detection at the pixel read rate, and is designed to supply edge point sets for the received image in real time. A layered processor structure is needed to implement the edge point detection. Three layers are identified. The first layer computes spatial derivatives for the pixels in a selected column within the array. In the second layer, the spatial derivatives are compared to

an externally supplied threshold to give a discrete derivative representation. In the third layer, logical processes are used to assign edge points to neighbourhoods of discrete derivatives. The architecture used for the Smart CMOS Camera was labelled as **Scanned Layer Architecture (SLA)**. This label reflected the fact that the image data was scanned out of the image-sensing array and processed through layered circuits that formed the column processor of Figure 1.2. These layered processing circuits form a **Near Sensor Image Processor (NSIP)**.

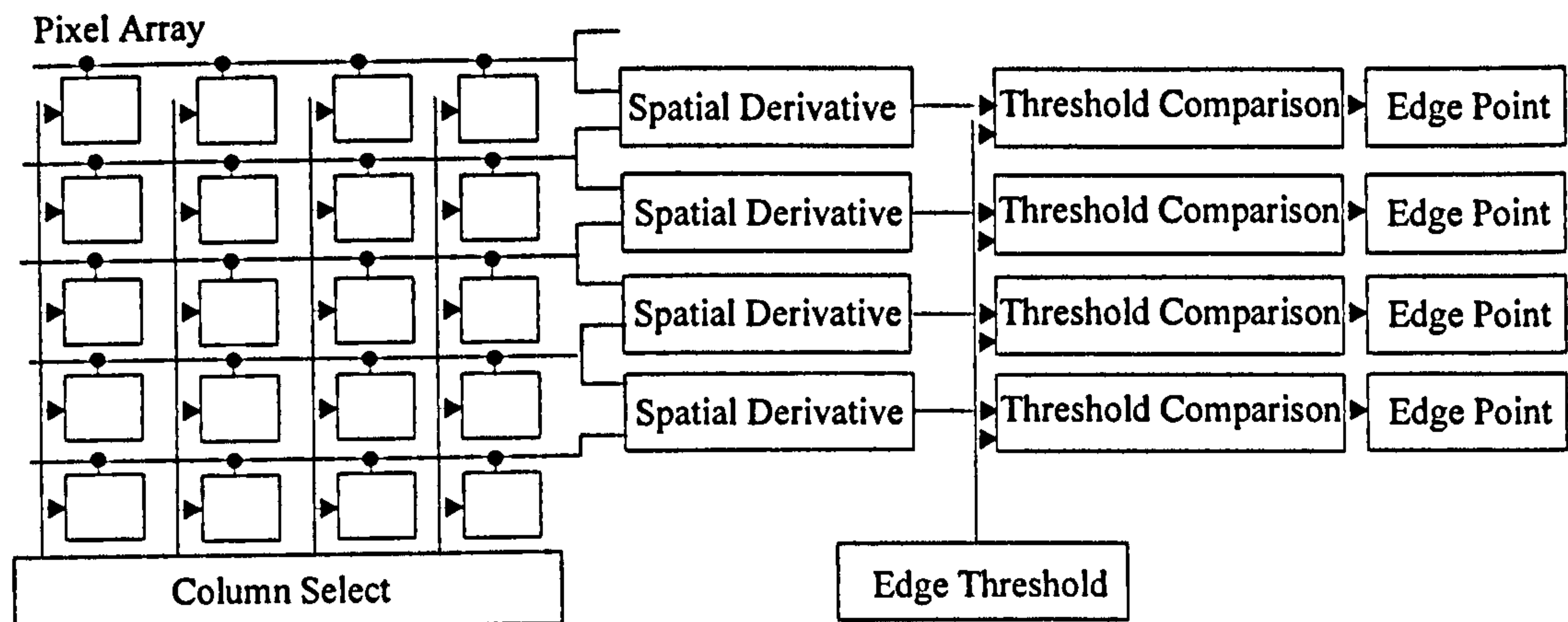


Figure 1.2 Smart CMOS Camera, Pixel Array and NSIP

1.3 Thesis Overview

The SLA NSIP edge detector was developed through image processing simulation described in Chapter 3. The edge detection and post edge detection processes developed through this simulation was labelled as the SLA algorithm. This new algorithm detected edges through parallel extraction of 1st and 2nd order spatial derivatives. It was developed from a study of gradient based edge detection algorithms [11,12,13]. In order to ensure that the algorithm could be realised within mixed signal CMOS environment, the spatial derivative convolution coefficients were limited to integer values, and the number of product terms was minimised [14].

The SLA algorithm simulation described in Chapter 3 includes post detection processes that extract line vectors from the edge point sets. A demonstration of

Beveridge's [15] pose recovery algorithm is used to illustrate how the line vectors can be matched to an environmental model. It is further shown how the recovered pose can be used in the implementation of navigation decisions.

In order to optimise parameter settings for the SLA edge point detection algorithm, and to test its performance against system level specifications a new **Edge Point Metric (EPM)** was developed. The implementation of this metric is described in Chapter 4. This metric was designed to embody Forstner's minimum quality specification [16]. This ensures that the metric is not limited to the quantitative comparison of detectors, but is of use in the selection and optimisation of a detector, for given vision system specifications.

Chapter 5 describes the circuits designed to implement the Smart Camera of Figure 1.2 on a CMOS substrate. This circuit implementation required the development of a random access pixel array and the design of a new contrast-sensitive current mode circuit. Results demonstrate the detection of edge points by a test implementation of the Smart CMOS Camera.

Chapter 2 Autonomous Vision Systems Review

2.1 Introduction

The general concept of an autonomous robot that can operate within the human environment and perform human type tasks provides the motivation for research into the field of autonomous vision. However, there is a problem, in that the energy consumed by current image processing systems is several orders of magnitude greater than that required by biological vision systems, whereas the acuity of these artificial systems is significantly less than that obtained from biological systems [6].

Section 2.2 reviews research into autonomous robots that depend on visual information for the implementation of navigation tasks. In these robotic systems there is a trade off between the degree of autonomy that can be obtained, and the electrical power required to process vision information. It is shown that if the robotic system is expected to operate in an office style environment then under current battery and processing technology limitations, it is not possible to implement autonomous operation. The operational systems reviewed used **Digital Signal Processing (DSP)** processing to implement their navigation decisions.

The implementation of vision processing through DSP systems is reviewed in Section 2.3. This establishes that for DSP a processing bottleneck exist in the low-level vision processes. The input data rate for low-level vision processes is set by the image sampling rate, as a result the low-level processing requirements are found to outstrip the capacity of current processor technology.

In order to address the low-level processing bottleneck Neuromorphic vision processors have been developed [14,17,18]. Research into neuromorphic vision processors is examined in Section 2.4. These devices seek to mimic the physiology of biological vision processors. They implement the low-level vision processing through a uniform array of processors with a processor assigned to each sample space in the imaging array. This is classed as massively parallel processing. A 100x100 imaging array will have 10,000 processors. In keeping with the biological model, and in order

to limit the substrate space occupied by the array processors, analogue circuitry is used to implement the low-level vision processes. However, the CMOS medium used to implement the neuromorphic devices exhibits significant variations in response across the analogue processing arrays [19]. This variation in response limits the practical application of the neuromorphic devices.

Near Sensor Image Processors (NSIP) have been developed to address the response variations in the neuromorphic arrays, whilst exploiting the light sensing properties and dense component integration features of the CMOS medium [20]. Like the neuromorphic arrays these devices are designed to implement the low-level vision processes. However, they differ in that they employ circuit implementations of algorithms previously developed for DSP implementations. A review of NSIP developments is given in Section 2.5.

2.2 Autonomous Navigation and Passive Vision

2.2.1 Navigation in Controlled Environments

Environments where navigational cues, such as visible flags are placed to assist autonomous systems in resolving their locations are classed as controlled environments. The marker flags are placed at predetermined locations within the navigation environment. The correspondence of the position of the flags with known locations in the robot's internal map are used to estimate the robots current location. On the basis of the current location estimate, future movement of the robot can be determined.

If the environment that the robot is required to operate within is primarily populated with fixed obstacles, and the illumination is controlled, then an autonomous system that relies upon flagged locations can be realised with current technology. The **Mobile Detection Assessment Response System (MDARS)** programme [21,22] was aimed at improving the effectiveness of unmanned security by deploying autonomous robots within a warehouse to detect intruders, fires and to monitor stock items. The MDARS robot navigates through a controlled indoor environment using optical tagging and sonar to assess its location and proceed with its patrol plan.

The MDARS project demonstrated that by exploiting sensor fusion techniques the processing burden of the robot can be minimised and practical service robots realised. It was reported that the MDRAS robot could navigate through a warehouse interior by using a vision system to detect reflective strips placed upon walls and shelving uprights. The reflective strips mark critical junctions. At these junctions the sonar system is used to evaluate the robot's location and determine its next movements. The MDRAS navigation processor is a Zilog Z80 and its navigation processing consumes approximately 70mA. The system was reported as capable of following a predefined patrol path for extended periods of operation.

The Artificial Intelligence Laboratory (AILab) at the University of Zurich has been investigating insect responses to visual stimuli [23]. They have sought to mimic these responses through robotic test beds with implementations of the compound eye.

Behaviour analysis of insects reveals that simple decision mechanisms explain their intelligent behaviour. The AILab robotic models limit the complexity of the on board processing circuitry by using relatively few light sensors.

The AILab have reported a visual homing-robot [23] that employs analogue processing to find its home location within a given space. This space has a set of visual cues or landmarks. These are observed from a ring of 32 photo diodes that are mounted on the robot. The robot responds to the observed cues by generating an **Average Landmark Vector (AL-Vector)**. The analogue circuit that implemented the AL-Vector processing used 91 op-amps and 12 analogue multipliers.

The robot is returned to its home location by comparing the current AL-Vector with a stored home location AL-Vector. This comparison gives a motion direction for the robot. Results show that the homing action returned the robot to within 68mm of the home position when the test environment was 1m square. If the home position is central to the 1m square then the 68mm error equates to a positional uncertainty of 14%. For an autonomous navigation system this is a relatively high level of positional uncertainty, but it is attributable in the homing-robot to the coverage of a 360° field of view with 32 sensors.

The MDRAS and AILab robots illustrate that for controlled environments with a known navigation map, low complexity decision processes can be used to give an approximation to autonomous activity. However, the decision processes that rely on a structured environment have more in common with machine vision systems than the versatile navigation operation expected from autonomous systems.

2.2.2 Navigation in Uncontrolled Environments

In an uncontrolled environment no flags or markers are added to the operational environment to assist autonomous systems in their navigation tasks. This section examines two autonomous vehicles that employ passive vision as the main source for their navigation information in uncontrolled environments. In these petrol-powered vehicles, the capacity to supply electrical energy to the navigation processing is

significantly greater than for the battery powered systems reviewed in Section 2.2.1. The navigation processing examined in this section relies upon visual information similar to that used by a human driver of the test vehicles.

In the Parma University ARGO project, a Lancia Thema 2000 car was converted into an autonomous vehicle by supplementing the manual controls with motor drives [24]. A pair of cameras mounted at the front of this car provided the navigation images. These navigation images were captured by a frame grabber board, mounted in a Pentium-I Personal Computer (PC). A **Generic Obstacle and Lane Detection (GOLD)** algorithm was implemented on this PC and control signals returned to the autonomous motor drives [25,26]. An override switch allowed a human supervisor to take control and operate the car as a normal road vehicle. The capabilities of this autonomous vehicle were demonstrated by its successful navigation of 2000km of Italian highways under normal traffic conditions.

The GOLD algorithm employed by the ARGO vehicle utilised an **Inverse Perspective Transform (IPT)** which was applied to both camera inputs. In the IPT images, the road surface acted as a ground plane. Translation and subtraction of the two IPT images returned pointers to obstacles on the road surface. Thus obstacle avoidance measures could be activated by the PC control program. One of the IPT images was further processed to register the road markings in a binary format. Morphological operators were employed to extract the tracks of the lane markings and this data was used to maintain the car position in the centre of the near side traffic lane.

The PC implemented the GOLD algorithm within 10ms. This low latency in lane detection processing allowed ARGO to travel at speeds of up to 140km/hr. There were two important factors in the delivery of the ARGO vehicle performance. The first was that the GOLD algorithm was coded in assembly language and it exploited the pipeline processes available on the Pentium processor. This maximised the usage of the processor capacity. Secondly, the GOLD algorithm focuses upon the critical information clues that are available in the IPT road-traffic scenes. This passive vision

system is limited by the restrictions imposed by the IPT and thus has limited applications beyond the detection of obstacles on a uniform tarmac ground plane.

An upgrade of the ARGO processor to a Pentium II processor permitted the inclusion of a stereo disparity algorithm without an increase in the processing lag [27]. The stereo disparity permitted accurate assessment of the distance to the other vehicles and thus enabled the autonomous vehicle to travel in convoy traffic conditions, at normal traffic speeds.

The Carnegie Mellon Robotics Institute has been researching the use of stereo vision for off-road navigation [28]. The research work was funded by the Suffield Canadian Defence Research Establishment. A cross-country vehicle has been equipped with a pair of stereo cameras that provide the prime range and obstacle information for the navigation computations [28,29]. Stereo disparity information is extracted from the camera data by a dedicated Pentium I processor. The images were sub-sampled and the disparity width was limited to ensure that the system could generate depth maps at a rate of 2Hz. A SPARC 20 processor, also mounted within the vehicle, provided the navigation processing. Under rough terrain conditions the vehicle travelled 200 meters in 6 minutes whilst avoiding 80 separate objects.

It was noted that the electrical energy required by the processors to implement visual based navigation in an uncontrolled environment, places a minimum size limitation on the host autonomous system. The workstation processing structure reported by the Carnegie Mellon Robotics Institute is unsuitable for integration into a battery-powered robot that could operate within an office environment. The PC based navigation processor reported by the ARGO team is suitable for integration into a battery-powered office style robot. However, the reliance of this system on the road texture and road markings for navigational cues limits the practical operation of this system when it is removed from the highway environment.

2.2.3 Navigation in a Pedestrian Environment

Autonomous systems that can operate in a pedestrian environment and interact with the humans are examined. These systems are required to operate using battery power, move at walking pace and extract their navigational cues from the positions of static objects, walls and doors. To implement this form of navigation they need a map of the operational environment. Furthermore they need an on-board electrical supply with sufficient capacity to provide locomotion and implement the processing necessary for human interaction and navigation.

The development of an autonomous robotic system that can interact with humans is of commercial interest. The operational principle of these systems is that, upon receipt of a command the robot will commence a task and require no further command input until the task is completed. The household applications for these robots include cleaning floors and monitoring the well-being of elderly people. Public applications include giving portage assistance to travellers in train stations, and for giving advice and guidance to visitors to exhibition centres.

The Minerva robot [30,31] was designed to interact with people in public spaces. It perceives its environment through cameras, laser rangefinders and ultrasonic sensors. This robot has been deployed in the Smithsonian's National Museum of American History to approach visitors, offer them tours and then lead them to the exhibits. Minerva maintains a sense of its location through a comparison of its assumed location with that derived from an analysis of a ceiling image acquired from an upward looking camera. Laser range scans give an alternative estimate of the robot's position. The two position estimates are compared and an aggregate position calculated. The use of an upward looking camera and the fusion of this information with laser range data provide a working autonomous navigation implementation.

The Minerva robot was an extension of the RHINO-Project [32] researched by the Institute of Computer Science III, University of Bonn into the synthesis of complex adaptive systems. The vehicle for the study was an autonomous mobile robot called RHINO. This robot was successfully deployed in the Deutsches Museum, Bonn. In

this environment, it guided hundreds of visitors through the museum during a six-day period.

The RHINO and Minerva robots demonstrated the feasibility of autonomous robots navigating within a pedestrian environment and interacting with humans. They have shown that it possible for a robot to map an environment, and then navigate through this environment providing a useful service to the public. These robots relied upon sensor fusion and active sensing to limit the complexity of their navigation algorithm [32]. The robots were battery powered and used three onboard PCs to process the active sensor data. A telemetry link provided access to off-board processing. In the case of the RHINO robot the off-board processing was used to implement stereo disparity evaluation [33]. Here, the camera data was first processed through a Datacube DSP system to detect the edge points within the images. This edge data was then communicated via a VME-S bus to a Sun workstation where a stereo disparity algorithm was used to extract depth information. The stereo disparity processing was performed on images sub-sampled to 244x58 pixels per image and processed at a frame rate of 4Hz.

The architecture for an office delivery robot was reported in 1994 by the Laboratory of Image analysis, Alborg University [34,35]. This architecture split the function of the robot into room, hallway, and door navigation. Each of these functions accessed navigation subsystems that included obstacle avoidance, a path finder and uncertainty management. For a given task, the sub-systems were used to compute a trajectory. This architecture avoided the high computational costs of vision-only processing through the use of sonar sensors to avoid obstacles and to follow walls. The wall following is particularly sensitive to process lags because a small error in the robots trajectory can give rise to wall collisions.

The designers of the Alborg, Minerva and RHINO systems used active sensing and off-board processing as a means of limiting the drain on the battery power capacity of the robots. If the active sensing was replaced by passive means then the systems processing requirements would be increased due to the greater complexity of the

perception algorithm. These robots highlight a significant deficiency in the current autonomous systems in that it is not practical to realise a system that can operate in a public space whilst relying solely upon passive vision for its navigation input.

2.2.4 Indoor Vision Based Autonomous Navigation

The discussion in Section 2.2.3 established that for autonomous systems the processing overheads associated with vision based navigation precludes the use of battery-powered pedestrian style robots. However, the ARGO implementation of Section 2.2.2 and the Minerva implementation of Section 2.2.3 demonstrate that significant processing efficiencies can be achieved if the navigation algorithm is designed to exploit structural features within the navigation environment.

In this section algorithms designed for the realisation of vision based indoor autonomous navigation are reviewed [36-39]. These algorithms are characterised by goal orientated behaviour. A typical goal would be the movement of the robot to a new room location. The robot's current location is determined through the sensing of landmarks and environment features. As the robot moves, these are tracked through local searches. By matching the landmarks with models of the environment, the trajectory is modified. In this, consideration is given to uncertainty of the perceived location of the robot. Landmarks that are critical to the navigation algorithms include fixed structural items such as doors, windows, and furniture. Important features of the navigation environment include the free floor space in the direction of travel and the distance to side obstructions. The robot needs to possess the manoeuvrability to pass all obstructions in the environment. It also needs to have a low latency decision process, so that it can correct for the uncertainty in its trajectory estimates and so avoid collision with structural features in the environment. The low latency decision process is also important if the robot is to avoid collision with other users of the environment.

The Active Vision methods proposed by Davison and Murray [36] employ an active stereo platform carrying two CCD cameras. The system chooses a set of high contrast landmark features. The pan and tilt functions of the camera platform allowed the robot

to maintain its fixation on the chosen features and thus track its own movement through the environment. Results from an implementation of this system demonstrated the robot moving at 20cm/sec when a single fixation point was used. PC processing was used to implement the vision functions. The stereo processing and the requirement to shift the camera pair between fixation points limited the practical operation of this robot. Davison and Murray's algorithm was implemented on a test bed that was joined through an umbilical to a static processing facility. Its use of depth perception to identify isolated key features was seen as a limitation of its practical implementation, as the stereo disparity computation is expensive in terms of processor capacity.

The autonomous navigation model proposed by Kosaka *etal* [37,38], employed a wire frame model of the environment that is matched with extracted features from a single camera mounted on a robot. A comparison between the model and extracted features allows the position of the robot to be estimated. Experimental results for this navigation method within a corridor environment demonstrated a correct location hit rate of 90%. In contrast to the Davison and Murray [36] approach, this method demonstrated that depth perception was not necessary for navigating an environment where model matches for the detected structural features were stored by the vision system. In the corridor example given by Kosaka [37,38], door uprights were marked by vertical wires and the floor to wall boundaries were marked by diagonal wires. Navigation processing was affected through matching these wires with structural lines within a 3D model environment. This approach to autonomous navigation is limited by the tolerance of the model matching process to environmental variations.

Gavriley *etal* [39] developed a model based navigation system that employed a single camera to collect image data that was subsequently processed to reveal the most significant gradients in a given scene. The direction and location of the gradients was used to infer the positions of doors and floor to wall boundaries in a corridor environment. This architecture proposes the use of a single camera to provide the visual information necessary for active perception. A hierarchical processing structure is used to match gradient profiles found within the observed scene to known features

within a visual data-base. A PC based implementation was used to demonstrate this architecture. The approach taken by Gavriley was similar to that proposed by Kosaka [37,38].

The navigation algorithms that rely upon the positional clues collected from a single camera require a model of the environment in which they operate. If this model is required to reflect the exact dimensions of the environment, then the system has a relatively low degree of autonomy. If the model is generalised to environment classes such as corridor, room and concourse then the robot can be said to have a high degree of autonomy. Systems with a high degree of autonomy need to build a map of the environment in which they are located. The subjects of map building and pose recovery from a single camera viewpoint are addressed by Beveridge [15]. He demonstrated that an iterative method of matching estimates of the floor and wall features to boundary lines extracted from captured scenes provided for the recovery of a robot's pose.

Indoor autonomous robot systems require battery power sources for locomotion and information processing. The demonstration models reported in this section were limited in their operation by either requiring external processing or needing long pauses in operation when updating their location estimates. In the following section the source of these processing limitations are examined.

Section 2.3 DSP Implementation of Vision Algorithms

2.3.1 Low-Level Processing Bottleneck

The autonomous robots reviewed in Section 2.2 used DSP methods for implementing their visual perception processes. A typical DSP implementation is composed of a camera, a frame-grabber and signal processing as illustrated in Figure 2.1 [40]. These DSP methods have been primarily developed for the implementation of machine vision systems. In the past twenty years DSP technology has evolved to provide an extensive range of object recognition systems that are employed in medical screening and industrial inspection.

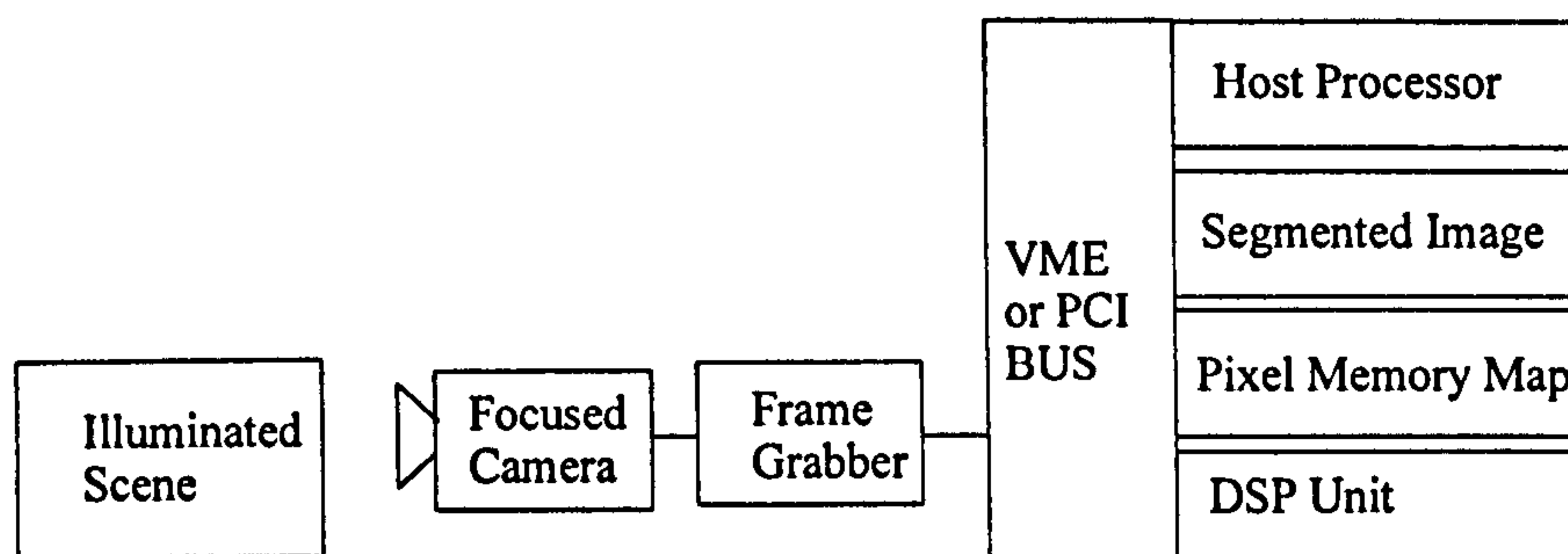


Figure 2.1 DSP Implementation of a Vision System

The frame grabber loads the serial stream of pixels generated by the camera into a memory mapped pixel array. This pixel array stores a two dimensional intensity profile of the illuminated scene. The DSP unit applies spatial filters and threshold functions to the pixel array to remove noise and generate a segmented image. In machine vision the segmentation processing requirements are minimised by maintaining a significant reflectance differential between the scene objects and the background.

In autonomous vision there is limited *a-priori* information about the operational environment. The illumination direction and magnitude are variables. The illumination can be from single or multiple sources with the magnitude for natural lighting ranging over 60dB [41]. The observed objects are presented with rotations

about the vertical axis, and at distances that range beyond the camera's depth of field. This variability in the presentation of the image data gives rise to the high processing requirements of the perception algorithms used in autonomous vision.

An example structure for an autonomous vision perception algorithm is illustrated in Figure 2.2. The algorithm employs a hierarchical data structure. In level '1' the segmentation primitives of edge points or textured regions are extracted. Also in level '1', optical flow may be extracted through the comparison of successive frames. In level '2' the segmentation primitives are then combined to provide partial object outlines, and depth information can be extracted from texture frequencies or pairs of stereo edge maps. In level '3' object recognition is implemented through vector matching and collision alerts are computed. At the top level '4', the navigation decision processes required by the autonomous systems are implemented.

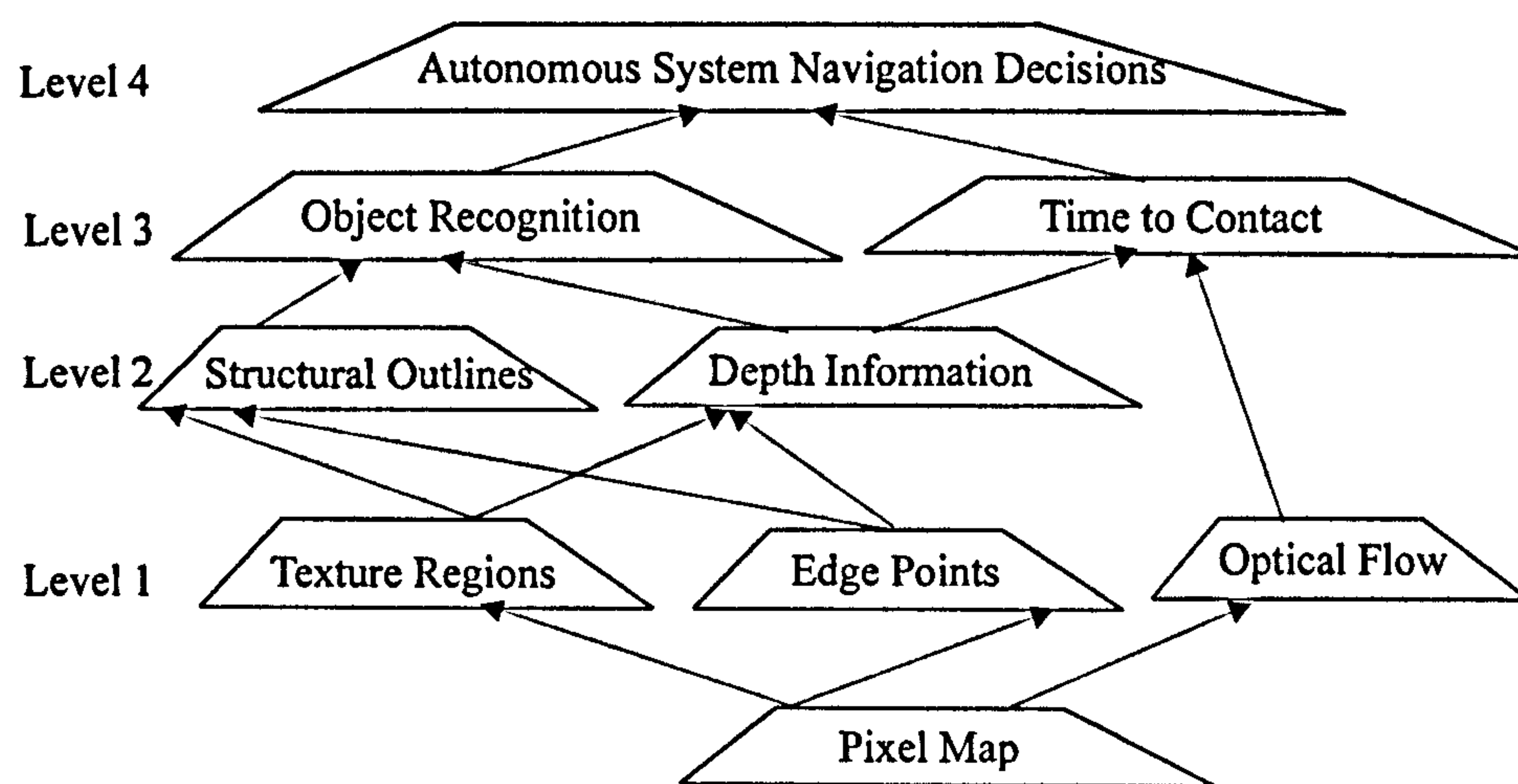


Figure 2.2 Autonomous Vision Hierarchical Data Structure

The processes in *Level 1* of the Figure 2.2, vision structure are classed as low-level vision processes [5]. These operate on each pixel location within the sampled image and give rise to high processing requirements. If the system processes the data from a 512x512 camera at 25 frames per second then the pixel data rate is 6MHz. The overall processor requirements are then evaluated as a multiple of this pixel data rate. For each processed pixel the area of pixels surrounding that processed pixel site are

accessed. Spatial averaging and spatial derivatives are applied to this accessed data. A low complexity pixel process such as the Sobel [12] detector requires 26 machine instructions per pixel giving a processor capacity requirement of 156×10^6 MIPS.

The quality of the segmentation data generated by the edge point detector is critical to the performance of the autonomous vision system. Erroneous or missing outlines will cause the objects to be wrongly classified and structural features to be missed, leading to incorrect navigation decisions. Noise from the sampling process and multiple path illumination of the object boundaries gives rise to uncertainty in the location of the object boundaries. In order to enhance the quality of the segmentation results the connectivity of the edge detector can be increased, but this increases the systems processing requirements.

An analysis of the processing requirements of autonomous robots, machine vision and image coding was carried out by Erten [3]. This analysis shows that the processing requirements for real-time autonomous vision are orders of magnitude greater than current DSP systems capability. Erten gives the example of a 2D correlation between two frames where a 7×7 pixel area in one frame is checked for the best match in a 21×21 pixel area in a second frame. If this process is repeated for each pixel location on 512×512 images at a framing rate of 30Hz then the processor requirements are 400 billion instructions per second. Erten points out that the solution to low-level vision problems is between three and four orders of processing magnitude beyond current DSP systems and thus argues that alternative processing solutions should be sought.

2.3.2 DSP Vision Processors

A technology leader in the supply of DSP machine vision systems is Datacube of Danvers, MA, USA. Datacube provides vision processing products to aerospace, defence and medical instrumentation markets. It manufactures VME and PCI boards which employ pipeline processing structures to perform vision processing [42]. In this structure, the vision task is split into a sequence of operations which are implemented through a series of processors that form the pipe. At a given instant, the pipe will be processing the data for several pixel locations. The time delay between

the results generated by the pipe is set by the longest process within the pipe. The processing of a typical set of 3x3 kernel convolution upon an image will run at a frame rate of 25Hz with a processing lag of 7 ms.

The Texas Instruments TMS320C80 Multi Video Processor (MVP) [43] provides an alternative to the Datacube pipelined systems. This MVP chip incorporates four parallel processors each of which has the facility to manipulate pixel data through arithmetic operations, bit field extraction and look up tables. A fifth master processor provides control of the four parallel processors. This chip provides the facility to apply standard integer based operations such as median filtering and Laplacian edge detection in real-time.

The vision processors supplied by Datacube and Texas Instruments are targeted at machine vision applications where the controlled environment eliminates the need for depth perception and structured lighting maintains a high contrast between the observed objects and the background. Under these conditions the integer based operations and small area convolutions of 3x3 pixels are sufficient to generate segmented image data.

Through advances in technology it is predicted that the capacity of DSP systems will increase and enhance the performance of the algorithms employed in the reviewed autonomous systems. However, it is worth considering that the low level processes employed in autonomous vision are relatively simple on an individual basis. The high processing requirements are derived from the large number of these processes that are needed to process an image frame. The implementation of these low level processes through massively parallel processors has been investigated through the development of neuromorphic systems. These are reviewed in the following section.

Section 2.4 Neuromorphic Vision Processing

2.4.1 Neuromorphic Processors

It has been observed that biological systems are much more efficient in their use of energy when processing visual information than the DSP approaches adopted in vision systems [17]. In order to exploit these observed efficiencies neuromorphic systems have been developed. In a neuromorphic system light sensing and early vision processing are performed by analogue circuits that mimic the cellular structures found in living creatures.

The image capture and the early vision processing circuits are combined on a single silicon substrate. This type of sensor has been referred to as a silicon retina [47], as a Focal Plane Processor [44,45] and as a retinomorphic sensor [46]. The biological functions that researchers have sought to incorporate into these sensors have included spatial enhancement, temporal displacement, lateral inhibition and sensitivity adaptation.

The principal medium used for the development of neuromorphic vision sensors has been VLSI CMOS. The VLSI CMOS substrate provides the opportunity of integrating a light-sensing array with analogue and digital circuits. The component packing density for a CMOS process is amongst the highest available and the foundry costs are not excessive. However the VLSI CMOS foundry processes have been developed for the production of discrete circuits which tolerate significant variations in device parameters. Analogue circuits are particularly susceptible to these variations in device parameters and this limits the performance of neuromorphic circuits.

2.4.2 Mead's Silicon Retina

The creation of a silicon equivalent of the biological retina, 'a silicon retina', was first proposed by Carver Mead in 1988 [47]. Mead's silicon retina was one of the first vision chips to implement retinal style processing on a VLSI substrate. This chip, illustrated in Figure 2.3, integrated light transduction and an array of early vision processing circuits onto a single substrate. The array processor employed a resistive network to simulate the spatial averaging associated with the retina horizontal cells.

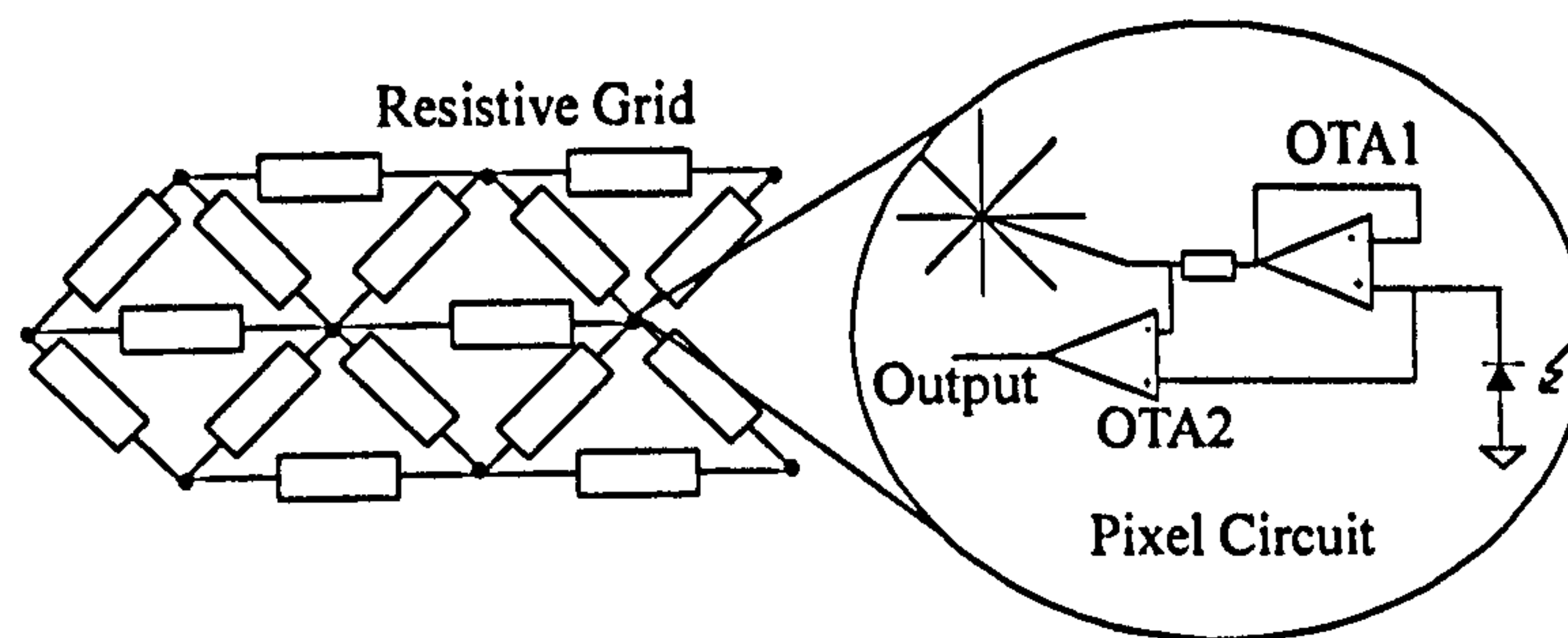


Figure 2.3 Meads Silicon Retina.

The illuminance sensed by each pixel is buffered by the Operational Transconductance Amplifier, *OTA1*, and summed into the resistive network. In the second amplifier, *OTA2*, the signal at the summing node is compared to the pixel luminosity signal to give the enhanced spatial contrast output. This comparison output is equivalent to the bipolar cell response found within biological retinas. In order to limit the power consumption of the array all the *OTA*'s are operated in sub-threshold mode. No images captured by Meads Silicon Retina have been found in the reported results, it was therefore concluded that the output generated by the silicon retina was of low quality.

This first silicon retina from Mead was designed to provide light transduction spatial contrast enhancement similar to that provided by the outer plexiform layers in a vertebrate retina. However the circuit was limited in the spatial resolution and by the variability in response across the array. Mead and Delbruck developed a time derivative pixel array [49-51]. This array provides temporal high pass filtering of the incident image and is designed for use as a pre-processor for a motion detection system.

The pixels illustrated in Figure 2.4, adapt to the slow variation in the background illumination in a manner similar to that found in biological retinas. Thus the full dynamic range of the sensor is available for the communication of movement. Light is sensed through a reverse biased diode *DI*. An active load *MI* is used to maintain the bias volts on the diode. Variation in this bias voltage is amplified through the cascode

circuit of $M2$, $M3$ and $M4$. This output voltage is feedback through a low pass circuit to control the active load. Results have been reported for single Adaptive Photoreceptor's but no results are reported for an array of these devices.

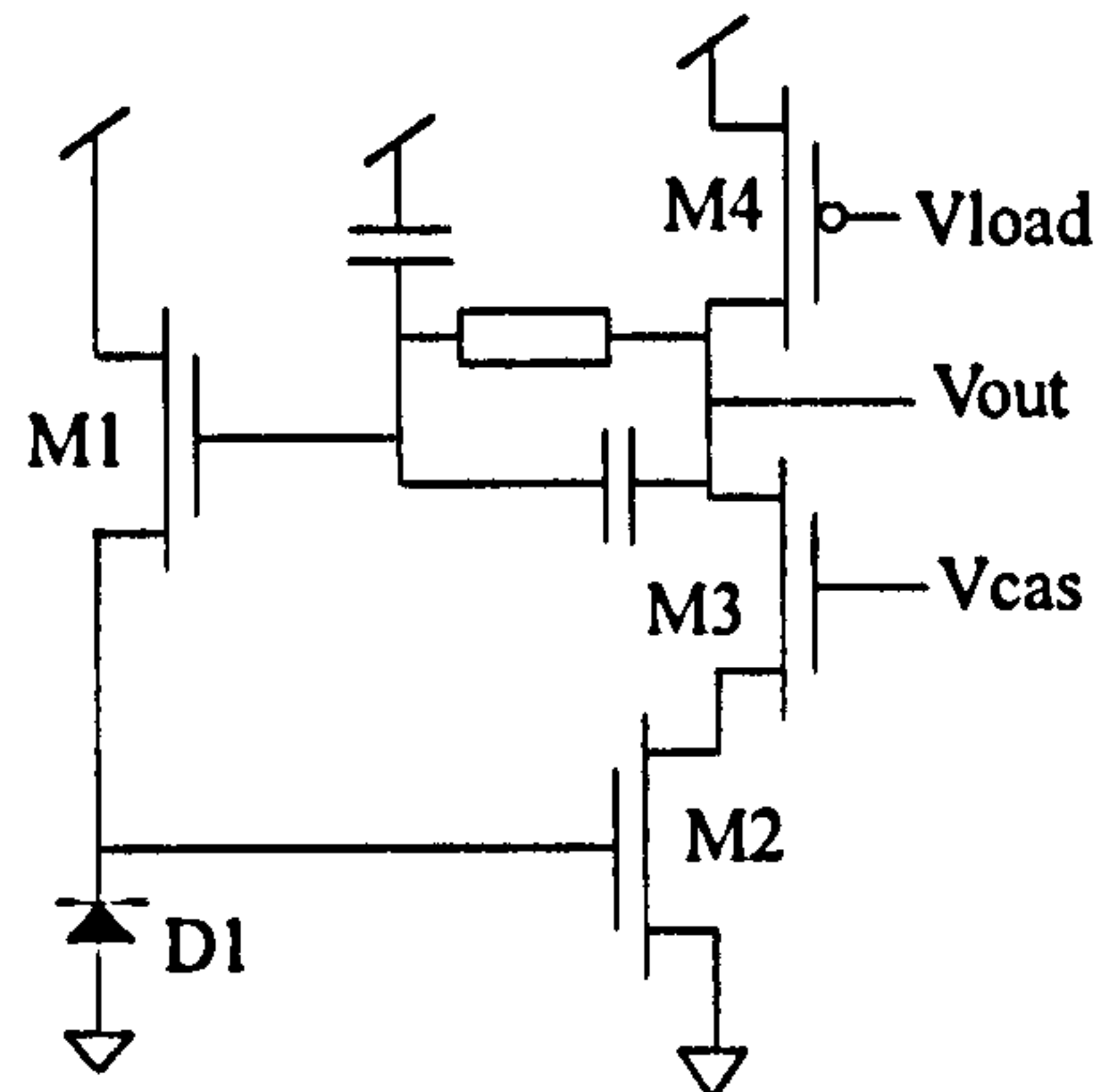


Figure 2.4 Delbruck and Mead's Adaptive Photoreceptor

2.4.3 Andreou and Boahen's Spatial-Temporal Retina

Andreou and Boahen developed a spatial temporal silicon retina [52,53]. This sensor exploits the native properties of sub-threshold CMOS circuits to realise an implementation of the vertebrate retina's outer plexiform layers. These are the layers of cone cells, horizontal cells, and bipolar cells [2]. The early vision spatial contrast enhancement function is achieved through low precision analogue circuitry.

The core cell structure of this silicon retina is illustrated in Figure 2.5. There are two separate diffuse networks that represent the cones and horizontal cells of a biological retina. The horizontal network implemented by the $M1$ transistors provides a wide area average of the sensed light and the $M2$ network provides a local average of the sensed light intensity. The $M3$ device compares the local to wide networks at each pixel site to give the output current I_{out} . The light transduction and its active load are provided by $M4$ and $T1$. A normalisation current is supplied into the I_{out} leg through $M5$. The control voltages V_n , V_c and V_h are globally supplied to the full array. The sensitivity of the array to local contrast is controlled by setting these control voltages.

The Andreou and Boahen spatial contrast retina has been implemented in an array of 210x230 pixels with the diffuse networks connected to six neighbours at each pixel

site. This sensor gives a uniform contrast sensitivity when there is a wide range of background illumination across the sensing area. Part of the image can be brightly illuminated and another part in deep shade. The sensor is designed to register contrast boundaries that occur in either region.

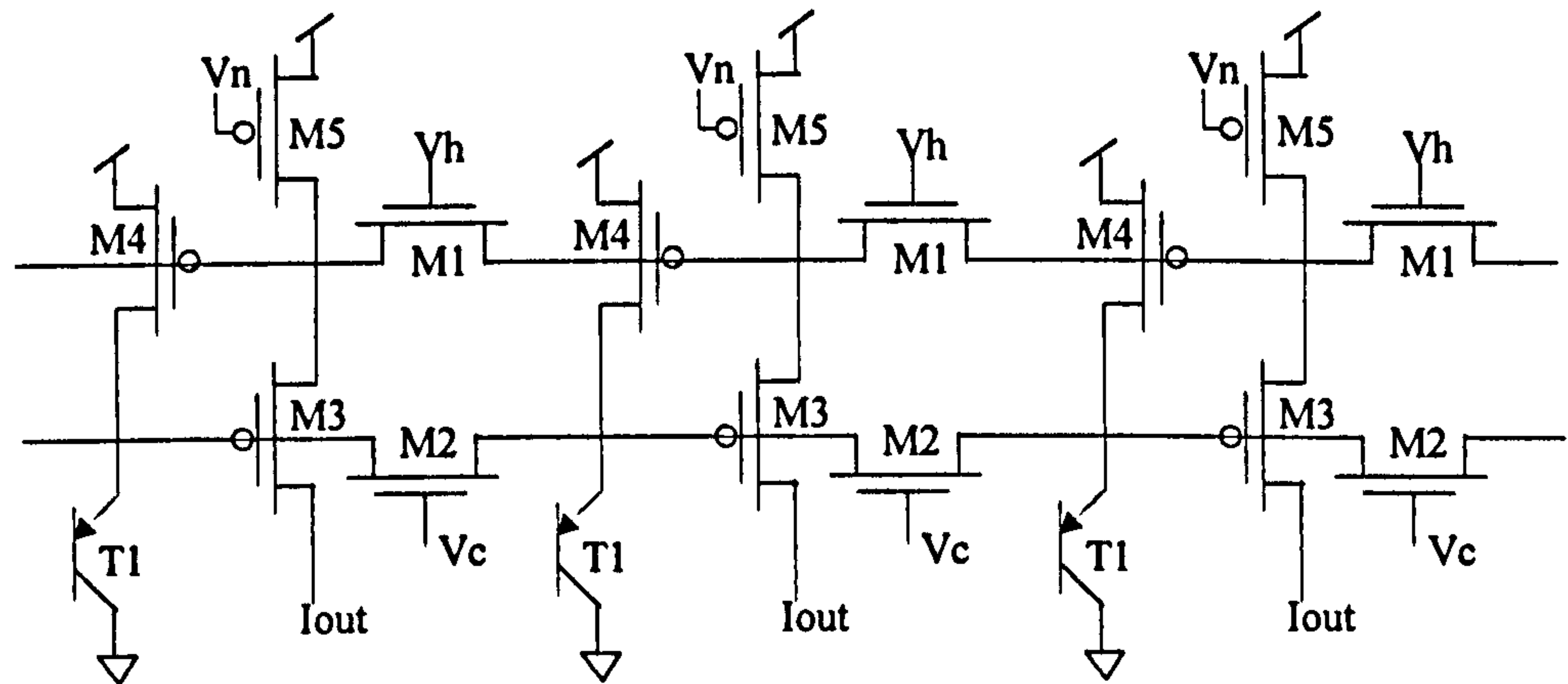


Figure 2.5 the Andreou & Boahen Silicon Retina.

The practical application of the Andreou and Boahen silicon retina was evaluated at Bonn University [54] using a real time face recognition system. The system employed a DATACUBE MaxVideo 20 pipeline image processor to interface the silicon retina to a host workstation which implemented a face matching algorithm. The predicted recognition time for a face from a database of twenty faces was three seconds. No results from an operational Andreou & Boahen silicon retina have been reported from this research.

There is currently ongoing research into neuromorphic sensors in a number of academic institutions [44,45,55-59]. The University of Seville has reported research on a mixed signal focal plane processing array [44,45]. Here the concept of a cellular neural network has been integrated into a 2D image acquisition array [55]. A neuromorphic linear sensor for visual tracking has been developed at the Institute of Neuroinformatics in Zurich [59]. This device employs the adaptive pixel developed by Delbruck. Analogue processing is applied to the pixel outputs to detect the location of edges on the linear array. The most significant of these edges is located through a winner-takes-all circuit. This discrete location within the array is then converted into

an analogue voltage. This device requires a high contrast level for successful edge detection and it has been incorporated into a track following robot. This type of device provides a means of reducing the computational load in a navigation or control system.

2.4.4 CMOS Analogue Array Response Variation

The practical use of neuromorphic sensors is limited through device parameter variations across an array of analogue processing circuits [19,60-62]. These variations are process dependent and remain an unresolved problem of the CMOS foundries. A study carried out by Pavasovic [19] where the sub-threshold current I_{ST} dependence on V_{GB} was measured for arrays of 1024 $4 \times 4 \mu\text{m}$ transistors illustrated this response variation.

It was reported by Pavasovic [19] that I_{ST} exhibited a 30% variation across the array when V_{GB} was an array constant. This variation exhibited a spatial period of between $100 \mu\text{m}$ and $200 \mu\text{m}$, which was labeled as a striation effect. In addition to the striation effect n-devices at the periphery of the array showed a reduction in I_{ST} of up to 15%. At the periphery of the array, p-devices showed an increase in I_{ST} of up to 50%. These are the combined results for tests on a total of 150,000 transistors. The above quoted percentage variations are the worst case results as I_{ST} ranged from 10pA to 100nA.

In the interests of the Smart CMOS Camera research a numerical study was used to investigate the possible causes of the I_{ST} variation reported by Pavasovic. The relationships which determine I_{ST} are given in equations (2.1) to (2.7) [63,64]. The sub-threshold operation of a MOSFET is defined as the region of operation where the surface potential ϕ_{sw} given by equations (2.2) varies between ϕ_f and $2\phi_f$, where ϕ_f is the Fermi level for the substrate and is given by equation (2.6). In this region the transconductance of the MOSFET reaches a maximum. The scaling current I_s given by equation (2.2), takes the Ebers-Moll form so that the MOSFET sub-threshold response resembles that of a Bipolar Junction Transistor.

$$I_{ST} = I_s \left(e^{-V_{SB}/\phi_T} - e^{-V_{DB}/\phi_T} \right) \quad (2.1)$$

$$I_s = \frac{\mu_n W C_o \gamma \phi_T^2}{2\sqrt{\varphi_{SW}} L} e^{\left(\frac{\varphi_{SW} - 2\phi_f}{\phi_T}\right)} \quad (2.2)$$

$$\varphi_{SW} = \left[-\frac{\gamma}{2} + \left(\frac{\gamma^2}{4} + V_{GB} - V_{FB} \right)^{1/2} \right]^2 \quad (2.3)$$

A number of the parameters in equations (2.2) and (2.3) are directly affected by the CMOS processes. The capacitance C_o given by equation (2.4) is determined by the channel oxide diffusion depth x_o . The body factor γ given by equation (2.5) is determined by C_o and by the substrate doping concentration N_{sub} . The Fermi potential ϕ_f given by equation (2.6) is also dependent upon N_{sub} . The flat band voltage V_{FB} given by equation (2.7) is dependent upon the parasitic charge Q_o . This parasitic charge is due to a combination of the surface states and the trapped charge in the oxide layer.

$$C_o = \frac{K_o \epsilon_0}{x_o} \quad (2.4)$$

$$\gamma \equiv \frac{\sqrt{2qK_s \epsilon_0 N_{sub}}}{C_o} \quad (2.5)$$

$$\phi_f = \phi_T \ln \frac{N_{sub}}{n_i} \quad (2.6)$$

$$V_{FB} = \phi_{MS} - \frac{Q_o}{C_o} \quad (2.7)$$

Silicon foundries will not divulge parameters such as the doping concentrations or the variation of these parameters in their manufactured devices. In order to evaluate possible causes of the 30% variation in I_{ST} reported by Pavasovic [19], it was decided to numerically analyse the dependence of I_{ST} on x_o , N_{sub} and Q_o . In this analysis these process parameters were individually allowed to vary by 10% and the resultant variations in I_{ST} are noted Table 2.1. The dependence of I_{ST} was repeated for four settings of φ_{SW} in the sub threshold region. The φ_{SW} settings and the associated nominal values for I_{ST} are given in the first two rows of Table 2.1. The results indicate

that the nominal 10% variation in any of these parameters will result in a variation in I_{ST} greater than the 30% variation reported by Pavasovic [19].

Nom. I_{ST} (A)	2.33×10^{-12}	6.28×10^{-10}	1.12×10^{-8}	1.5×10^{-7}
ϕ_{sw}	$1.01\phi_f$	$1.52\phi_f$	$1.78\phi_f$	$2.01\phi_f$
N_{sub}	-34.7%	-49.3%	-42.8%	-44.9%
x_o	-39.3%	-50.5%	-55.5%	-59.7%
V_{FB}	-36.3%	-38.1%	-38.7%	-39.1%

Table 2.1 Percentage Change of I_{ST} as CMOS Parameters Vary by 10%

Pavasovic indicated that the striations in the I_{ST} response correlated with the surface preparation of the substrate prior to the implementation of lithographic processes. The level of response variation across the presented neuromorphic arrays is an important factor in the poor uptake of these substrate based mixed signal processors. Evidently there is a need for further investigation of these response variations in CMOS processes. However the commercial sensitivity of the foundry processes means that such an investigation may only be carried out under the auspices of an interested foundry.

The neuromorphic substrate based processor has a relatively low density of cell interconnections when compared with biological cellular structures. The processors realised in the planar VLSI environment are limited to forming connections with the neighbouring cells. Thus the pixel connectivity is limited to connections between four or six local pixels. This level of connectivity cannot replicate the connectivity that is found in 3-D biological structures where the neurons are widely connected. The result is that it is not practical to implement the full cell structure found within a vertebrate retina on silicon. Neuromorphic research does however provide a valuable insight into biological processing structures. It is worth noting that these biological structures have been evolving since the earliest appearance of vertebrate life on earth.

Section 2.5 Near Sensor Image Processors

An alternative to the focal plane processing adopted in neuromorphic processors is that of Near Sensor Image Processors (NSIP). As with the neuromorphic devices the image is sensed through a photo sensing array formed on a CMOS substrate. In the NSIP devices a mixed signal processor is placed on the substrate adjacent to the image sensing array. The image data in analogue format is loaded into the processing circuits where the low-level vision tasks are performed.

2.5.1 Matrix Array Picture Processors

A series of NSIP devices have been developed jointly by Linkoping University and Integrated Vision Products Inc (IVP) [20,65-67]. The IVP **Matrix Array Picture Processor (MAPP)** combines an image sensor and a general purpose image processor on a single substrate [68]. This sensor applies adaptive thresholds to the received image and programmable logic circuits to process the received data. The MAPP sensors have been successfully integrated into web inspection and process control machine systems.

The research at Linkoping University pioneered the development of NSIP devices. They have sought to overcome the trade-off between spatial resolution and processor complexity through non-destructive pixel readouts and local processing. They limit the processing complexity by applying thresholds to the analogue pixel outputs to generate a binary readout from the array. They have exploited the local connectivity in the parallel array readouts to realise high-speed image processing algorithms required in machine vision. The MAPP sensors are of limited use in autonomous vision applications because they require structured illumination to successfully implement their segmentation functions.

2.5.2 Mixed-Signal Array Processor

A variation of the NSIP theme was reported by Martin *et al* [69] where the early vision tasks were implemented in a mixed-signal array processor. This processor was formed from an array of cells each of which utilises a programmable analogue

arithmetic unit. The arithmetic unit employs digital conversion to perform addition, subtraction and multiplication. Each cell within the array is independently programmed to give a multiple-instruction, multiple-data MIMD processor. The analogue pixel data is read into the first column of cells in the array. In each cell the analogue signal is converted to digital format for processing and reconverted to analogue format to be passed to the next column of cells in the processor array. The multiple conversions between analogue to digital formats limits the quality of the low-level vision processes.

2.5.3 General Purpose Visual Computational Sensor

The Sensory-Motor-Systems Laboratory at the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, has been researching a **General Purpose Visual Computational Sensor (GPCS)** [70]. In the GPCS spatial processing and temporal processing circuits are integrated into a NSIP structure. The GPCS also includes analogue to digital conversion for the pixel outputs and global pixel gain control.

The GPCS employs current mode processing at the pixel level. Each pixel presents multiple current mode outputs, which are selectively summed through a set of nodes that form a spatial convolution mask. The programming of the convolution mask allows the GPCS to implement a series of vision convolution algorithms. The convolution algorithm can be set to be a pair of orthogonal Gabor filters, a smoothing filter, a Laplacian edge detector or a pair of directional edge detectors.

The stated objectives for the development of the GPCS is that it should be integrated with other intelligent systems such as neural networks or expert systems to provide VLSI real time solutions to dynamic vision tasks. These dynamic vision tasks can range from video coding to autonomous vehicle navigation. The performance of these vision tasks is set by the quality of real time edge detection provided by the GPCS [71]. It was noted from the reported results that the GPCS required a high level of contrast to register an edge in its processed images.

2.6 Conclusion

The selection of robot research programmes reviewed pointed to the broad scope of the autonomous navigation problem. It was illustrated that the nature of the solution was dependent upon the level of structure in the navigation environment and the electrical power available to process the vision data. As the level of structure was reduced the processing requirements increased and thus the power consumption of the robot increased.

It was evident from the review that the realisation of an autonomous vision system that can mimic human visual acuity and operate from a mobile, pedestrian style platform is still an open problem. A critical deficiency with the current technology is the lack of an energy efficient method of performing low-level vision tasks. Designers of autonomous vision systems have to make a trade off between the energy consumed by the system and the quality of vision processing that they employ.

In the reviewed systems, the autonomous vehicles reported from Parma University and Carnegie Mellon University employed passive vision to implement their navigation algorithms. These systems could draw on electrical power generated by the vehicle alternators. The reviewed systems that depended upon battery power such as the Minerva and RHINO robots resorted to sensor fusion and off-board processing to implement their navigation algorithms.

The Carnegie Mellon cross-country vehicle was considered the most complete autonomous system in the review and with an onboard workstation it was limited to a 2Hz framing rate. These limited framing rates, despite the use of significant processing resources, are typical of the compromises that autonomous systems designers need to make in order to implement an operational system. The ARGO road vehicle demonstrated that exploitation of the dark road colouring considerably reduced the computational requirements, and that a 25Hz framing rate was necessary to mimic human road control activity.

The reviewed systems that achieved autonomous operation relied upon image capture and DSP processing similar to that found in machine vision systems. However, the low levels of scene structure available for autonomous vision operation significantly increases the complexity of the vision algorithm and hence the processing requirements. In order to overcome these limitations neuromorphic sensors that mimic the cell structure of vertebrate retinas have been developed.

Research into neuromorphic vision systems capable of realising autonomous operation has been inspired by studies of biological vision systems. This research has been ongoing over the past ten years. In this, engineers seek to mimic on silicon circuits the cell structures and neural processes found within biological retinas. In the vertebrate retina a dense 3D cell structure implements the early vision tasks of light transduction, spatial contrast enhancement and motion detection with a fraction of the energy required by a DSP system.

Neuromorphic researchers have exploited the high integration density of VLSI CMOS to implement up to three layers of retinal processing at each pixel site in an imaging array. They have demonstrated the replication of light transduction, spatial contrast enhancement and motion detection within a single silicon retina. The robustness of these processes is limited by the planar nature of the substrate upon which the processing circuits are formed.

In order to overcome the limitations of the VLSI planar environment, an approach known as the Near Sensor Image Processor (NSIP) has been adopted for the realisation of a retinal equivalent sensor. In NSIP research the low level vision tasks are implemented through mixed signal processing circuits that are sited adjacent to the image sensing array on the CMOS substrate. In a NSIP device the image is read from the sensing array and loaded directly into the processing circuits. This architecture allows the spatial connectivity of the sensor to be extended beyond that found in neuromorphic structures.

The NSIP and neuromorphic sensors offer a solution to the size and power consumption problems associated with DSP implementations. The quality of the early

vision processing provided by these sensors is inferior to that provided by DSP vision systems. These quality problems arise from the CMOS fabrication technology where process parameter variations give rise to noise in the integrated analogue processes. This noise limits the usefulness of these sensors. Hence there is a need for refinement of the CMOS foundry process before these analogue and mixed signal solutions can replace the current DSP implementations with equivalent quality in vision processing.

It was concluded from the review that given current technology limitations, the research into a vision system front end accelerator should adopt a holistic approach to development of an accelerator for vision based navigation. In this the data structure needed for navigation, the quality of image primitives used by this structure and a sensor capable of delivering the image primitives should be examined. Thus the research proceeded on three fronts. These were the development of the development of an edge detection algorithm suitable for integration into a NSIP structure. The analysis of this detector's results with respect to the requirements of vision based navigation. The implementation of this detector within a compact, low power consumption, camera.

The following Chapters detail the design and testing of a Smart CMOS Camera for use within autonomous and machine vision systems. The prime requirement for this smart camera was the generation of edge points sets for the captured images. These edge points sets were to be of sufficient quality to permit autonomous navigation to be realised. This research draws on the neuromorphic work and on the established computational methods employed in machine vision. The smart camera employs a new mixed signal processing architecture referred to as Scanned Layer Architecture (SLA) which is aimed at overcoming the tradeoff between spatial resolution and noise susceptibility of CMOS analogue processing.

Chapter 3 SLA Algorithm Simulation

3.1 Introduction

The review of autonomous vision systems in Chapter 2 demonstrated that current robotic systems are limited in their application because of the need to employ high-speed processors to implement low-level vision tasks. Designers of autonomous vision systems are forced to compromise between the quality of edge point data that they can extract from the received images and the energy consumed by the processing required to implement the edge extraction. In order to address these compromises between quality and processing power the Smart CMOS Camera was developed. This is a VLSI CMOS sensor designed to combine an imaging array with the processing necessary to implement edge point detection. The SLA edge detection algorithm developed for integration into this sensor is described in this chapter.

The block diagram of Figure 3.1 illustrates the major processing stages and the memory blocks required by the SLA algorithm. Furthermore Figure 3.1 illustrates the integration of the SLA algorithm into a pose recovery algorithm. The layout constraints imposed by the CMOS implementation meant that the SLA algorithm was required to process the received images through two orthogonal scans. These have been labelled as the *Horizontal* and *Vertical* scans, each of which produces its own directional edge map for the image. These edge maps are operated on by *Post Detection Processes* to generate *Horizontal and Vertical line lists*. The list contents have a vector format and they provide the primitive structural outline information needed to recover the robot's pose, through a geometric model matching algorithm [15].

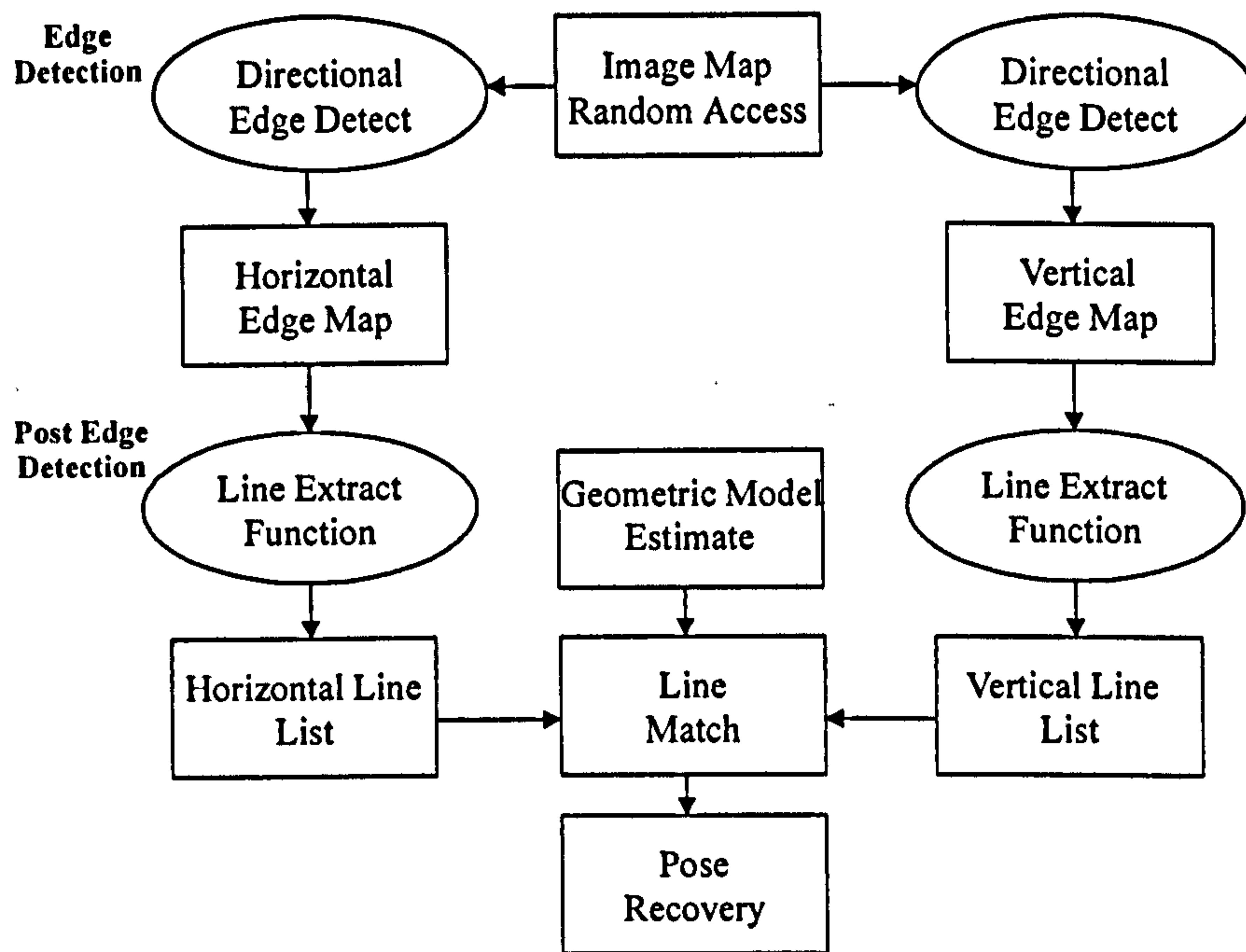


Figure 3.1 Overview of SLA Algorithm

The SLA directional edge point detector locates edge points through a distributed decision process illustrated in Figure 3.2. This process is initialised by the application of 1st and 2nd order spatial derivative convolutions to the sampled intensity profile. Adaptive thresholds are computed for each pixel location. These are used to convert the spatial derivatives into a discrete format. An area based logical operation is then applied to the discrete derivative results to assign the image edge points. The spatial derivative convolutions are described in Section 3.2. The adaptive threshold evaluations and the edge decision logic are described in Section 3.3. The Post Detection processes are described in Section 3.4

Although the SLA simulations were primarily aimed at developing an algorithm for integration into a CMOS VLSI, it was evident from the simulation results that the algorithm also represented an efficient DSP implementation. In Section 3.5 the computation resources necessary to implement the SLA edge detection and post edge detection processes are examined. It is shown that a real time implementation of the edge detection and line list generation is viable with current processor technology.

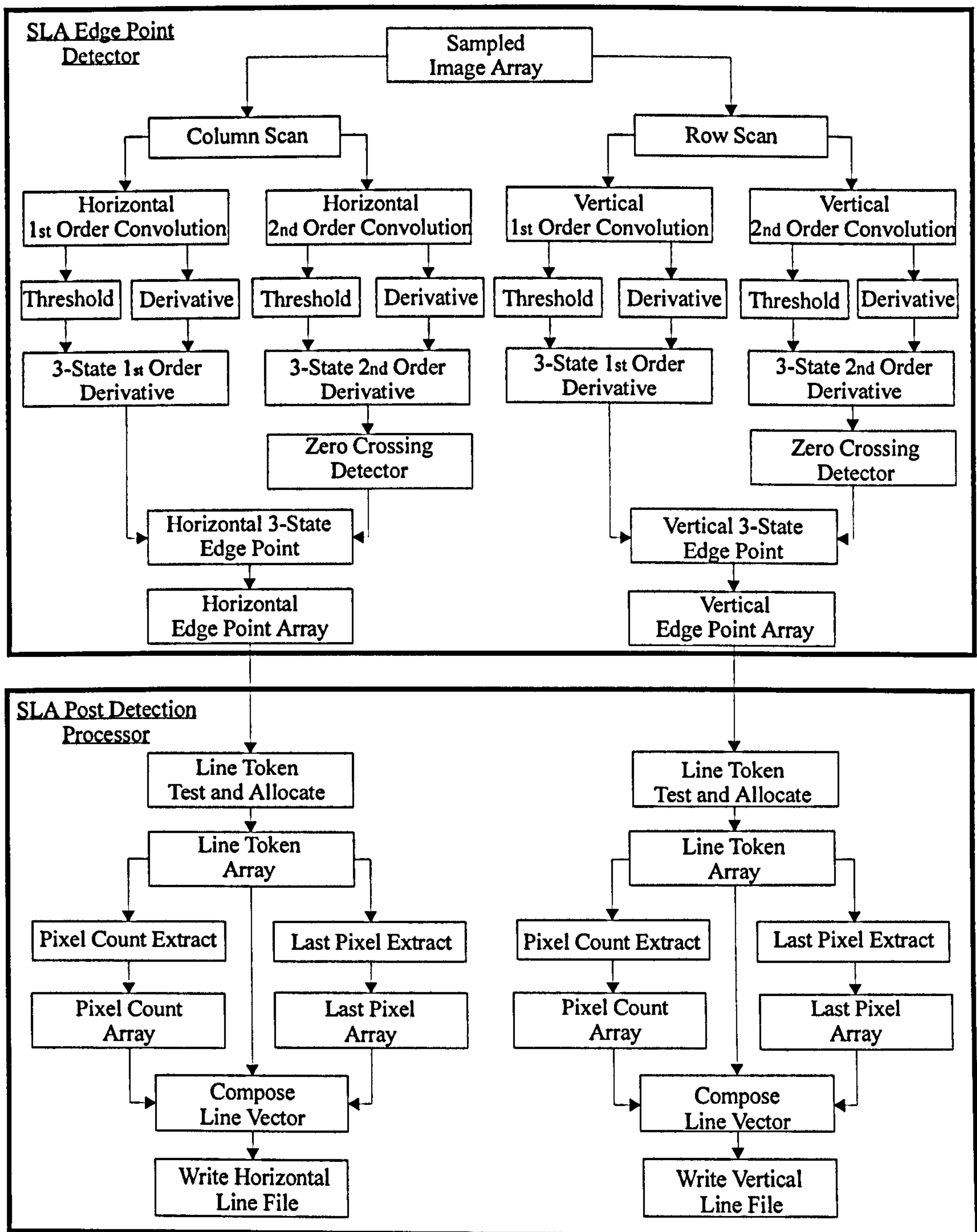


Figure 3.2 SLA Algorithm Processing Structure

In Section 3.6 the use of the SLA line lists for the recovery of a robot's pose from within a corridor environment is studied. This study demonstrates that the retention of

edge sense information in the line lists allows co-linear line segments to be grouped. These groupings reduce the uncertainty of the match between the model and the structural features detected within captured images. In Section 3.7 consideration is given to the operational specifications for the SLA implementation. In Chapter 5 these specifications are used to set the operational parameters for the Smart CMOS Camera circuit implementation

The SLA algorithm development borrowed heavily from the considerations of an edge detector that could be realised within the restrictive processing environment of analogue CMOS circuits. Particular effort was made to avoid product functions that would prove expensive in terms of substrate layout space [14]. The practicality of routing signals across the substrate was also considered in the choice of the data transfers used by the algorithm.

Section 3.2 The SLA Derivative Operators

3.2.1 1st and 2nd Order Sparse Convolutions

The algorithm employs a combination of 1st and 2nd order spatial derivatives to locate edge points. This dual derivative method was adapted from the Canny algorithm [11] which selects edge points from within its 1st order results through non maximal suppression. The non-maximal suppression of the 1st order results is equivalent to searching for zero crossings within the 2nd order spatial derivatives. In the SLA algorithm edge points are located by combining the results of a 1st order derivative operator with zero crossings detected within 2nd order derivative results.

The SLA algorithm requires a total of four derivative operators to process each pixel in of the image intensity profile. The operators are given by 1st and 2nd order derivative convolutions applied in both the horizontal and vertical directions. The application of these derivative operators to a sampled image profile $I(x,y)$ has the general form of a two dimensional convolution given by equation (3.1) [72].

$$d_{direction}^{order}(x,y) = \sum_{i,j \in H} I(x-i, y-j) \cdot h_{direction}^{order}(i,j) \quad (3.1)$$

The derivatives of $I(x,y)$ are given as $d_{direction}^{order}(x,y)$. The subscript *direction* is set to *v* for the vertical direction and *h* for the horizontal direction. The superscript *order* is set to 1 for a 1st order derivative and 2 for a 2nd order derivative. The impulse response of the derivative convolution is given by $h(i,j)$. The masks that define the four derivative convolutions employed by the SLA algorithm are illustrated in Figure 3.3. Equation's (3.2) to (3.5) give the numerical operations applied to each pixel in the processed image.

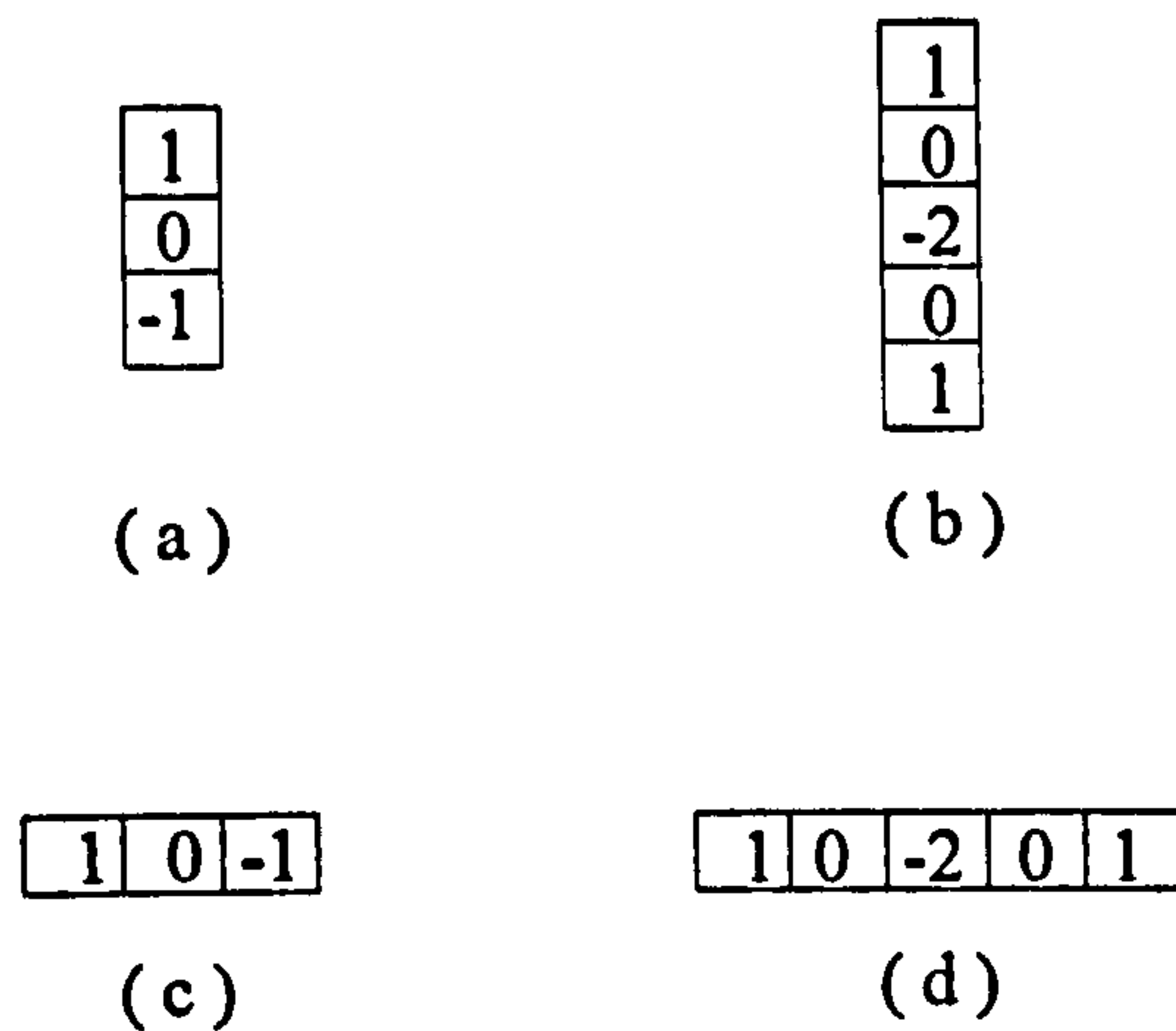


Figure 3.3 SLA derivative masks (a) horizontal 1st order convolution, (b) horizontal 2nd order convolution, (c) vertical 1st order convolution, (d) vertical 2nd order convolution

$$d_h^1(x, y) = [I(x, y - 1) - I(x, y + 1)] \quad (3.2)$$

$$d_h^2(x, y) = [I(x, y - 2) - 2I(x, y) + I(x, y + 2)] \quad (3.3)$$

$$d_v^1(x, y) = [I(x - 1, y) - I(x + 1, y)] \quad (3.4)$$

$$d_v^2(x, y) = [I(x - 2, y) - 2I(x, y) + I(x + 2, y)] \quad (3.5)$$

The convolution masks illustrated in Figure 3.3 are themselves the product of a series of convolutions. Consider the 1st order derivative {1,0,-1}; this is derived from the convolution of the short uniform average {+1,+1} with the short derivative {+1,-1} as illustrated in equation (3.6). The 2nd order operator is given by two additional convolutions of these short operators as illustrated in equation (3.7).

Thus the SLA algorithm derivatives are the realisation of a series of uniform averages and derivatives. If the length of the uniform average is increased, then the span of the convolutions are increased but the number of coefficients remain at two for the 1st order derivative and three for the 2nd order derivative. As a result, the SLA algorithm

has been classed as a sparse convolution algorithm. Equations (3.8) and (3.9) illustrate the convolutions for a uniform filter of length 4. This length parameter controls the connectivity of the detector across the direction of the detected edge. The sparse convolutions of equations 3.8 and 3.9 afford the SLA algorithm, wide connectivity, with a minimal amount of routing [73].

$$\{1, 0, -1\} = \{1, 1\} * \{1, -1\} \quad (3.6)$$

$$\{1, 0, -2, 0, 1\} = \{1, 0, -1\} * \{1, 1\} * \{1, -1\} \quad (3.7)$$

$$\{1, 0, 0, 0, -1\} = \{1, 1, 1, 1\} * \{1, -1\} \quad (3.8)$$

$$\{1, 0, 0, 0, -2, 0, 0, 0, 1\} = \{1, 0, 0, 0, -1\} * \{1, 1, 1, 1\} * \{1, -1\} \quad (3.9)$$

3.2.2 Averaged Sparse Convolutions

A feature of the SLA NSIP orthogonal scans is that averaging normal to the spatial derivative direction is available without need for additional routing. In the case of the horizontal convolutions additional columns about the processed column can be enabled and summed to give uniform averaging. The convolution masks for averaging of width 3 are illustrated in Figure 3.4. Equation's (3.10) to (3.13) give the numerical operations when the convolution masks are applied to the image.

1	1	1
0	0	0
-1	-1	-1

(a)

1	1	1
0	0	0
-2	-2	-2
0	0	0
1	1	1

(b)

1	0	-1
1	0	-1
1	0	-1

(c)

1	0	-2	0	1
1	0	-2	0	1
1	0	-2	0	1

(d)

Figure 3.4 SLA Derivative Masks Length =2, Width = 3, (a) Horizontal 1st Order Convolution Mask, (b) Horizontal 2nd Order Convolution Mask, (c) Vertical 1st Order Convolution Mask, (d) Vertical 2nd Order Convolution Mask

$$d_h^1(x, y) = \begin{bmatrix} I(x-1, y-1) - I(x-1, y+1) \\ I(x, y-1) - I(x, y+1) \\ I(x+1, y-1) - I(x+1, y+1) \end{bmatrix} \quad (3.10)$$

$$d_h^2(x, y) = \begin{bmatrix} I(x-1, y-2) - 2I(x-1, y) + I(x-1, y+2) \\ I(x, y-2) - 2I(x, y) + I(x, y+2) \\ I(x+1, y-2) - 2I(x+1, y) + I(x+1, y+2) \end{bmatrix} \quad (3.11)$$

$$d_v^1(x, y) = \begin{bmatrix} I(x-1, y-1) - I(x+1, y-1) \\ I(x-1, y) - I(x+1, y) \\ I(x-1, y+1) - I(x+1, y+1) \end{bmatrix} \quad (3.12)$$

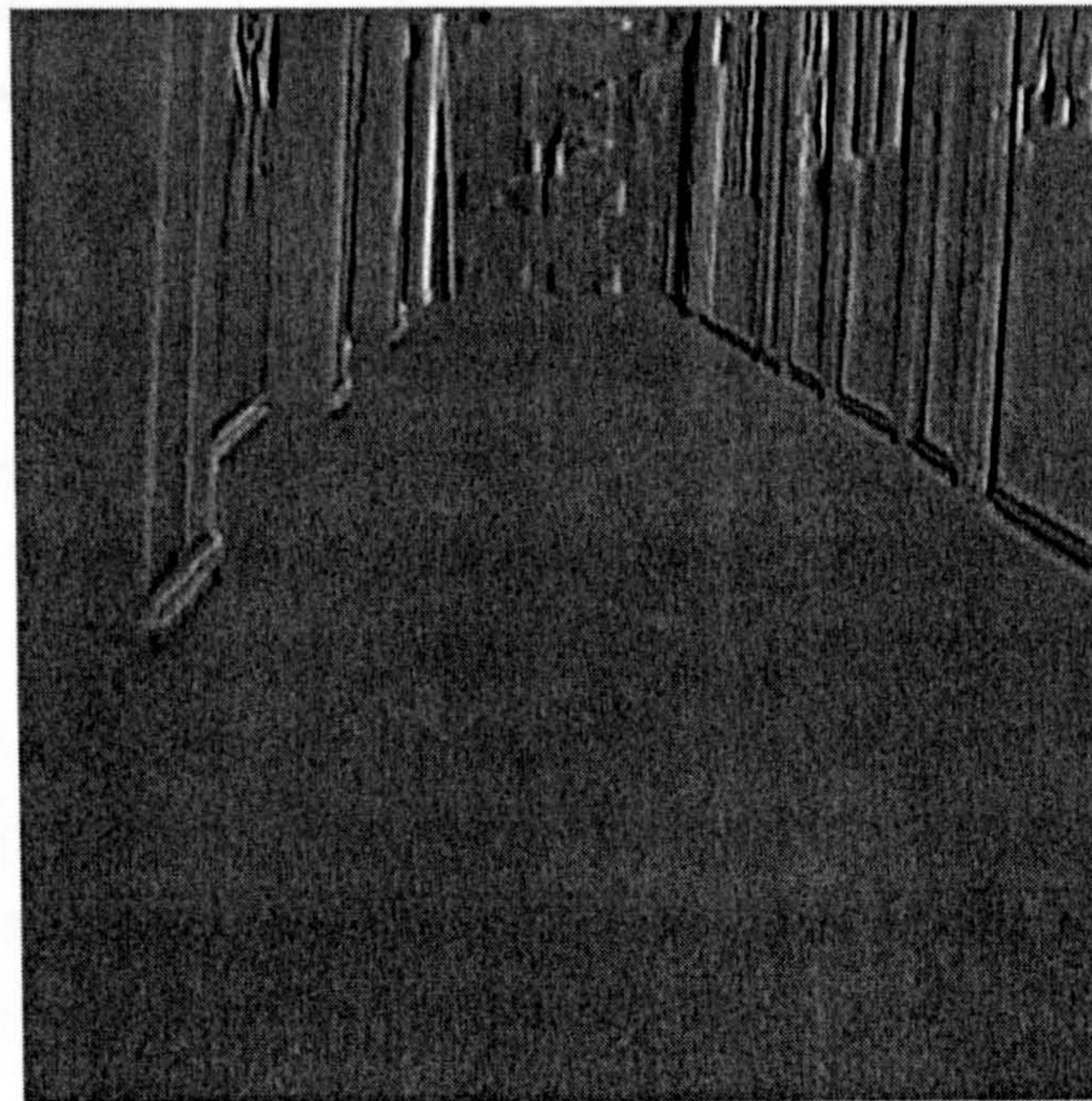
$$d_v^2(x, y) = \begin{bmatrix} I(x-2, y-1) - 2I(x, y-1) + I(x+2, y-1) \\ I(x-2, y) - 2I(x, y) + I(x+2, y) \\ I(x-2, y+1) - 2I(x, y+1) + I(x+2, y+1) \end{bmatrix} \quad (3.13)$$

3.2.3 Directional Derivative Sense Retention

The image in Figure 3.5(a) is representative of the data that an autonomous navigation system will need to recover its pose from, if corridor navigation is to be implemented. The image is sampled at a resolution of 512x512 pixels. This image is used to illustrate the operation of the SLA algorithm. An important aspect of the distributed processing employed in the SLA algorithm is the retention of derivative sense information [74,75]. The retention of derivative sense information is illustrated in Figure 3.5(b) where a corridor image is processed by the vertical direction 1st order derivative of equation (3.12). On the plane surfaces, the derivative returns the mid-tone grey level to register no gradient. The derivative scan is from left to right. At dark to light surface transitions, the derivative returns negative gradients represented by the dark lines. At a light to dark surface transition the derivative returns positive gradients represented by the white lines.



(a)



(b)

Figure 3.5 Sense Information, SLA Length =2 Width=3, **(a)** Corridor Image, **(b)** 1st Order Vertical Derivative

Section 3.3 Adaptive Thresholds and Edge Assignment

3.3.1 Adaptive Thresholds

The SLA algorithm converts the analogue directional derivatives, given by the application of the equations derived in Section 3.2, into discrete signals through comparisons with local thresholds. The 1st order discrete derivatives mark the regions of the image where edge points are located. The positions of the edge points within these regions are refined through the extraction of zero crossing points from the 2nd order discrete derivatives.

In the SLA algorithm simulations the thresholds were adaptively set. The thresholds for each pixel derivative were set to a percentage of the average pixel intensity that contributed to that pixel derivative. The percentage component in the threshold calculation was set globally for the image. The use of the adaptive threshold was used to reflect the response of the contrast sensitive derivative circuits described in Section 5.4.

The adaptive threshold $t(x,y)$ for the horizontal 1st order derivative convolution of Figure 3.4(a) is given by equation (3.14). This equation finds the average intensity of the pixels that contribute to the derivative and sets the threshold to a percentage of that average intensity. The percentage parameter Per_1^{st} is globally set for the image. The adaptive threshold $t(x,y)$ for the horizontal 2nd order derivative of Figure 3.4(b) is evaluated through equation (3.15), where the global percentage threshold is given by Per_2^{nd} . The evaluations for the adaptive thresholds of the Figure 3.4 vertical derivatives are given in equations (3.16) and (3.17).

$$t_h^1(x, y) = \left(\frac{Per\ 1^{st}}{3 \times 2 \times 100} \right) \begin{bmatrix} I(x-1, y-1) + I(x-1, y+1) \\ I(x, y-1) + I(x, y+1) \\ I(x+1, y-1) + I(x+1, y+1) \end{bmatrix} \quad (3.14)$$

$$t_h^2(x, y) = \left(\frac{Per\ 2^{nd}}{3 \times 4 \times 100} \right) \begin{bmatrix} I(x-1, y-2) + 2I(x-1, y) + I(x-1, y+2) \\ I(x, y-2) + 2I(x, y) + I(x, y+2) \\ I(x+1, y-2) + 2I(x+1, y) + I(x+1, y+2) \end{bmatrix} \quad (3.15)$$

$$t_v^1(x, y) = \left(\frac{Per\ 1^{st}}{3 \times 2 \times 100} \right) \begin{bmatrix} I(x-1, y-1) + I(x+1, y-1) \\ I(x-1, y) + I(x+1, y) \\ I(x-1, y+1) + I(x+1, y+1) \end{bmatrix} \quad (3.16)$$

$$t_v^2(x, y) = \left(\frac{Per\ 2^{nd}}{3 \times 4 \times 100} \right) \begin{bmatrix} I(x-2, y-1) + 2I(x, y-1) + I(x+2, y-1) \\ I(x-2, y) + 2I(x, y) + I(x+2, y) \\ I(x-2, y+1) + 2I(x, y+1) + I(x+2, y+1) \end{bmatrix} \quad (3.17)$$

The percentage parameters in the 1st and 2nd order adaptive thresholds are used to control the type of edge set detected by the SLA algorithm. The structural outlines that the autonomous navigation system needs to identify are found by setting the *Per_1st* term to 5% and setting the *Per_2nd* term to 1%. By reducing both these terms by a factor of 5 the detector can be made to detect faint outlines and textured details as well as the structural outlines.

3.3.2 Discrete Derivatives

The conversions of the analogue derivatives, with the retained sense information, to a discrete format gives three possible states for each derivative. If the derivative is greater than $t(x,y)$ then the derivative is assigned the Positive (*P*) state. If the derivative is less than $-t(x,y)$ then the derivative is assigned to the Negative (*N*) state, otherwise the derivative is assigned to the Zero (*Z*) state. In equation (3.18) to (3.21) the discrete conversion for the vertical and horizontal derivatives are defined.

$$D_h^1(x, y) = \begin{cases} P & \text{for } (d_h^1(x, y) \geq t_h^1(x, y)) \\ N & \text{for } (d_h^1(x, y) \leq -t_h^1(x, y)) \\ Z & \text{otherwise} \end{cases} \quad (3.18)$$

$$D_h^2(x, y) = \begin{cases} P & \text{for } (d_h^2(x, y) \geq t_h^2(x, y)) \\ N & \text{for } (d_h^2(x, y) \leq -t_h^2(x, y)) \\ Z & \text{otherwise} \end{cases} \quad (3.19)$$

$$D_v^1(x, y) = \begin{cases} P & \text{for } (d_v^1(x, y) \geq t_v^1(x, y)) \\ N & \text{for } (d_v^1(x, y) \leq -t_v^1(x, y)) \\ Z & \text{otherwise} \end{cases} \quad (3.20)$$

$$D_v^2(x, y) = \begin{cases} P & \text{for } (d_v^2(x, y) \geq t_v^2(x, y)) \\ N & \text{for } (d_v^2(x, y) \leq -t_v^2(x, y)) \\ Z & \text{otherwise} \end{cases} \quad (3.21)$$

3.3.3 Edge Point Assignment

The equations that define the SLA horizontal edge points $EP_h(x, y)$ and the vertical edge points $EP_v(x, y)$ are given in equations (3.22) and (3.23). These edge points have the same three state discrete format used in the derivative processing. If an edge point exists it is either Positive (P), or Negative (N), otherwise no edge exists and the Zero (Z) state is assigned. A horizontal edge point assignment $EP_h(x, y)$ requires a non zero $D^1(x, y)$ and a zero crossing between $D^2(x, y-1)$ and $D^2(x, y+1)$. The derivative sense information is used to ensure that edges are only assigned when the direction of the zero crossing is valid for the sense of the 1st order derivative.

$$EP_h(x, y) = \begin{cases} P & \text{for } ([D_h^1(x, y) = P] \text{And} [D_h^2(x, y-1) = P] \text{And} [D_h^2(x, y+1) = N]) \\ N & \text{for } ([D_h^1(x, y) = N] \text{And} [D_h^2(x, y-1) = N] \text{And} [D_h^2(x, y+1) = P]) \\ Z & \text{otherwise} \end{cases} \quad (3.22)$$

$$EP_v(x, y) = \begin{cases} P & \text{for } ([D_v^1(x, y) = P] \text{And} [D_v^2(x-1, y) = P] \text{And} [D_v^2(x+1, y) = N]) \\ N & \text{for } ([D_v^1(x, y) = N] \text{And} [D_v^2(x-1, y) = N] \text{And} [D_v^2(x+1, y) = P]) \\ Z & \text{otherwise} \end{cases} \quad (3.23)$$

The zero crossing tests employed in equation (3.22) test for 2nd order derivatives that exceed the local thresholds at pixel sites $(x, y-1)$ and $(x, y+1)$. This avoids the need to determine if the 2nd derivative is zero at site (x, y) . In theory a zero condition at site (x, y) indicates a zero crossing [13] and thus marks an edge point. However, due to the

sampled nature of the processed data there can be no certainty that a zero crossing will be marked by a zero in the 2nd order derivative profile. The approach adopted in the SLA detector avoids this potential source of noise in the determination of edge locations. In equation (3.23) the zero crossing for site (x,y) in the vertical scan is determined by testing the 2nd order derivatives at sites $(x-1,y)$ and $(x+1,y)$. As a result of employing this three-pixel spread in the zero crossing detection a double edge point is allocated where the edge point occurs with an abrupt intensity discontinuity.

3.3.4 Thresholds for Structural Edges

The application of the SLA analogue derivatives and discrete conversions to the corridor image are illustrated in Figure 3.6. The derivatives are given by equation's (3.10) to (3.13) and the adaptive thresholds by equations (3.14) to (3.17). The Per_1^{st} was set to 5% the Per_2^{nd} was set to 1%. The three-state results for the vertical and horizontal edge detection processes are given in Figure 3.6(a) and (b). The adaptive nature of the SLA derivative thresholds ensures the detection of the double door features even though they are in a corridor region that is dimly illuminated.

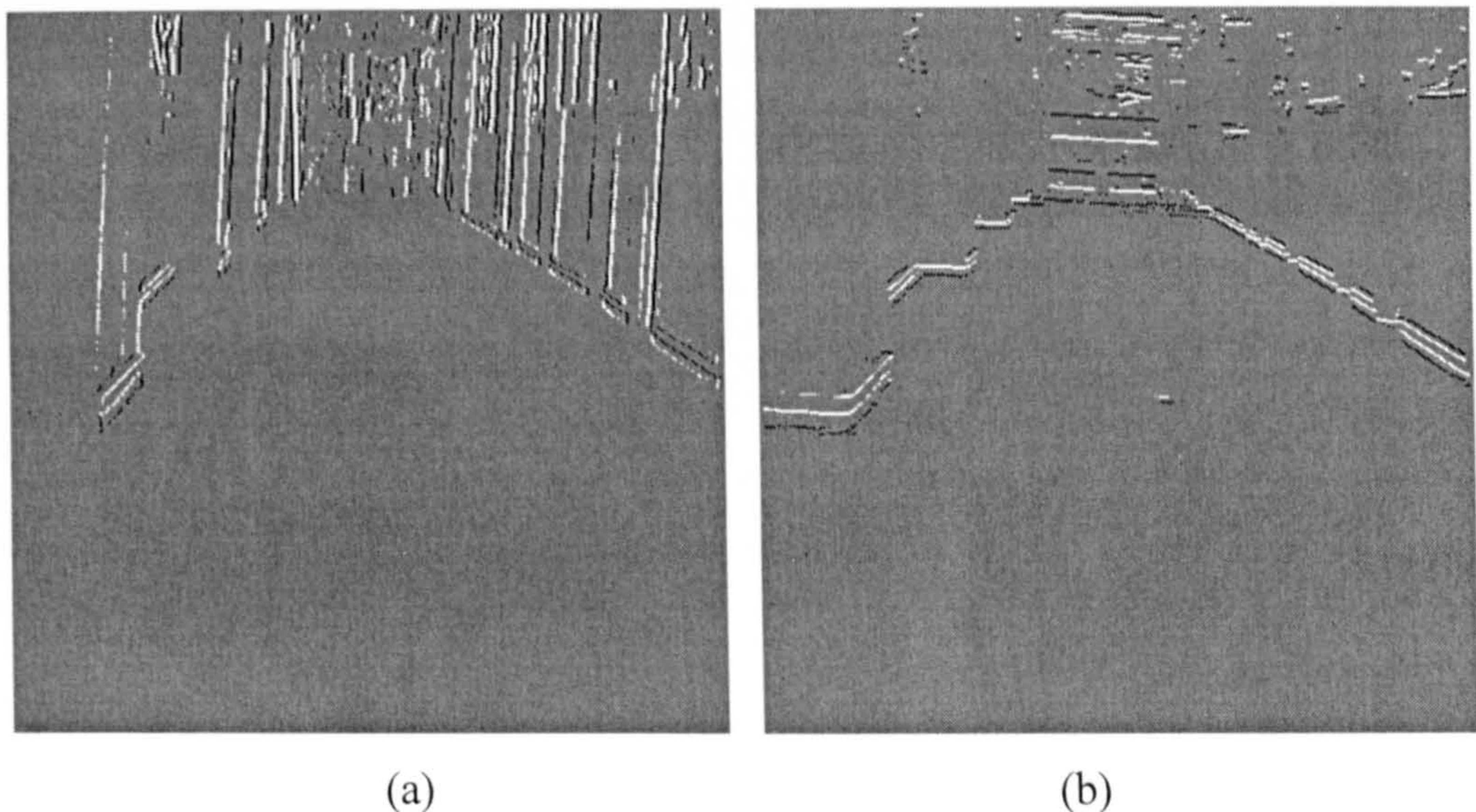


Figure 3.6 SLA Directional Edge Sets, **(a)** Vertical Edges, **(b)** Horizontal Edges

The Per_1^{st} and Per_2^{nd} settings used for the Figure 3.6 results give the detector a low susceptibility to noise. The main structural outlines are retained, but corner features

are missed. The missed corner features give rise to an edge set that is not aesthetically pleasing however, the suppression of corner details is critical to the efficient implementation of Beveridge's [15] pose recovery algorithm. The processing overheads are reduced if the pose recovery is limited to finding matches between straight lines in the model and the image extracted data. These threshold settings were used for the navigation analysis carried out in Section 3.6.

Section 3.4 Post Edge Point Detection Processing

3.4.1 Test and Allocate Process

The SLA Post Detection Processing is centred on a line extraction algorithm that converts the SLA direction edge maps, generated by equations (3.22) and (3.23), into line vector lists [76]. These line vectors have a total of six integer parameters that are used to describe the line's position and attitude. These vectors are written into a text file for use in the robot navigation algorithm [37,38]. It is demonstrated that the line length parameter provides a useful means of limiting the noise content of the line listing. The structure of the SLA post detection processing of the horizontal edge set is illustrated in Figure 3.7.

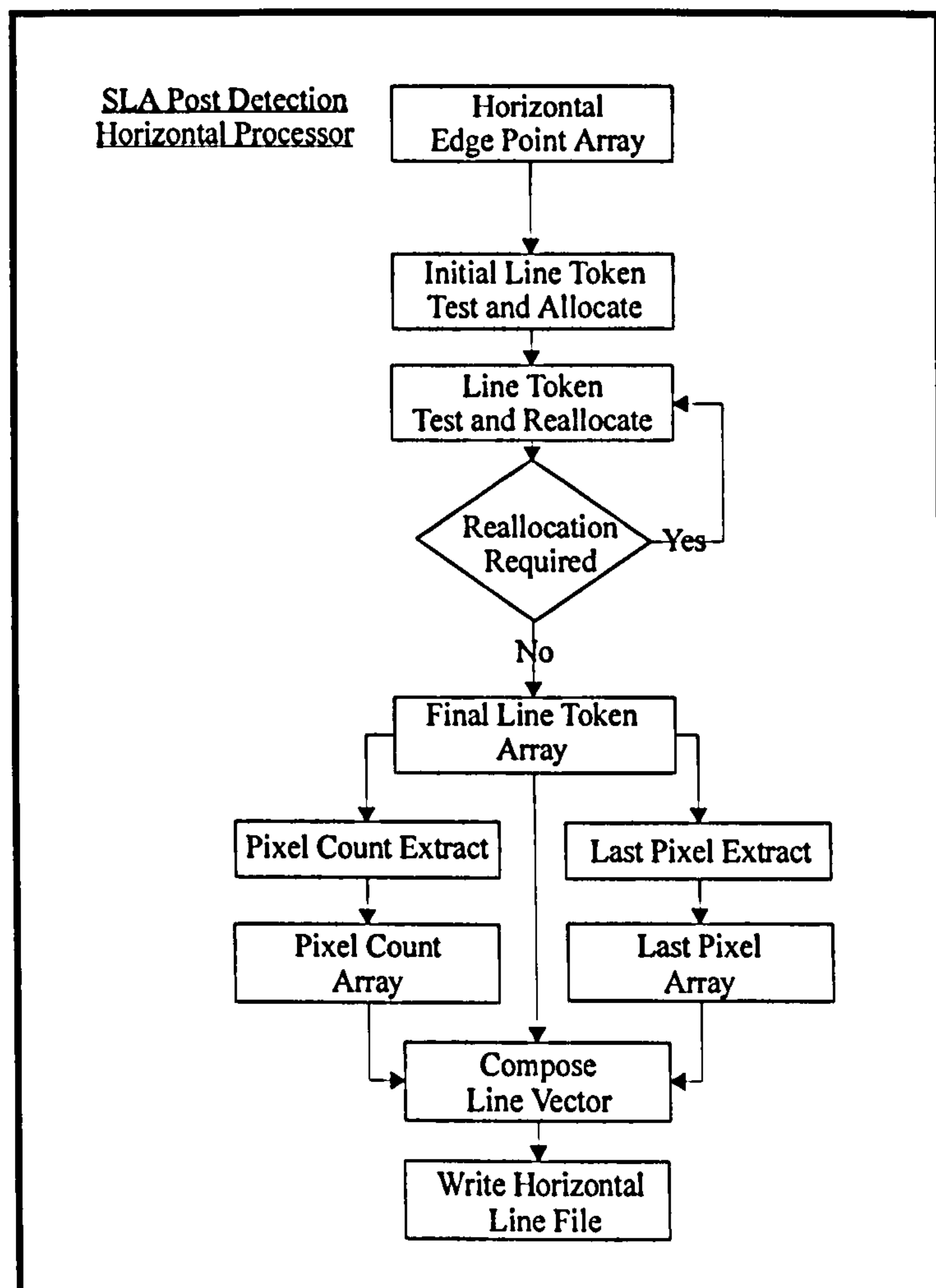


Figure 3.7 Horizontal Edge Set Post Detection Processing

The line extraction algorithm is implemented through multiple scans of the edge map. In the first scan all connected edge points are allocated a token value. The token value gives the lowest address of a pixel that is linked to the Allocation Pixel (AP). The tokens are assigned through a test and allocate function that is raster scanned through the edge point set. In Figure 3.8 the *AP* and its spatial relationship to the connection Test Pixels (TP) is illustrated. The directions of the raster scan relative to *AP* are noted. If *AP* is not linked to a previously assigned pixel then it's token is set to the current pixel address. If one of the *TP*'s has a token assigned to it then *AP* inherits this token. If more than one *TP* has a token assigned then the lowest of these is assigned to the *AP* site.

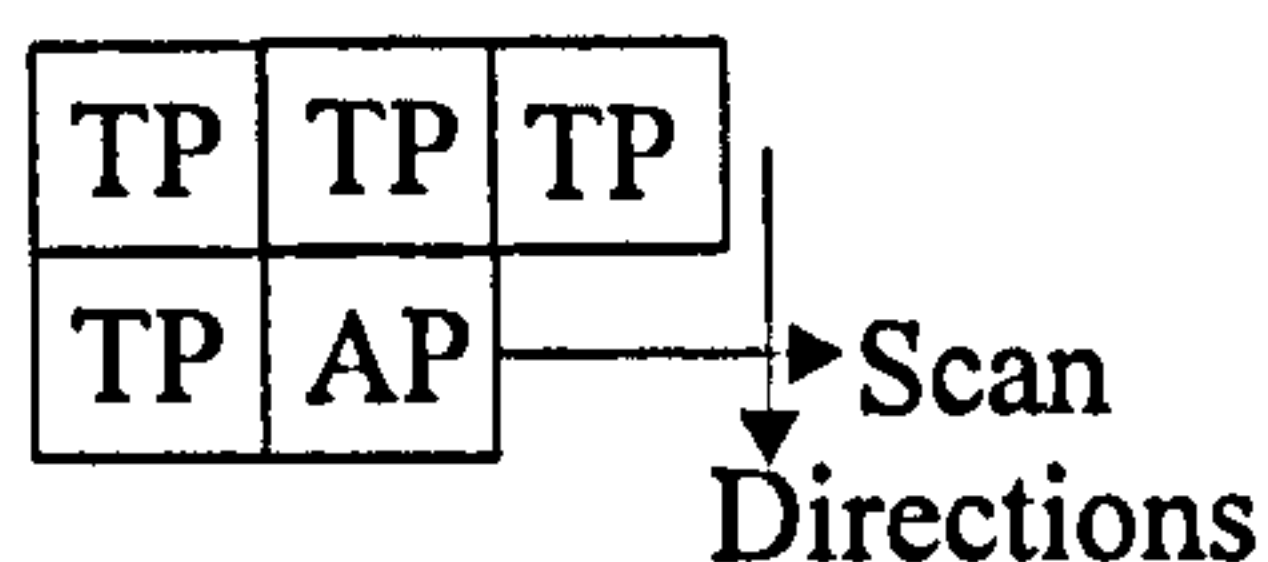


Figure 3.8 Test Pixel and Allocate Pixel Spatial Relations

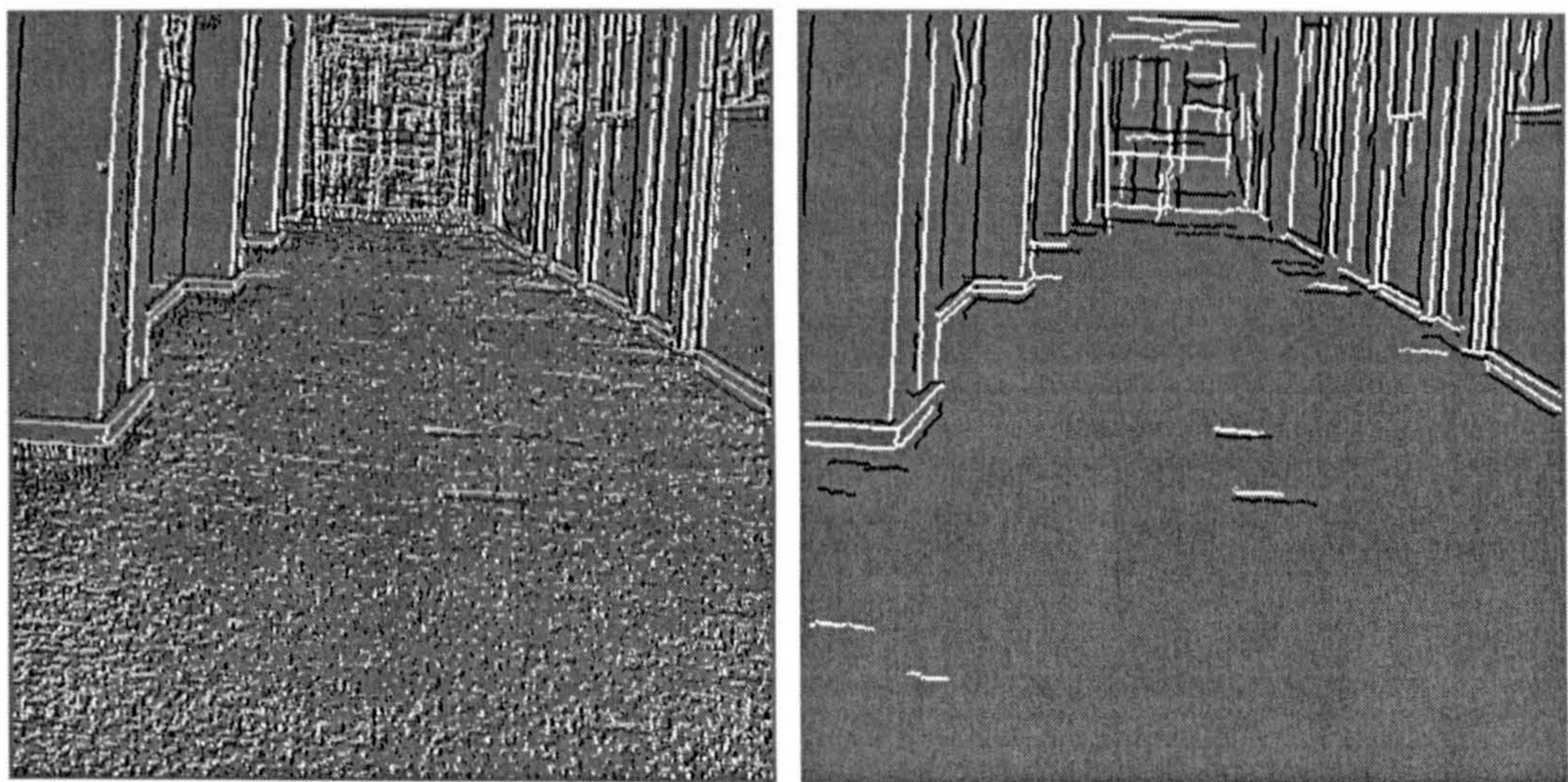
If the test and allocate function is limited to a single scan of the edge set then it is possible that an image line will have multiple tokens assigned to it. In order to remove this effect a redirect process is used. In this, the test and allocate function is scanned through the assigned token set to locate all lines with multiple token designation. Where multiple token assignments are located, the higher address token is redirected to the lower address token. If no lines with multiple tokens are found the line extraction process is complete. This second phase redirect process is repeated until all the multiple token lines are removed from the line set. Results from processed images showed that the redirect process is typically repeated twice to resolve the line token clashes.

The final token sets given by the application of the line extraction function are processed to form lists of line vectors that are loaded into text files. A total of six integer values make up the line vectors. Four of these values were given by the line start and stop co-ordinates. The remaining vector values are given by the line sense

and the pixel count results. The pixel count parameter is used to thin the line list by setting a minimum pixel count for the line vectors to be written into the text file.

3.4.2 Pixel Count Threshold for Noise Removal

The results given in Figure 3.9 illustrate the usefulness of the pixel count parameter. The corridor image of Figure 3.5(a) was processed with SLA set to detect faint outlines. For this $Per_{1^{st}}$ was set to 1% and $Per_{2^{nd}}$ was set to 0.5%. These settings ensure that the faint outlines are detected. However, they also cause the texture components in the carpet to be registered. The faint outlines differ from the textured lines in that they are composed of continuous lines with the same edge sense. In contrast the direction of the texture components are constantly changing. Thus they are characterised by short lines which can be removed through the application of a pixel count threshold to the extracted line sets.



(a)

(b)

Figure 3.9 Combined Faint Outline Horizontal and Vertical Edge Sets, (a) No Pixel Count Threshold, (b) Pixel Count Threshold Set to 40.

Figure 3.9(a) illustrates the results of the faint outline processing without the application of a pixel count threshold. The clutter created by the textured carpet surface obscures the important floor to wall boundary lines. In Figure 3.9(b) the application of a pixel line count threshold of 40 pixels removes the clutter associated with the floor texture and leaves the faint outlines in the processed image.

A comparison between the results of Figures 3.9(b) and Figure 3.6 demonstrates that the faint outline settings combined with line length thresholds gives a more complete segmentation. A subjective assessment will give the SLA detector set to the faint outline settings a higher quality rating than the structural setting of the SLA detector used in the Figure 3.6 results. This gain in qualitative response has been facilitated by the distributed nature of the SLA algorithm. By deferring the noise filtering operation until after the primitive line information has been collated, the SLA algorithm can retain low contrast outlines whilst removing high contrast noise. This strategy contrasts with established detectors [11,77-80] where the chief noise suppression mechanism is the application of low pass filtering prior to the derivative computations.

Section 3.5 SLA Computational Requirements

The SLA edge detector was designed for incorporation in a CMOS NSIP. Its processes were therefore based on integer summations of pixel intensities in order to simplify the circuit implementation of the algorithm. The computational requirements of this algorithm were of interest, as in addition to the CMOS implementation there is the possibility that the algorithm could be integrated into a DSP structure.

The structural overview of the SLA algorithm given in Figure 3.2 illustrates the sequence of pixel processes that transforms the sampled intensity profile into a succinct line vector listing. In Table 3.1 the processing requirements for the constituent parts of this algorithm are summarised. It is assumed that the horizontal and vertical processes are implemented separately. The tabulated results are given for the horizontal processing. It is further assumed that the frame capture process is complete, so that the data within the sampled image array is unchanged while the SLA edge detection process is implemented. The convolutions were assumed to have a width of 3 and the image resolution was set to a resolution of 512x512 pixels.

Process	Array Reads	Array Writes	Plus/Minus Operations	Product Operation
Convolution	15			
Derivatives			12	
Thresholds			12	2
3-State Conv.			4	
Zero X Det.			2	
Edge allocate		1	2	1
Token allocate	4	1	4	
Reallocate	4	2	4	
Refresh	2	1	1	
Pixel Count	2	1	1	
Last Pixel	2	1	1	
Vector Form	3	6	2	4
Totals per Pixel	32	13	45	7

Table 3.1 SLA Algorithm Processing Requirements

The tabulated results given in the 1st and 2nd columns of Table 3.1 detail the number of image *read* and *write* operations required per processed pixel at each level in the algorithm. The *plus/minus operations* column gives the number summations, subtractions and comparisons needed to process each pixel address. The number of processor *product operations* was noted in the final column.

A total of 97 machine instructions per pixel are required to implement the horizontal processing of the SLA algorithm. If a 512x512 image is processed with a 10Hz frame rate then a DSP system with a 2.54×10^8 MIPS would be required to fully process all the horizontal directional data generated by the sensor in real time. Critical to the DSP implementation of the SLA algorithm is the memory management necessary for the post detection processing. In the implementation of the SLA algorithm a maximum of 12 memory bytes are required for each processed pixel. Thus the process memory required for a 512x512 image is 3.2Mbytes. These processor requirements would require a dedicated DSP processor. A Pentium III with processor capacity in excess of 5×10^8 MIPS could implement the SLA algorithm for both horizontal and vertical directions.

The review of navigation algorithms in Section 2.2.4 demonstrated that the process lag between image capture and the instigation of directional correction was critical to the implementation of vision based navigation. Consider the case of a robot travelling at walking pace of 1m/sec. A process lag of 100ms, given by a 10Hz update rate is equivalent to 10cm of movement. This degree of overshoot would be insignificant in the operation of an indoor robot.

3.6 SLA in Autonomous Navigation

3.6.1 Integration of SLA into a Pose Recovery Algorithm

It was decided to investigate the means by which the SLA algorithm results could be used to provide an efficient implementation of vision based autonomous navigation. The autonomous navigation architecture reported by Kosaka *etal* [37,38] was chosen as a target environment for the SLA algorithm. The 3D pose recovery algorithm proposed by Beveridge [15] was chosen as the mechanism for transferring the SLA scene description primitives into the model space required by the [37-38] algorithm. An analysis of the positional uncertainty was made for the SLA based implementation.

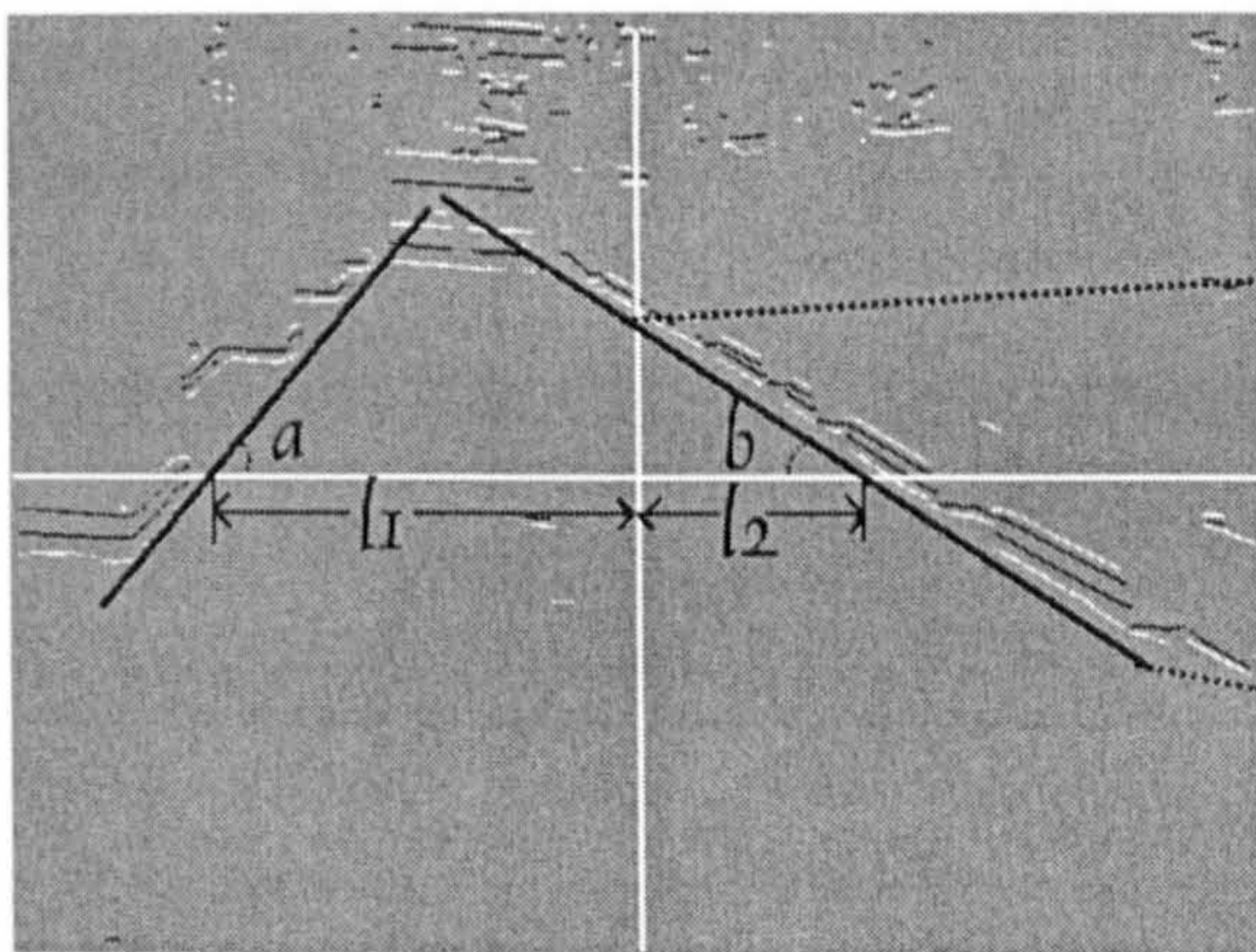
The algorithms reviewed in Section 2.2.4 indicated that there were two major tasks required of the vision based indoor navigation system. These were door location and wall following. The forward view provided by a single camera as given in Figure 3.5 allows the structure of environment to be mapped [37-39]. In this image the double doors provide a main target for the navigation algorithm and a comparison between successive frame mappings of the door location allows the robot to maintain a straight path towards the door. The estimate of the minimum distances between the robot and the walls of the corridor is determined by projections of the floor to wall boundaries. Uncertainties in the match between the model and the extracted image lines are amplified through these projections.

In order to limit the errors in the robot to wall distance estimates it was decided to employ a wide-angle view of the corridor. The corridor image given in Figure 3.10(a) was used to evaluate the uncertainty that would result from the use of the SLA algorithm results to resolve the floor to wall boundaries and estimate the robot to wall distances. This image was processed through a SLA edge detector with the convolution uniform filter length set to 2 and the average filter width set to 3. The convolutions are given by equations (3.10) to (3.13). The thresholds were set for structural outlines with $Per_{1^{st}}$ was set to 5% and $Per_{2^{nd}}$ set to 1%. The horizontal edge set for the image is given Figure 3.10(b). To facilitate the pose recovery explanation a set of (x,y) axis with $(0,0)$ central to the image have been drawn on

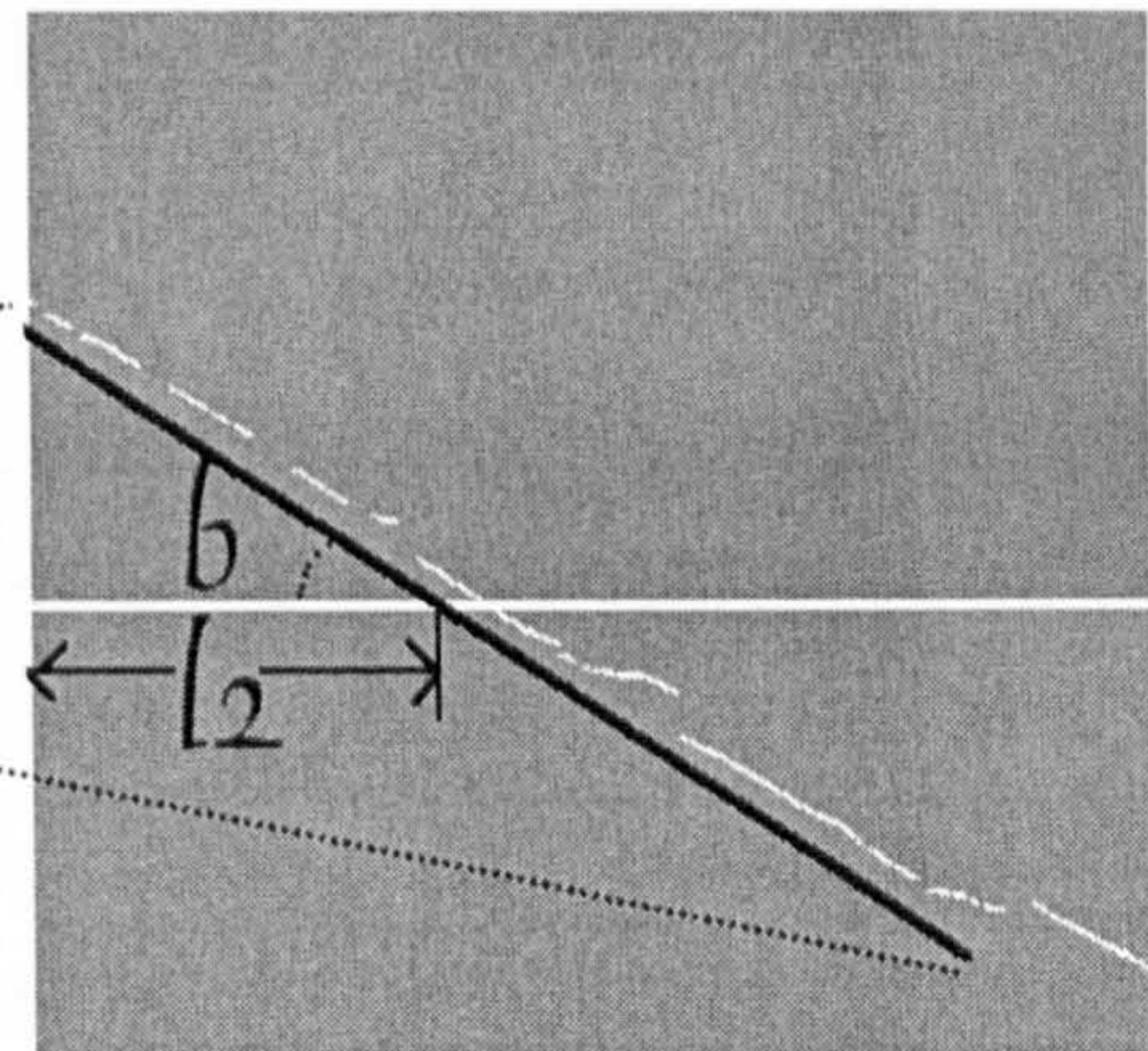
Figure 3.10(b). Model estimates of the corridor floor to wall boundaries have also been superimposed upon the Figure 3.10(b) image, these are given by the two broad lines with a disappearance point central to the double doors.



(a)



(b)



(c)

Figure 3.10 Geometric Model Comparison to Extracted Lines. (a) Wide Angle Corridor View, (b) The Model Estimate and Horizontal Edge Set, (c) Boundary Model Estimate and Selected Lines

The SLA algorithm retention of edge sense information provides an efficient means of selecting the image lines to be paired with the model estimate. The lines within the

extracted line set are grouped as a consequence of their angle, point of crossing the vertical y axis and their sense. One of these groupings and an initial model estimate for the left-hand floor to wall boundary are given in Figure 3.10(c). The Beveridge [15] method proceeded by evaluating the error between the model and the grouping of image lines. This error was then reduced through refining the model estimate and changing the selection of image lines paired with the model.

When the robot's view of the doors in Figure 3.10(a) is obstructed by other corridor traffic then the floor to wall boundaries become the main navigational clue. For a given camera fixing these boundary lines give the position of the robot within the corridor and its heading along the corridor. The application of the Beveridge [15] geometric model matching to these lines allows the robot's pose to be recovered. If the extracted floor to wall boundary lines are corrupted through missing lines or cluttered through multiple returns from skirting boards then the geometric match processing requirements for the robot's navigation can become prohibitive. An efficient implementation of Beveridge's [15] algorithm requires the extracted image lines to exclude corner features. This can be achieved, as was noted in Section 3.3, through the use of the SLA structural line detector. This is the detector set-up used in Figure 3.10 results.

The objective in the SLA post edge detection processing was to generate succinct line sets in a vector format that included a majority of the floor to wall boundary lines in the image. The extracted line vectors hold the line start and stop locations from which the line's axis crossing and the angle to the image (x,y) co-ordinate's can be calculated. The position of the robot within the corridor is resolved by minimising the error between selected image lines and the model estimate. In Figure 3.10, for a given camera fixing, the angles a and b and the lengths l_1 and l_2 uniquely define the position of the robot and its heading.

3.6.2 Evaluation of Model Match Errors

The pairing of the selected image lines with the model estimate gives displacement and angular errors. The errors calculated for the pairings of Figure 3.10(c) are

tabulated in Table 3.2. These errors are rms summed to give an overall displacement error of 24.44 pixels and an angular error of 11.94 degrees. An example of the refinement of the Figure 3.10(c) model estimate is given in Table 3.3. The rms errors are reduced by shifting and rotating the model estimate and changing the set of lines to be paired with the model. The new rms displacement error is 8 pixels and the rms angular error is 3.89 degrees.

The results given in Tables 3.2 and 3.3 illustrate that the model refinement and local search techniques proposed by Beveridge [15] are readily implemented when the SLA extracted line lists are used as the source image information. The facility to group image lines on the basis of their sense as well as their direction and axis crossings ensured that the paired set is not cluttered by unconnected lines. The SLA sense information contributes to the model refinement process by limiting the search space for missing line segments.

Line List				Line Errors	
x start	y start	x end	y end	Angular Error (degree)	Disp. Error (pixel)
358	171	388	185	8.79	10.84
391	188	400	192	9.84	15.40
407	195	422	205	0.11	19.45
433	212	460	228	3.15	21.56
474	240	491	251	0.90	19.44
500	255	508	257	19.77	25.24
515	269	562	298	2.13	18.64
562	302	567	304	12.00	15.49
570	306	601	314	19.33	26.02
602	329	681	376	3.05	18.49
684	380	696	382	24.34	24.39
720	384	758	406	3.74	52.46
Model Estimate				RMS Errors	
265	113	697	402	11.95	24.44

Table 3.2 Initial Model Estimate Line Pairing Results

In Figure 3.10 on the x axis, an error of 8 pixels in the evaluation of l_2 is equal to 6 cm on the robot floor. When the distance from the robot to the l_2 floor location is 3.5m the

combined effect of the 6cm and the 3.89° errors is to give an uncertainty of $\pm 0.45\text{m}$. The error in the angle estimate is the more significant of the two errors with respect to the uncertainty in the robot's position. Additional cycles of the local search algorithm may be used to reduce this error. However, there is a limit to the reduction in uncertainty that can be achieved through refinement of the model estimate. When the camera view of the floor to wall boundary is restricted to a length of 50 pixels and a potential error of ± 1 pixels exist at either end of the line match, then the model estimate uncertainty is $\pm 1.6^\circ$. Thus an analysis based on the Figure 3.10 image is limited to an uncertainty of $\pm 0.19\text{m}$ in the distance of the robot from the corridor wall.

Line List				Line Errors	
x start	y start	x end	y end	Angular Error (degree)	Disp. Error (pixel)
358	171	388	185	6.90	6.90
391	188	400	192	7.95	7.95
407	195	422	205	-1.78	-1.78
433	212	460	228	1.26	1.26
474	240	491	251	-0.99	-0.99
515	269	562	298	0.24	0.24
602	329	681	376	1.16	1.16
720	384	758	406	1.84	1.84
Model Estimate				RMS Errors	
270	113	734	402	3.89	8.00

Table 3.3 Refined Model Estimate Line Pairing Results

3.6.3 Smart CMOS Camera Specifications

If the robot relies upon a single forward-looking camera, with the view given by Figure 3.10, to implement the wall following and door finding it must take up a position at least 0.19m away from the corridor wall. It was noted that the wall following could be implemented with less uncertainty if the camera was directed to view towards the wall. This will reduce the distance to the l_2 measurement. Changing this distance from 3.5m to 1.17m will reduce the robot to wall distance uncertainty by a factor of 3.

The conflicting viewing requirements of the two tasks of wall following and door finding brought about the realisation that a single fixed camera could not gather the navigation information for both tasks. A single camera could be set to shift its view between three different positions, one forward view for door finding and two side views to monitor the distances to the side walls.

An alternative strategy was chosen for the implementation of the vision based navigation. It was decided to proceed on the basis of a system that employed three fixed cameras to provide the navigation information. A central forward looking camera would provide door finding information while two side view cameras would provide the wall following information. The Smart CMOS Camera was seen as providing a solution to the integration of multiple fixed cameras onto a robotic platform. These devices described in Chapter 5 integrate the SLA edge detection algorithm into substrate NSIP structures. In order to optimise the camera function specifications were generated for the spatial resolution and frame rate of the Smart CMOS Camera that would act as a vision accelerator for the navigation tasks.

The SLA algorithm, designed for structural line extraction, employs a width parameter that requires the data from a given pixel to be read on three successive accesses to the array. The accepted practice in CMOS image sensor is to use integrating pixels with destructive readouts. This form of readout cannot support the SLA algorithm. In order to facilitate the implement of the SLA algorithm it was necessary that the Smart CMOS Camera had a random access pixel array that permitted successive reads of the pixels. The use of a random access pixel array permits sub sampling of the image space. Thus in wall following mode the navigation processor does not need to access the full frame, instead the neighbourhood of the current model estimate of the floor to wall boundary can be processed to update the navigation information.

If the wall following camera is fixed to view the floor to wall boundary, with the line that marks this boundary being the main feature in its field view, then the angular error constraints are reduced. If the robot takes up station 0.3m away from the wall

and the camera is angled so that the l_2 measurement is made at 1.17m forward from the robots location then a 50 pixel section of floor to wall boundary will result in an uncertainty of 0.06m. Thus the spatial resolution of the camera can be reduced to the order of 100x100 pixels.

In a camera that permits frame sub-sampling the 50 pixel line could be expected to exist within a 40x40 area of the image. An image search for this line could be limited to a space of 65x65 pixels given that the system remembers the location of the last estimate match. The NSIP structure of Chapter 5 employs parallel processing of the row and column data read from the image sensing array. The result of this parallel processing and the dual scan of the SLA algorithm is to set the pixel access rate to 1300Hz, if the 65x65 search area is to be refreshed at 10Hz. The 1300Hz pixel access rate translates into a 100x100 full frame rate of 6.5Hz.

When the edge detection function is implemented within the Smart CMOS Camera, the remaining post detection processes require 46 machine instructions per processed pixel (Table 3.1). Given a search space of 65x65 pixels with a frame refresh rate of 10Hz the process lag given by the post edge detection processing is 0.33ms for a PC based processor with a capacity of 500×10^6 MIPS.

In Figure 3.10(c) there are 12 candidate line segments that form two groups to be matched with the model estimate. The implementation of the geometric match described in Section 3.6.2 required 4 multiply and 10 sum operations per matched pair. Thus a total of 168 machine instructions are required to implement a model refinement cycle when there are 12 line match segments. If it is assumed that 5 refinement cycles are needed to reduce the model error to an acceptable level then a total of 840 machine instructions are needed to adjust the model. The process lag given by the line matching algorithm is $1.7 \mu\text{s}$ for a PC based processor with a capacity of 500×10^6 MIPS.

It was concluded that if the proposed Smart CMOS Camera can be realised, then the additional processing lags incurred of 0.33ms for post edge detection processing and

the $1.7\mu\text{s}$ for the line match can be neglected, as the camera 65×65 sub-frame refresh occurs at 10Hz . This will allow the wall following distance uncertainty to be limited to 0.06m . This indicates a successful implementation of the Smart CMOS Camera will prove of benefit to the development of vision based navigation.

3.7 Conclusion

Simulation was used to develop the scanned layer solution to the implementation of an edge point detector. The scanned layer term was adopted from the substrate layout constraints imposed by the implementation of the NSIP processors adjacent to an image-sensing array. The label of Scanned Layer Architecture (SLA) was assigned to the edge detector that was created for incorporation in the Smart CMOS Camera. This device provides full image edge point results at the sensor frame rate.

The SLA edge detector of Section 3.2 employs sparse convolutions to extract 1st and 2nd order derivatives from the image profile. Adaptive thresholds convert these derivatives into a discrete format and logical operations locate the edge points within the discrete derivative results. It was demonstrated, in Section 3.3, that the SLA algorithm could detect edge points with sufficient quality for implementation of autonomous navigation.

In Section 3.4 the SLA algorithm was extended through a post detection process to extract line information from the edge point data. The application of a line length threshold was used to demonstrate that the edge sense information retained in the SLA detection results facilitated the removal of noise and the retention of faint outlines. The edge sense information was also retained in the extracted line data. It was shown in Section 3.6 that the retained sense information allowed the grouping of line segments into extended lines. These extended lines were then matched with model estimates of major structural features and thus the pose of a robot could be recovered.

Analysis of the positional uncertainty [37,38] in the pose recovery algorithm demonstrated that three fixed cameras were required for autonomous navigation. These cameras would monitor forward, left and right views around the robot. The forward looking camera providing long range target information. The side view cameras provided the information necessary to implement wall following.

The research objective chosen for the Smart CMOS Camera was the implementation of autonomous navigation in a compact and low power consumption system. The

integration of edge detection into a NSIP device would ensure that the power consumption of the low-level vision processes was minimised. The analysis of the wall following navigation function in Section 3.6 generated a set of specifications for the NSIP device. The analysis determined that the image sensing array should be set to 100x100 pixels. This resolution was sufficient for the uncertainty of the robot to wall distance to be 0.06m. The process lags could be maintained at an acceptable level for corridor travel speeds of 1m/s, if the full frame refresh rate is set to 6.5Hz with a pixel access rate of 1300Hz.

In Section 3.5 a DSP based implementation was considered for the realisation of autonomous navigation using the SLA edge and line detection processes. This established that sub-sampling and selective processing of the image data allowed for real-time operation of a robot within a corridor. The real time designation is given as a robot travelling along the corridor at 1m/s whilst using visual sensing to maintain its track along the corridor.

Chapter 4 Analysis of Edge Point Detectors

4.1 Introduction

The review of autonomous vision systems demonstrated that edge point detection is a critical element in the implementation of autonomous vision. Edge point maps have been demonstrated as an efficient method of segmenting the intensity profiles and thereby generating useful scene descriptions. The binary nature of the edge point data provides a succinct representation of the scene's contents and thus limits the computational burden of subsequent processing. An important factor in the development of the SLA detector was the analysis of the quality of its edge point results [81]. Existing methods for assessing the performance of edge detectors are classed as either subjective [82-86] or objective [87-90].

In subjective analysis, results given by an edge detector are compared to the original image by an experienced observer who assesses the quality of the detectors results. This assessment is made on the basis of the completeness of the object and feature outlines given by the detector. Also considered are the detectors susceptibility to noise where it generate false edge points and it's propensity to displace the object outlines from their true image positions. There are two major deficiencies in this method. Firstly, the quality of the original image presented to the experienced observers determines their ability to assess the detector results. Secondly, it is expected that variance will exist in the assessment given by two experienced observers, because there are no established methods for this subjective analysis to be carried out. The difficulties in setting up an expert observer assessment are evidenced in the research reported by Heath [89].

In objective assessment, the edge detector is applied to an image for which a ground truth set exists. The ground truth set marks the ideal locations of all edge points in the test image. A comparison between the ground truth set and the edge detector results allows a quantitative analysis of the detector to be made. This comparison can be made through a computation process. A typical metric compares the edge results on a

pixel by pixel basis to the ground truth. The correspondence between the detector and ground truth results is measured through a figure of merit assigned to the detector. This Figure of Merit has a maximum of unity for a perfect detector, and a minimum of zero for a worst-case detector result.

A number of metrics designed for the quantitative evaluation of edge detector results and a ground truth set have been reported [82-86]. None of these has gained wide acceptance in the field of image processing. This is in part due to the processing overheads associated with their implementation and systematic errors produced by the edge detectors that give erroneous performance ratings. In Section 4.2 the metrics proposed by Pratt [82] and Kitchen-Rosenfeld [83] are examined. The examination reveals shortcomings in the metric's facility to deal with systematic errors generated by edge detectors and the ambiguous nature of their figure of merit zero condition.

A new metric labelled the Edge Point Metric (EPM) that was designed to address the shortcomings found in the Pratt and Kitchen-Rosenfeld metrics. A full description of this new metric is given in Section 4.3. Comparisons are drawn between the EPM and the Pratt and Kitchen-Rosenfeld metrics in Section 4.4. Edge point results given by the SUSAN Sobel and SLA edge detectors were used for the comparison tests. In Section 4.5 the EPM metric is used to optimise the SLA detector for use in vision based autonomous navigation.

4.2 Edge Point Metrics

4.2.1 Pratt Metric

In the past twenty years a number of quantitative methods for assessing the performance of edge detectors have been reported [82-86]. Unfortunately, none of the metrics has gained wide acceptance and as a result subjective analysis predominates in the assessment of edge detection algorithms. In this section, the Pratt [82] and the Kitchen-Rosenfeld [83] metrics are reviewed.

Pratt proposed a figure of merit for edge detectors that evaluated contributions from three types of error:-

A missed edge

An edge generated as a result of noise

The displacement of a valid edge.

This quantitative measure required the use of a ground truth set that contains the location of all valid edges. The Pratt metric requires:-

The Number of Ideal Edges (N_{IE}) in the image.

The Number of Actual Edges (N_{AE}).

The displacement d_i between the Actual Edge and the nearest ideal edge.

The Figure of merit (F) was defined as given in equation (4.1) where the Scaling Constant is set to ($C_S=0.111$) to penalise offset edges. At a value of unity F is a maximum and the detectors response is ideal. The figure of merit has a minimum of zero but the edge results necessary to give this minimum result were not specified.

$$F = \frac{1}{\max(N_{IE}, N_{AE})} \sum_{i=1}^{N_{AE}} \frac{1}{1 + C_S (d_i)^2} \quad (4.1)$$

The Pratt metric tests for edges at all orientations [82]. It provides quantitative results that allow for the comparison of two or more edge detectors on a test image, for which a ground truth set exists. If the result given by one of these detectors has a systematic displacement of one or two pixels then these displacements will result in a low figure of merit rating from the Pratt metric. The inability of the Pratt metric to

deal with systematic displacements led to the development of the Kitchen-Rosenfeld metric.

4.2.2 Kitchen and Rosenfeld Metric

A detector evaluation method was reported by Kitchen and Rosenfeld which relied upon local edge coherence to give the Evaluation measure ($E_{(image)}$). This metric does not need a ground truth set, but it requires a synthetic test image populated by vertical running edges [83]. The valid edges within this synthetic image are given as vertical lines. Thus there is no requirement for a ground truth, and lateral systematic displacements within the edge detector results are not registered as errors.

The value of $E_{(image)}$ is given by an average of local coherence results taken from each pixel site in the image calculated as $E_{(x,y)}$ in equation (4.2). In these $E_{(x,y)}$ varies from a maximum of one, where the edges form thin continuous vertical lines, to a minimum of zero. It was reported that the parameter γ should be set to 0.8 to give a best compromise between continuation and thinness.

$$E_{(x,y)} = \gamma C + (1 - \gamma)T \quad (4.2)$$

The Continuity value (C) and the Thinness value (T) are calculated separately for a 3x3 pixel neighbourhood given in Figure 4.1. The parameter (γ) is chosen to adjust the bias of E towards well connected edges or towards thin contours, γ may be set to any real value between zero and one. The continuity value C for each 3x3 neighbourhood is in equation (4.3) given by the average of the left and right continuity measures ($L(k_{max})$ and $R(k_{max})$).

$$C = (L(k_{max}) + R(k_{max}))/2 \quad (4.3)$$

The thinness value T is given by equation 4.6, where (N_R) is the Number of the Remaining edge pixels. These are the kernel edges not used in the evaluation of $(L(k_{max}))$ and $R(k_{max})$ in 4.4(a) and 4.4(b).

$$T = 1 - \frac{N_R}{6} \quad (4.6)$$

The Kitchen-Rosenfeld metric [83] overcomes the sensitivity of the Pratt metric [82] to systematic displacements by limiting the metric to the analysis of vertical lines within a synthetic test image. However, this limitation will have the effect of skewing the performance of detectors optimised through analysis given by the Kitchen-Rosenfeld metric to detect vertical lines.

The Kitchen-Rosenfeld and Pratt metrics provide quantitative figures of merit that range from unity to zero. At a figure of merit of unity the detector has a perfect performance and there are no false returns registered by the detector. However, at a figure of merit of zero there is no definition of the levels of false returns required to give rise to this performance rating. Indeed, it is difficult to devise a detector results set that would register a zero with either of these metrics.

4.3 Edge Point Metric

4.3.1 EPM Error Classification

The EPM figure of merit was designed to provide a means by which the performance of an edge detector could be judged against the specifications of a vision problem. In this the EPM's zero value was set to indicate when the detector's results were not adequate for the vision problem under review. There are seven types of pixel classification registered by the EPM. The error types are:-

True Positive (TP)

True Negative (TN)

False Positives (FP)

False Negatives (FN)

Displaced Positives (DP)

Displaced Negative (DN)

Wide Positives (WP)

The EPM figure of merit was based upon a linear scaled sum of the conditional probabilities of a *FP*, or a *FN* occurring within the detector results. Additional qualitative evaluation of the tested detectors was based on the conditional probabilities of a *DP*, or a *WP* occurring within the detector's results.

Edge point detectors operate on sampled images and assign edge points through a series of discrete operations. These operations can give rise to systematic displacements between the detected edge points and the image ground truth set, a typical systematic displacement being a shift of one or two pixels. These minor systematic shifts in the edge sets do not reduce the performance of a vision system, thus it is necessary that they are not classed error returns. As a result of sampling a detector may register an edge location at two adjacent pixels. This is line broadening and its effect on a vision system performance is determined by the post edge detection processes adopted by the system. If the *WP* returns are classed as errors then the detector evaluation can be compromised.

4.3.2 DGC Algorithm Structure

The Detector to Ground-truth Comparison algorithm (DGC) was designed to detect the presence of systematic displacements and line broadening within a detectors results through the application of a set of heuristics. An overview of the hierarchical structure of the DGC algorithm is given in Figure 4.2. The DGC algorithm employs three allocation phases to classify each pixel within the map into one of the seven states noted in Section 4.3.1.

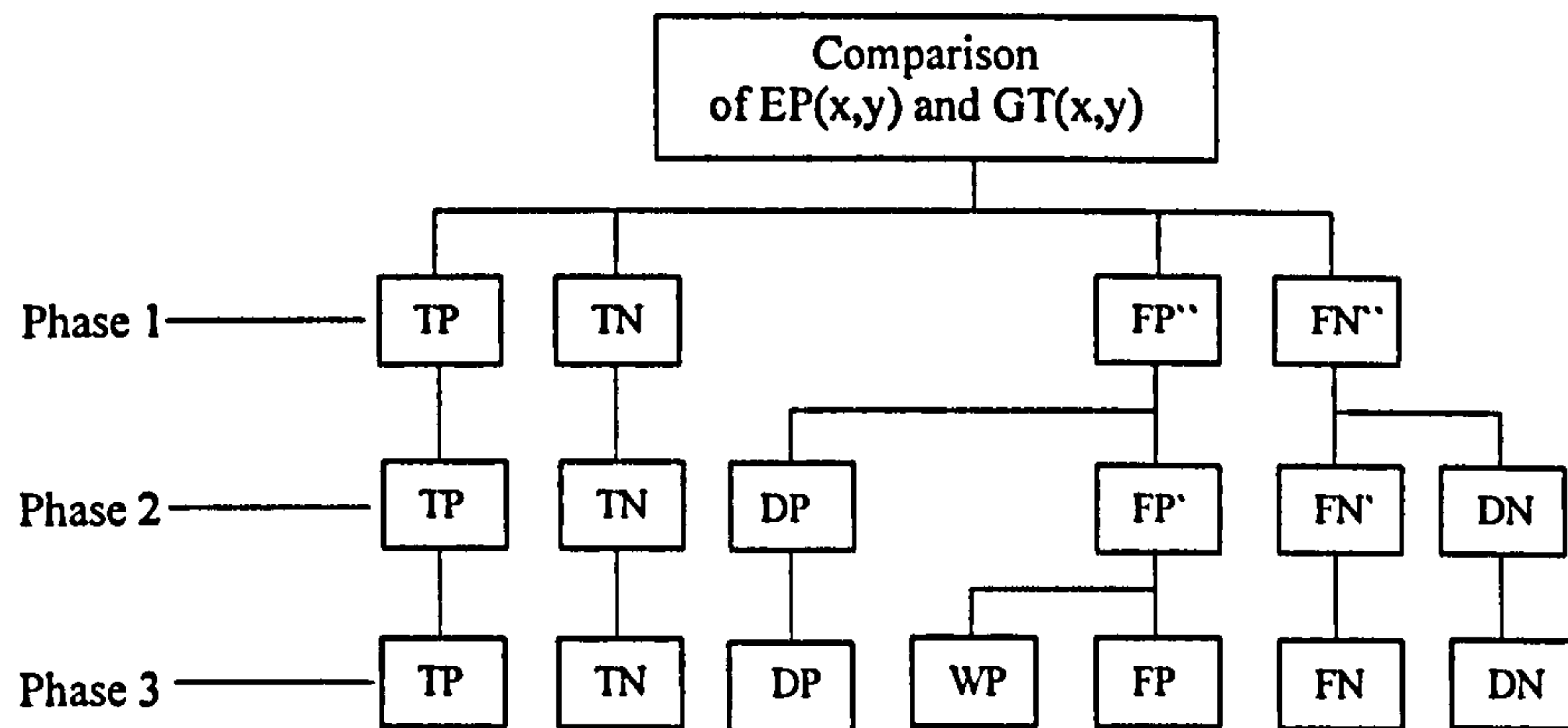


Figure 4.2 DGC Algorithm Decision Hierarchy

For a given intensity profile $I_{(x,y)}$ the DGC algorithm takes two input sets. An Edge Point set $EP_{(x,y)}$ generated by the application of an edge detector to the intensity profile, and a Ground Truth set $GT_{(x,y)}$ that marks all the valid edge points in the intensity profile.

The ground truth for the image profile $I_{(x,y)}$ is given as $GT_{(x,y)}$

$$GT_{(x,y)} = 1 \text{ for valid edge pixel}$$

$$GT_{(x,y)} = 0 \text{ for valid non edge pixel}$$

The detector edge set for the image profile $I_{(x,y)}$ is given as $EP_{(x,y)}$

$$EP_{(x,y)} = 1 \text{ for a detected edge point}$$

$$EP_{(x,y)} = 0 \text{ for no detected edge point.}$$

In order for comparisons to be made between the SLA, Sobel and SUSAN detectors it was necessary to remove the SLA edge sense information. This edge sense removal was achieved by assigning the value 1 to the SLA edge points registered as P or N in equations 3.22 and 3.23, and assigning the value 0 to the points registered as Z in equations 3.22 and 3.23. The Horizontal and Vertical results are then combined through a logical *OR* function to give a binary edge set for the image.

4.3.3 DGC Phase One

In the first phase the $EP_{(x,y)}$ and $GT_{(x,y)}$ sets are processed through a heuristic given by Truth Table 4.1 to give an interim $Map1_{(x,y)}$ populated by TP, TN, FP, FN states.

$EP_{(x,y)}$	$GT_{(x,y)}$	$Map1_{(x,y)}$
1	1	TP
0	0	TN
0	1	FN
1	0	FP

Table 4.1 Phase1 Heuristic Truth Table

In the second and third phases the DGC algorithm requires the 10 heuristic tests given by equations (4.7) to (4.16). These tests are performed on a pixel basis and they use the convolution mask illustrated in Figure 4.3. The **Central Pixel** to this kernel is labelled as PC . For each heuristic test, the assignment result is loaded into the PC location.

P14	P13	P12	P11	P10
P15	P3	P2	P1	P9
P16	P4	PC	P0	P8
P17	P5	P6	P7	P23
P18	P19	P20	P21	P22

Figure 4.3 DGC Convolution Kernel

4.3.4 DGC Phase Two

In the second phase, $Map1_{(x,y)}$ is processed through the heuristics to allocate DP and DN states within the new image $Map2_{(x,y)}$. In these heuristics the false returns from the first phase are tested for displacements of one or two pixels from uncovered ground truth pixels. Where a displacement exists the FP return is reallocated as a DP state and the FN return is reallocated as a DN state. The TP and TN returns from the first phase are unchanged by the heuristics used in the second phase.

Phase 2 Test 1 uses the convolution kernel elements highlighted in Figure 4.4. The test heuristic, given in equation (4.7a), tests for a 4-connected displaced positive. The heuristic, given in equation (4.7b), tests for a 4-connected displaced negative.

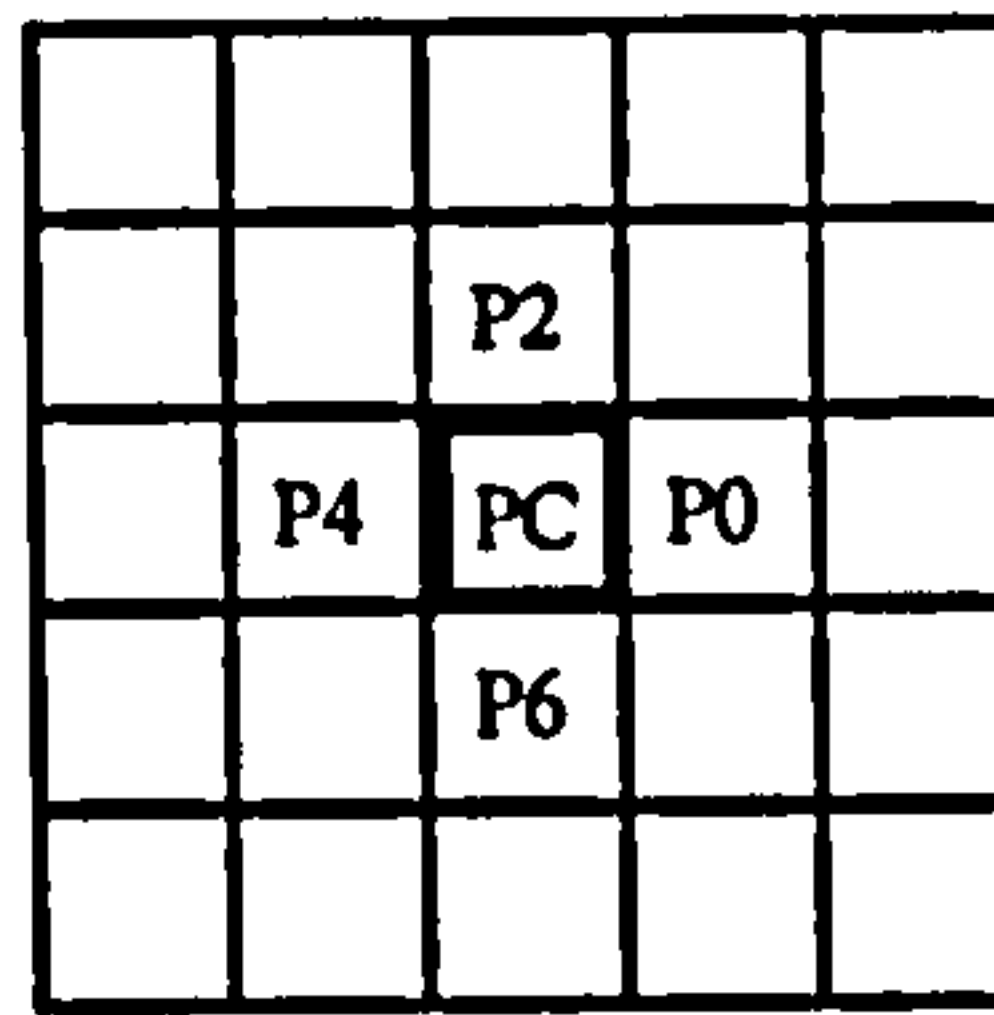


Figure 4.4 Kernel for Phase 2 Test 1

$$AssignDP \quad if \quad \left\{ (PC = FP) \text{ and } \left\{ \begin{array}{l} (P0 = FN) \\ or (P2 = FN) \\ or (P4 = FN) \\ or (P6 = FN) \end{array} \right\} \right\} \quad (4.7a)$$

$$AssignDN \quad if \quad \left\{ (PC = FN) \text{ and } \left\{ \begin{array}{l} (P0 = FP) \\ or (P2 = FP) \\ or (P4 = FP) \\ or (P6 = FP) \end{array} \right\} \right\} \quad (4.7b)$$

Phase 2 Test 2 uses the convolution kernel elements highlighted in Figure 4.5. The test employs two heuristic given in equations (4.8a) and (4.8b). The heuristics assign

DP and DN values to the central pixel for a two-pixel displacement to the *right* direction.

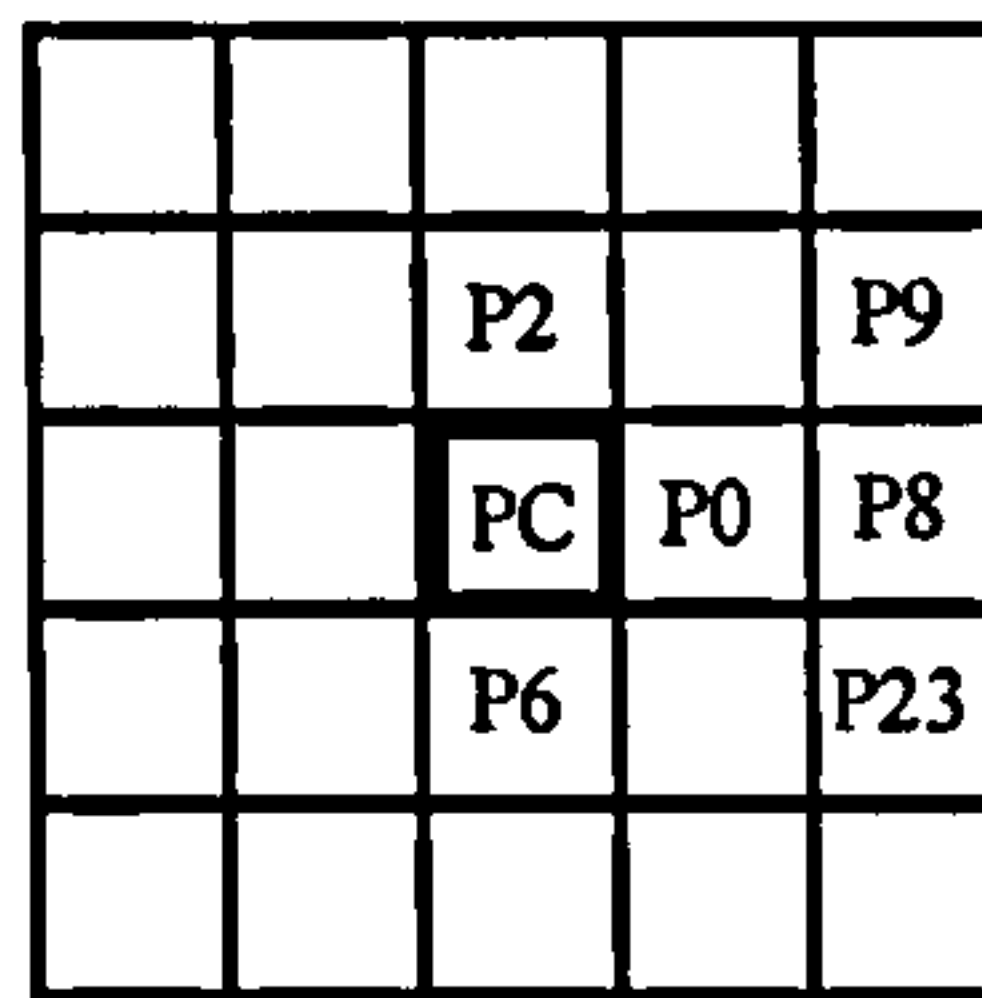


Figure 4.5 Kernel for Phase 2 Test 2

$$AssignDP \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FP) \\ and(P0 = TN) \\ and(P8 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P2 = FP) \\ or(P6 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P9 = FN) \\ or(P23 = FN) \end{array} \right\} \right\} \quad (4.8a)$$

$$AssignDN \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FN) \\ and(P0 = TN) \\ and(P8 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P2 = FN) \\ or(P6 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P9 = FP) \\ or(P23 = FP) \end{array} \right\} \right\} \quad (4.8b)$$

Phase 2 Test 3 uses the convolution kernel elements highlighted in Figure 4.6. The test employs two heuristic given in equations (4.9a) and (4.9b). The heuristics assign DP and DN values to the central pixel for a two-pixel displacement in the *up* direction.

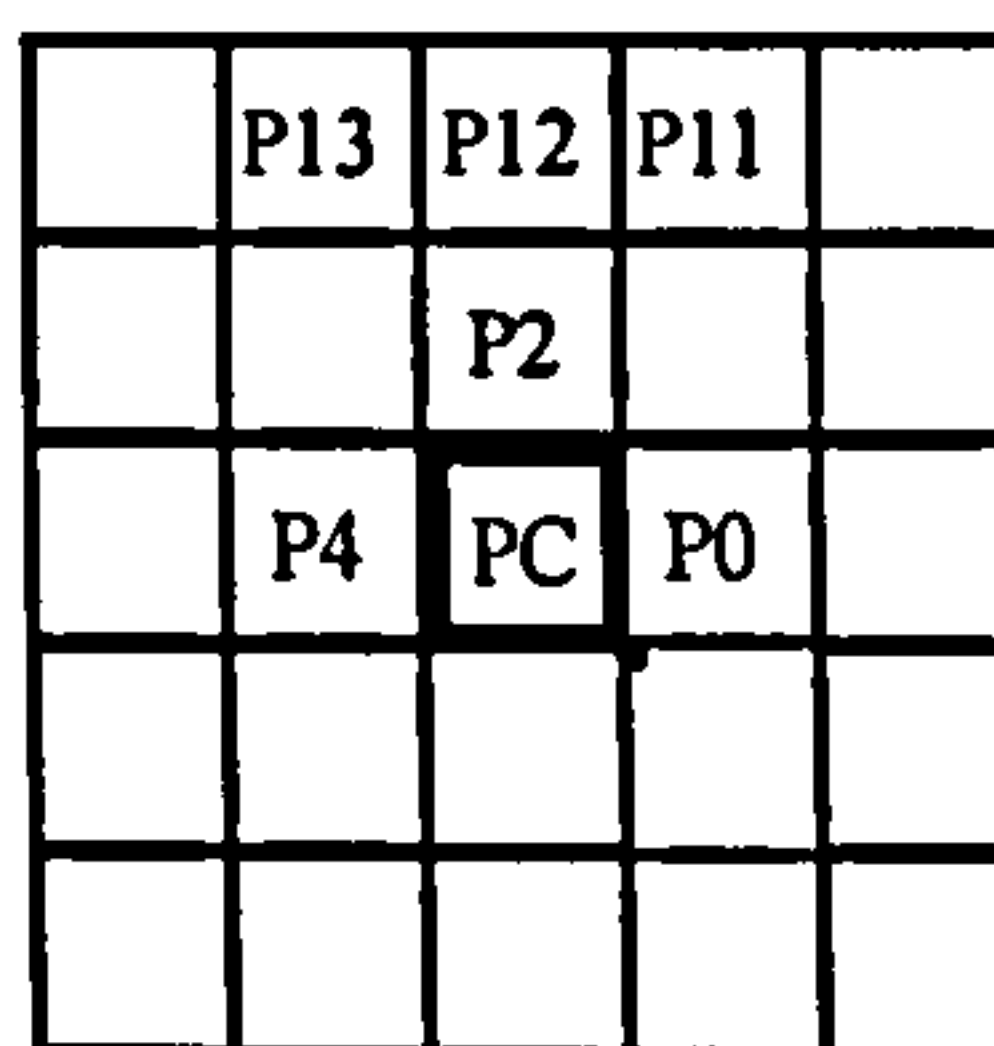


Figure 4.6 Kernel for Phase 2 Test 3

$$AssignDP \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FP) \\ and(P2 = TN) \\ and(P12 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P0 = FP) \\ or(P4 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P11 = FN) \\ or(P13 = FN) \end{array} \right\} \right\} \quad (4.9a)$$

$$AssignDN \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FN) \\ and(P2 = TN) \\ and(P12 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P0 = FN) \\ or(P4 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P11 = FP) \\ or(P13 = FP) \end{array} \right\} \right\} \quad (4.9b)$$

Phase 2 Test 4 uses the convolution kernel elements highlighted in Figure 4.7. The test employs two heuristic given in equations (4.10a) and (4.10b). The heuristics assign *DP* and *DN* values to the central pixel for a two pixel displacement in the *left* direction.

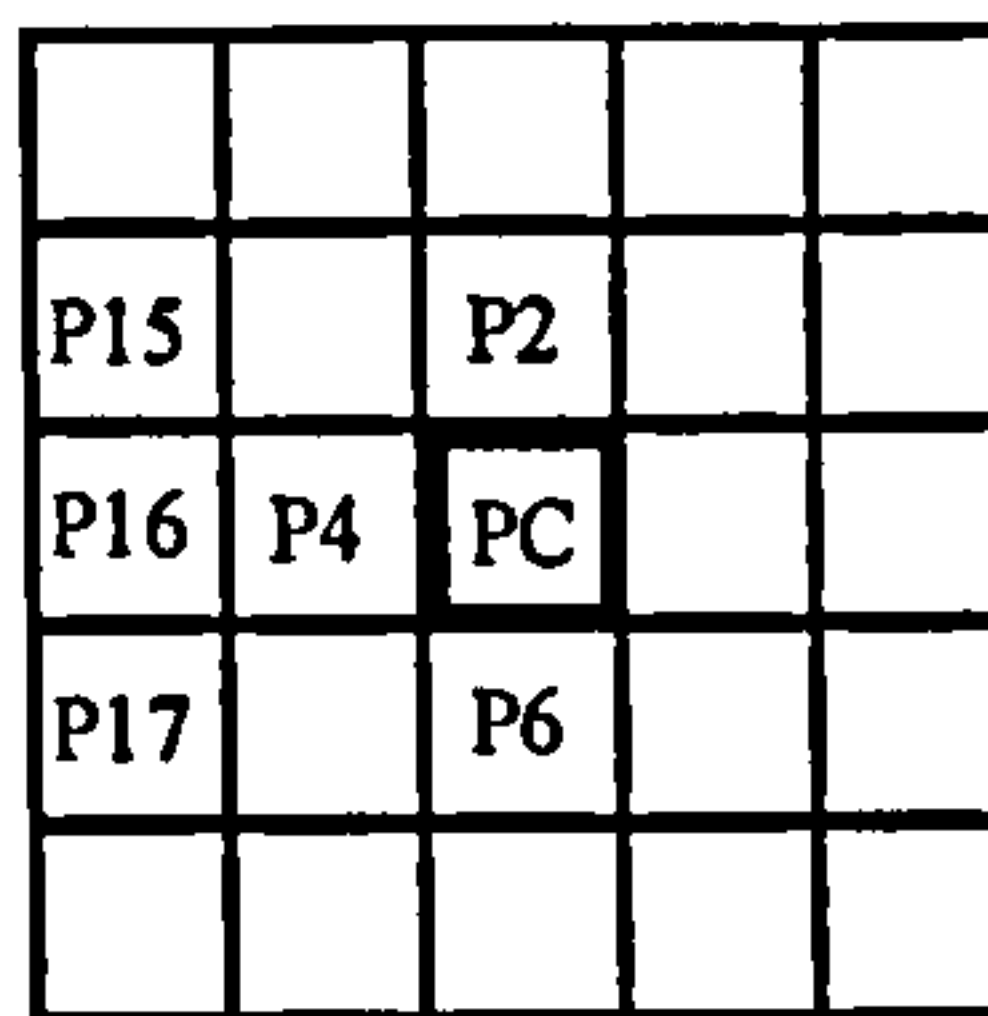


Figure 4.7 Kernel for Phase 2 Test 4

$$AssignDP \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FP) \\ and(P4 = TN) \\ and(P16 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P2 = FP) \\ or(P6 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P15 = FN) \\ or(P17 = FN) \end{array} \right\} \right\} \quad (4.10a)$$

$$AssignDN \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FN) \\ and(P4 = TN) \\ and(P16 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P2 = FN) \\ or(P6 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P15 = FP) \\ or(P17 = FP) \end{array} \right\} \right\} \quad (4.10b)$$

Phase 2 Test 5 uses the convolution kernel elements highlighted in Figure 4.8 The test employs two heuristic given in equations (4.11a) and (4.11b). The heuristics assign *DP* and *DN* values to the central pixel for a two-pixel displacement in the *down* direction.

	P4	PC	P0	
		P6		
	P19	P20	P21	

Figure 4.8 Kernel for Phase 2 Test 5

$$AssignDP \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FP) \\ and(P6 = TN) \\ and(P20 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P4 = FP) \\ or(P0 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P19 = FN) \\ or(P21 = FN) \end{array} \right\} \right\} \quad (4.11a)$$

$$AssignDN \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FN) \\ and(P6 = TN) \\ and(P20 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P4 = FN) \\ or(P0 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P19 = FP) \\ or(P21 = FP) \end{array} \right\} \right\} \quad (4.11b)$$

Phase 2 Test 6 uses the convolution kernel elements highlighted in Figure 4.9. The test employs two heuristic given in equations (4.12a) and (4.12b). The heuristics assign *DP* and *DN* values to the central pixel for a diagonal displacement in the *up-right* direction.

		P12		
	P3		P1	
		PC		P8
			P7	

Figure 4.9 Kernel for Phase 2 Test 6

$$AssignDP \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FP) \\ and(P1 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P3 = FP) \\ or(P7 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P12 = FN) \\ or(P8 = FN) \end{array} \right\} \right\} \quad (4.12a)$$

$$AssignDN \text{ if } \left\{ \left\{ \begin{array}{l} (PC = FN) \\ and(P1 = FP) \end{array} \right\} and \left\{ \begin{array}{l} (P3 = FN) \\ or(P7 = FN) \end{array} \right\} and \left\{ \begin{array}{l} (P12 = FP) \\ or(P8 = FP) \end{array} \right\} \right\} \quad (4.12b)$$

Phase 2 Test 7 uses the convolution kernel elements highlighted in Figure 4.10. The test employs two heuristic given in equations (4.13a) and (4.13b). The heuristics assign DP and DN values to the central pixel for a diagonal displacement in the *up-left* direction.

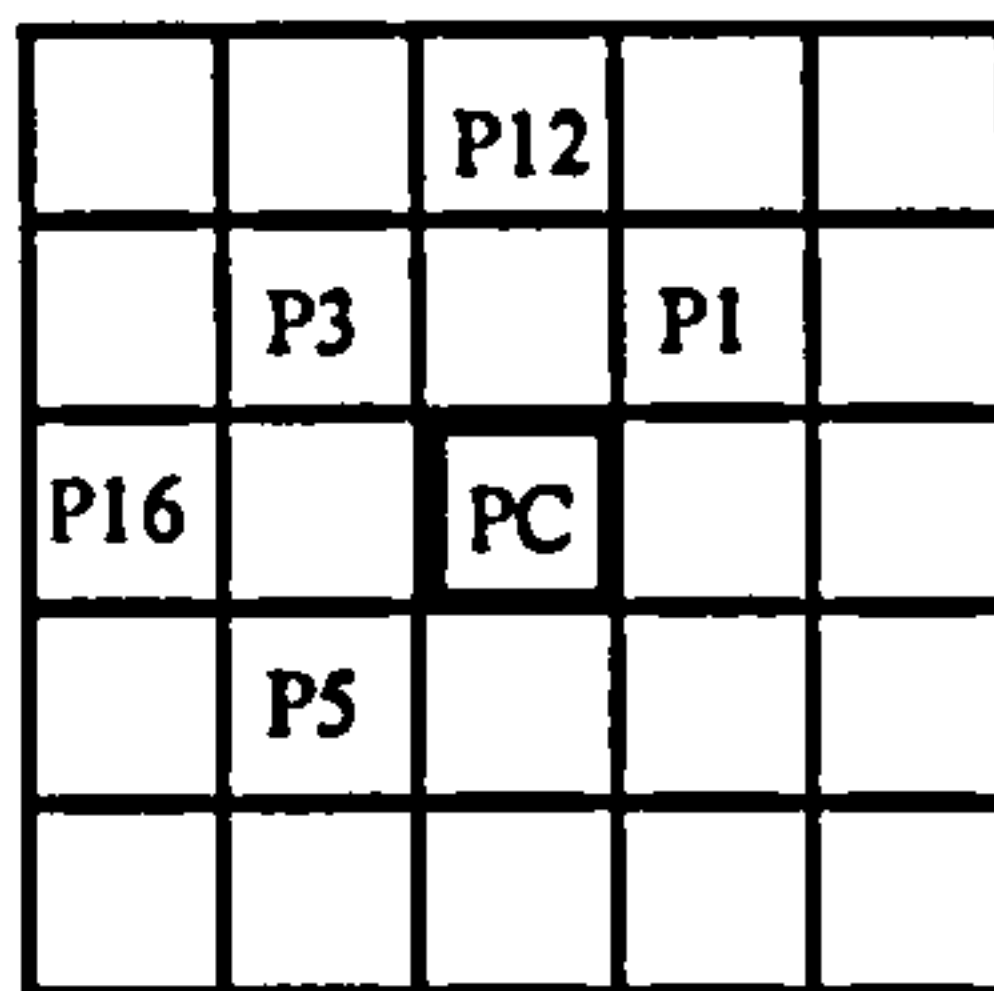


Figure 4.10 Kernel for Phase 2 Test 7

$$AssignDP \quad if \quad \left\{ \left\{ (PC = FP) \right\} \right. \left. \left\{ \left\{ and(P3 = FN) \right\} \right\} \right. \left. \left\{ \left\{ (P1 = FP) \right\} \right\} \right. \left. \left\{ \left\{ and(P5 = FP) \right\} \right\} \right. \left. \left\{ \left\{ (P12 = FN) \right\} \right\} \right. \left. \left\{ \left\{ and(P16 = FN) \right\} \right\} \right\} \quad (4.13a)$$

$$AssignDN \quad if \quad \left\{ \left\{ (PC = FN) \right\} \right. \left. \left\{ \left\{ and(P3 = FP) \right\} \right\} \right. \left. \left\{ \left\{ (P1 = FN) \right\} \right\} \right. \left. \left\{ \left\{ and(P5 = FN) \right\} \right\} \right. \left. \left\{ \left\{ (P12 = FP) \right\} \right\} \right. \left. \left\{ \left\{ and(P16 = FP) \right\} \right\} \right\} \quad (4.13b)$$

Phase 2 Test 8 uses the convolution kernel elements highlighted in Figure 4.11. The test employs two heuristic given in equations (4.14a) and (4.14b). The heuristics assign DP and DN values to the central pixel for a diagonal displacement in the *down-left* direction.

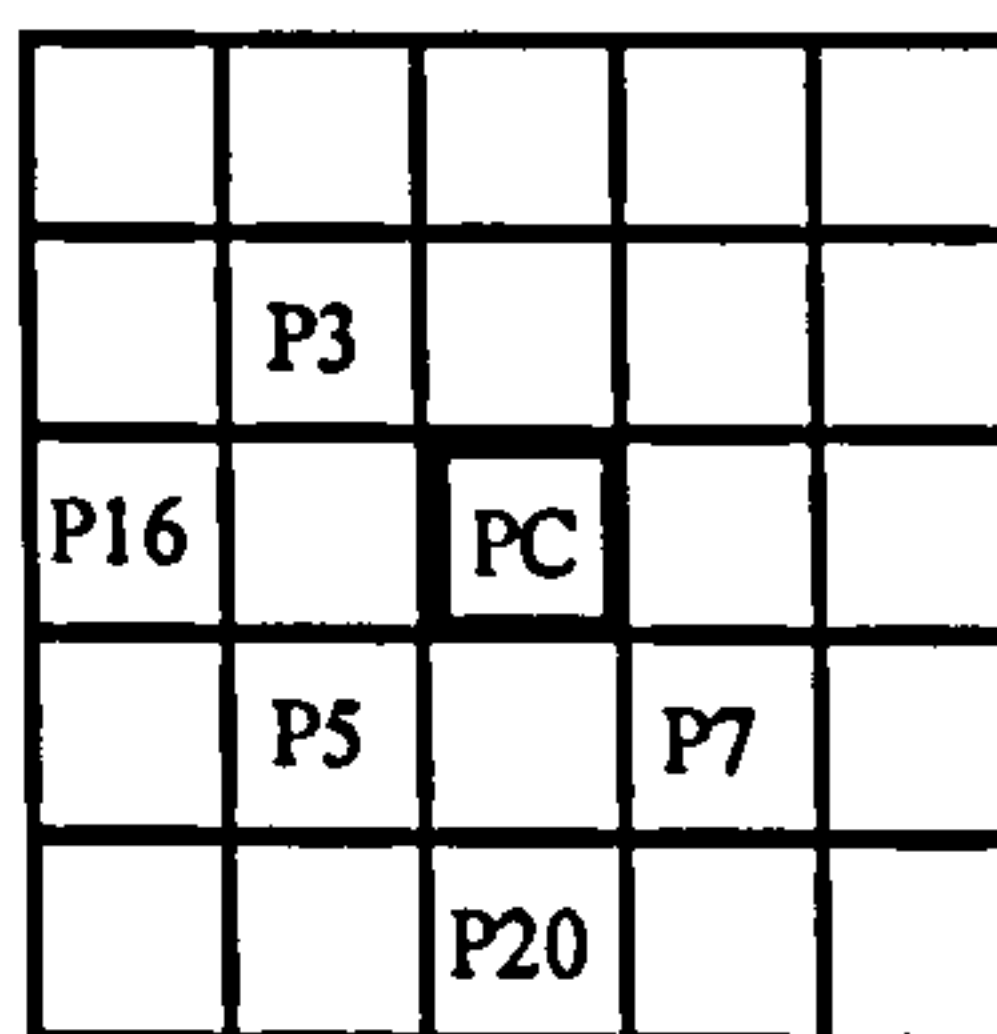


Figure 4.11 Kernel for Phase 2 Test 8

$$AssignDP \quad if \quad \left\{ \left\{ (PC = FP) \right\} \right. \left. \left\{ \left\{ and(P5 = FN) \right\} \right\} \right. \left. \left\{ \left\{ or(P7 = FP) \right\} \right\} \right. \left. \left\{ \left\{ or(P20 = FN) \right\} \right\} \right\} \quad (4.14a)$$

$$AssignDN \text{ if } \left\{ \left\{ (PC = FN) \right\} \text{and} \left\{ (P3 = FN) \right\} \text{and} \left\{ (P16 = FP) \right\} \right\} \left\{ \text{and}(P5 = FP) \right\} \text{and} \left\{ \text{or}(P7 = FN) \right\} \text{and} \left\{ \text{or}(P20 = FP) \right\} \right\} \quad (4.14b)$$

Phase 2 Test 9 uses the convolution kernel elements highlighted in Figure 4.12. The test employs two heuristic given in equations (4.15a) and (4.15b). The heuristics assign *DP* and *DN* values to the central pixel for a diagonal displacement in the *down-right* direction.

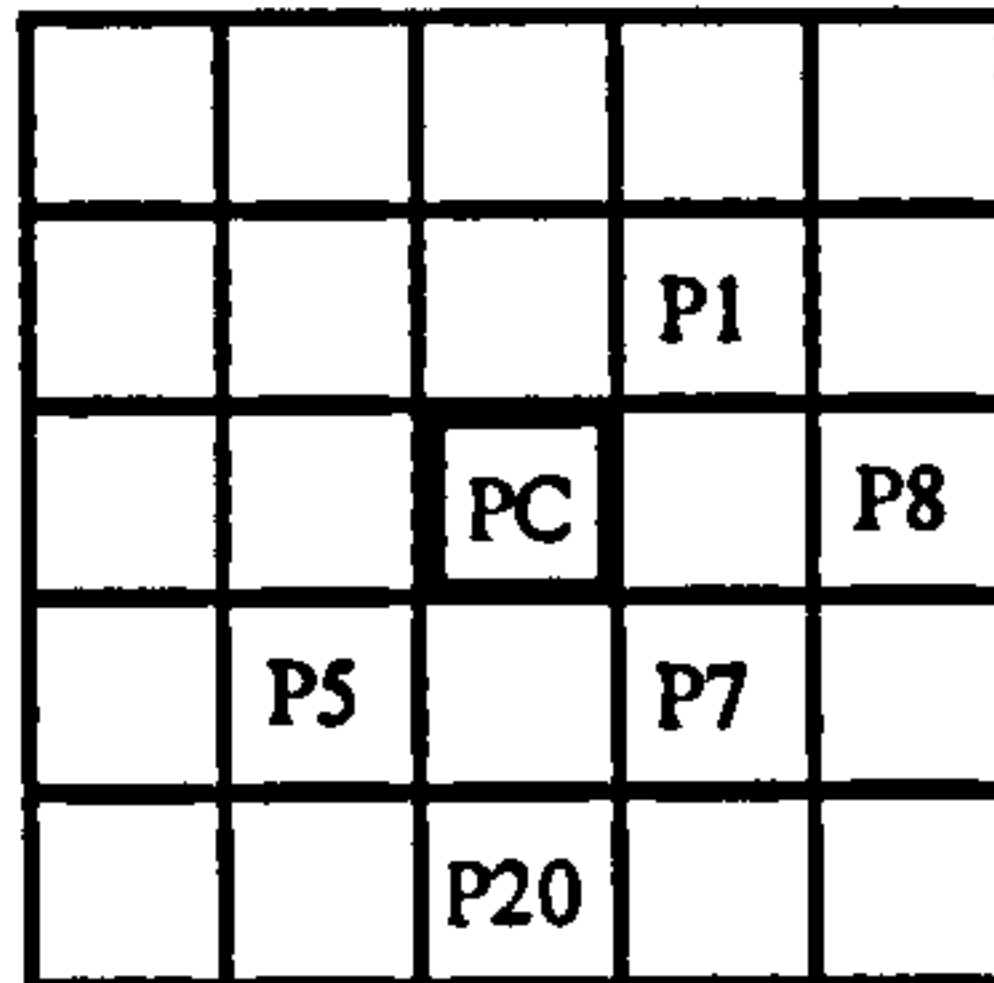


Figure 4.12 Kernel for Phase 2 Test 9

$$AssignDP \text{ if } \left\{ \left\{ (PC = FP) \right\} \text{and} \left\{ (P1 = FP) \right\} \text{and} \left\{ (P8 = FN) \right\} \right\} \left\{ \text{and}(P7 = FN) \right\} \text{and} \left\{ \text{or}(P5 = FP) \right\} \text{and} \left\{ \text{or}(P20 = FN) \right\} \right\} \quad (4.15a)$$

$$AssignDN \text{ if } \left\{ \left\{ (PC = FN) \right\} \text{and} \left\{ (P1 = FN) \right\} \text{and} \left\{ (P8 = FP) \right\} \right\} \left\{ \text{and}(P7 = FP) \right\} \text{and} \left\{ \text{and}(P5 = FN) \right\} \text{and} \left\{ \text{and}(P20 = FP) \right\} \right\} \quad (4.15b)$$

4.3.5 DGC Phase Three

In the third phase the *FP* returns of $Map2_{(x,y)}$ are tested to see if they can be allocated as width modulation pixels. In this a *FP* return that increases the width of the detected line is reallocated as a Wide Positives (*WP*). The DGC algorithm Phase 3 uses a single heuristic test. The convolution kernel for this test is illustrated in Figure 4.13. This heuristic applies the 8-connected test of equation (4.16), to check for line broadening pixels, these are assigned the *WP* value. The *TP*, *TN*, *DP*, *FN* and *DN* returns in $Map2_{(x,y)}$ are not changed by the operation of this heuristic. The results from the third heuristic phase are loaded into $Map3_{(x,y)}$.

	P3	P2	P1	
	P4	PC	P0	
	P5	P6	P7	

Figure 4.13 Kernel for Phase 3 Test

$$\text{AssignWP} \quad \text{if} \quad \left\{ (PC = FP) \text{and} \left\{ \begin{array}{l} (P0 = TP) \text{or} (P0 = DP) \\ \text{or} (P1 = TP) \text{or} (P1 = DP) \\ \text{or} (P2 = TP) \text{or} (P2 = DP) \\ \text{or} (P3 = TP) \text{or} (P3 = DP) \\ \text{or} (P4 = TP) \text{or} (P4 = DP) \\ \text{or} (P5 = TP) \text{or} (P5 = DP) \\ \text{or} (P6 = TP) \text{or} (P6 = DP) \\ \text{or} (P7 = TP) \text{or} (P7 = DP) \\ \text{or} (P8 = TP) \text{or} (P8 = DP) \end{array} \right\} \right\} \quad (4.16)$$

4.3.6 DGC Example Results

The operation of the DGC algorithm is illustrated in Figure 4.14. In Figure 4.14 the dashed line that crosses the 5x6 pixel-grid, marks the hairline separation of the two regions of differing intensity. The pixels with bold outlines mark the ground truth pixels for this intensity discontinuity. The grey filled pixels in Figures 4.14(a), (b) and (c) illustrate results from three different detectors with systematic shifts, line broadening and noise related returns. The DGC allocation of states is given by the labels assigned to the Figure 4.14 pixels. In order to highlight the operation of the algorithm, the *TN* assignment labels have been omitted, so that all unlabeled pixels are in the *TN* state.

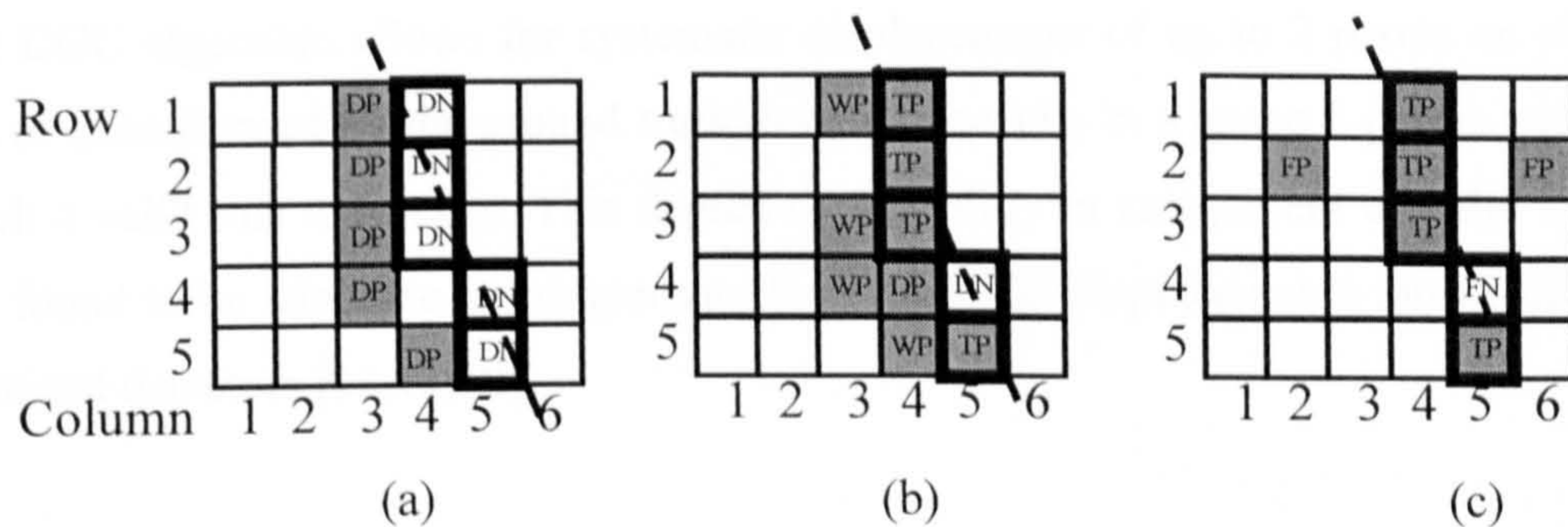


Figure 4.14 DGC Algorithm results for three edge detectors. (a) detector with systematic shift, (b) detector with line broadening, (c) detector with false returns.

In Figure 4.14(a) the detector's results are shifted to the left of the ground truth pixels. The DGC algorithm employs the 4-connected heuristic test of equation 4.7 to match the single shifted pixels to adjacent ground truth pixels and gives the *DP* and *DN* states of rows 1, 2 and 5. The edge pixel at row 4, column 3 or (4,3) is displaced by two pixels from the (4,5) ground truth location. This pixel is allocated to the *DP* state through a double pixel displacement heuristic given by equation (4.10). Similarly, the uncovered ground truth pixel (4,5) is part of a line segment displaced by two pixels from a vertical line segment and this is assigned to the *DN* state. Figure 4.14(b) illustrates results from a detector that generates line-broadening pixels. A *FP* return that is adjacent to a *TP* or *DP* pixel on an 8-connected test is reallocated to the *WP* state.

The false returns that remain in Figure 4.14(c) after the DGC reallocation indicate errors in the image segmentation. These false returns contribute to a reduction in the usefulness of the edge detector. In contrast the detector results of Figure 4.14(a) and (b) have no false returns remaining after the reallocation phases, and these give complete segmentations of the original image. An assessment of the quality of these two complete segmentations shows Figure 4.14(b) to be of lower quality, because it has a higher density of width modulation pixels. The width modulation pixels disrupt the operation of line detection processes. These pixels may be removed through thinning processes. The high density of displacement pixels in the Figure 4.14(a) results have a relatively low effect on the quality of segmentation as long as the image is over sampled.

The DGC algorithm allows for systematic displacements of up to 2 pixels on either side of lines formed by the ground truth line. This results in a space 5 pixels wide in which a valid line may exist. This degree of flexibility in assignment of valid edges was found to be adequate to compensate for systematic displacements introduced by the tested detectors [11,12,80].

4.4 EPM Figure of Merit

4.4.1 Minimum Quality Specification

Förstner [16] expressed the relationship between the edge detection algorithm and the minimum quality of result through equation (4.17).

$$q(r|d;a,t) \geq q_0 \quad (4.17)$$

Where:-
 q = the qualitative result for a given test
 q_0 = the minimum quality that can be tolerated by the vision system.
 r = the edge detector result
 a = the edge detector algorithm
 d = the test data set
 t = the tuning parameters

The analysis is repeated for a series of data sets that encompass the full range of edge characteristics encountered by the vision system. The data sets are drawn up through inspection of the images encountered by the vision system. The algorithm, the tuning factors and the minimum quality levels are fixed for a given empirical assessment.

The EPM metric was designed to conform to the accepted practice of generating a figure of merit that ranged between unity and zero. A perfect detector would register a figure of merit equal to unity. However, it was decided to allocate Förstner's [16] minimum quality level to the zero result for the figure of merit. This allows the EPM to act as an edge metric and to implement Förstner's minimum quality test. Thus a detector that did not conform to the system requirements would register a negative figure of merit. This approach was adopted because it exploited the full dynamic range of the metric and provided an unambiguous representation of the host system specifications.

4.4.2 EPM Scale Factors

The SLA structural detector described in Section 3.3 presents a configuration problem common to many threshold dependent edge detectors, in that a compromise needs to be sought between the detectors susceptibility to False Negative and False Positive

returns. Thus in a metric that assesses the performance of the detector, the relative significance of the two types of false returns needs to be considered. This was achieved in the EPM metric by separately scaling the probabilities of the false returns before combining these results to give the EPM figure of merit.

Analysed image results established that the extraction of valid line segments from the SLA detector of Section 3.3 was mainly limited through clusters of false returns. In the analysed results, false negative returns were found to cluster to form extended breaks in the image results and the false positive returns clustered to form false line segments. These error clusters present a significant problem for the line extraction process. As breaks occur in the valid outlines, this can result in sections of the outlines being removed by the minimum line length threshold (Section 3.4). However if the minimum length threshold is reduced then false line segments, due to groupings of false edge points, will be retained within the extracted line results and corrupt the operation of the pose recovery algorithm.

An examination of the effect of error clustering reveals that the *FP* errors are four times more likely to link and form an error cluster than the *FN* errors. The *FP* have the facility to connect with eight adjacent pixels whereas the false negatives are limited to connecting with two adjacent pixels. This higher degree of connectivity in the *FP* returns means that the metric needs to apply higher weighting to these errors in the evaluation of the figure of merit. It should be noted that if the false returns occur as isolated single pixel errors, then a simple morphological process can be used to remove these errors.

The autonomous vision problem of Section 3.6 matched image line segments to model estimates of structural features. The model estimates were required to be of the order of 50 pixels long to ensure a pose recovery that was adequate for the task of navigating along a corridor. It was decided that the minimum quality level for the matched line segments should be set to one pixel in six pixels being missed. Thus as a result of error clustering, an image line of 50 pixels length could be split into a maximum four sections. This minimum quality level can also be expressed in terms of

the probability of a *FN* occurring as 0.167. Under the rule that the *FP* returns are four times more likely to link and form false line segments, a minimum level of 0.042 was assigned to the *FP* error probability in the autonomous navigation problem.

4.4.3 Figure of Merit Evaluation

The EPM figure of merit is calculated from the DGC results using equation (4.18). This equation employs two scaling factors that produce a result of zero or less when the false negative probability is 0.167, or the false positive probability is 0.042. The scaling factor, S_1 , of equation (4.18) is set to 6. The scale factor S_2 is set to 4 to reflect the relative weighting of the false positive and false negative returns. The conditional probabilities for a false positive $P(FP)$ and a false negative $P(FN)$ were calculated from the DGC results as given in equations (4.19) and (4.20). The Totals of the *TP*, *TN*, *FP*, *FN*, *DP*, *DN* and *WP* were found by accumulating the number of pixels in each of these states in the detector results after the application of the DGC algorithm.

$$EPM = 1 - S_1(P(FN) + S_2 P(FP)) \quad (4.18)$$

$$P(FN) = \frac{TotalFN}{TotalTP + TotalFN + TotalDP} \quad (4.19)$$

$$P(FP) = \frac{TotalFP}{TotalTN + TotalFP + TotalDN} \quad (4.20)$$

Equation (4.19) and equation (4.20) demonstrate the function of the DGC algorithm. It reclassifies the false positive returns as *DP* returns. These are then treated as true positives in the $P(FN)$ probability evaluation. The false negative returns were reclassified as *DN*. These are then treated as true negatives in the $P(FP)$ probability evaluation. However, it is important to know the degree to which these effects occur within the detector's results. Thus the EPM figure of merit results were augmented through the evaluation of the conditional probabilities of width modulation and edge displacement. The width modulation quality measure is given by the $P(WP)$ evaluation of equation (4.21). The displacement quality measure is given by the $P(DP)$ evaluation of equation (4.22).

$$P(WP) = \frac{TotalWP}{TotalTP + TotalWP + TotalDP} \quad (4.21)$$

$$P(DP) = \frac{TotalDP}{TotalTP + TotalWP + TotalDP} \quad (4.22)$$

The combination of the DGC algorithm and the EPM evaluations provided a means for evaluating the performance of an edge detector against a problem specification. There are two methods of deriving test images for this evaluation. In the first, captured images are used and hand editing of these provides the ground truth against which the detectors performance is assessed [88,90]. In the second, a graphical draw package is used to create synthetic test images and the ground truth is extracted from the image hairline outlines [82,85]. In the captured image method, the hand edit is labour intensive and the validity of the assessment is dependent upon the choice of test images. In the synthetic image method, validity of the assessment is dependent upon whether the characteristics of the constructed test edges are representative of the edges encountered in the captured images.

Synthetic test images were used in the analysis of edge detectors for the autonomous navigation problem. These synthetic test images were constructed from analysis of the edge results taken from captured images. Edge profiles representative of the autonomous navigation problem were used to populate the test images. Noise with a Gaussian profile and a zero mean was added to the test images to exercise the detectors over the SNR range noted in the captured image results. The *SNR* in *dB* evaluation is given by equation (4.23), where *d* is the depth of the intensity discontinuity and σ is the standard deviation of added noise.

$$SNR = 20 \log\left(\frac{d}{\sigma}\right) \quad (4.23)$$

4.5 Metric Results Edge Detector Comparisons

4.5.1 Synthetic Image for Metric Comparison

In this section, comparisons are drawn between Pratt, Kitchen-Rosenfeld and EPM metrics [82,83]. A synthetic vertical bar image was generated for these comparisons. The synthetic test image, exclusive of added noise, is illustrated in Figure 4.15(a). The profile from the cross section line AA is shown in Figure 4.15(c). The profile steps oscillate about the mean grey level of 127. The step amplitude is set to 13 grey levels to give a 10% contrast shift at each edge point. Analysis of captured images from corridor environments established this 10% level as a minimum contrast level that marked structural outlines. A ground truth set was created by extracting the hairline outlines from the vertical bar image. As the Kitchen-Rosenfeld metric [83] is limited to the analysis of vertical lines, the vertical bar test image ensured that useful comparisons could be drawn between the metrics. The Pratt [82] and the EPM metrics are capable of analysing edge points at any orientation.

Noise with a zero mean and Gaussian distribution was added to the vertical bar image to give a series of test images. The standard deviation σ of the added noise was varied from 1 to 10 grey levels to create a total of ten test images. The image of Figure 4.15(b) has a standard deviation of 8 grey levels. A vertical striation component was included in the added noise to exercise the facility of the detectors to correctly locate the vertical lines of edges.

The detectors chosen for these comparison tests were the SLA algorithm, detailed in Chapter 3, the Sobel detector [12] and the SUSAN detector [80]. The Sobel detector is a low connectivity derivative based detector. This detector has limited scope for dealing with the added noise in the synthetic image tests. However, it is similar to the SLA detector in that it employs integer coefficients and relatively few product operations. The SUSAN detector is an area-based detector that also relies on integer operations to locate edge points. The degree of connectivity used by the SUSAN detector matches that used by the SLA implementation.

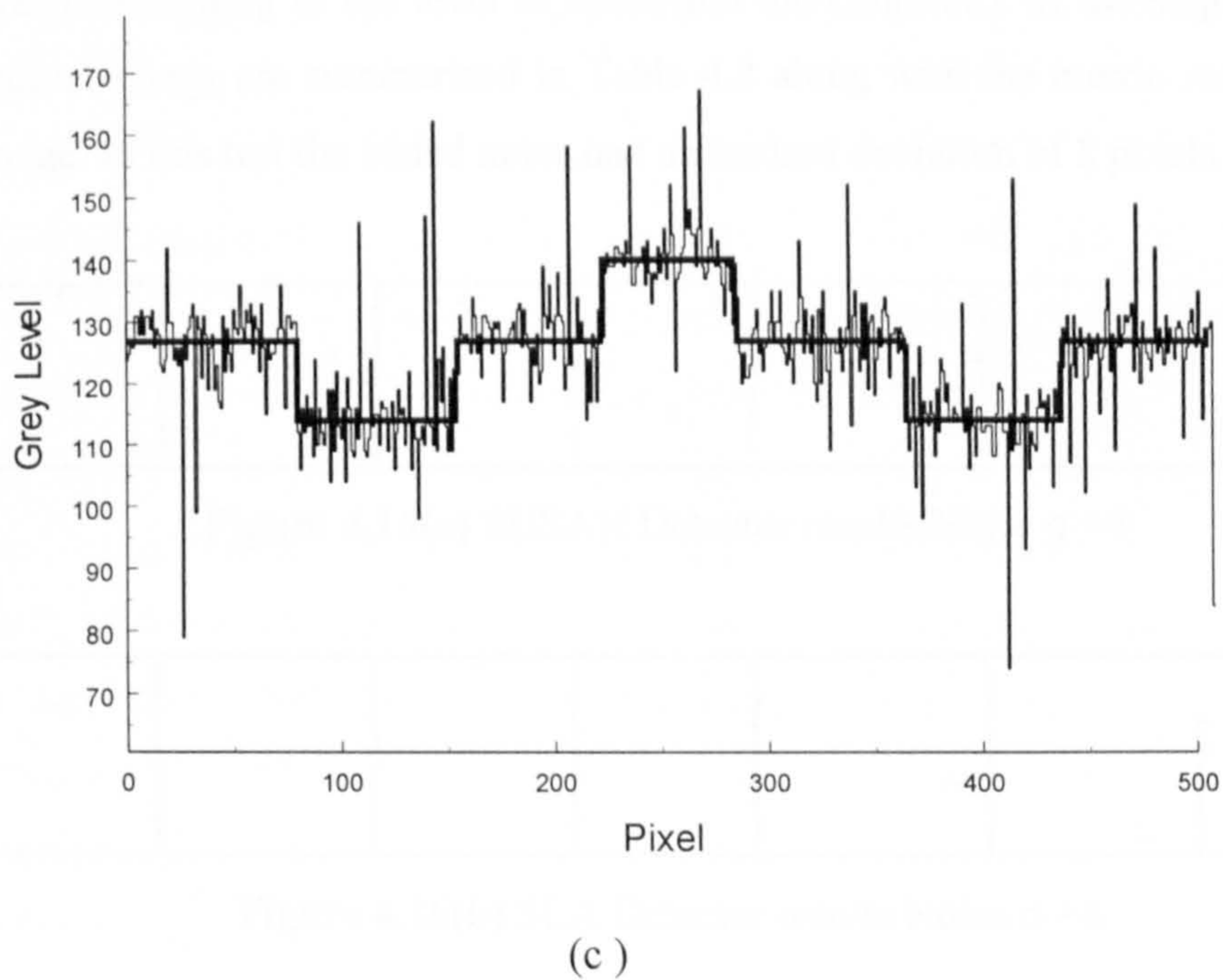
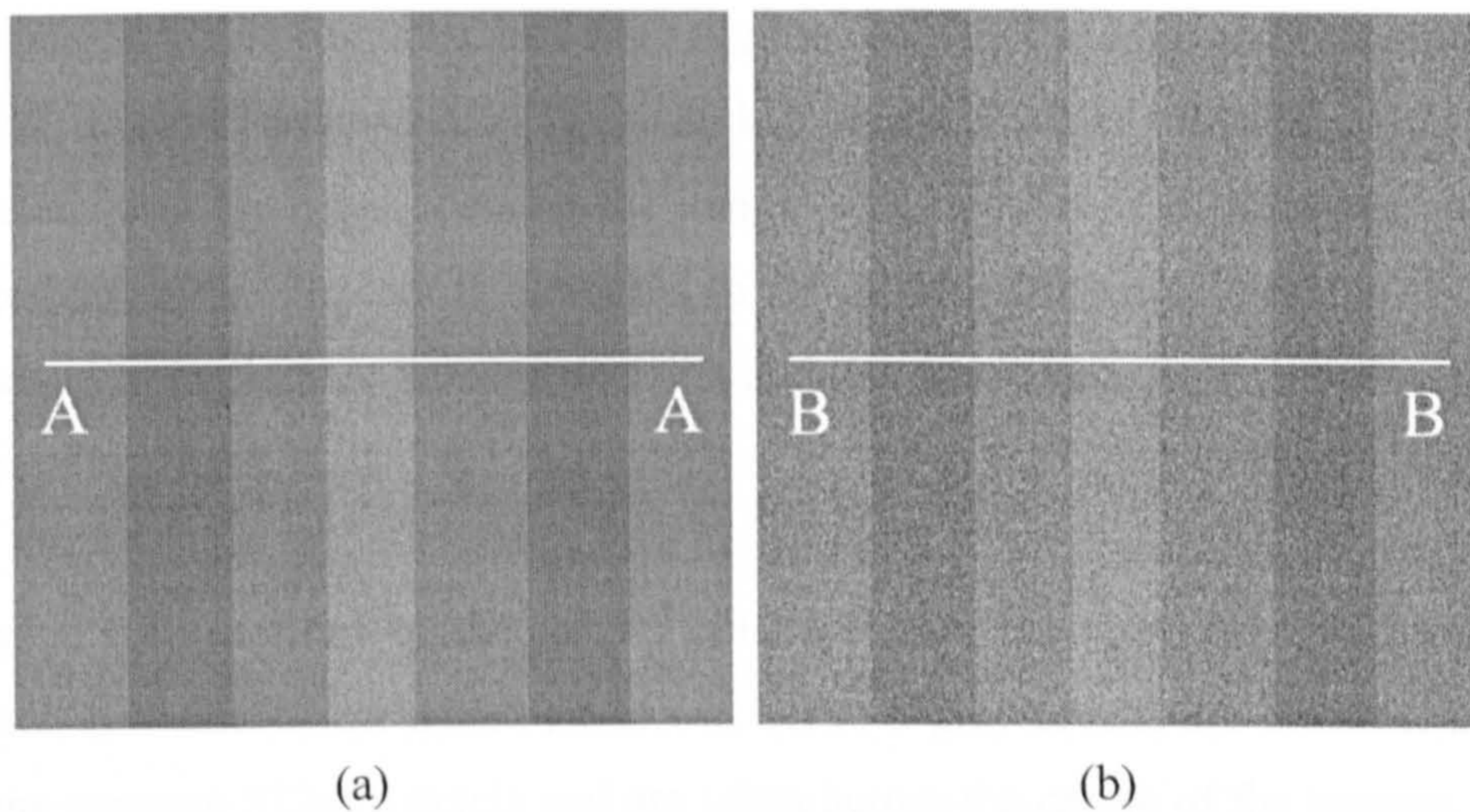


Figure 4.15 Vertical Bar Synthetic Test Image. (a) Vertical Bar no added Noise (b) Vertical Bar Noise $\sigma = 8$, (c) Cross Section Profiles AA and BB

4.5.2 Qualitative Analysis of SLA, SUSAN and Sobel Detectors

The SLA configuration was set to uniform filter length of 6 pixels and the convolution width was set to 7 pixels. This gave rise to a detector that covered an area of 15x7

pixels and 56 pixels from this area were used to evaluate the edge point locations. The SUSAN detector employed a 37 pixel kernel upon which the edge decision is computed [80]. By pre-processing the image with a 5x5 median filter, the effective connectivity of the detector is increased to 109 pixels. The Sobel detector employs a 3x3 kernel to implement its edge decision and this low connectivity is reflected in its poor results [12].

In Figure 4.16 sections of the results obtained from the three detectors, when the standard deviation of the added noise was set to 8 grey levels, are illustrated. These sections measure 512x60 pixels and are taken across the centre of the images. These results were used to implement a qualitative analysis of the detectors. The detectors were rated according to the level of noise and the continuity of the step edges. The qualitative ratings are summarised in Table 4.2 along with the metric results for the test image. In this test the added noise had a standard deviation of 8 pixels.

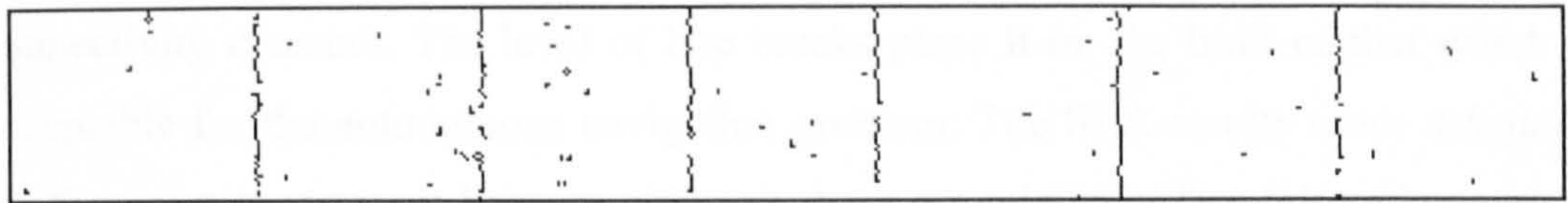


Figure 4.16(a) SUSAN Detector results Noise $\sigma = 8$

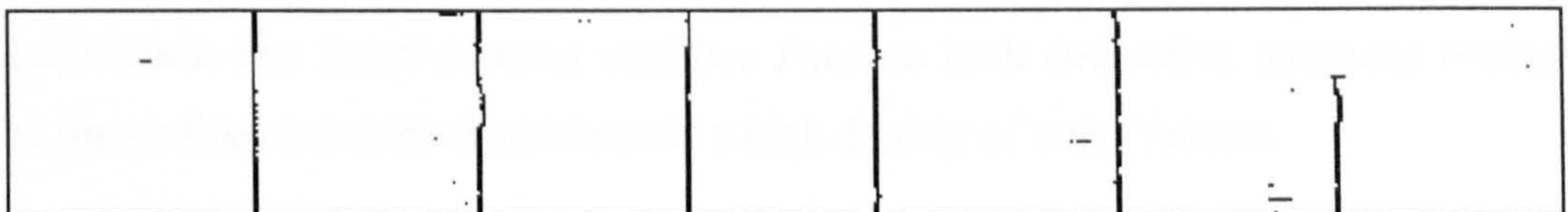


Figure 4.16(b) SLA Detector results Noise $\sigma = 8$

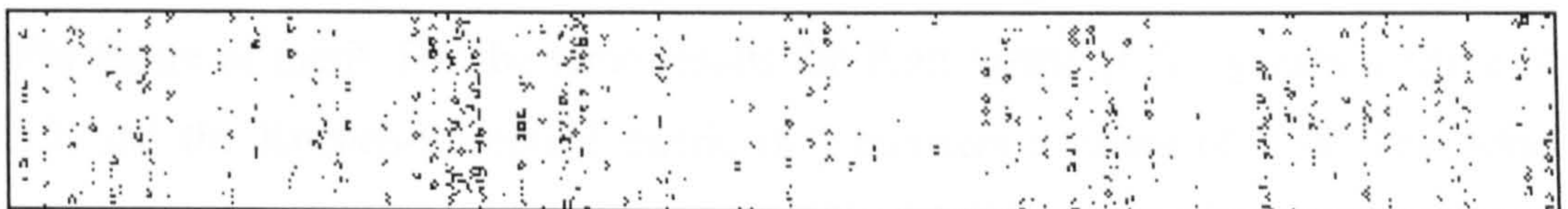


Figure 4.16(c) Sobel Detector results Noise $\sigma = 8$

Vertical Bar $\sigma = 8$	Qualitative Noise	Qualitative Continuity	EPM	Pratt	KR
SUSAN	Adequate	Poor	0.02	0.66	0.66
SLA	Adequate	Adequate	0.42	0.82	0.80
Sobel	Poor	Poor	-4.69	0.12	0.48

Table 4.2 Metric Comparison to Qualitative Results

The qualitative noise ratings was based on an assessment of the level of false returns in the image and their facility to link and form short line segments. The qualitative continuity rating was based on an assessment of the level of the breaks occurring in the image lines. The ratings for each of these qualitative measures were set to *Good*, *Adequate* and *Poor*.

The SUSAN detector is registered as *Adequate* on noise measures and *Poor* on the connectivity measure. The level of line breaks place it on the limit of that which is acceptable for the autonomous navigation problem. The SLA results score *Adequate* for the noise assessment because although there are relatively few false edge returns they do link to form significant false line segments. The SLA detector is also rated as *Adequate* in the continuity measurement because of the large break in one of the image lines. The Sobel detector registers *Poor* on both qualitative measures because the image lines are missed and there is a high density of noise returns.

The effects of the scaling factors applied to the EPM results is illustrated in the Sobel results where *Poor* qualitative ratings are ascribed a negative result of -4.69 by the EPM figure of merit. For the same results the Pratt metric [82] registers a rating of 0.12, and the Kitchen-Rosenfeld metric [83] registers a rating of 0.48. The Sobel detector is not capable of resolving the vertical lines when σ is set to 8. By specifying the levels of false returns that give a zero result the EPM metric is not limited to comparing detectors but allows a detector to be assessed against vision problem specifications.

4.5.3 Metric Comparisons

In Figure 4.17 results from the three metrics for the SLA's responses to the vertical bar image are given. These results demonstrate good agreement between the metrics until the SNR reaches 10dB. Here, the EPM result indicates a marked decline as the added noise disrupts the operation of the detector and renders it unsuitable for the autonomous vision problem. The Pratt and Kitchen-Rosenfeld metrics do not exhibit this marked decline.

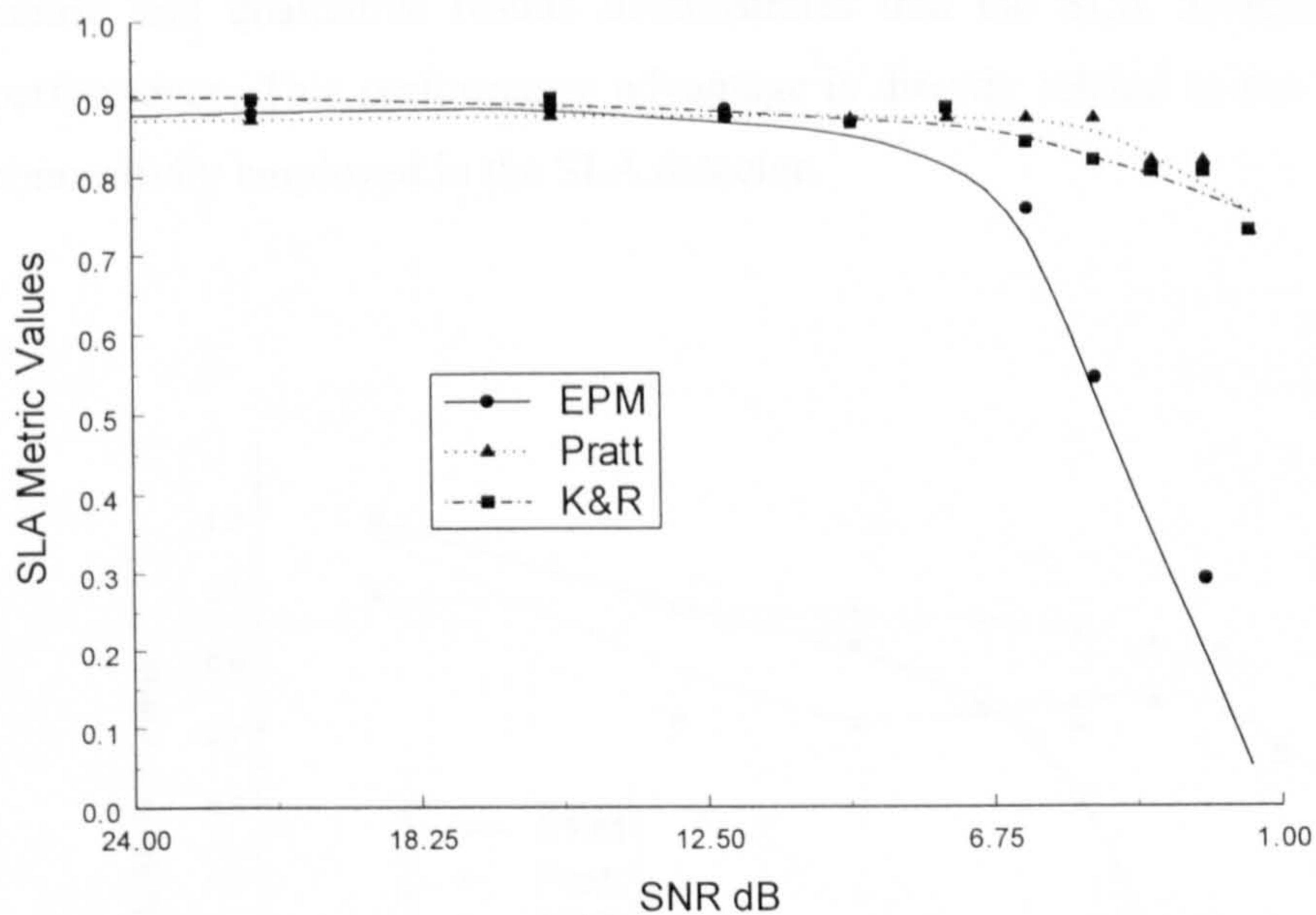


Figure 4.17 SLA Metric Results for Vertical Bar Image

When in Figure 4.17 the SNR reaches 1.6dB, the SLA results are at the limit of their practical use in the autonomous navigation problem. This is evidenced by Figure 4.18 which shows the SLA results given for the vertical bar test with $\sigma=10$ and SNR=1.6dB. The qualitative noise test was assigned an *Adequate* rating and the qualitative continuity test was assigned a *Poor* rating. The level of breaks in the vertical lines place these results at the limit of the acceptability for the autonomous vision navigation.

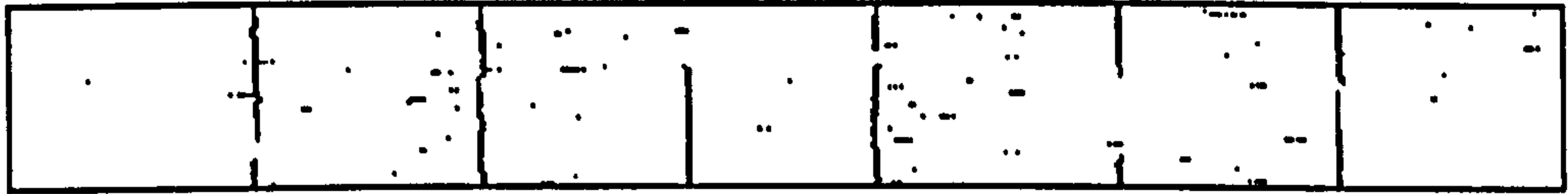


Figure 4.18 SLA results Noise $\sigma=10$ SNR=1.6dB

In Figure 4.19, the full metric results for the SUSAN detector are given for the vertical bar test images. The three metrics follow similar curves until the added noise gives rise to significant error levels and then the scaling factors in the EPM metric give rise to a steep roll off in the results. A comparison of the SLA, SUSAN detector metric and qualitative results demonstrates that the SLA detector has the better performance. This performance advantage is directly related to the higher degree of connectivity employed in the SLA detector.

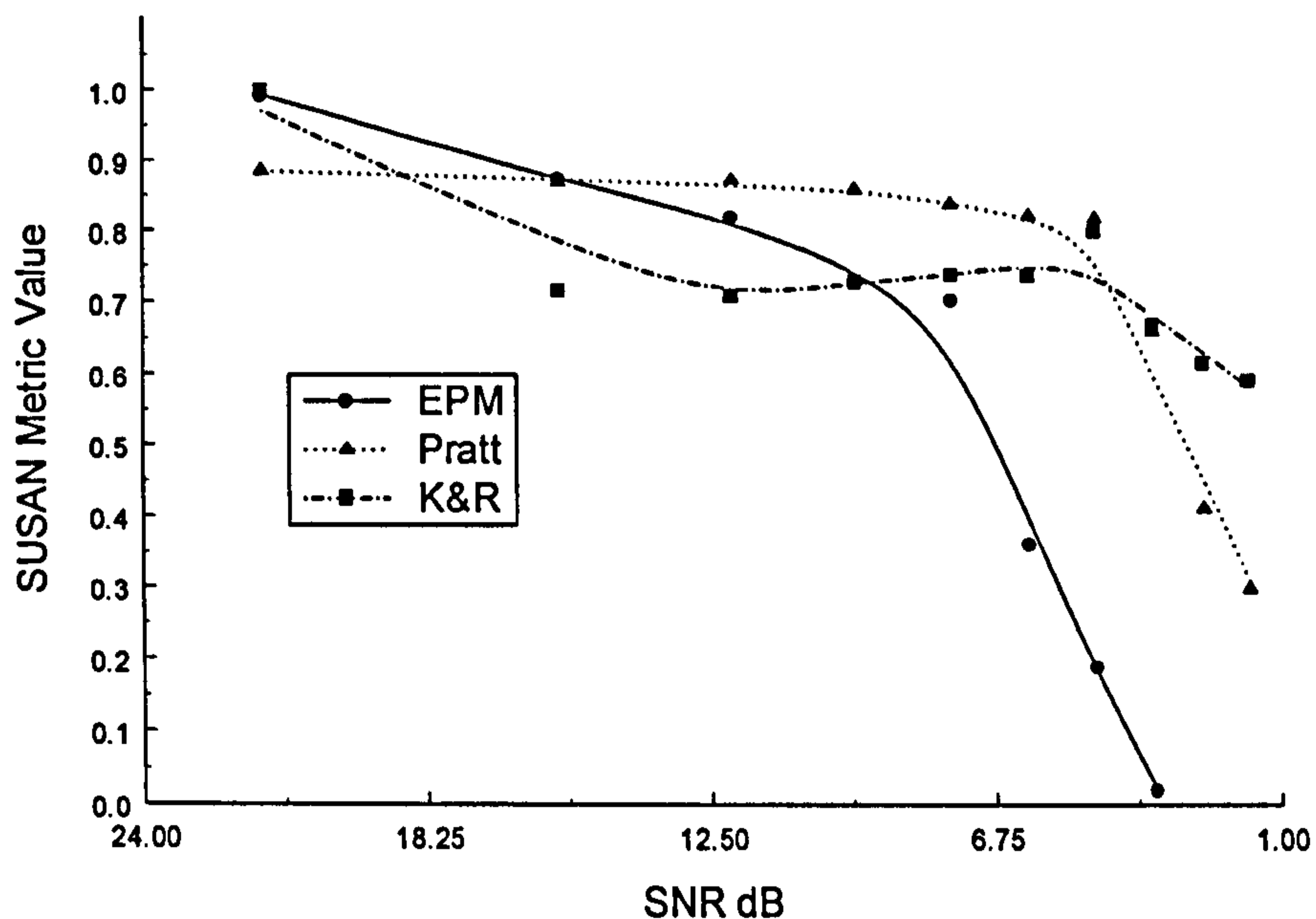


Figure 4.19 SUSAN Metric Results for the Vertical Bar Test Images

The low connectivity Sobel detector is more susceptible to the added noise than the SLA or SUSAN detectors. When the noise level is low, the Sobel detector performs as well as the other tested detectors. However, when the SNR reaches 15dB the metrics record a decline in the detector performance. The Sobel detector falls outwith the

autonomous navigation specification at a SNR of 10dB whereas the SUSAN detector maintains the navigation specifications until the SNR reaches 3.5dB and the SLA detector extends the specification compliance to a SNR of 1.6dB.

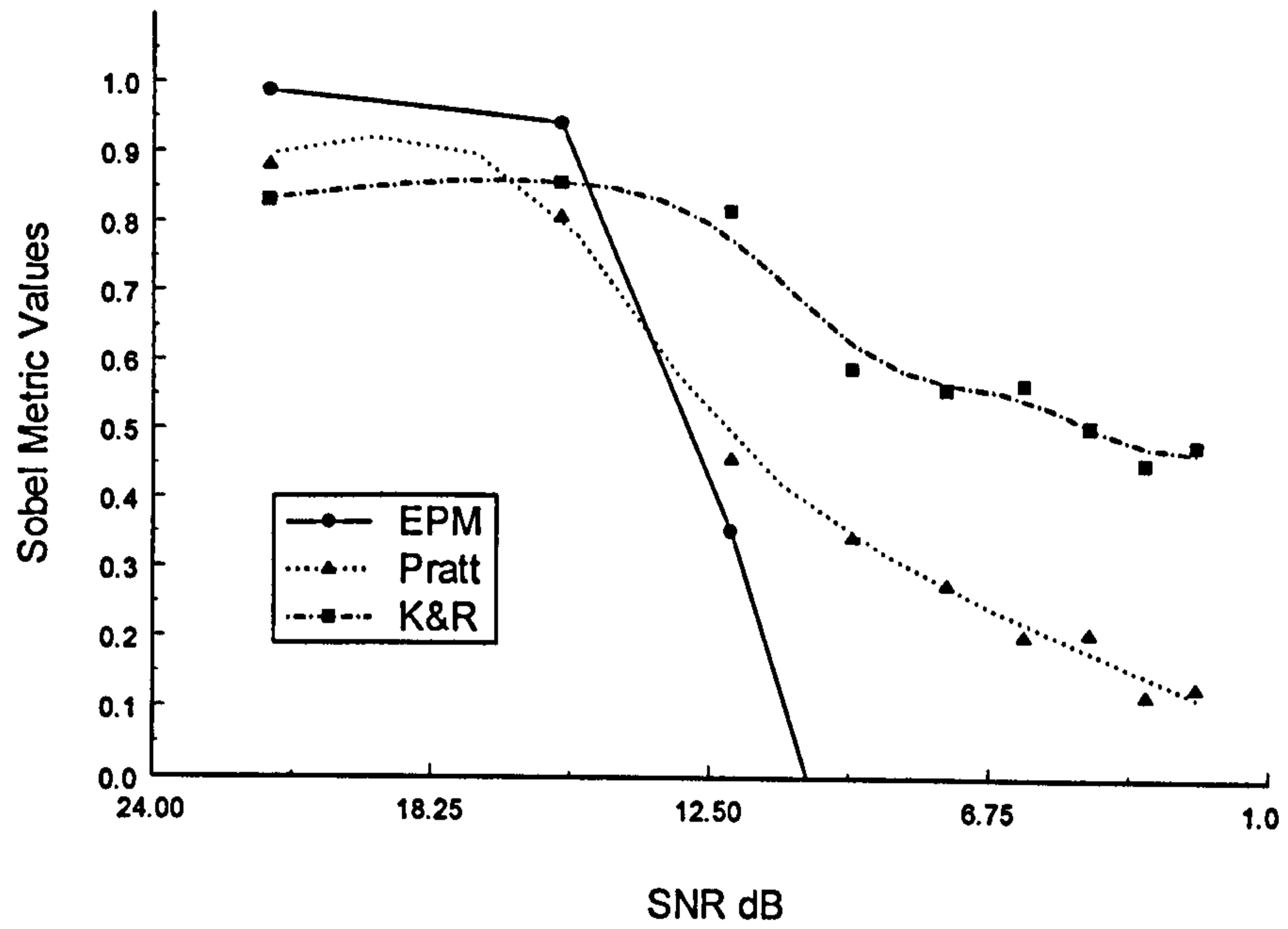


Figure 4.20 Sobel Metric Results for the Vertical Bar Test Images.

4.6 SLA Algorithm Configuration for NSIP implementation

4.6.1 SLA NSIP Test and Optimisation Synthetic Images

The use of fixed focus lenses in the SLA NSIP sensor results in defocusing occurring within scenes, such as the double door image given in Figure 4.21a. This is an out of focus section of the corridor image given in Figure 3.5. Objects that are outwith the lens's depth of field will be delineated by **spread-edges** [79]. These are characterised by an intensity discontinuity that occurs across three or more pixels. The spread-edge has the general form of a sampled hyperbolic *tan* curve. The edge point for this curve being allocated at the point of steepest gradient in the profile.

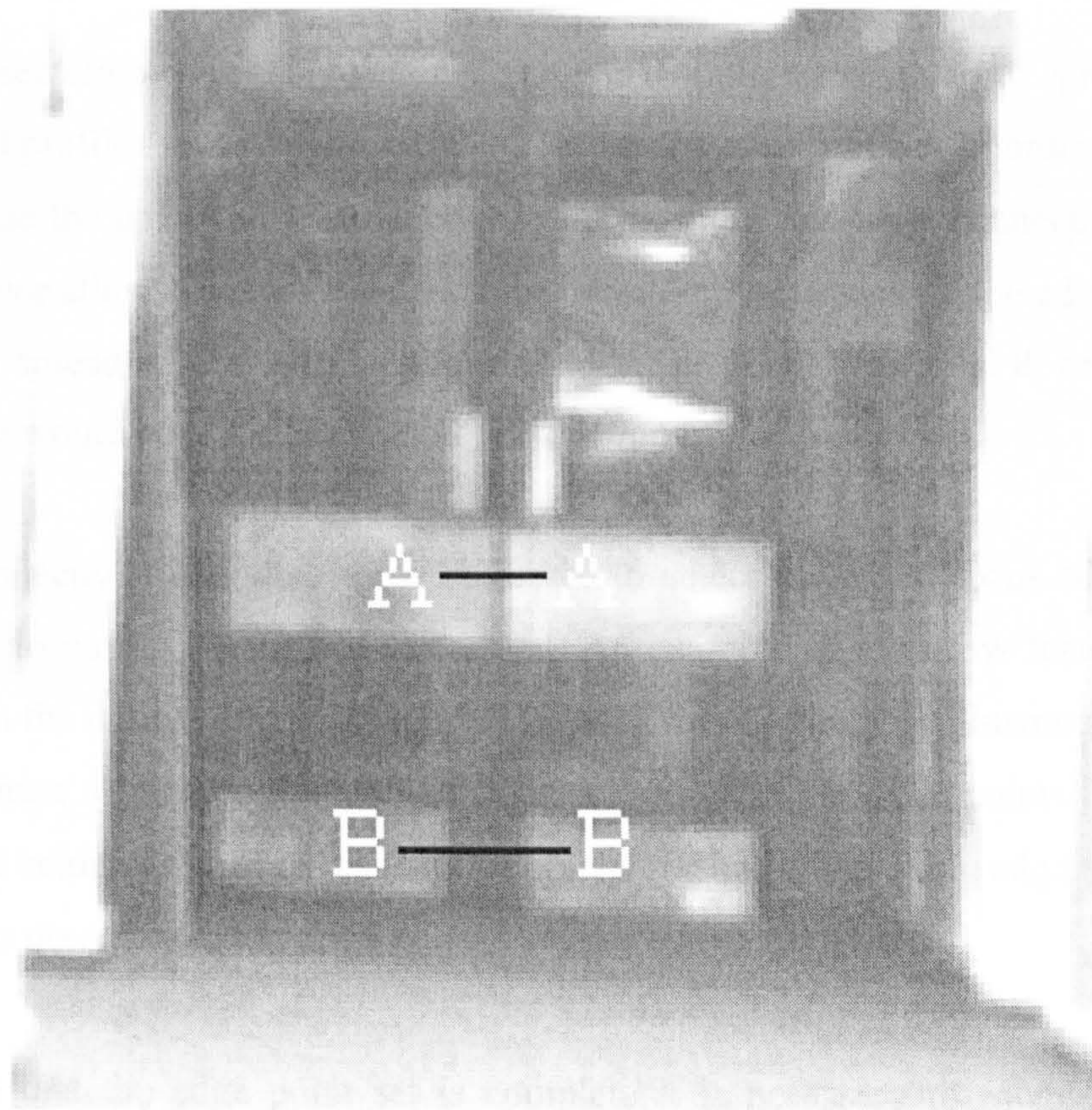


Figure 4.21(a) Double Door Image

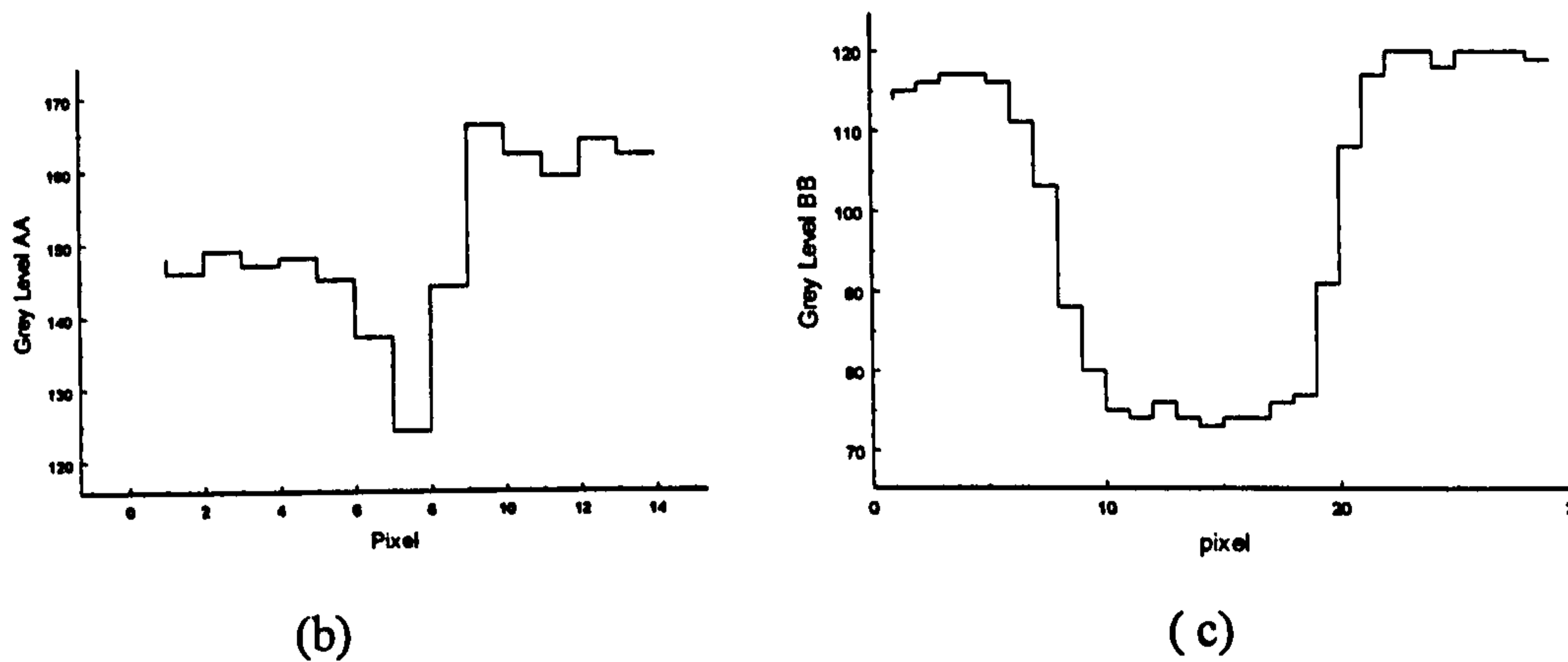


Figure 4.21(b) Section AA Narrow Groove Feature, (c) Section BB Pair of Spread-Edges

The cross section profile BB given in Figure 4.21(c) illustrates a pair of spread-edges. The spread profiles give rise to a reduction in the magnitude of the 1st order derivative and increase the uncertainty in the edge point location. The wide connectivity of the SLA detector allows the spread-edges to be detected. By inspection, the edge point for the falling spread-edge would be allocated to either pixel 7, or pixel 8, or both. The rising edge would be allocated to either pixel 19, or pixel 20, or both.

As the connectivity of a detector is increased to detect the out of focus edges so it's sensitivity to narrow features is decreased. An example of a narrow feature is also taken from the double doors of Figure 4.21(b). Cross section AA illustrates a narrow groove feature. By inspection, the groove is delineated by two edge points. The falling edge point being assigned to pixels 6, or pixel 7, or both. The rising edge point being assigned to pixel 7, pixel 8, or both.

To ensure that the edge point set is complete it is necessary to establish that the narrow features, as well as the spread-edges are detected. These conflicting requirements result in a compromise in the configuration of the detector. The option of using multiple passes of the detector set to different connectivity scales was not practical in the SLA NSIP as the routing of the derivatives and the logical operations are fixed in the substrate layout.

In order to configure the SLA algorithm for the autonomous vision problem the synthetic images of Figure 4.22 were created using a graphical draw package. The hairline outlines of the draw package were processed to give ground truth sets for the synthetic images. The concentric ring profiles ensured that edge point at all possible angles are tested. In Figure 4.22(a), narrow features were created with a width of 3 pixels, similar to the V shaped groove given in Figure 4.21(b). The edge profiles in Figure 4.22(b) were smoothed to give edge spreads of 5 pixels, similar to those illustrated in Figure 4.21(c). Noise with a Gaussian distribution was added to these synthetic images to give a set of test images against which the SLA detector configurations could be assessed.

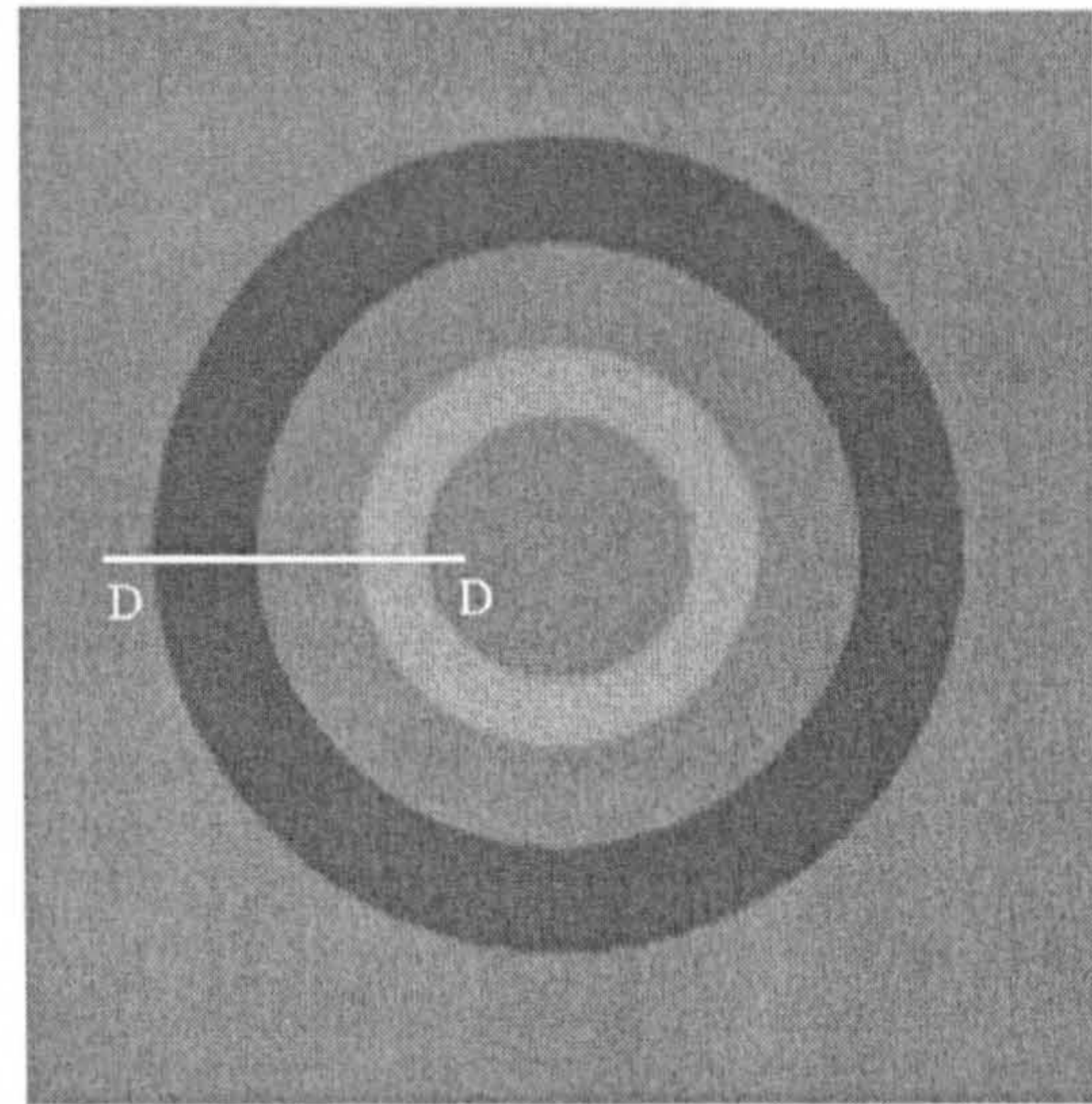
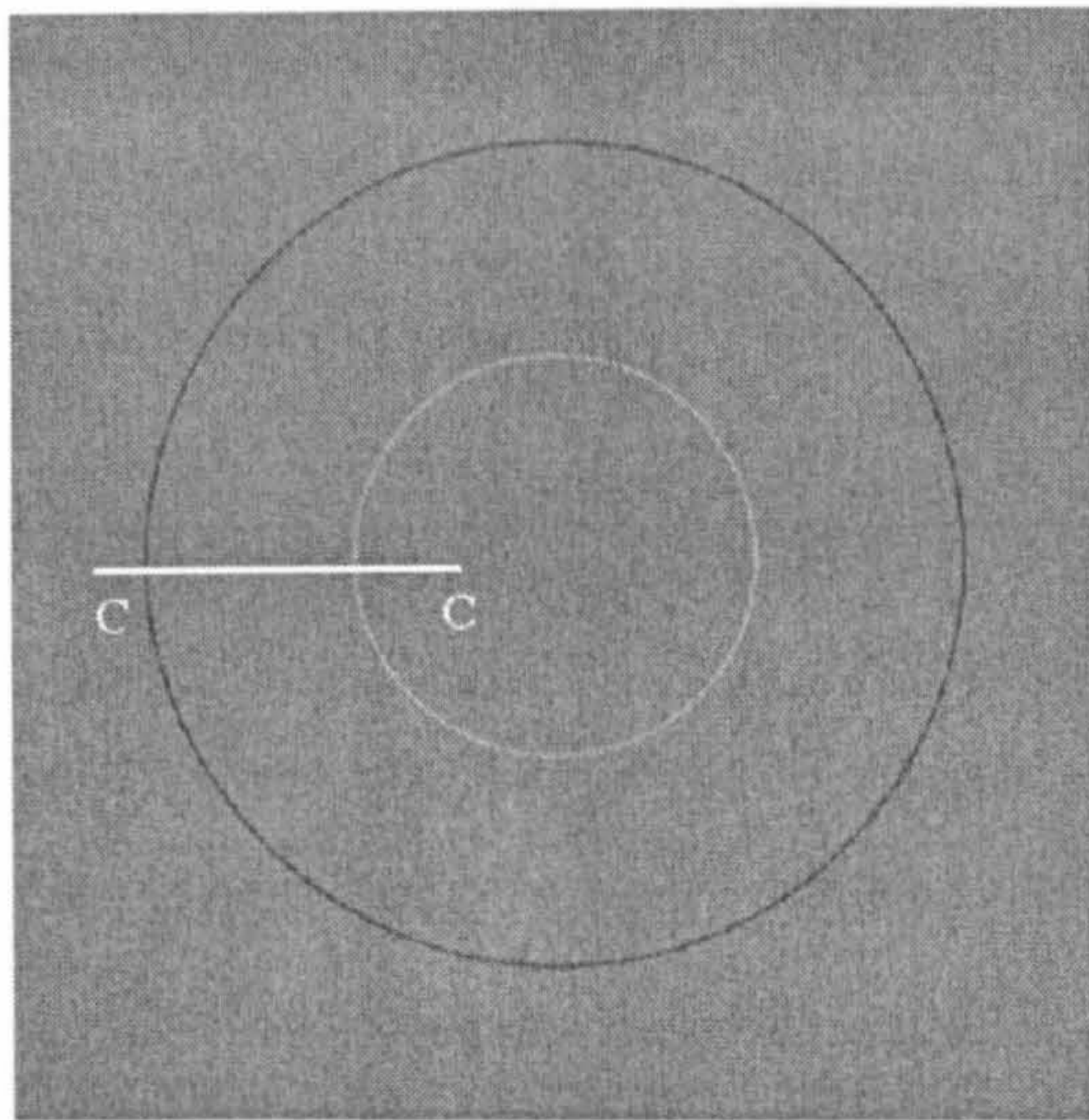


Figure 4.22(a) Narrow Feature Test

Figure 4.22(b) Spread Edge Test

The cross sections CC and DD from image 4.22 are illustrated in Figure 4.23. Cross section CC illustrates the narrow features used to configure the SLA algorithm. The narrow feature is delineated by a pair of edge points. These edge points are separated by a single pixel space. Cross section DD illustrates the spread edges used to configure the SLA algorithm. The edges are spread over 5 pixels with the maximum gradient central to this range.

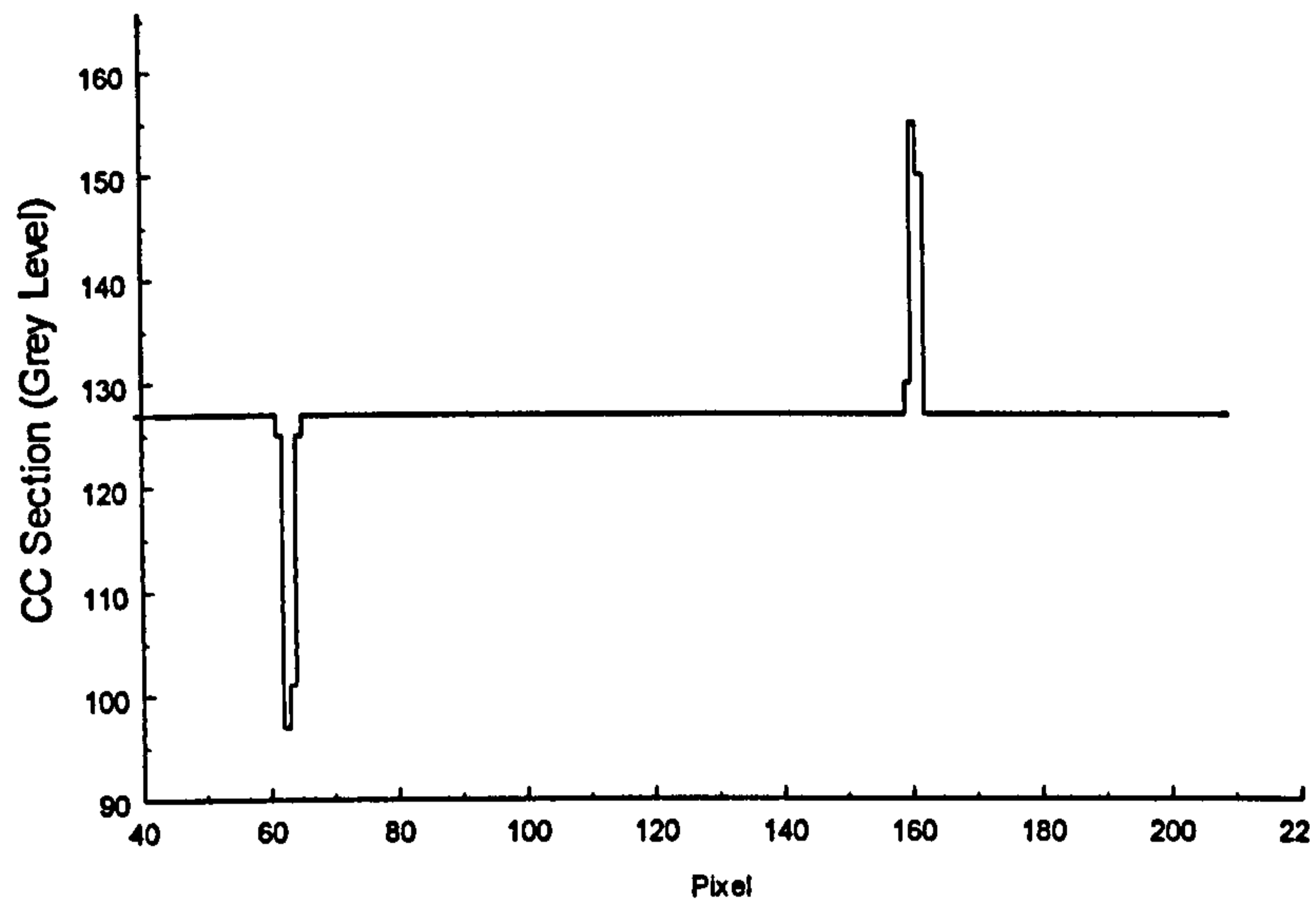


Figure 4.23(a) Narrow Feature Cross Section CC

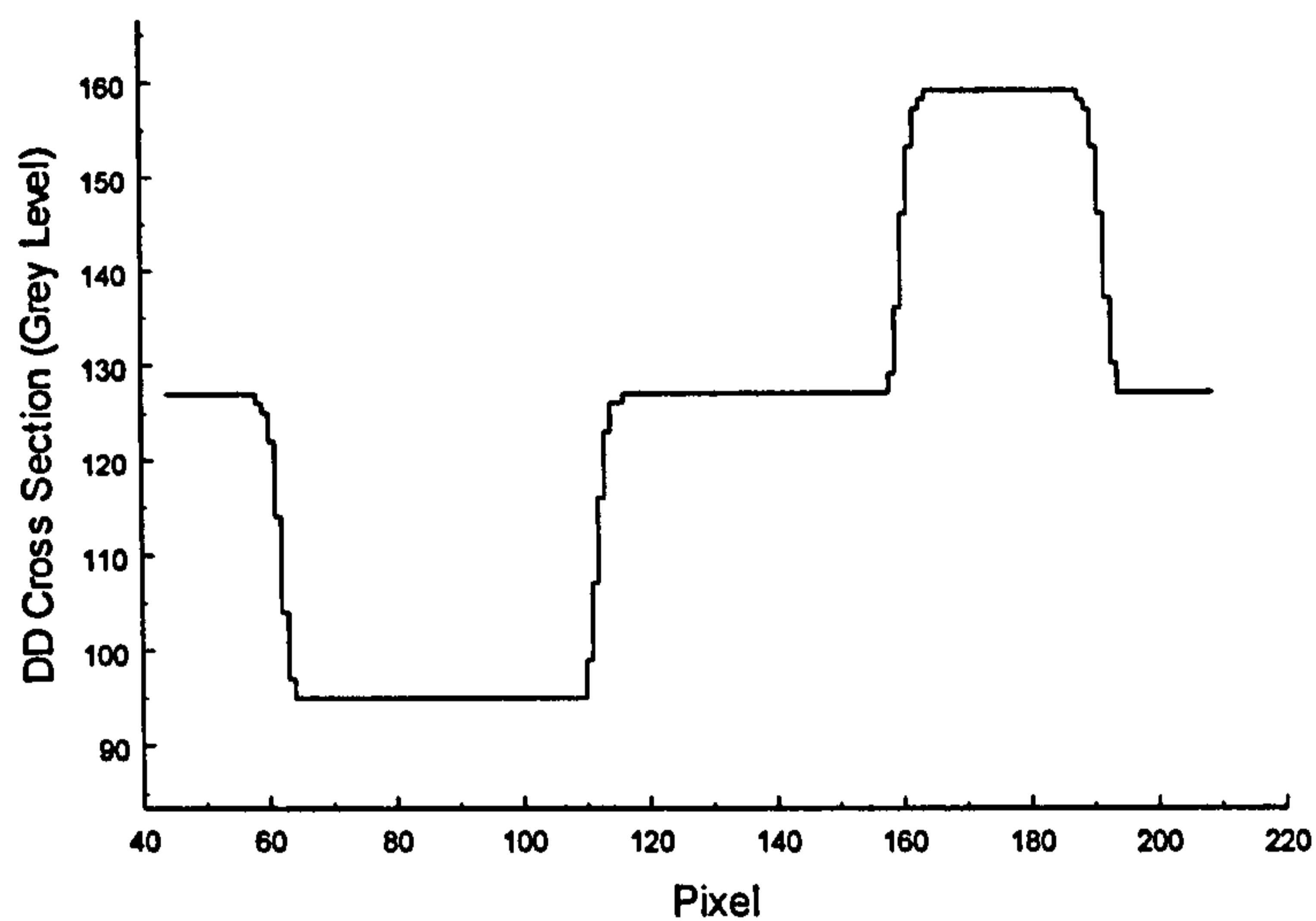


Figure 4.23 (b) Spread Edge Cross Section DD

4.6.2 SLA Detector Uniform Filter Length of 4 Average Width of 3

The widely connected SLA detector employed in the metric comparisons of Section 4.4 was configured to detect vertical and horizontal lines. The width setting of 7, in this detector limited its facility to detect lines that lie on a diagonal. The analysis of Section 3.6 established that the diagonal lines are critical to the successful implementation of the autonomous navigation problem. In order to ensure that

diagonal lines were not missed by the SLA detector its width parameter was limited to a maximum of 3 pixels. The width, the $Per_{1^{st}}$ and $Per_{2^{nd}}$ parameters of the SLA algorithm are adaptive in the NSIP configuration. The uniform filter length is fixed through the CMOS layout and this parameter was the main focus of the configuration analysis.

EPM results for the SLA algorithm, with the length of the uniform filter set to 4 pixels, are illustrated in Figure 4.24. In this implementation the 1st order directional derivative has a span of 5 pixels and the 2nd order derivative has a span of 9 pixels. A convolution of this form maximises the 1st order response to the spread edges of Figure 4.22(b). The $Per_{1^{st}}$ was set to 5% and the $Per_{2^{nd}}$ was set to 0.1%. The minimum line pixel count was set to 3.

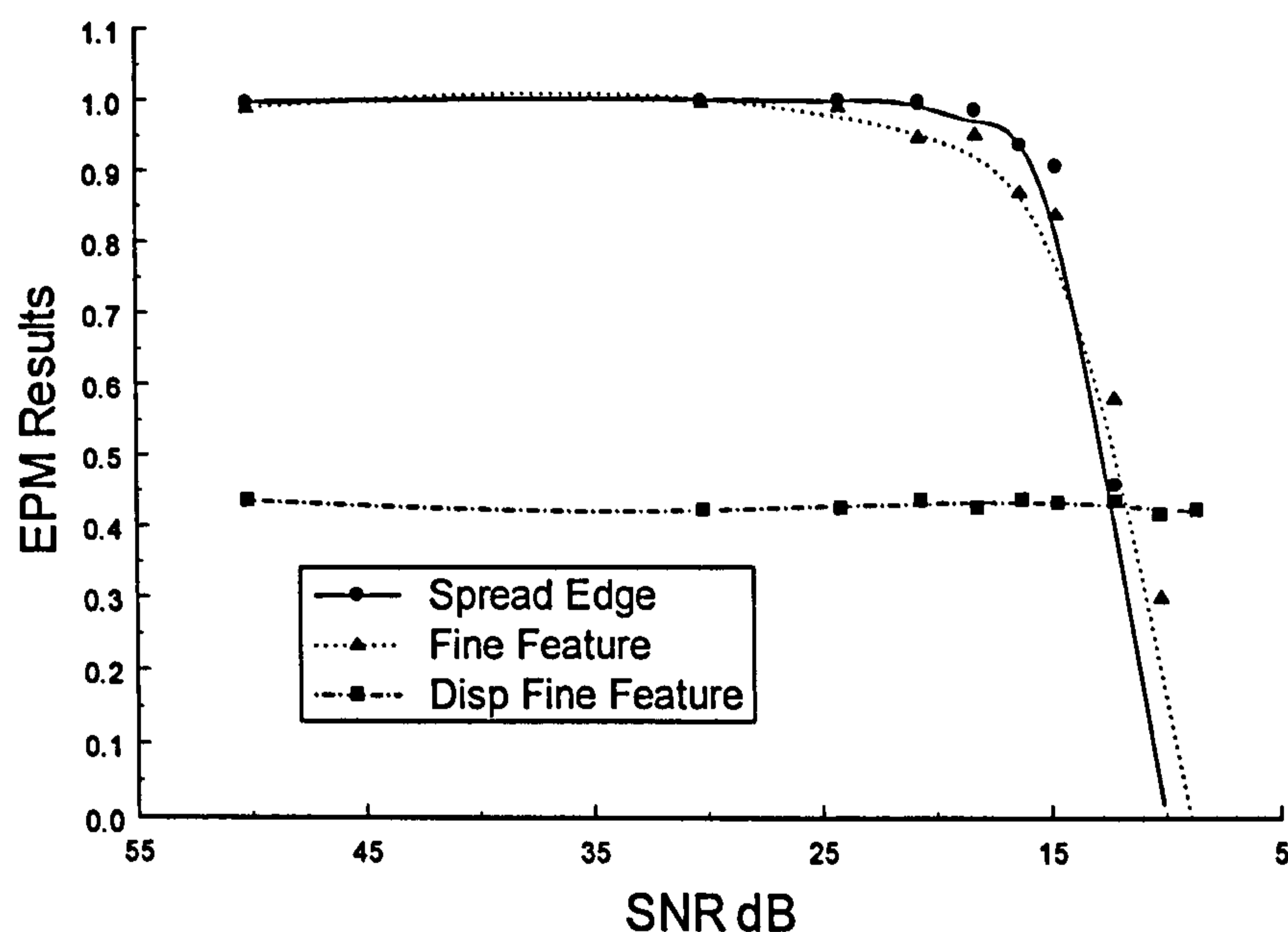


Figure 4.24 SLA Detector Filter Length of 4

The EPM results given in Figure 4.24 show that this SLA configuration maintains high metric returns until the SNR reaches 15dB. At a SNR of 10dB and 9dB, the spread edge and narrow feature results cross zero. This zero crossing indicates that when the SNR of the captured image is of the order of 10dB that the SLA detector

fails to segment the images with sufficient quality to allow the autonomous vision algorithm to be implemented. Further illustrated in Figure 4.24 is the displacement probability for edge points detected in the narrow feature test images. This shows that approximately 45% of the edges are displaced. This is due to broadening of the narrow feature. Inspection of the detector results demonstrated that the SLA, with convolutions based on uniform filter of length 4, caused the width of the Figure 4.22(a) narrow feature to be increased from 3 pixels to 5 pixels.

4.6.3 SLA Detector Uniform Filter Length of 2 Average Width of 3

In order to avoid broadening of the narrow feature the SLA detector was reconfigured with a uniform filter length of 2. The convolution width was maintained at 3, the line pixel count threshold was set to 3. $Per_{1^{st}}$ was set to 4% and $Per_{2^{nd}}$ was set to 0.1%. The EPM results for this amended configuration are given in Figure 4.25. As a result of reducing the degree of connectivity in the SLA detector, the probability of a displacement occurring in the narrow feature results was reduced to approximately 10%. However, the reduced connectivity also causes the zero crossing points of the EPM results to be increased to 14dB for the spread edge tests and 13dB for the narrow feature tests.

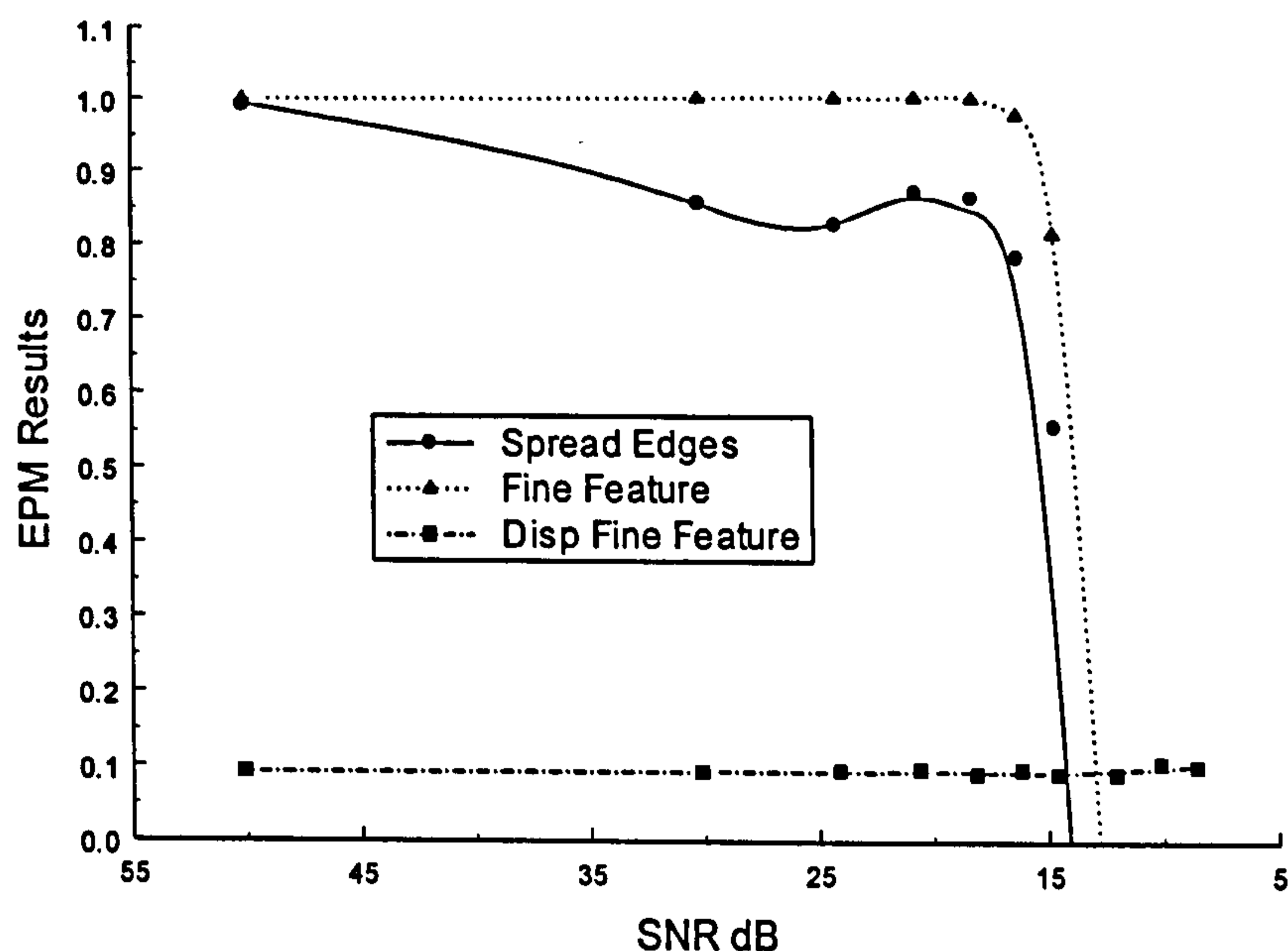


Figure 4.25 SLA Detector Filter Length 2

Analysis of the corridor test image of Figure 3.5 gave a worst case SNR of 16.5dB for structural edges. These worst case results were found in the neighbourhood of the dimly illuminated double door. Thus the synthetic test results established that the SLA NSIP should be configured with convolutions of uniform filter length 2. The equations that describe these convolutions were derived in Section 3.2. The 1st order derivative convolution is given by equation (3.10) and the 2nd order derivative convolution given by equation (3.11).

The EPM quantitative measure proved an effective tool for the assessment of the detector. It allowed a quantitative assessment of the detector to be made and alternative configurations to be compared. By setting the gain in the linear combination of the error probabilities to give a zero crossing metric the useful range of the detector was clearly indicated.

In Appendix A, five examples of the operation of the SLA algorithm with the span set to 2, width set 3, $Per_{1^{st}}$ set to 4% and $Per_{2^{nd}}$ set to 0.1% are given. The example images include three indoor views that are representative of the scenes likely to be encountered by an autonomous navigation robot. Also included in Appendix A are the Lena and Clare images. These are accepted as standard segmentation test images in the field of vision processing. The results in Appendix A demonstrate that the SLA edge detector which was optimised for the detection of narrow features and spread edges is effective in the segmentation of the standard images and indoor scenes.

4.7 Conclusion

The quality of the information contained in an edge point data set is critical to the overall performance of the vision system. An assessment of the quantitative edge detector metrics proposed by Pratt [82] and Kitchen-Rosenfeld [83] was carried out. It was noted that these metrics were limited to comparing edge detectors and are not designed to compare a detectors performance to a vision systems specification. A new metric called the Edge Point Metric was developed. This metric allows for the inclusion of a minimum quality specification [16] for the edge detector. The EPM results were used to select the convolution span, filter width and threshold parameters of a SLA detector for use in autonomous navigation.

The new metric employs a Detector to Ground truth Comparison algorithm (DGC), described in Section 4.3, that compensates for systematic displacements of up to two pixels on either side of the ground truth line. This DGC algorithm ensures that the metric assessment is not skewed as a result of image sampling or the discrete detector processes. The DGC compensation employs heuristics that can be implemented through an image processing package that allows kernel based logical operations to be applied to binary images. Accumulated results from the DGC algorithm allow the EPM figure of merit to be evaluated from the conditional probabilities of the detector registering a *FP* or *FN* return.

A comparison of the vertical bar image results given by the SLA detector in Figure 4.17 and the SUSAN detector in Figure 4.19 demonstrates that the SLA detector returns higher figures of merit results across the tested SNR ranging down to 2dB. The connectivity of the SUSAN detector used in these tests was 109 pixels and the connectivity of the SLA detector extended over an area of 105 pixels of which 56 were sampled to give recorded results. This comparison establishes that the SLA detector with its sparse convolution operators and distributed decision logic give a higher quality performance than the SUSAN detector.

Chapter 5 Smart CMOS Camera Implementation

5.1 Introduction

This Chapter presents research into the circuit realisation of the Smart CMOS Camera. The medium chosen for this camera research was the Mietec 2.4 μ m CMOS process. The research covers the development of a random access pixel array and the circuit implementation of a Near Sensor Image Processor (NSIP) that incorporates the SLA edge detection algorithm. The operational specifications for the camera were determined from the analysis of the requirements of autonomous corridor navigation given in Section 3.6.

A block diagram of the Smart CMOS Camera is given in Figure 5.1. The camera hosts a pixel array and two implementations of the SLA NSIP. The array pixel selection circuits allow the image data to be readout in two orthogonal scans. These have been nominated as the *Vertical Scan* and the *Horizontal Scan*. In the horizontal scan columns of edge points are extracted by the *Horizontal SLA NSIP*. In the vertical scan rows of edge points are extracted by the *Vertical SLA NSIP*.

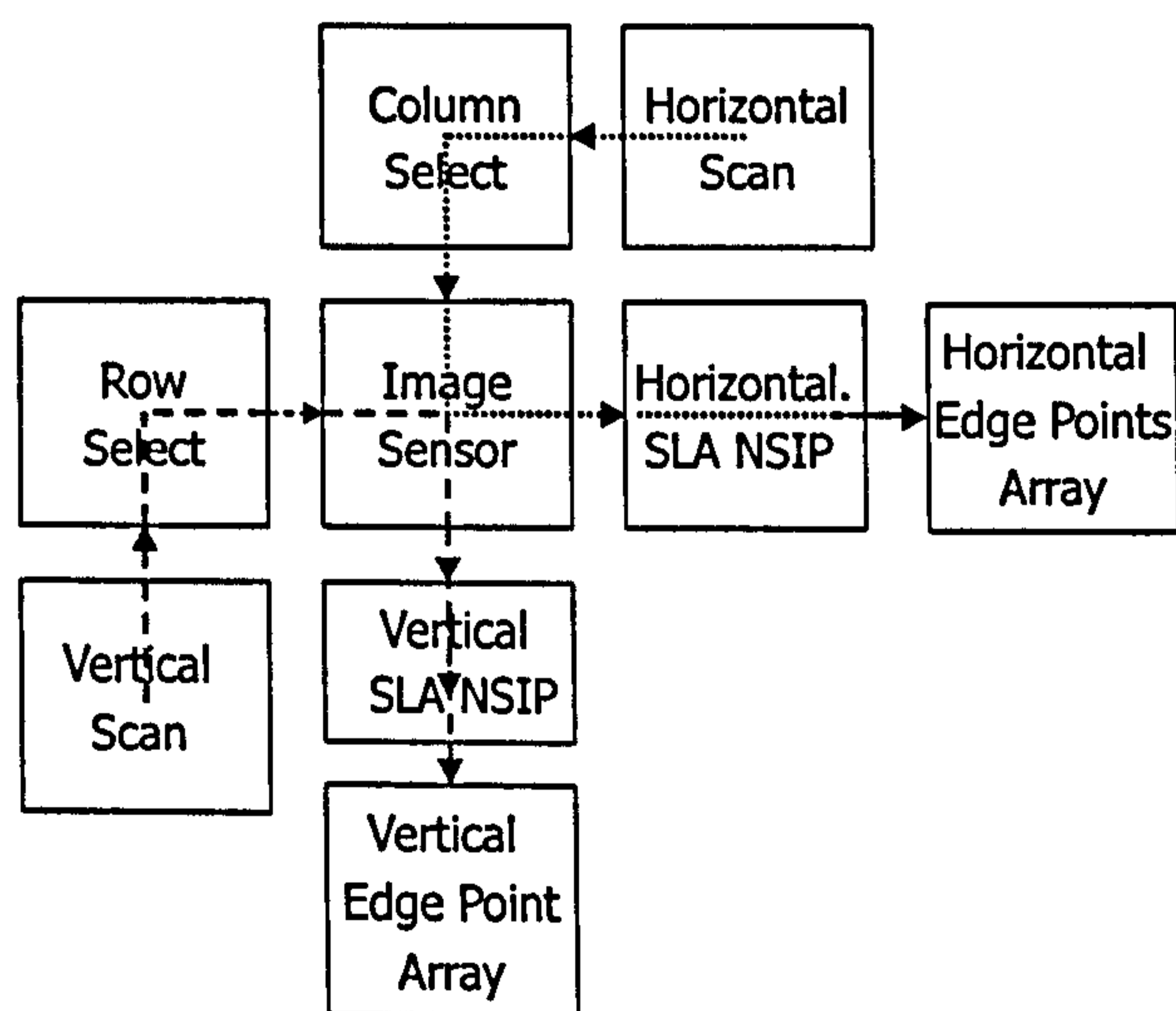


Figure 5.1. Smart CMOS Camera Block Diagram

The SLA NSIP was designed to allow parallel detection of edge points on a row or column basis, as the image data was readout from the array. The timing diagram of Figure 5.2 illustrates the edge point readout sequence. In the camera *Frame Period*

the two orthogonal scans are implemented. In the first scan, all the pixels from each accessed column are processed in parallel through the *Horizontal SLA NSIP* to give columns of edge points, that are loaded into the *Horizontal Edge Points Array*. In the second scan, all the pixels from each accessed row are processed in parallel through the *Vertical SLA NSIP* to give rows of edge points, that are loaded into the *Vertical Edge Points Array*. The *Edge Output* sequence of Figure 5.2 illustrates the order in which the edge points are generated by the two scans of the image array.

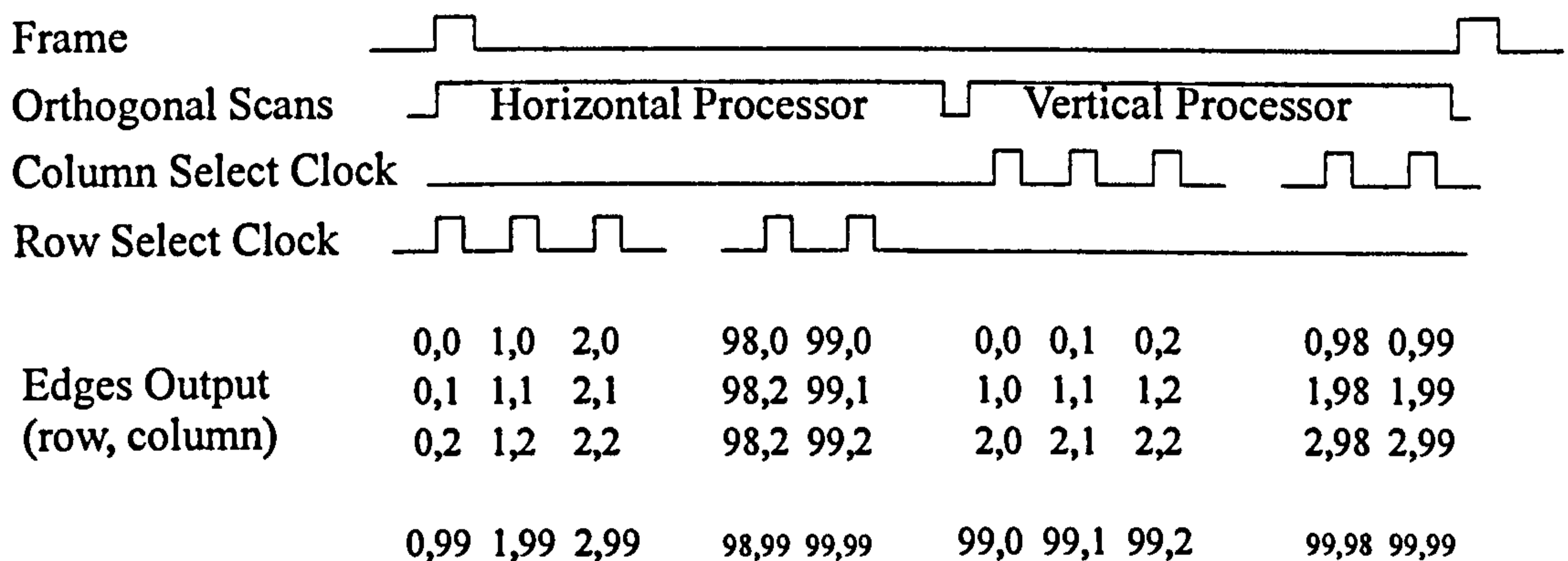


Figure 5.2 Smart CMOS Camera Timing Diagram

In the design of the autonomous navigation sensor the full frame process rate was set to 6.5Hz, to comply with navigation uncertainty requirements detailed in Section 3.6. The uncertainty requirements also set the array spatial resolution to 100x100 pixels. In keeping with these settings the period of the *Row* and *Column Select Clocks* were set to 0.77ms to give a pixel read rate of 1300Hz.

In keeping with the constraints imposed by commercially available optics it was elected to limit the image sensing area to a maximum of 0.8x0.8cm. This receive area could be accommodated by a 1" format lens F/1.2. Thus the square pixel pitch for the 100x100 array was set to 80 μ m with each pixel measuring 80x80 μ m.

The random access operation of the pixel array meant that the instantaneous photo currents generated by the pixels would set the signal levels readout from the sensor array. In Section 5.2 the photo currents generated by CMOS light sensing diodes are evaluated. It is noted that these diode currents limit the SLA NSIP framing rate to less

than the required 6.5Hz. To overcome this limitation, it was decided to integrate a vertical Bipolar Junction Transistor (BJT) into a light sensing diode structure to provide gain at each pixel site. An evaluation of the gain available through the BJT pixel is given in Section 5.3. Measurements made on a test pixel structure established that the BJT pixel would comply with the random access requirement of the Smart CMOS Camera.

The NSIP circuit implementation of the spatial derivatives required by the SLA algorithm is described in Section 5.4. The circuits are designed to provide both 1st and 2nd order spatial derivatives. The layout of the derivative circuits was designed to fit within the 80 μ m pitch of the image-sensing array. This allowed the spatial derivative processes to be implemented in parallel for the columns and rows of image data readout from the array.

A test implementation of the Smart CMOS camera created in the Mietec 2.4 μ m CMOS process is described in Section 5.6. In this, an array of 10 rows and 4 columns of 80x80 μ m pixels provided the image sensing. Column selection circuitry and a horizontal SLA NSIP were integrated on the CMOS substrate alongside the image sensing. The NSIP circuits used to convert the spatial derivatives into a discrete format and the edge decision logic are described in Section 5.5 Results from this test implementation of the Smart CMOS Camera demonstrate the detection of edge points.

5.2 CMOS Phototransduction Diodes

The random access operational requirement specified for the Smart CMOS Camera meant that the sensor function was dependent upon the instantaneous photo-transduction processes within the silicon substrate [91]. In a silicon substrate, photo-transduction occurs when absorbed photons of incident light excite electrons from the valence to conduction band within a diodes depletion region. Photo-transduction also occurs when this excitation occurs within a diffusion length of the diode depletion region. A reversed-bias applied to the diode allows it to act as a current source giving a photo generated current I_{ph} . Three types of CMOS light sensing diode structure are illustrated in Figure 5.3. These are the N- well to P- substrate structure of Figure 5.3(a), the N+ diffusion to P- substrate of Figure 5.3(b) and the N- well to P+ diffusion of Figure 5.3(c).

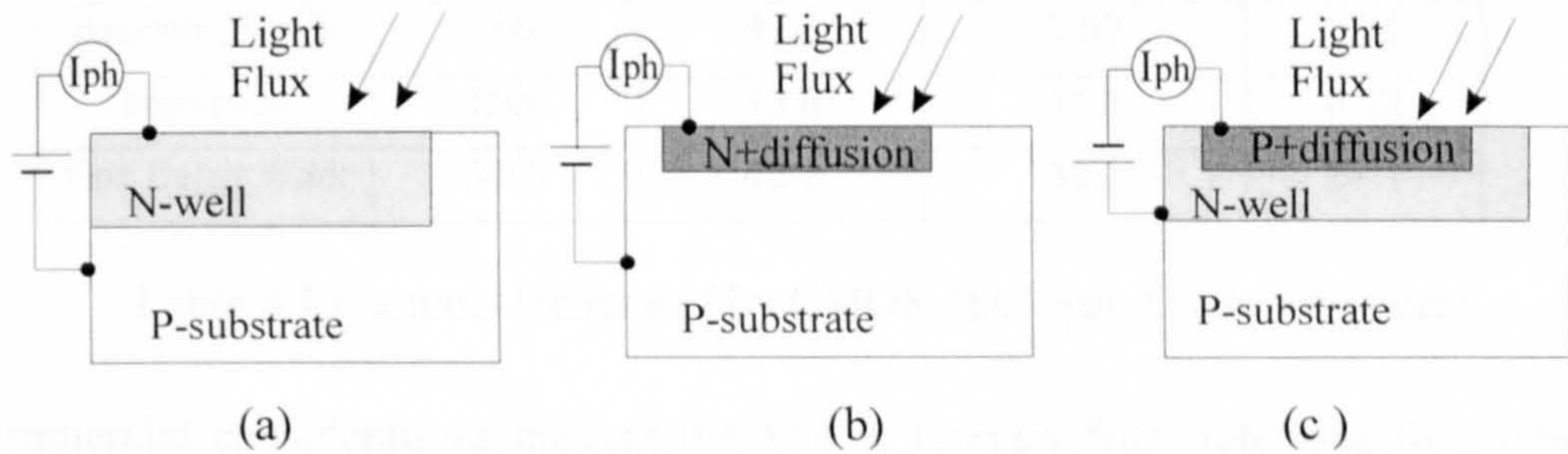


Figure 5.3 CMOS diode structures, **(a)** N- well to P- substrate diode **(b)** N+ diffusion to P-substrate diode **(c)** N- well to P+ diffusion diode

The relationship between the photo current I_{ph} , incident flux intensity P_o and the receive area of the diode A is given in equation (5.1) [91]. The energy E_p of the absorbed photons is dependent upon the wavelength of the incident light. The factor R is the reflection coefficient for silicon. The internal quantum efficiency η_i is the probability that a photon will excite an electron from the valence to conduction band to create an excess minority carrier. The factor F gives the probability that this minority carrier is collected and contributes to I_{ph} .

$$I_{ph} = q \frac{P_o}{E_p} (1 - R) \eta_i F \quad (5.1)$$

A theoretical analysis of photo-transduction in CMOS diode structures was carried out by Moini [92]. This analysis reports the probability product ($\eta_i F$) for the diode structures of Figure 5.3. These probability products were evaluated for wavelengths ranging from 300nm to 1 μ m. Integrals were taken of Moini's ($\eta_i F$) probability products for the photopic range of 400nm to 700nm. These were used to evaluate the currents generated by the Figure 5.3 diode structures, under white light illumination, for the Illumination Engineering Society (IES) lighting conditions given in Table 5.1 [42]. The lighting conditions assessed in Table 5.1 are representative of the environments that the smart camera is expected to operate within.

Environment	Illuminance Minimum (lx)	Diff-Substrate Diode (nA)	Well-Substrate Diode (nA)	Diff-Well Diode (nA)
Storage Areas	50	0.46	0.39	0.27
General Office	200	1.85	1.13	1.08
Assembly Work	500	4.63	3.92	2.71
Inspection	1500	13.9	11.8	8.12
Fine Detail Work	5000	46.3	39.2	27.1

Table 5.1 Currents Generated by CMOS 56x56 μ m Diode Structures

Commercial considerations prevent the Mietec foundry from releasing the diffusion depths or doping concentrations necessary for a full analysis of the light detection properties of diode structures formed within their 2.4 μ m process. The Moini analysis was made on a process equivalent to the Mietec 2.4 μ m CMOS process. Thus it was decided to base the analysis of the light detection diode structures on the data supplied by Moini.

The array of 80x80 μ m pixels incorporates readout circuitry that has the effect of reducing the area available for the detection of light in each pixel. In this analysis, the pixels were assumed to have a fill factor of 50%, allowing half of the pixel area to be taken over by the readout circuitry and routing. The light sensing area would be a central 56x56 μ m region within the pixel. The currents for CMOS diodes given in Table 5.1 are calculated for diodes with a sensing area of 56x56 μ m. In the calculation

of pixel currents given in Table 5.1 the optical gain supplied by the camera's image focus lens was assumed to be unity.

The rate at which the pixels can be accessed from within the array is dependent upon the photo current generated by the pixels and upon the line readout capacitance. The line readout capacitance was calculated from the Mietec 2.4 μ m Electrical Characteristics [93]. This line readout capacitance is composed of three components;

Metal to Substrate Capacitance $C_M = 512$ fF

ON-MOSFET Capacitance $C_{ON} = 19.7$ fF per MOSFET

OFF-MOSFET Capacitance $C_{OFF} = 11.8$ fF per MOSFET

The SLA implementation described in Section 3.7, shows that for each processed pixel location three pixels are connected to each readout line. The resultant *LineCapacitance* for the array readout line is given by equation 5.2 as 1.68pF.

$$LineCapacitance = C_M + 3C_{ON} + 97C_{OFF} \quad (5.2)$$

The slew rate that could be expected for the readout line when the sensor was operated under the *General Office* condition of Table 5.1, was used to assess the practicality of a random access array constructed from diffusion to substrate diodes. An illumination level of 200lx on the diffusion to substrate diode gives rise to a maximum slew rate of 1.1V/ms when the diode output current charges a 1.68pF capacitance. If the pixel to pixel voltage swing is 1V, the maximum pixel read rate that can be supported by this diode structure is 908Hz. This read rate is less than the 1300Hz required for a 6.5Hz framing of the 100x100 sensor. As a result of this limited pixel read rate it was decided to consider the practicality of introducing gain into the pixel structure.

5.3 Pixel with Integral Gain

5.3.1 BJT Pixel

In the N-well to substrate diode structure it is possible to integrate a vertical Bipolar Junction Transistor (BJT) capable of providing current gain. These BJTs have been sited as viable light detectors, with reported theoretical gains in the order of 300 Moini [92]. For the development of the Smart CMOS Camera random access array, it was decided to investigate the pixel structure illustrated in Figure 5.4(a). The equivalent circuit for this structure is given in Figure 5.4(b).

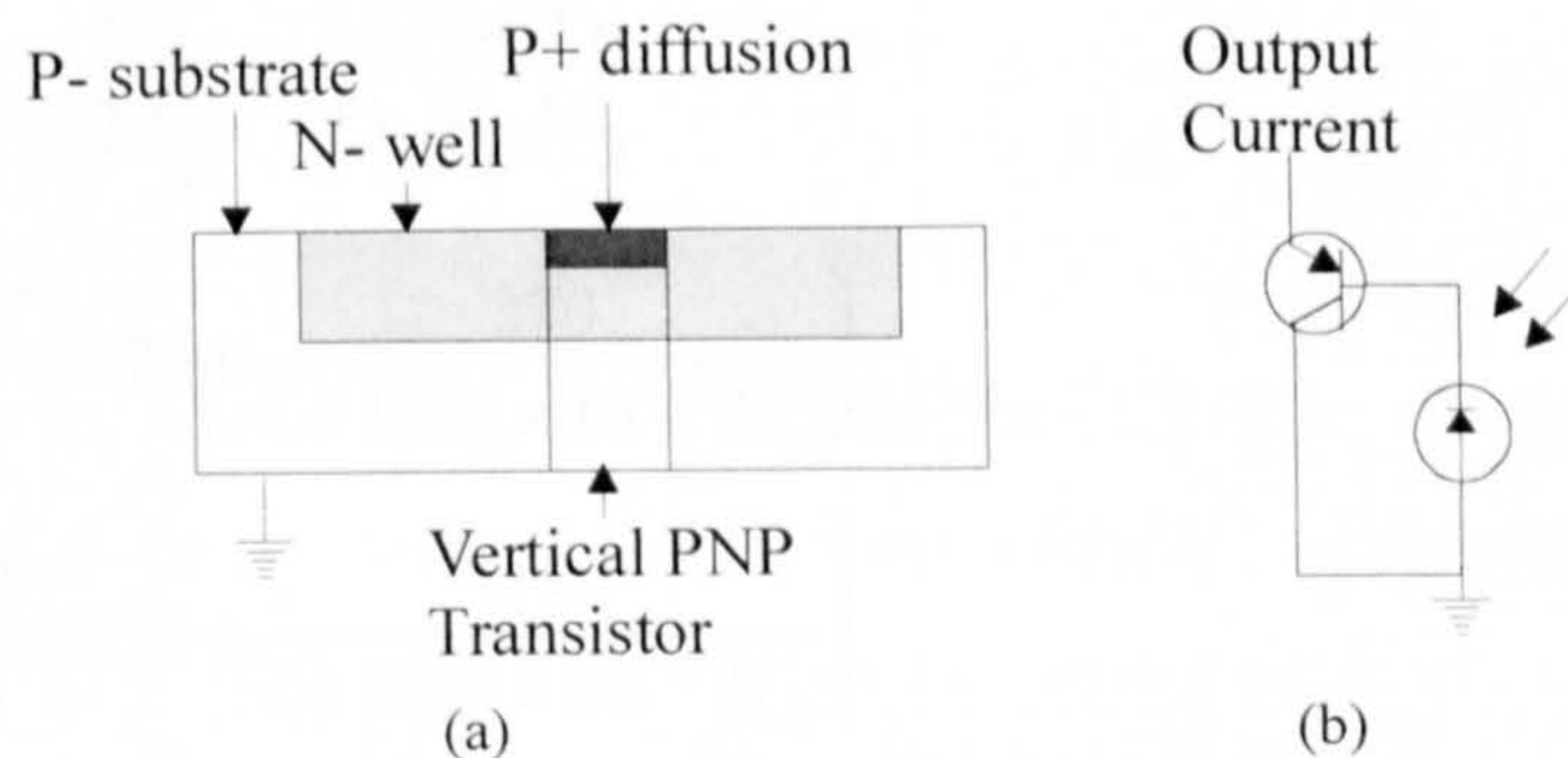


Figure 5.4(a) Cross Section of BJT Pixel, **(b)** Equivalent Circuit for BJT Pixel

In the Figure 5.4(a) configuration, light is sensed by the N- well to substrate diode and current gain is effected by the vertical PNP transistor created by the P+ diffusion on the well surface. The P+ diffusion forms the transistor emitter. The N- well underneath the diffused emitter forms the base and the adjacent P-substrate forms the collector. The substrate is grounded and a positive potential applied to the emitter forward biases the base emitter junction and reverse biases the N- well to P- substrate junction diode. This diode jointly forms the transistor collector and the light sensing diode structure of the pixel. Photons that are absorbed within the diode depletion region or within a diffusion length of the depletion region give rise to an excess carrier current in the well that is conducted through the base emitter junction. The transistor current gain results from the fact that under the forward active bias conditions the emitter collector current is nominally 100 times greater than the base emitter current.

5.3.2 BJT Pixel Gain Analysis

The maximum current gain available from a BJT is given by equation (5.3) [94]. Gray and Meyer [94] state that the gain is dependent upon the base width W_B , the emitter width W_E and upon the N_A/N_D ratio of the emitter/base doping densities. The diffusion coefficients given by D_p and D_n were calculated from the empirical equations (5.4) and (5.5) given by Moini [92]. The diffusion length of minority carriers in the base is L_p . This length was also calculated from the empirical equations (5.4) (5.6) and (5.7) given by Moini.

$$\beta_{F\max} = \frac{1}{\frac{W_B^2}{2L_p^2} + \frac{D_n W_B N_D}{D_p W_E N_A}} \quad (5.3)$$

$$D_p = \frac{kT}{q} \left(370 + \frac{370}{1 + 1.563 \times 10^{-18} N_D} \right) \quad (5.4)$$

$$D_n = \frac{kT}{q} \left(232 + \frac{1180}{1 + 1.125 \times 10^{-17} N_A} \right) \quad (5.5)$$

$$\tau_p = \frac{1}{7.8 \times 10^{-13} N_D + 1.8 \times 10^{-31} N_D^2} \quad (5.6)$$

$$L_p = \sqrt{D_p \tau_p} \quad (5.7)$$

Due to commercial considerations, CMOS foundries will not divulge the doping concentrations that they employ in the creation of wells or surface diffusions. However, based on published parameters for CMOS processes [63,64,92,94,95] estimates were made of the doping concentrations and the well and diffusion depths for a 2.4 μm CMOS process. The P+ doping concentration N_A was assumed to be $1 \times 10^{19} \text{ cm}^{-3}$ and the N- doping concentration N_D was assumed to be $2.5 \times 10^{16} \text{ cm}^{-3}$. The P+ diffusion depth W_E was set to 0.4 μm and the base width W_B was set to 2.6 μm .

Substitution of the assumed values in the empirical equations gives a $\beta_{F\max}$ for the vertical PNP transistor of 180. Yang [64], and Gray and Meyer [94] describe the fall of β_F as the level of forward active bias is reduced. This is expressed by the inclusion

of a third ratio term in the denominator of equation (5.8). The β_F reduction is a result of recombination within the base emitter depletion region. The third term in the β_F denominator gives the ratio of the emitter current I_{pE} equation (5.9) to the recombination current I_{rg} equation (5.10).

$$\beta_F = \frac{1}{\frac{W_B^2}{2L_p^2} + \frac{D_n W_B N_D}{D_p W_E N_A} + \frac{I_{pE}}{I_{rg}}} \quad (5.8)$$

$$I_{pE} = -qAD_p \frac{n_i^2}{N_D W_B} e^{\frac{V_{BE}}{\Phi_T}} \quad (5.9)$$

$$I_{rg} = \frac{qAn_i W_{dE}}{2\tau_o} e^{\frac{V_{BE}}{2\Phi_T}} \quad (5.10)$$

Equations (5.9) and (5.10), as given by Yang [64], illustrate the dependence of the emitter and recombination currents on the bias potential V_{BE} . In these equations, the area of the emitter is given by A , τ_o [64] is the carrier lifetime in the base emitter depletion region and W_{dE} is the width of this depletion region. The factor '2' in the denominator of the I_{rg} exponential term causes the recombination current to become more significant as the level of bias applied to the base emitter junction is reduced. As the level of bias is reduced the transistor gain is decreased. It was reported by Yang [64] that at high bias levels, the transistor gain would fall off as the base injected minority and majority carrier densities become comparable. As a result of these reductions in gain the useful range of the transistor is limited to a variation in collector currents of three or four orders of magnitude.

In order to exploit the useful gain range of the vertical BJT within the CMOS N-well, it was necessary to match the moderate bias condition of $V_{BE}=0.63$ to the mid range pixel current. This current was set for the Smart CMOS Camera by the Table 5.1 *Assembly Work* illumination of 500lx. An initial estimate of this mid range current was taken as 0.71 μ A, given by a transistor β_F of 180 and the N-well to P-substrate pixel current evaluated at 500lx in Table 5.1. In the CMOS fabrication process the

area of the emitter A is available for adjustment. The emitter area that will satisfy this bias condition of $V_{BE}=0.63$ volts and $I_{pE} = 0.71\mu\text{A}$ is given by equation (5.11) as a $6\times 6\mu\text{m}$ square area.

$$A = \frac{I_{pE} N_D W_B}{q D_p n_i^2} e^{\frac{V_{BE}}{\Phi_T}} \quad (5.11)$$

In Figure 5.5 a simulation gain plot for a BJT with a $6\times 6\mu\text{m}$ emitter is given. The plot was calculated from equation (5.8) with:-

$$\begin{aligned} A &= 36\mu\text{m}^2 \\ N_A &= 1\times 10^{19} \text{ cm}^{-3}, \\ N_D &= 2.5\times 10^{16} \text{ cm}^{-3} \\ W_E &= 0.4\mu\text{m} \\ W_B &= 2.6\mu\text{m} \\ \tau_o &= 15\times 10^{-6} \text{ sec.} \end{aligned}$$

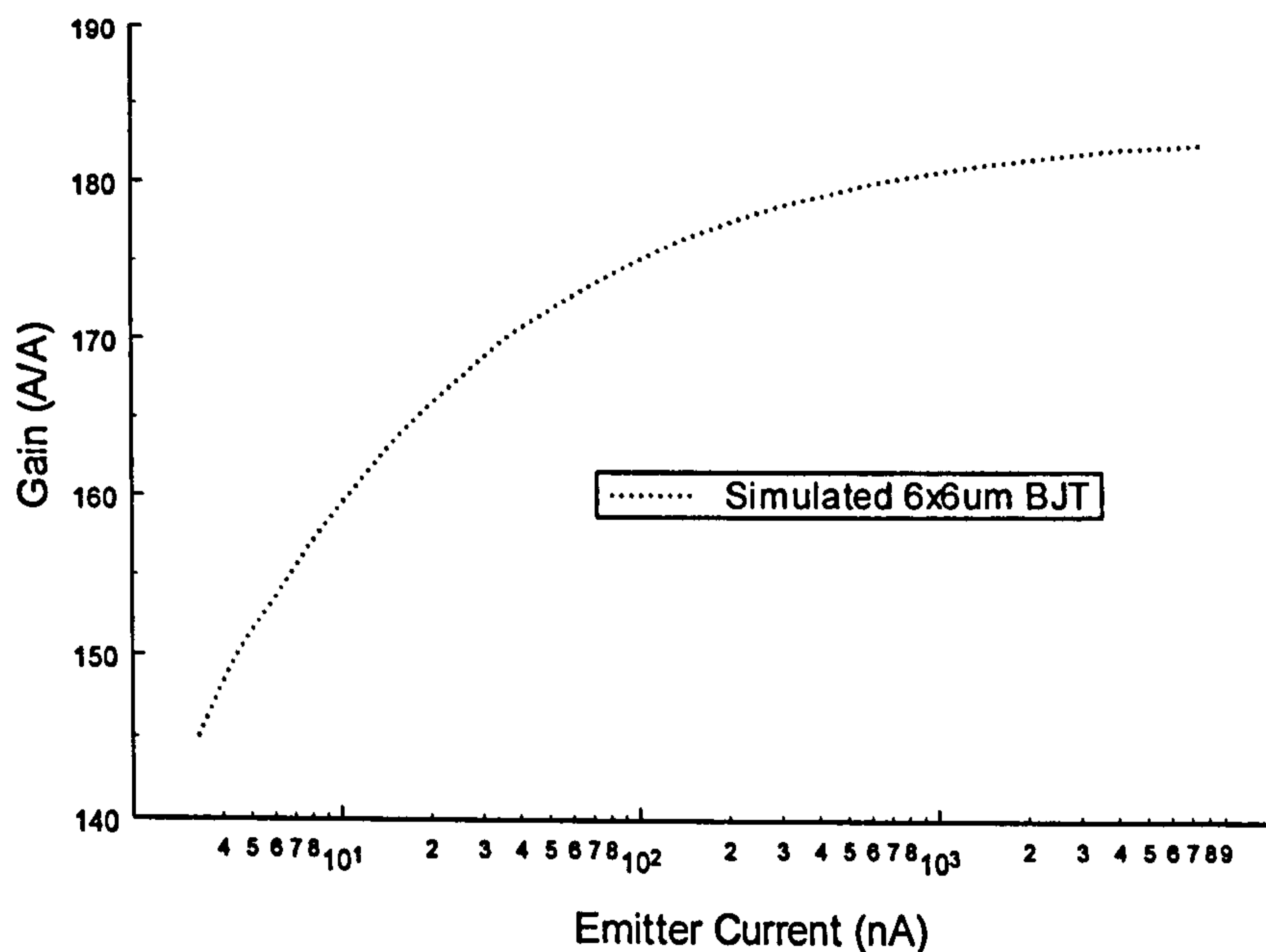


Figure 5.5 Simulated Gain for BJT Emitter $6\times 6\mu\text{m}$

The gain is plotted against emitter currents ranging from 2nA to $5\mu\text{A}$. This range is representative of the currents to be expected from a $80\times 80\mu\text{m}$ BJT pixel with a 50%

fill factor. The evaluation indicates that gains of greater than 145 are available across this current range when the emitter size is formed by a $6 \times 6 \mu\text{m}$ diffusion area.

5.3.3 BJT Pixel Gain Measurement

An experiment was set up to test the assumptions of the gain characteristics of the vertical BJT embedded in the N- well to substrate pixel structure. On a Mietec $2.4 \mu\text{m}$ CMOS substrate, two N- wells both measuring $120 \times 160 \mu\text{m}$ were formed. One of these was used to create a N- well to P- substrate diode sensor. The other was used to create a diode sensor with integral BJT as illustrated in Figure 5.4. These devices exceed the Smart CMOS Camera pixel $56 \times 56 \mu\text{m}$ sense area by a factor of 6. This was done to ensure that the expected diode currents could be measured. The layout of these N- well detectors is illustrated in Figure 5.6. The diode detector was formed by connecting to the N- well sense surface through N+ diffusions. The BJT detector was formed by connecting to the N- well sensing surface through P+ diffusions. The emitter for this pixel measured $435 \mu\text{m}^2$.

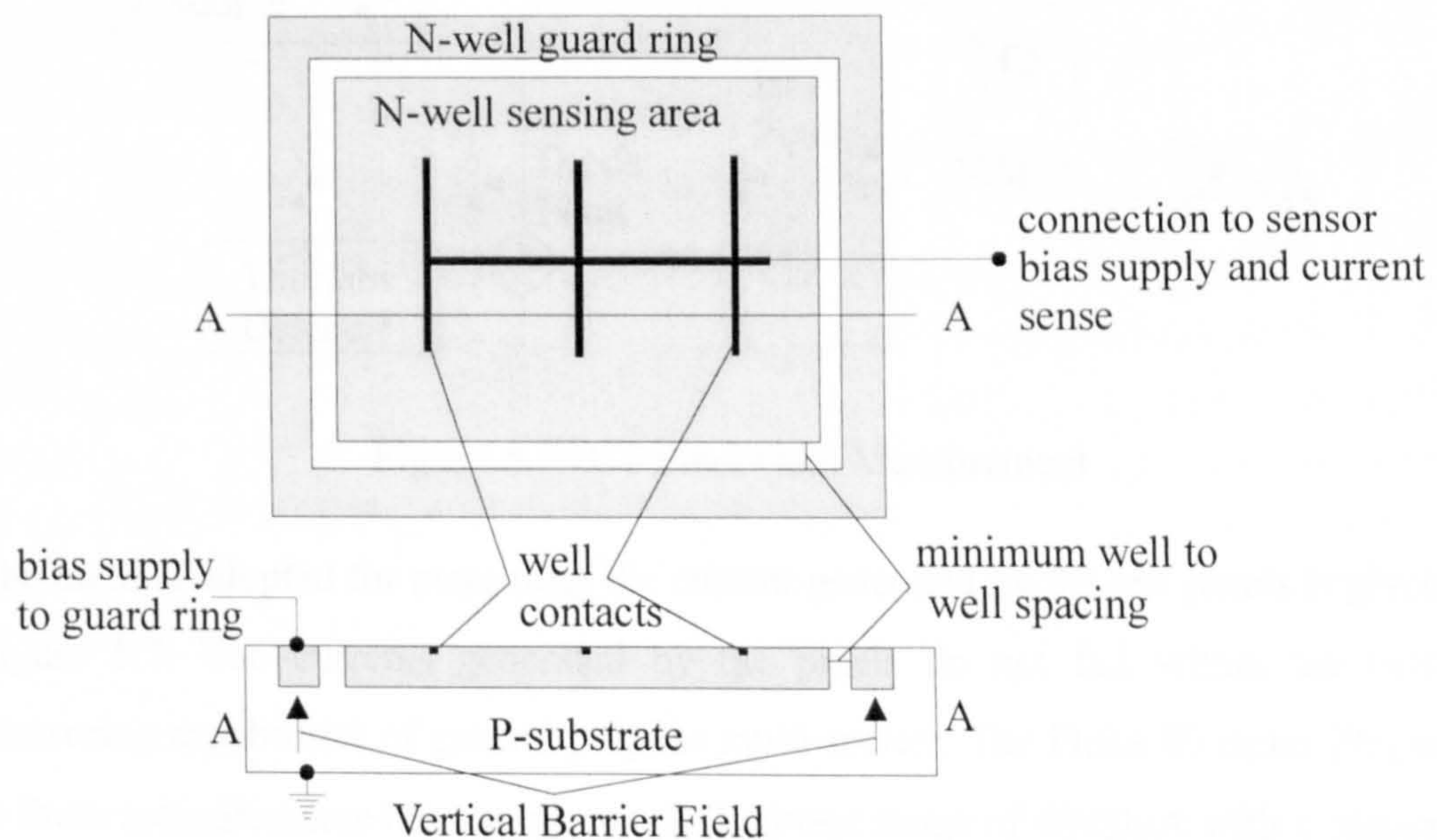


Figure 5.6 Layout of the $120 \times 160 \mu\text{m}$ Diode Detectors

In the test structure the maximum separation between the contacts to the N- well sensing area was set to $35 \mu\text{m}$ to replicate the losses due to recombination centres

within a $56 \times 56 \mu\text{m}$ well with a $6 \times 6 \mu\text{m}$ central emitter. The maximum diffusion length of $310 \mu\text{m}$, calculated from equation 5.7, indicates that the detectors of Figure 5.6 can collect photon-excited carriers from an area spanning $730 \times 780 \mu\text{m}$. For the image-sensing array it is necessary to limit the pixel's response to light incident upon the pixel sense area. The N-well guard ring that encloses the N-well sensing area is biased to create a vertical field that will collect carriers from out with the sense area and conduct them away into the guard ring bias circuit.

The experimental set up to measure the BJT pixel gain is illustrated in Figure 5.7. A diffuse white light source illuminated a pair of the diode and BJT sensors, with layouts as illustrated in Figure 5.6. A Thorlabs Calibrated detector was used to measure the level of incident illumination. In the experiment the incident illumination was varied between 50lx and 5000lx. This range was chosen to reflect the levels of incident illumination to which the Smart CMOS Camera was expected to respond.

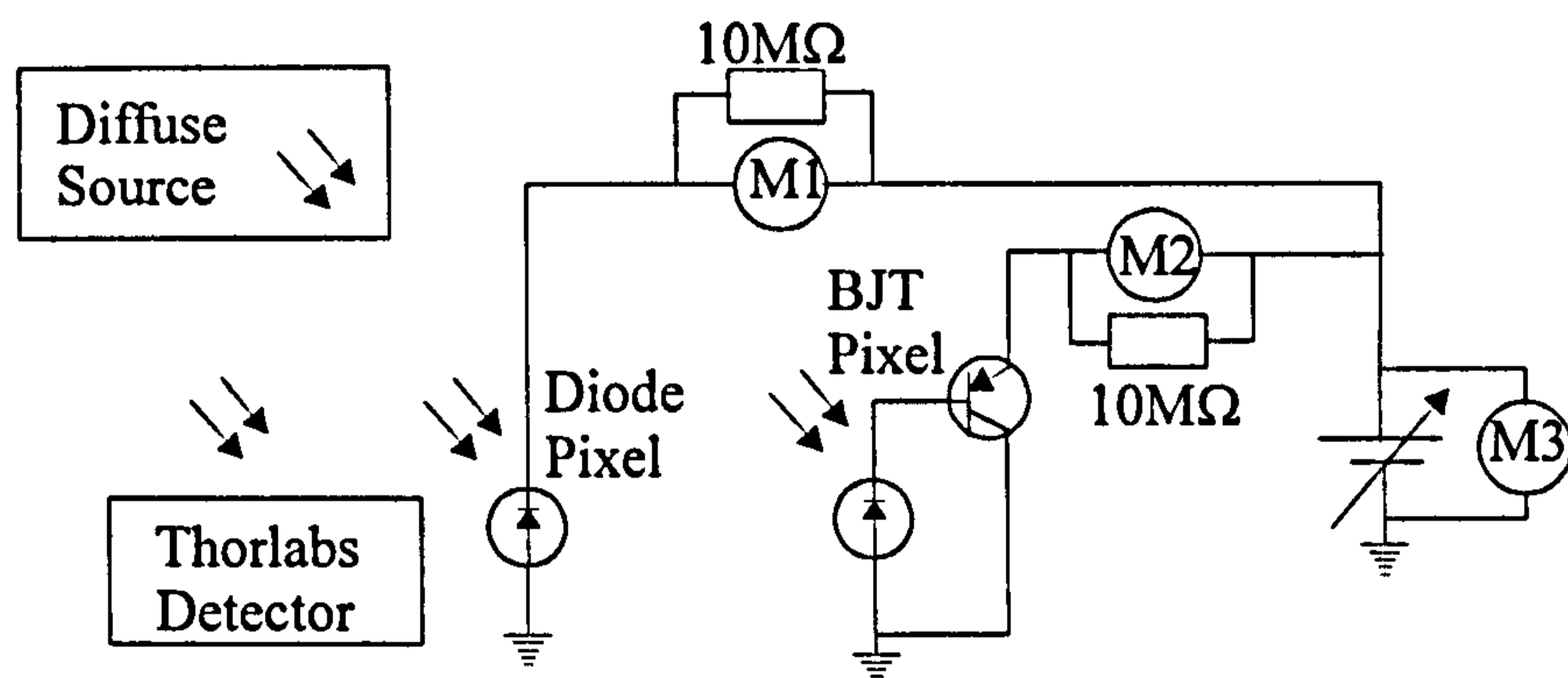


Figure 5.7 BJT Pixel Gain Measurement

The method adopted for measuring the current generated by the test pixels is given in Figure 5.7. The currents generated by the pixels do not fall within the current measuring capabilities of general purpose multi-meters. The Fluke 85 meter [96] used in these measurements has a minimum DC current range of $400.0 \mu\text{A}$ with a minimum resolution of $0.1 \mu\text{A}$. Thus the expected diode currents ranging from 2.4 nA to 240 nA cannot be measured directly by this meter.

The photo current measurement method of Figure 5.7 exploits the $10\text{M}\Omega$ standardised input impedance of the Fluke 85 multi-meter. When the meter is set to measure dc volts a standardised $10\text{M}\Omega$ resistance exists across the input terminals. In the battery powered Fluke 85 this $10\text{M}\Omega$ resistor accounts for input leakage currents. The M1 and M2 meters in Figure 5.7 are set to the DC volts mode. The $10\text{M}\Omega$ input impedance acts as a current sense resistor. When the Fluke 85 is set to the 400.0 mV range, this measurement technique gives DC current measurement with a scale maximum of 40.00nA and a resolution of 10pA. Setting the Fluke 85 to the 40.00V range gives a scale maximum of $4\mu\text{A}$ and a resolution 1nA. The bias applied to the pixel is evaluated from the volts registered by M3.

In the experiment the current sourced by both pixels was continuously monitored. A ratio was taken of these two currents to give the BJT gain. In Figure 5.8 the measured gain is plotted against the BJT emitter current. The $120\times 160\mu\text{m}$ pixel exhibited a gain that varied from 120 to 170 as the emitter current was varied between 20nA and $5\mu\text{A}$. Figure 5.8 includes simulation results for an equivalent BJT. The simulated results were given by equation (5.8) with:-

$$A = 435\mu\text{m}^2$$

$$N_A = 1\times 10^{19}\text{ cm}^{-3},$$

$$N_D = 2.5\times 10^{16}\text{ cm}^{-3}$$

$$W_E = 0.4\mu\text{m}$$

$$W_B = 2.6\mu\text{m}$$

$$\tau_o = 15\times 10^{-6}\text{sec.}$$

A good match is recorded between the simulated and measured results. This indicates that the general form of equation (5.8) is valid. Further verification of the listed diffusion, width and the lifetime terms used in equation (5.8) are required if this equation is to be generally applied. However, the results of Figures 5.8 and 5.5 demonstrate that the useful gain of the BJT pixel extends over three orders of magnitude. Through the selection of the pixel emitter area, this useful gain range can be centred on the illumination range that the Smart CMOS Camera is expected to respond to.

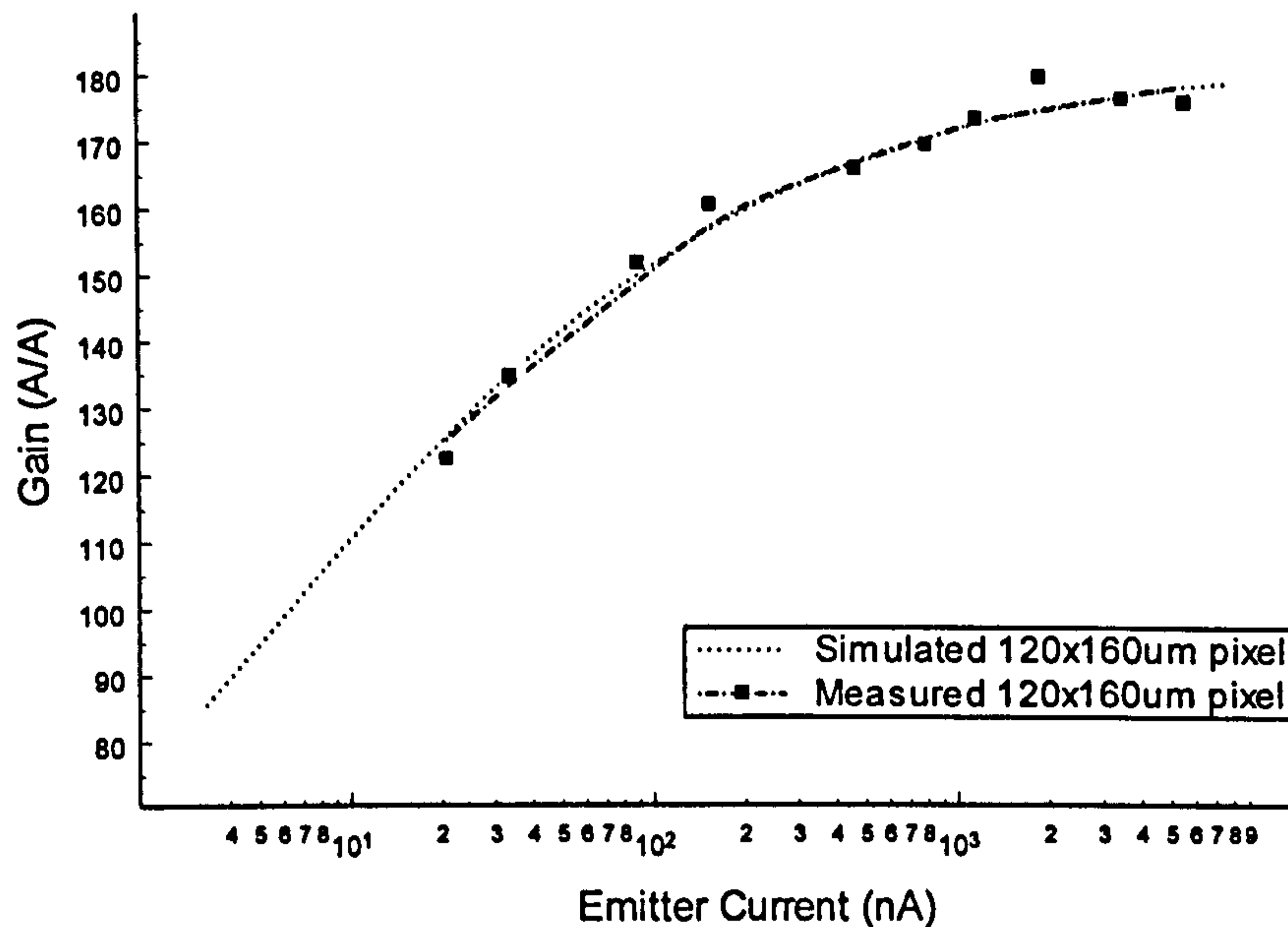


Figure 5.8 Measured and Simulated Gain of a 120x160 BJT Pixel

5.3.4 BJT Pixel Layout and Switching Circuit

The switching necessary to implement both the vertical and horizontal access to the BJT pixels is illustrated in Figure 5.9. In order for the BJT pixel to comply with the image sensing array random access requirement, it was necessary to ensure that when the device was not being accessed its bias conditions were maintained. If the bias is removed from the BJT pixel the charge stored in the light sensing reversed-bias diode will leak away. A recharge through the emitter base junction of the transistor will introduce a significant switch on time delay and thus reduce the pixel access rate. Figure 5.9 illustrates the means by which the pixel bias was maintained for the two orthogonal scans. Standby switching transistors were included to connect the BJT emitter to a bias supply in the periods when the pixel was not being accessed. These switching transistors were controlled by signals supplied by the row and column selection circuitry.

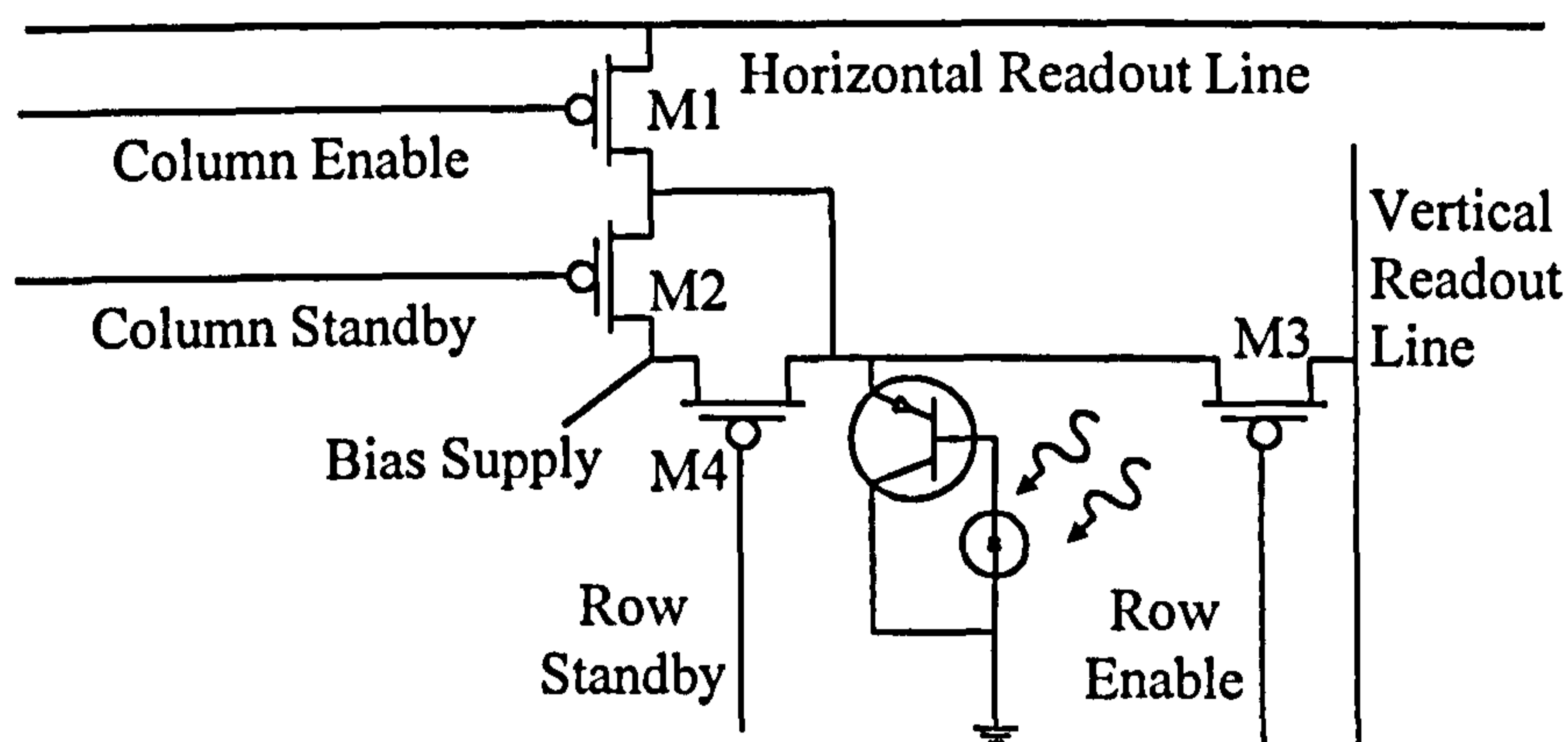


Figure 5.9 Pixel Readout Switching Circuit

The horizontal readout is implemented by switching MOSFETs $M1$ and $M2$. In standby mode, $M2$ is ON and $M1$ is OFF to connect the BJT to the bias supply and isolate it from the readout line. In read mode, $M2$ is OFF and $M1$ is ON. During the horizontal scan both $M3$ and $M4$ are OFF. The timing diagram of Figure 5.10 illustrates the switching of MOSFETs in columns 5 and 6 needed to place the photo currents from pixels 3,5 and 3,6 on the row 3 readout line.

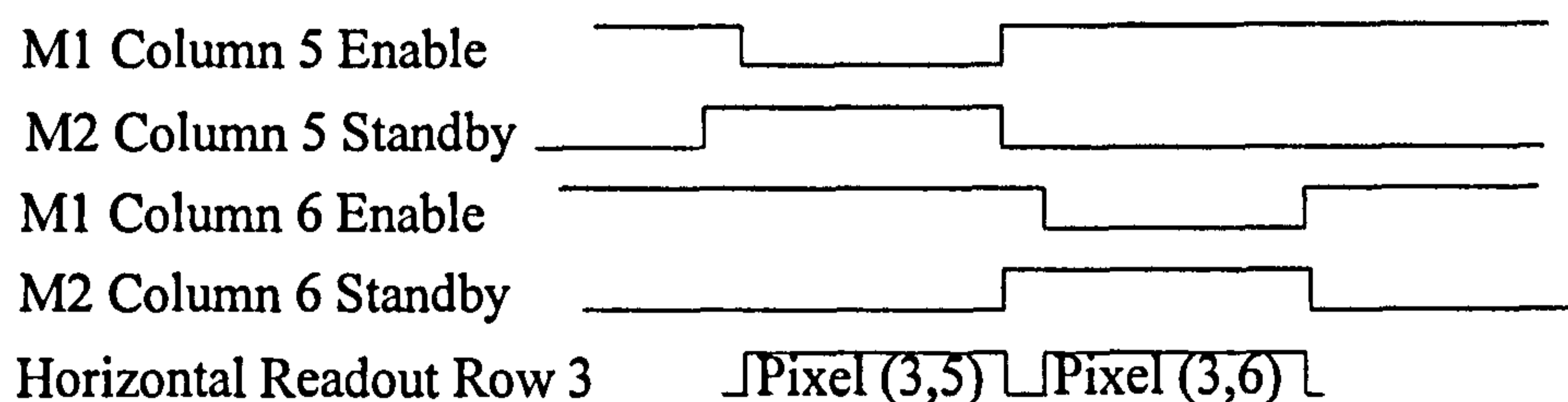


Figure 5.10 Pixel Access Timing Diagram

The layout of the BJT pixel, inclusive of row and column switching devices is illustrated in Figure 5.11. The central N-well is reverse-biased through the potential applied to the central P+ diffusion. This P+ diffusion, that forms the emitter of the BJT, is $4\mu\text{m}$ wide and $9\mu\text{m}$ long to comply with the SLA NSIP emitter area evaluated from equation (5.11). The guard ring that surrounds the pixel sensing area is connected to the sensor's V_{dd} potential. This provides a substrate for the P-channel

switching MOSFETs of the pixel readout circuitry. The guard ring also isolates the pixel receive area from light induced carriers that are created through absorption out-with the pixel $80 \times 80 \mu\text{m}$ sample space. This avoids smearing of the processed image.

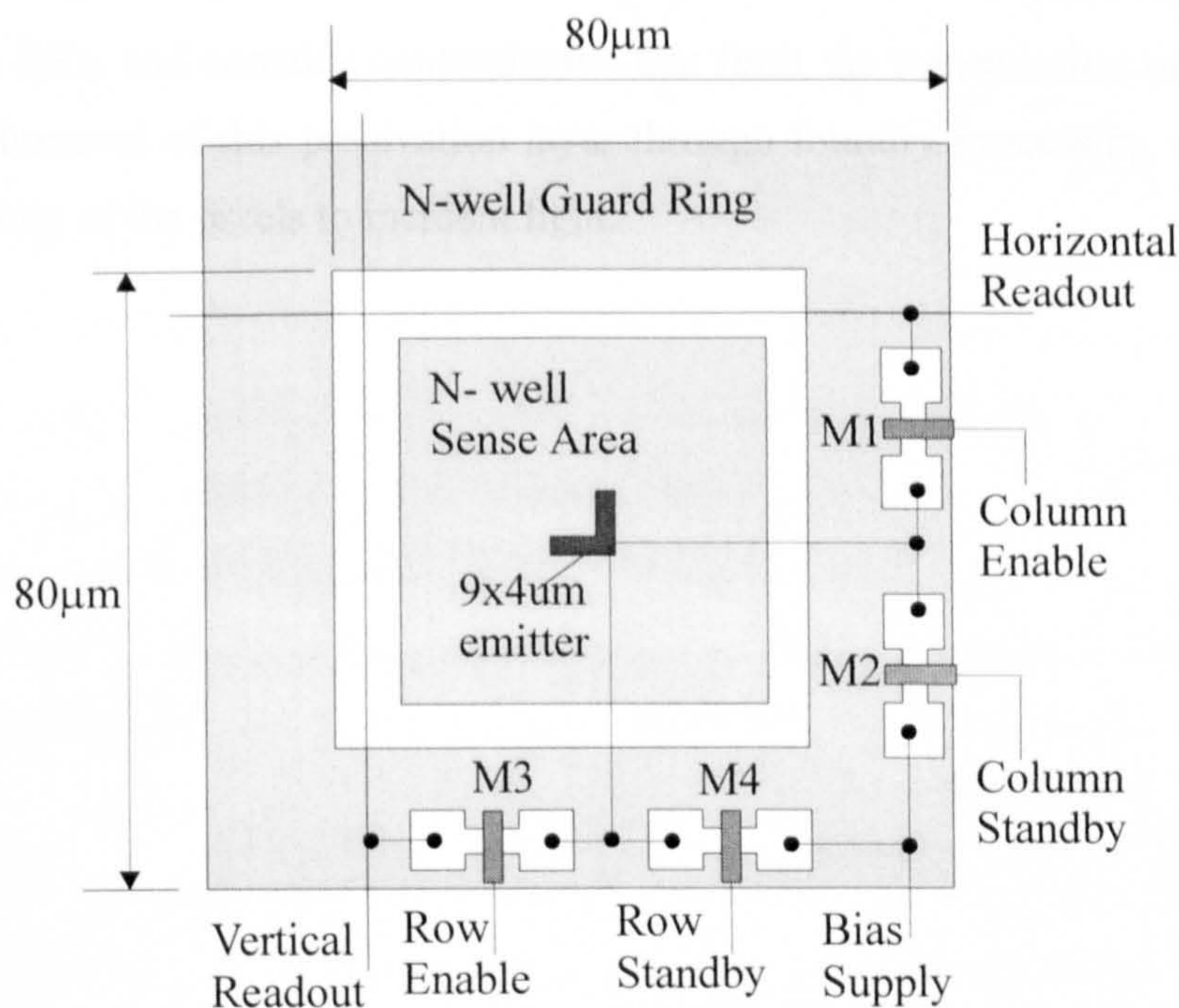


Figure 5.11 $80 \times 80 \mu\text{m}$ Pixel Layout

The switching circuit for each pixel occupies two wings of the N-well guard ring. This leaves two wings of the guard ring available for adjacent pixels to locate their row and column switching devices. The $80 \mu\text{m}$ dimensions for the pixel are marked on Figure 5.11. In this array layout it was found that the bias supply and the column and row selection lines could be routed to each pixel over the space occupied by the guard ring. Thus the shielding of the N-well sense area from incident light was limited to routing necessary for the sensed photo current.

An implementation of the $80 \times 80 \mu\text{m}$ pixel structure, illustrated in Figure 5.11, was fabricated in the Mietec $2.4 \mu\text{m}$ process. The photo current measurement set up given in Figure 5.7 was used to measure emitter current under General Office lighting conditions, Table 5.1. This nominal illuminance of 200lx gave an average emitter current of 100nA . This signal current is capable of supporting a maximum pixel read rate of 59.5KHz , as given by the pixel read rate calculation of Section 5.2. Thus the

random access requirement pixel access rate of 1.3KHz, can be met for scenes illuminated to the level of a *General Office*. The pixel test structure had the passivation layer in place over the 56x56 μ m sense area. The passivation layer is formed by SiO₂ and contains contaminants that limit the transmission of light to the detector. Removal of this passivation layer through foundry processing will increase the sensitivity of the pixels to incident light.

5.4 Current Mode Processing in the SLA NSIP

5.4.1 Sub-Threshold Operation for Spatial Derivative Processing

The results of Section 5.3 established that the BJT pixel structure would generate sufficient instantaneous photo-current to drive the readout line of the 100x100 array, at a frame rate of 6.5Hz, when the viewed scene was a general office environment. The photo currents generated by the BJT pixels are less than $2\mu\text{A}$ and as such are classed as sub-threshold currents. It was decided to let these photo currents set the quiescent currents in the spatial derivative circuits implemented by the SLA NSIP processors. This approach was adopted to facilitate the current mode circuit implementation for the spatial derivatives. In current mode operation the wide dynamic range of the pixel-generated currents can be accommodated without recourse to range switching.

A MOSFET operated in the sub-threshold mode does not have an inversion channel formed under the gate. Diffusion rather than drift accounts for the charge transport across the MOSFET [63]. The diffusion transport mechanism is slower than the drift mechanism and sub-threshold circuits are limited to low frequency processing. In Section 5.2 it was noted that the 6.5Hz frame rate, combined with the parallel processing of row and column data, set the pixel process rate to 1300Hz. This is low frequency processing which can be accommodated by the subthreshold mode of operation. In this mode the power consumed by the CMOS circuits is considerably less than that when the circuits operate under strong inversion. The low power consumption was also seen as beneficial in the realisation of the Smart CMOS camera.

The SLA algorithm detects edges on the basis of 1st and 2nd order spatial derivatives. The Smart CMOS Camera employs NSIP structures to implement these spatial derivatives. The derivatives are evaluated by summing and subtracting the photo-current generated by pixels local to the processed site. The horizontal and vertical readout lines given in Figure 5.11 route the localised pixel photo currents from within the array to the horizontal and vertical NSIP structures at the edge of the pixel array. Parallel processing within these NSIP structures required the layout of the spatial

derivative circuits to be realised within the 80 μ m pixel pitch of the image-sensing array.

5.4.2 1st Order Spatial Derivative

Current mode summations can be realised by forming a node that connects the high impedance outputs of two or more current sources. The SLA derivative convolutions introduced in Section 3.3 employ plus and minus integer coefficients. The negative coefficient requires an inversion of the current sourced from the readout line. Equation 5.12 expresses the dependence of the 1st order derivative at pixel site (n) on a pair of readout line outputs $I_{(n-1)}$ and $I_{(n+1)}$ selected from a row or column sequence. For the researched NSIP implementation the variable n ranges from 0 to 99.

$$d_n^1 = I_{(n-1)} - I_{(n+1)} \quad (5.12)$$

The current difference circuit developed to give a 1st order differential is illustrated in Figure 5.12. In this, a pair of pixel photo currents represented by $I_{(n-1)}$ and $I_{(n+1)}$ are processed through mirror circuits to give a differential output at V_o . The W/L ratios of the mirror circuits are set for a 1:1 ratio between the input and output currents. Variations in the summing node output voltage, v_o indicate the magnitude and sense current difference between the input pixel currents. Figure 5.12 is a transresistance circuit where the variation in the output voltage v_o is given by equation (5.13). The transresistance R_T is given by the parallel combination of the two MOSFET output resistances $Ro_{(n+1)}$ and $Ro_{(n-1)}$.

$$v_o = (I_{(n-1)} - I_{(n+1)}) \left(\frac{Ro_{(n-1)} * Ro_{(n+1)}}{Ro_{(n-1)} + Ro_{(n+1)}} \right) = (I_{(n-1)} - I_{(n+1)}) (R_T) \quad (5.13)$$

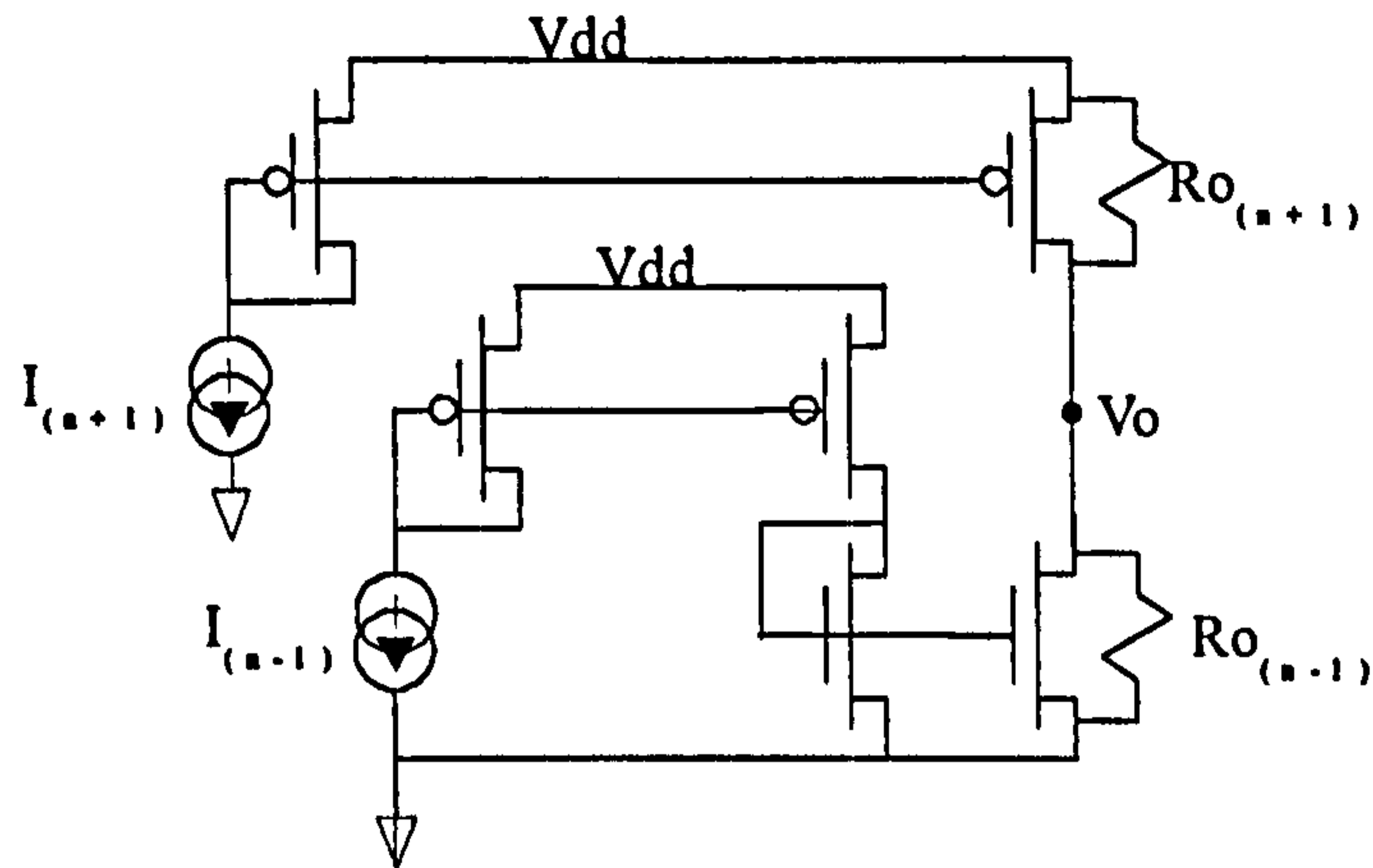


Figure 5.12 Current Mirror Implementation of 1st Order Derivative

The output impedance R_o of a MOSFET under sub-threshold operation is given by Tsividis [63] in equation (5.14). This is valid while the device remains in saturation; the saturation operation of sub-threshold devices being given by V_{DS} greater than 130mV at 300K [63]. The voltage V_{AW} is the equivalent of the BJT Early Voltage and it is dependent upon the length of the MOSFET channel. Tsividis evaluates V_{AW} from equation (5.15) where the L is the MOSFET length, and K_{AW} is an extracted device parameter with the units V/ μ m.

$$R_o = \frac{V_{AW}}{I_{DS}} \quad (5.14)$$

$$V_{AW} = K_{AW} L \quad (5.15)$$

If the output resistances $R_{O(n-1)}$ and $R_{O(n+1)}$ are matched and the input currents $I_{(n+1)}$ and $I_{(n-1)}$ are equal then the summing node V_o potential will be $V_{DD}/2$. Laker and Sansen [97] specify representative values of K_{AW} , as 4V/ μ m for an N-channel device and 7V/ μ m for a P-channel device. Thus for this set of representative constants a V_{AW} match is obtained when the length of the N-channel device is set to 1.75 times the length of the P-channel device.

When the current differential between $I_{(n-1)}$ and $I_{(n+1)}$ is small relative to the common mode current, the derivative output v_o of the circuit is given by equation (5.16) where

the common mode current is given by $(I_{(n-1)}+I_{(n+1)})/2$. In equation (5.16) the R_T is expressed in terms of V_{AW} and the input currents. In this R_T is inversely proportional to the common mode current. This inverse dependence gives rise to a constant contrast sensitivity response from the current differential circuit. A 1% derivative between $I_{(n-1)}$ and $I_{(n+1)}$ will give a v_o output of $0.005V_{AW}$. This contrast sensitive response closely mimics the spatial contrast response of the outer plexiform layers in biological retinas [98].

$$v_o = \left[(I_{(n-1)} - I_{(n+1)}) \right] \left[\frac{V_{AW}}{(I_{(n-1)} + I_{(n+1)})} \right] \quad (5.16)$$

5.4.3 Measurement of V_{AW} .

The contrast sensitivity response of the Figure 5.12 circuit is dependent upon the degree to which the V_{AW} of the output MOSFETs varies with I_{DS} . An experiment was set up to measure the relationship between V_{AW} and I_{DS} across the operational range of the current mode pixel. In this experiment, the drain current I_{DS} was monitored while the drain source voltage V_{DS} was varied between 150mV and 5V for a series of fixed Gate to Source voltages V_{GS} . The values of V_{GS} were chosen to bias the test MOSFET in the subthreshold mode. The value of V_{AW} was then found by extrapolation of the saturation $I_{DS}:V_{DS}$ curve to the point where it crossed the V_{DS} axis.

The tested device in this experiment was a 4 μ m long p-channel device fabricated in the Mietec 2.4 μ m process. Results from this experiment are illustrated in Figure 5.13. The results established that as the bias level of I_{DS} was increased so V_{AW} increased. At an I_{DS} of 10nA, V_{AW} was 12.5V and at 2 μ A, V_{AW} was 18V. Thus the contrast sensitivity of the Figure 5.12 circuit would vary by approximately 50% across the range of currents expected in the SLA NSIP operation. This variation in V_{AW} indicates that the single extracted parameter K_{AW} [63,97] gives a limited approximation to the value of R_o .

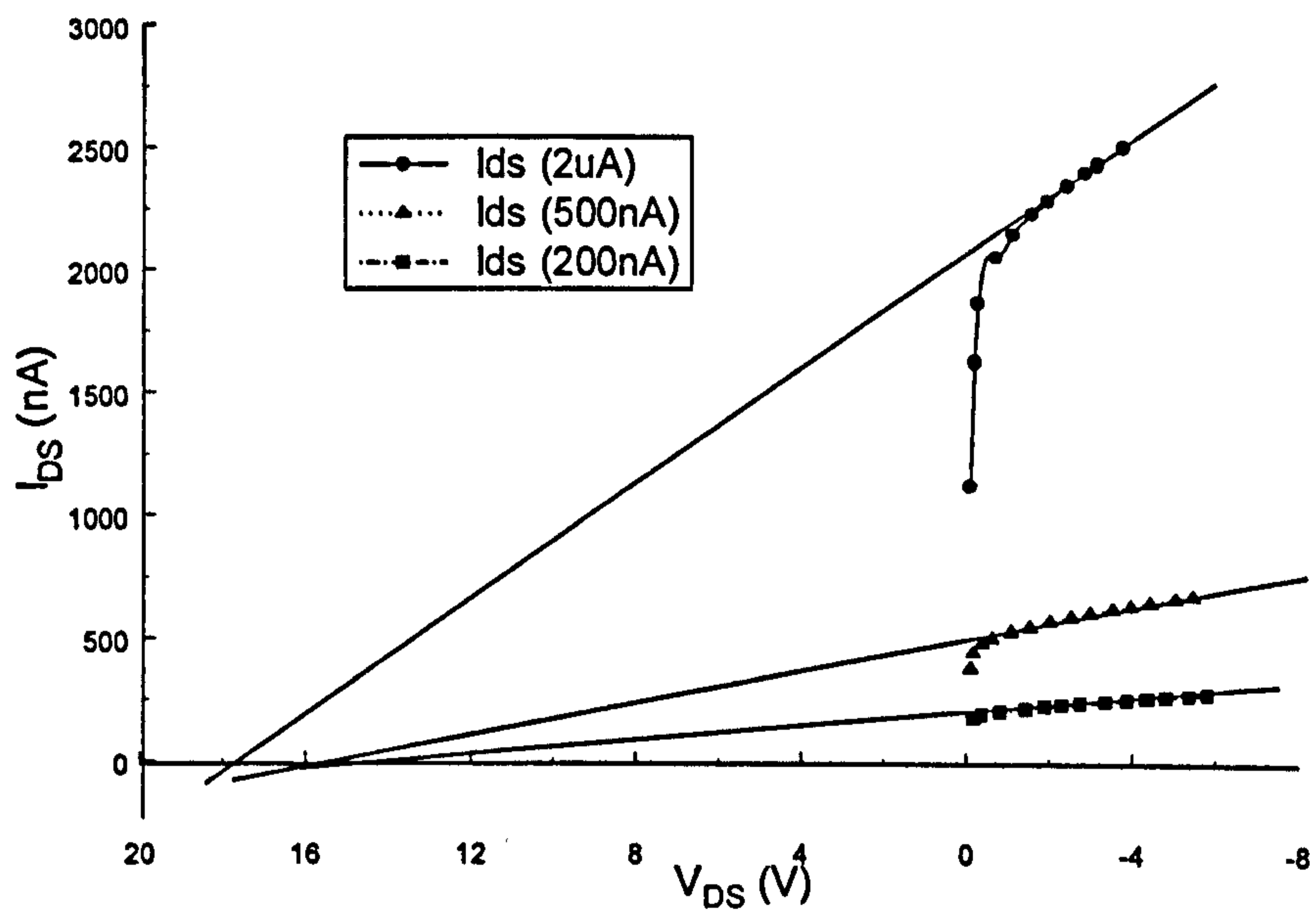


Figure 5.13(a) V_{AW} Measurements 2uA to 500nA

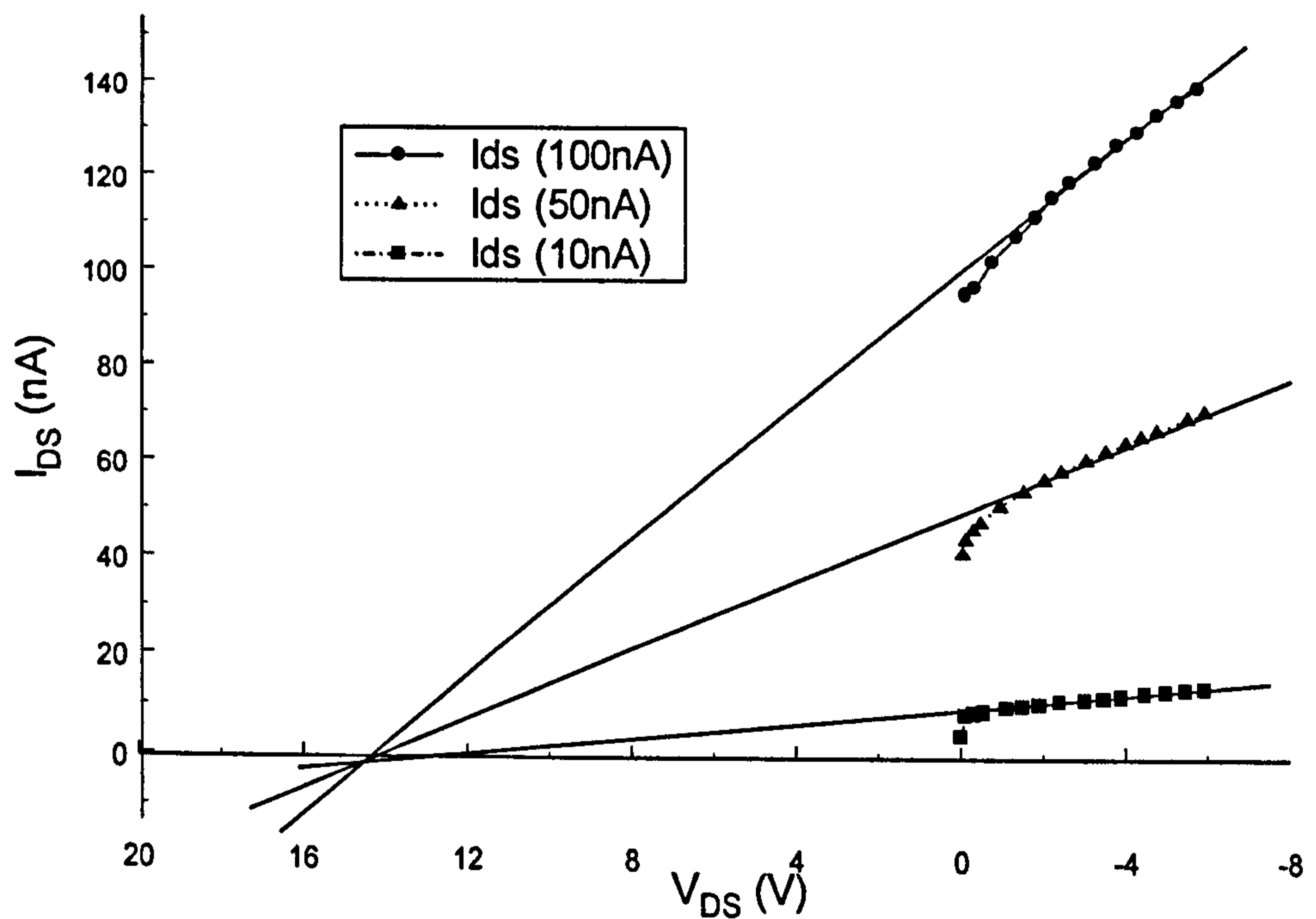


Figure 5.13(b) V_{AW} Measurements 100nA to 10nA

The SLA simulation results, of Section 3.3, established that for structural analysis of a scene, the 1st order differential circuit was required to detect percentage contrasts of

5% and the 2nd order differential was required to detect percentage contrasts of 1%. The scaling factor in equation (5.15) is given as $0.005V_{WA}$ per % contrast between the input photo currents. For the measured V_{AW} of 12.5V at a common mode input of 10nA, the scaling results in contrast sensitivity of 0.0625 V/ % contrast, and gives a v_o of 0.313V for a 5% contrast. If the common mode input is increased to 2 μ A the contrast sensitivity increases to 0.09 V/% contrast to, give a v_o of 0.45V for a 5% contrast.

5.4.4 Feedback Circuit to Match Complementary Output Conductances

The successful operation of the contrast sensitive spatial derivative circuit given in Figure 5.12 is dependent upon the matching of V_{AW} for the complementary MOSFET outputs. The measured results indicated that V_{AW} value was not dependent upon a single extracted device parameter. In order to limit the dependence of the spatial derivative circuit upon extracted device parameters it was decided to introduce feedback control into mirror circuits to match the output conductance of the complementary outputs at each readout line. The mirror circuit with back-gate feedback control [99] used to implement this conductance match is illustrated in Figure 5.14.

The Current Mirror circuit of Figure 5.14 was designed to provide conductance matching and implement the photo current signal inversion. In this circuit, the current from the array readout line (n) is processed to give two output sets. One of these sets gives three inverted versions of the pixel current $M9(a)$, (b) , (c) . The other gives three non-inverted versions of the pixel current $M10(a)$, (b) , (c) . These outputs may then be summed with complementary outputs from adjacent pixel readouts to give the 1st and 2nd order derivatives required for the implementation of the SLA edge point detector. The circuit of Figure 5.14 was designed to fit within the 80 μ m pitch of the SLA NSIP array, so that this circuit could be replicated on the substrate for each readout line.

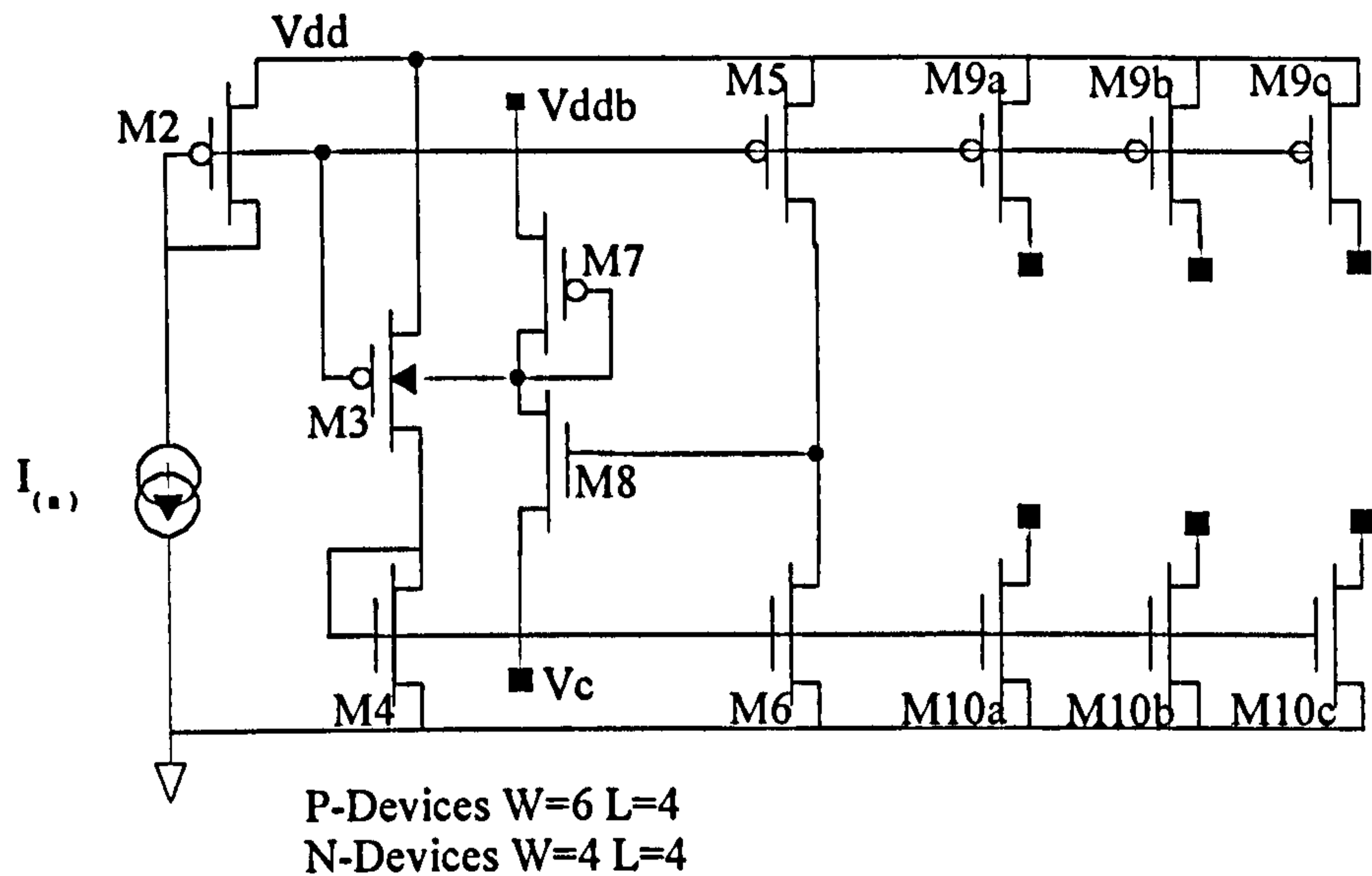


Figure 5.14 Line Readout, Current Mirror Circuit

The conductance of $M10$ devices is matched to that of $M9$ devices through the $M3$ - $M8$ circuit. The input pixel current is reflected into $M5$ through the 1:1 mirror of $M2$ and $M5$. A comparison between the external reference Vc and the $M5$ / $M6$ divider sets the back-gate voltage of $M3$ [99]. This back-gate voltage modulates a mirror of the pixel current that is supplied to $M4$ via $M3$. The $M4$ current is mirrored in $M6$ to close the control loop. The control loop maintains the $M5$ / $M6$ divider voltage at $(Vc + V_{GS8})$. When this voltage is set to $Vdd/2$, $M6$ and $M5$ have matched output conductance. The layout widths and lengths of $M2$, $M5$ and $M9$ devices are set equal so that the input current $I_{(n)}$ is reflected in the $M9$ outputs. The layout lengths of the $M6$ and $M10$ devices are set equal to match the output conductance of the $M10$ devices to the $M9$ devices.

5.4.5 Measurements on the 1st Order Spatial Derivative

The circuit configuration of Figure 5.15 illustrates the cross connection between two conductance match circuits necessary to give a 1st order derivative. Simulation tests were performed using the level 3 model cards supplied for the Mietec 2.4 μ m process. The tests were designed to test for the contrast sensitivity response predicted through the theory of Section 5.4.2. The current $I_{(n+1)}$, mirrored in $M9(a)$, was set as a

reference while the current $I_{(n-1)}$ mirrored in $M10(a)$ was set to vary by $\pm 1\%$ about the reference current.

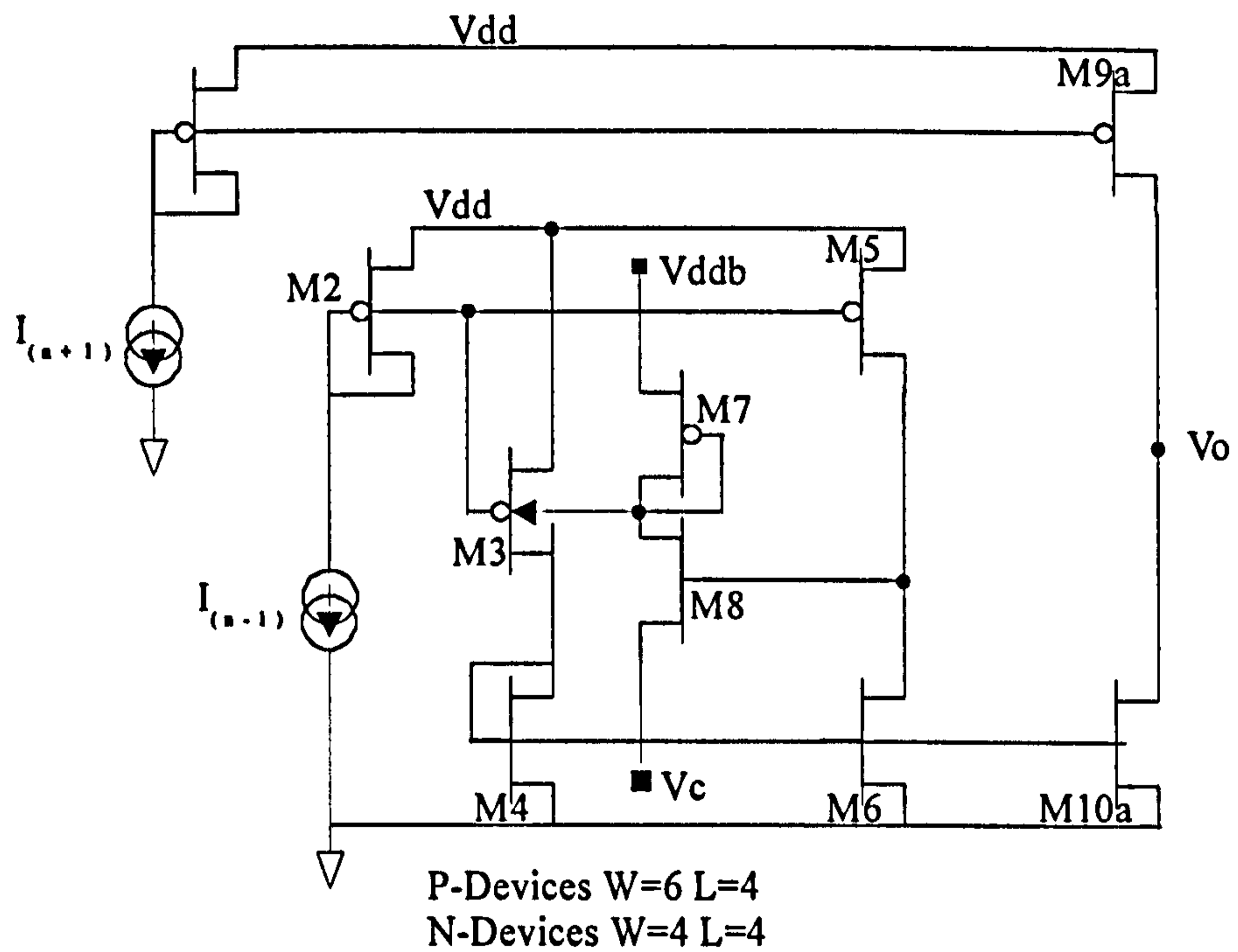


Figure 5.15 Contrast Sensitivity Evaluation Circuit

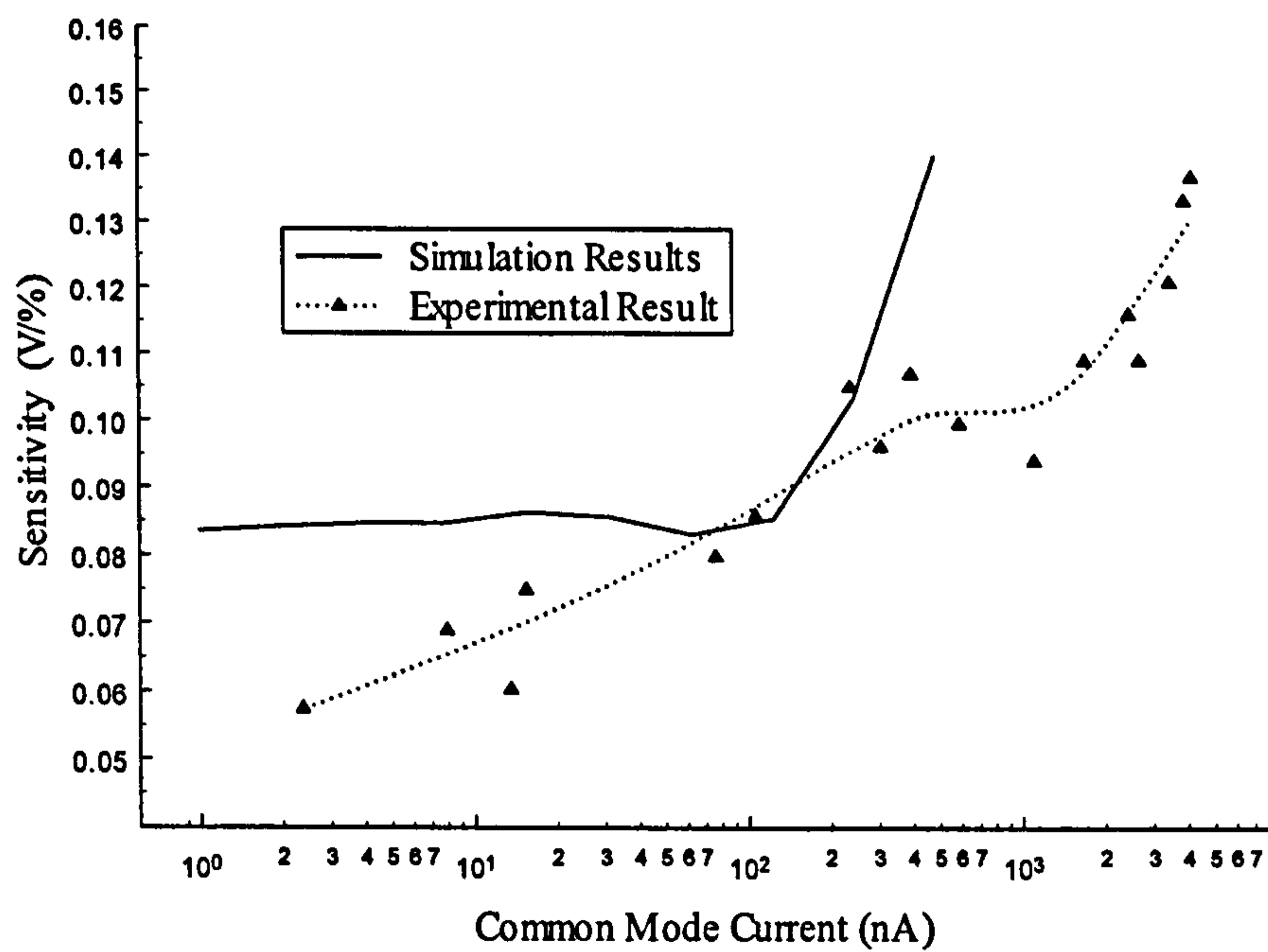


Figure 5.16 Contrast Sensitivity of Current Difference Circuit

The simulation results given in Figure 5.16 indicate a contrast sensitivity response of 0.08V/%contrast at an input reference current of 1nA. The contrast sensitivity response remains near constant at 0.08V/%contrast until the reference input is 100nA. At this point the simulation results register an abrupt change and reach 0.14V/%contrast when the reference input current is 500nA. The abrupt change was attributed to the level 3 simulation changing between sub-threshold and weak inversion modes.

An implementation of the Figure 5.15 test circuit was fabricated in the Mitec 2.4 μ m technology. In tests on this circuit the reference input $I_{(n-1)}$ was varied between 2nA and 4 μ A, while the $I_{(n+1)}$ input had a 1% variation applied about this reference current. The output V_O was monitored through a high input impedance FET probe. The contrast sensitivity results obtained from this experiment are plotted in Figure 5.16. The measured results confirm the direct relationship between contrast sensitivity and common mode current predicted from the measurements of V_{AW} of Section 5.4.3. At 2nA the contrast sensitivity measured 0.06V/%, whilst at 2.7 μ A, the contrast sensitivity measured 0.12V/%. The level 3 model used in the simulation does not include a parametric representation of the measured variation of V_{AW} . This limitation combined with the sub-threshold to weak inversion model change accounts for the limited agreement between the simulation and measured results of Figure 5.16.

The simulation tests were used to evaluate the frequency response of the 1st order derivative circuit given in Figure 5.15. The reference current was set to 10nA and a 10% contrast modulation was applied to the second differential input. The frequency of the 10% modulation was swept from 1Hz to 100kHz. The results of this frequency sweep are illustrated in Figure 5.17. The response is flat to within 5% until the modulation frequency reaches 10kHz and the 3dB frequency is at 30kHz. Additional tests demonstrated that if the reference current is increased, then the 3dB frequency is increased. These results indicate that the 1300Hz array sample frequency required to give a 6.5Hz framing rate can be achieved with the array pixel active areas set to 56x56 μ m in a 100x100 pixel array.

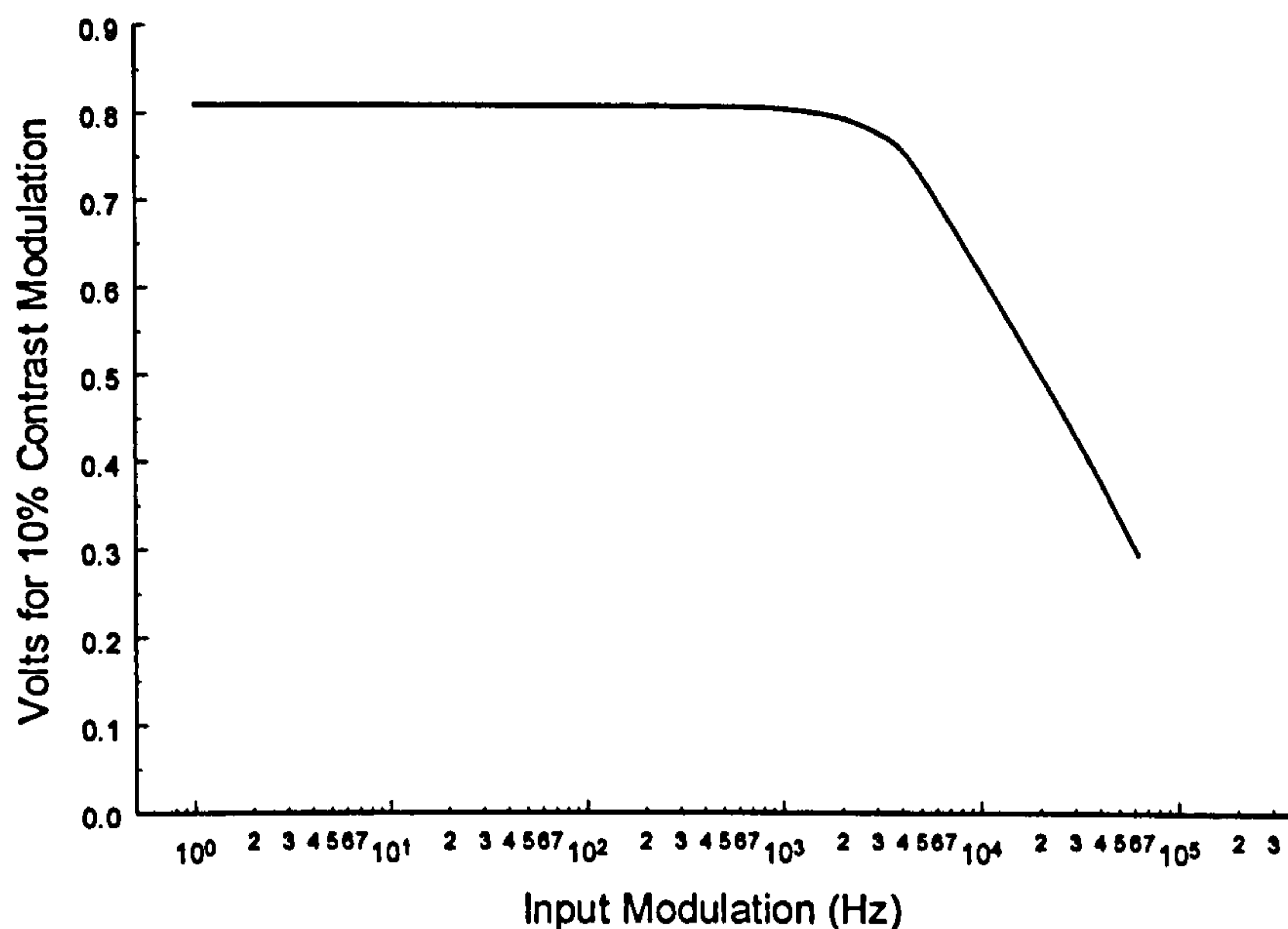


Figure 5.17 Frequency Response of 1st Order Derivative Circuit

The simulation and measured results for the 1st order spatial derivative circuit of Figure 5.15 demonstrate that this current differential configuration mimics the contrast sensitivity response found in biological retinas. The circuit does not require a range adjustment as the input currents are varied from 10nA to 5uA. Also within this range the circuit can support a 6.5Hz frame rate for arrays of pixels configured as illustrated in Figure 5.11.

5.4.6 2nd Order Derivative Circuit

Equation 5.17 expresses the dependence of the 2nd order derivative on three readout line outputs, $I_{(n)}$, $I_{(n-2)}$ and $I_{(n+2)}$. The 2nd order derivative is generated by summing two negative reflections of a central readout line with positive reflections from two adjacent readout lines. The circuit implementation for this derivative is given in Figure 5.18. The 2nd order derivative makes use of the mirrored outputs $M9(b)$, $M9(c)$, $M10(b)$ and $M10(c)$ given by the Figure 5.14 circuit. The 1st order derivative illustrated in Figure 5.15 used the mirrored outputs $M9(a)$ and $M10(a)$. Thus, from the circuit of Figure 5.14, at each readout line, two separate summing nodes are formed, one of these gives a 1st order spatial derivative and the other a 2nd order derivative.

5.5 Three Layer SLA NSIP Edge Detector.

The SLA edge detection algorithm is implemented through the three layer edge detection circuit illustrated in Figure 5.19. This is the NSIP implementation of the SLA algorithm. In *layer 1* the 1st and 2nd order derivatives are generated through summation nodes formed as illustrated in Figures 5.15 and 5.18. The summing node derivatives have an analogue signal format. In *layer 2* the analogue derivatives are converted into discrete format through window comparison circuits. In *layer 3* the 1st and 2nd order discrete derivatives are combined through logical operations to assign edge points. The circuit of Figure 5.19 illustrates the layered circuits necessary for edge point assignment at location (*n*). This edge detector operates with a span of 7 pixels.

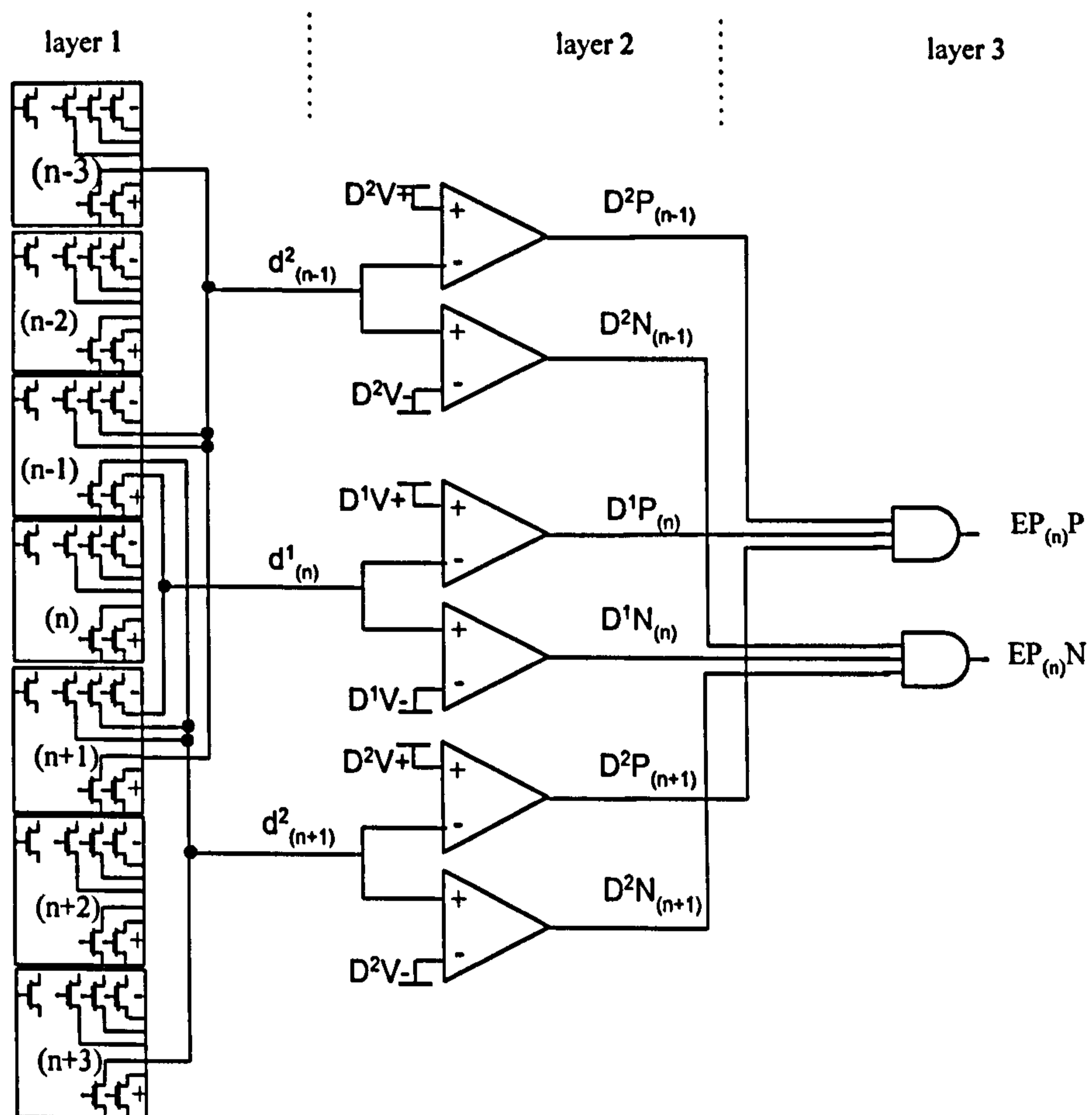


Figure 5.19 SLA NSIP Edge Point Detector

The *layer 1* summing node derivatives are coupled into the window circuits of *layer 2*. Each window circuit is comprised of two voltage mode comparators. These compare the summing node voltages to positive and negative thresholds. In the 1st order windowing operation applied to the $d'_{(n)}$ summing node output, the positive threshold $D'V+$ is set to $V_{dd}/2 + V_{threshold}$ and the negative threshold $D'V-$ is set to $V_{dd}/2 - V_{threshold}$. The value of $V_{threshold}$ is given by the contrast sensitivity threshold specified for the vision problem. If this contrast sensitivity is specified as 5% and V_{AW} for the circuit is 12.5V, then, as evaluated in Section 5.4.2, the value of $V_{threshold}$ is 0.313V. If the summing node input is greater than the positive threshold then the $D'P_{(n)}$ output is set high. If the summing node input is less than the negative threshold the $D'N_{(n)}$ output is set high. Otherwise both the $D'P_{(n)}$ and $D'N_{(n)}$ outputs are set low.

A schematic of the positive threshold comparator is given in Figure 5.20. Transistor $M1$ and R_{ext} provide the input stage bias current. The voltage generated at the drain gate connection of $M1$ is bus connected to all the positive threshold circuits. The value of V_{th} is set so that when V_o is greater than $V_{dd}/2 + V_{threshold}$ $M3$ changes to a low impedance state. A pair of inverters, given by device combinations $M4/M7$ and $M6/M5$, condition the switching signal generated by $M3$ and its load $M2$ into a logic compatible waveform. In Table 5.2 the results obtained from a test implementation of the positive threshold comparator are given. These demonstrate that a hysteresis of up to 0.15V exists between the high-low and low-high transitions of the comparator circuit.

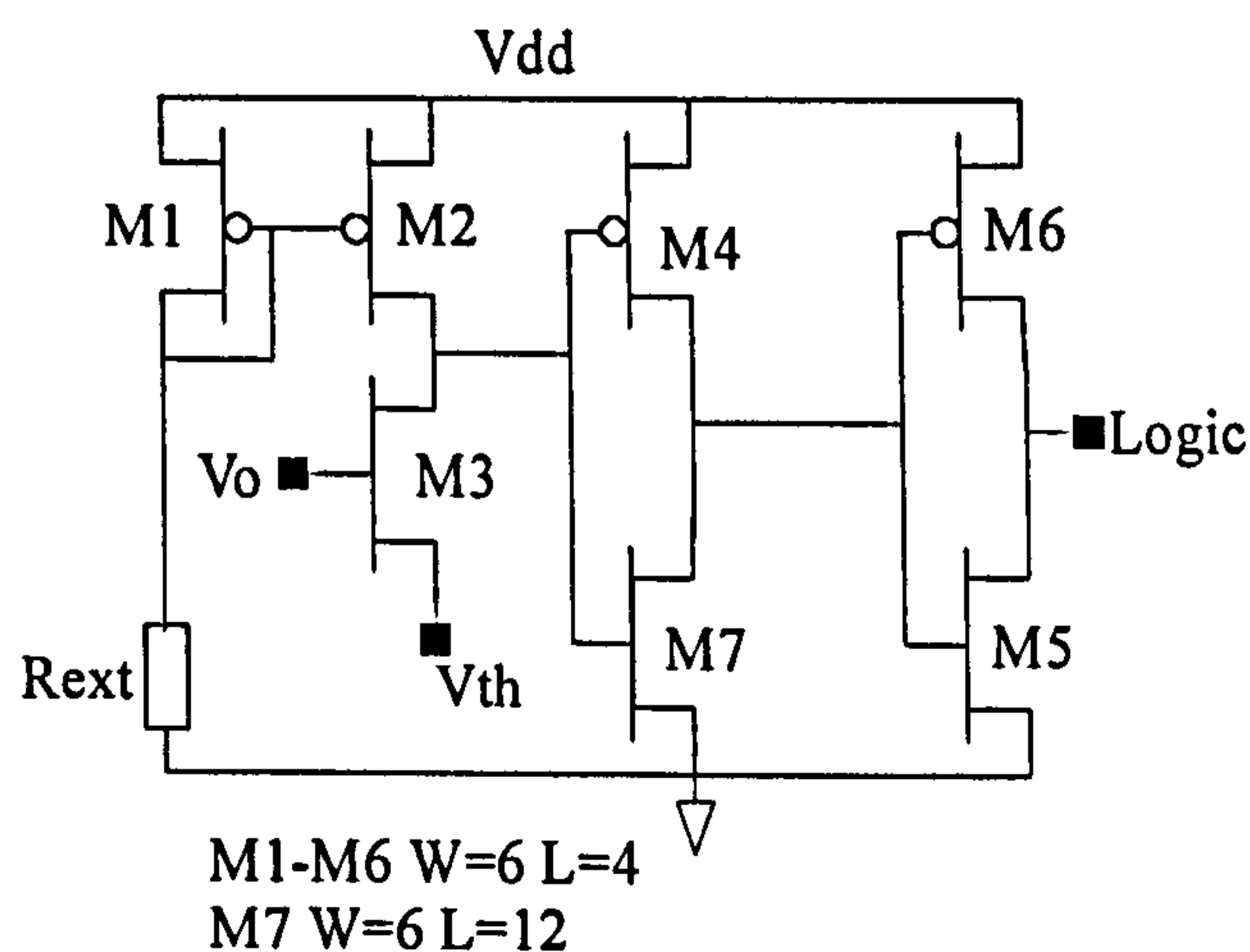


Figure 5.20 Positive Threshold Comparator Circuit

Vth (V)	Vo Low-High (V)	Vo High-Low (V)
0.4	1.95	2.10
0.6	2.25	2.39
0.8	2.50	2.65
1.0	2.76	2.91
1.2	3.02	3.12
1.4	3.31	3.39
1.6	3.70	3.75

Table 5.2 Comparator Switching Thresholds

In *layer 3*, the SLA algorithm completes the edge detection process through the edge point decision logic. The decision logic employs two three input AND gates. Truth tables for these gates are given in Table 5.3. This logic circuit implements the SLA discrete edge point assignment given by equation 3.22. The edge point has three possible states. The first is a zero state given by both the $EP_{(n)}P$ and $EP_{(n)}N$ outputs set low. The second is a positive state given by $EP_{(n)}P$ high and $EP_{(n)}N$ low. The third is a negative state given by $EP_{(n)}P$ low and $EP_{(n)}N$ high.

$D^2P_{(n-1)}$	$D^2N_{(n-1)}$	$D^1P_{(n)}$	$D^1N_{(n)}$	$D^2P_{(n+1)}$	$D^2N_{(n+1)}$	$EP_{(n)}P$	$EP_{(n)}N$
1	0	1	0	0	1	1	0
0	1	0	1	1	0	0	1
0	0	X	X	X	X	0	0
X	X	0	0	X	X	0	0
X	X	X	X	0	0	0	0

Table 5.3 Truth Table for SLA NSIP Layer 3 Edge Decision Logic

5.6 Smart CMOS Camera Edge Detection Test

An image sensing array composed of 10 rows and 4 columns of pixels was fabricated to test the operation of the NSIP SLA edge detection circuit. A layout overview of this test device is given in Figure 5.21. This illustrates the tiled structure of the pixel array and the adjacent layered NSIP structure. The three processing layers in the NSIP are interspersed with connection matrices that make the distributed connections necessary for the SLA algorithm.

Column selection circuitry was used to route the pixel outputs through horizontal readout lines to the NSIP structure. The circuits and routing of Figure 5.21 needed for the assignment of a single edge point were repeated in the NSIP test device for four edge point assignments. Column routing in the NSIP structure was used to supply the window circuit thresholds and the mirror circuit V_c control voltage.

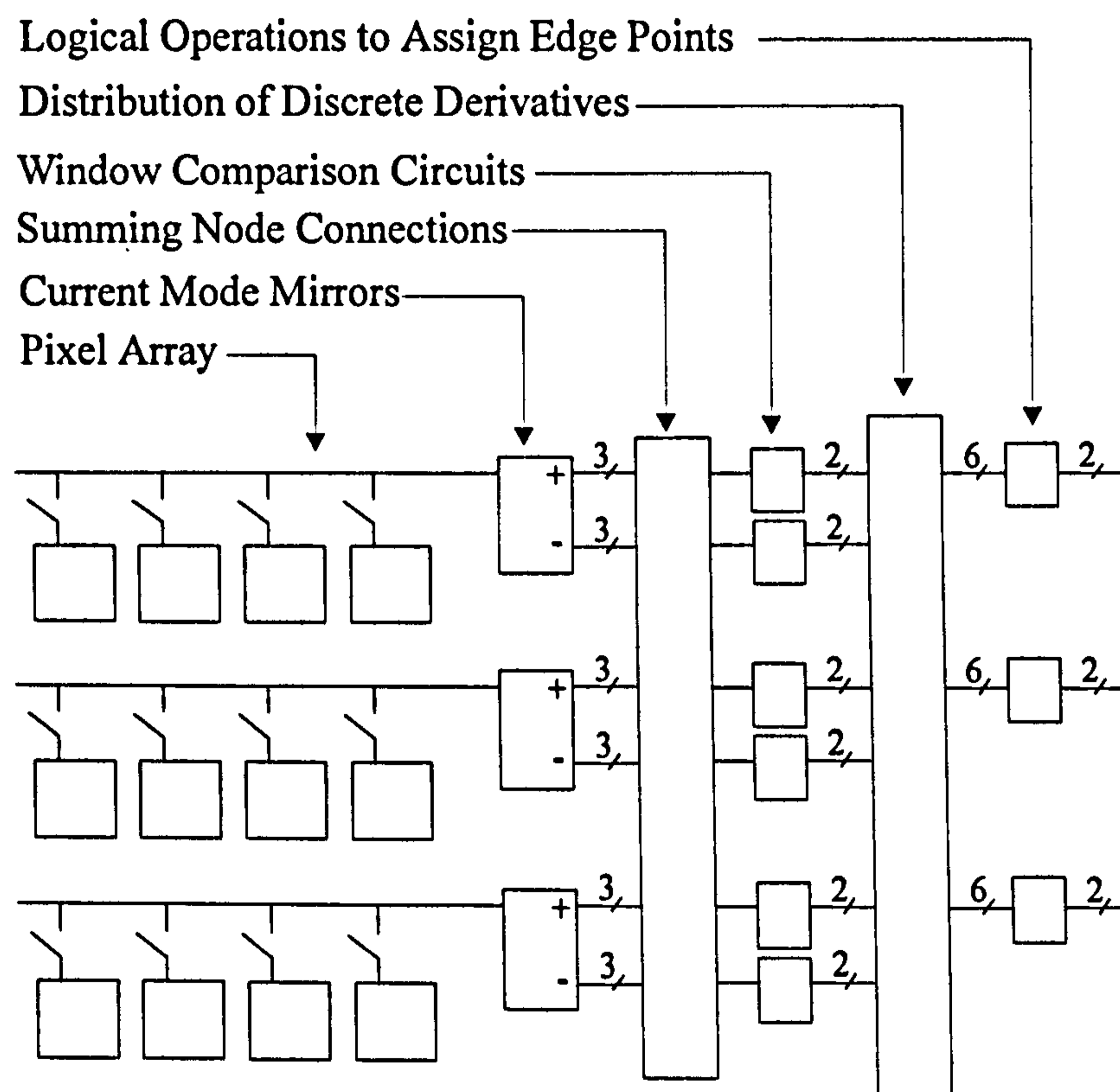


Figure 5.21 Layout Overview of the Smart CMOS Camera Test Chip

The measurement set up illustrated in Figure 5.22 was used to test the edge detector. The pixel array was directly illuminated by a white light source chopped by a rotating blade. As the shadow of this blade traversed the array the conditions for an edge with a contrast shift greater than 50% were created. Under these conditions it was found that the window threshold and the V_c control inputs could be adjusted so that one of the four edge point detectors would register both the positive and negative edges generated by the sweeping blade. Example Oscilloscope traces for the EP(n)P and EP(n)N edge outputs are given in Figure 5.23. The edge frequency was 10Hz. The phase shift between the traces result from EP(n)P registering the blade leading edge and EP(n)N registering the blade trailing edge. These results established the functionality of the SLA NSIP circuits.

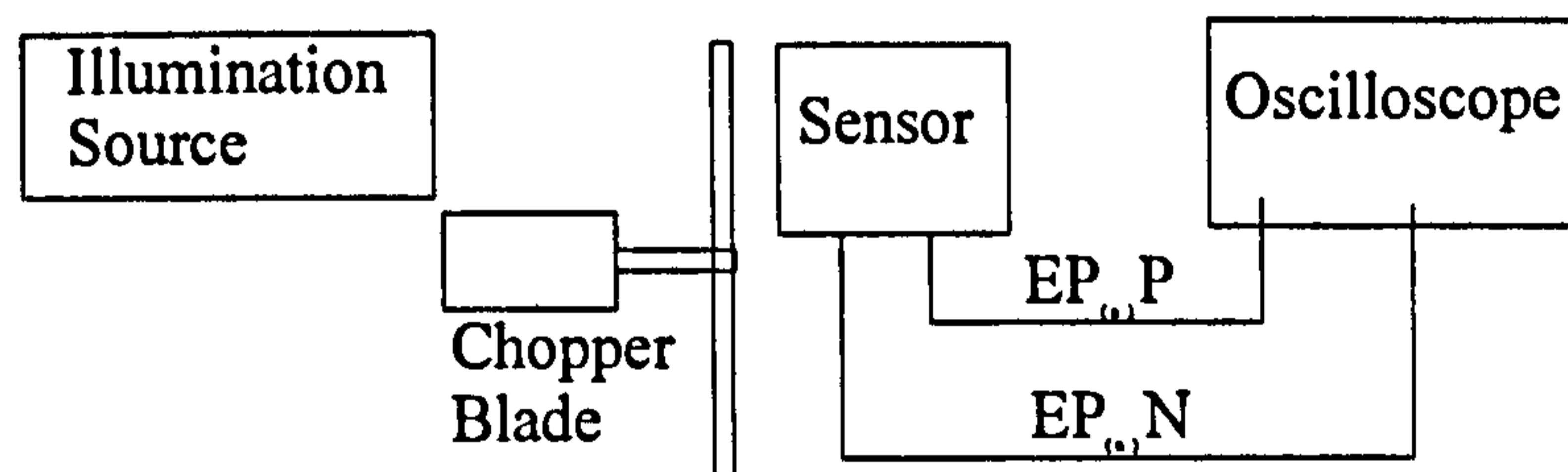


Figure 5.22 Edge Detection Test Instrumentation

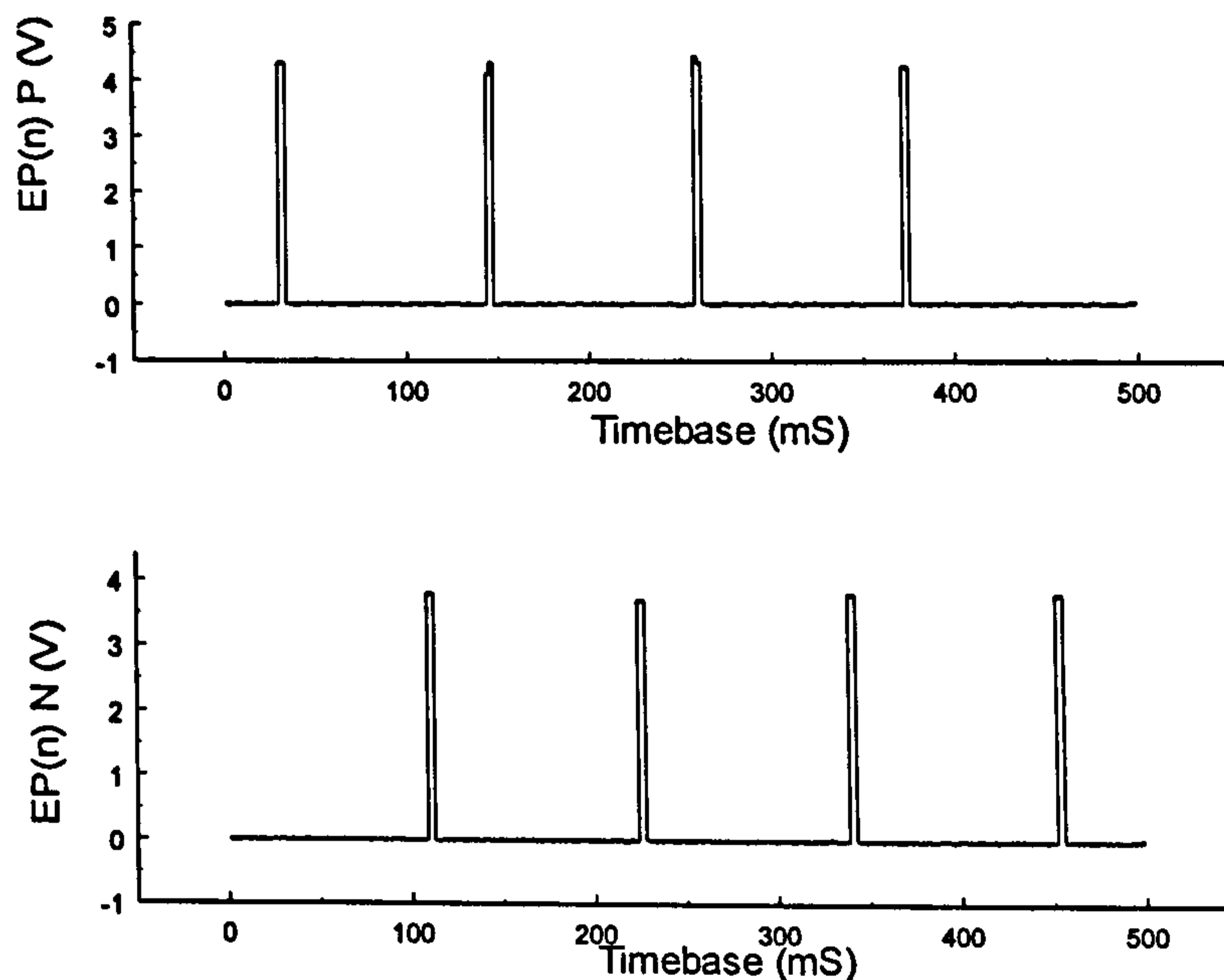


Figure 5.23 Results from the Edge Detection Tests

The Smart CMOS Camera test chip was limited to detection of edge points with spatial contrasts of greater than 50%. Tests carried out on four 1st order spatial derivative circuits established a variation in the summing node potentials of 1.7V, under uniform illumination of the array with V_c fixed. This order of variation in the summing node outputs would limit the facility of the windowing circuits to convert the derivative outputs into discrete representations. The results obtained in Figure 5.23 were obtained by setting the $V_{threshold}$ parameter to 1.1 V for both the 1st and 2nd order window circuits.

Possible causes of the variation in the summing node potential were considered. It was noted that a number of circuit features could contribute to the phenomena. Variations in the pixel gain due to differences in the doping densities of the P+ emitters across the array could contribute to the summing node voltage variations. In the current mirror circuits of Figure 5.14 variation in the MOSFET threshold voltages of $M2$ and $M5$ along the NSIP linear array could result in summing node voltage variation. Also in this circuit, the MOSFET $M8$ that compares V_c to a control sample of the summing node potential is susceptible to variation in the substrate potential. Further research is required to quantify the contribution that each of these mismatch factors makes to the noted variation in the summing potential.

5.7 Power Consumption

5.7.1 Vision Based Autonomous Navigation

A critical aspect of the development of the Smart CMOS Camera was the power consumption that could be expected for the device. The important query was; could such a device make a significant reduction in the power required to implement the low level vision processing necessary for autonomous navigation? A comparison is drawn between the power consumption of the CMOS implementation of the SLA algorithm and an equivalent low power **Complex Programmable Logic Device (CPLD)** implementation [100]. Consideration is given to the power consumption of a DSP devices in the implementation of low level vision processing. These considerations established that the Smart CMOS Camera does hold out the prospect of a providing a significant reduction in the power required to implement low level vision processing.

The workstation and PC based approaches to navigation based on visual sensing are indicative of a computationally complex problem [24-29]. The Smart CMOS Camera and its associated SLA algorithm implement a front-end accelerator for low level vision processing. This device was designed to reduce the power consumption due to low level vision processing in an autonomous battery powered system. The reported battery powered autonomous systems [30-35] employed sonar and laser ranging to deal with problems such as wall following and obstacle avoidance. These single point sensing mechanisms require considerably less processing capacity than that needed for pixel-array vision sensors. The battery powered autonomous systems [30-35] use lead acid batteries that provide 200 amp hours at 12V. This power resource was used to supply the motor drive, the human interface and the autonomous processing.

The subject of low level vision accelerators was addressed by Jordan and Holburn [7,8], when a CCD camera was closely coupled to a DSP device to minimise the signal transmission power costs. No specifics were quoted for the power consumption of this vision accelerator. However, Texas Instruments [43] quote the current drawn by the TMS320C80 as 1 amp for a typical low level vision process such as the Sobel edge detector. As noted in Section 2.3 the low connectivity of the processing afforded by DSP vision systems cannot generate the quality of segmentation results required by

the autonomous navigation processes. The SLA algorithm that was developed to meet the connectivity requirements of autonomous vision cannot be integrated into the 3x3 and 5x5 kernel structures that are provided by DSP vision processors [42,43].

5.7.2 CPLD Implementation of SLA Algorithm

The SLA algorithm was primarily developed for inclusion within a NSIP structure described in Sections 5.4 and 5.5. However, the diffuse processing structure and the use of integer only coefficients allows for a programmable logic implementation of this algorithm. The CPLD devices supplied by Xilinx [100] have low power consumption and are ideally suited to battery powered applications. Thus it was decided to investigate the power consumption of a SLA implementation based on these devices.

The system overview given in Figure 5.24 illustrates the use of four Xilinx XC95288XV devices in conjunction with a CMOS camera, a frame grabber and static RAM to implement the SLA algorithm. The image grabbed from the camera is loaded into a frame memory. The uniform averages and the horizontal and vertical scan mappings are implemented by CPLD1 and by CPLD3. The averaged and re-mapped data is loaded into horizontal and vertical frame buffers. An implementation of the SLA edge detector is programmed into CPLD2 and CPLD4, these devices generate horizontal and vertical edge sets for the input image and load this data into the output edge map memory.

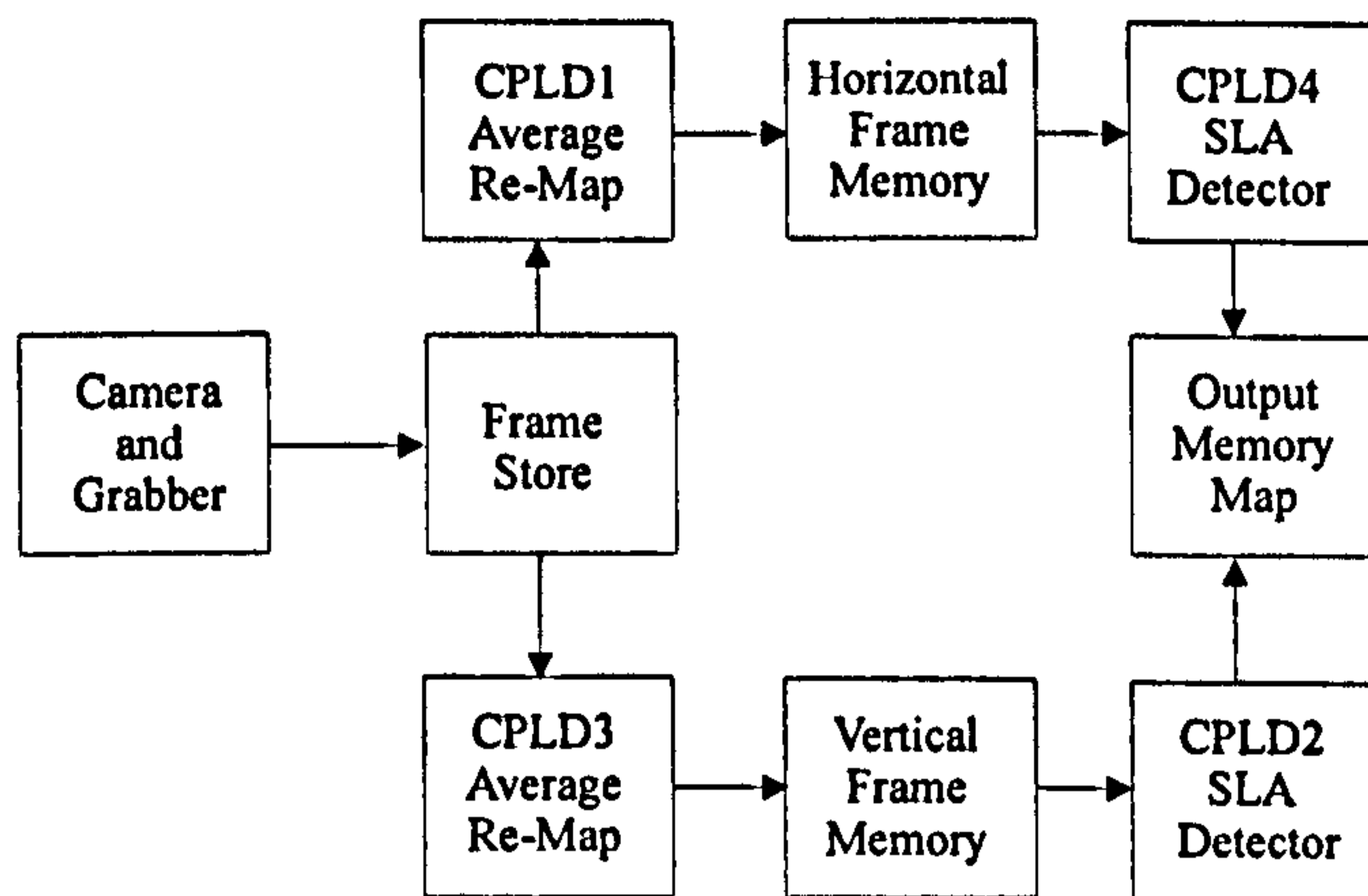


Figure 5.24 CLPD Implementation of SLA Edge Detector

The XC95288XV is the largest CPLD currently available from Xilinx, the 288 macro cells contained within this device do not allow a full implementation of the SLA algorithm within one device. Thus it was necessary to split the uniform averaging and derivative computations between two CPLD's for each scan direction. This device capacity restriction does not exist within currently available Field Programmable Gate Arrays (FPGA), however the FPGA's are not designed for low power implementations, and a single FPGA implementation would exceed the power consumption of the multiple CPLD solution.

In the power consumption evaluations for the CPLD implementation the processing requirements were set by the 100x100 array with a 6.5Hz frame rate specified for the Smart CMOS Camera in Section 3.6. The current draw evaluation for the XC95288XV is given in equation (5.18) [100]. In this equation MC_{HP} gives the number of macro cells in high-performance mode, MC_{LP} gives the number of cells in low-performance mode, MC is the total number of macro cells and f is the clock frequency in MHz.

$$I_{CC}(mA) = MC_{HP}(0.5) + MC_{LP}(0.3) + MC(0.0045 \text{ mA/MHz})f \quad (5.18)$$

Analysis of the registers and logic necessary to implement the scan re-mappings and a uniform filter of width 3 set the cell usage in CPLD1 and CPLD3 to 118 MC_{LP} . The

clock frequency that these devices are required to operate at is 3 times the edge point generation rate. There is limited scope for parallel operation in the CPLD implementation. Thus the edge point generation rate is set to 130KHz, that is 100 times the row and column access rate of the Smart CMOS Camera. Thus the clock rate for CPLD1 and CPLD3 is set to 390KHz. The re-mapping and uniform filtering processes give rise to a current draw of 35.5mA on the 2.5V supply used by the XC95288XV. The re-mapping and uniform average devices are enabled separately, each for 50% of the frame period. Thus the total power consumption due CLPD 1 and CPLD3 is 88.7mW.

Analysis of the registers and logic necessary to implement the derivatives, adaptive thresholds and edge decisions of the SLA edge point allocation gave a total of 125 MC_{LP} for the operation of CPLD2 and CPLD4. The clock frequency that these devices were required to operate at was five times the edge point generation rate, and this clock was set to 650KHz. Thus the current draw presented by both these devices given by equation (5.18) is 37.5mA. These two devices are separately enabled, each for 50% of the frame period. Thus the total power consumption due to CPLD2 and CPLD4 is 93.8mW. The total power consumption for the four CPLD devices is 182.5mW. The static memory power consumption was assumed to be negligible in comparison to the CPLD requirements. It was assumed that a low power CMOS camera and frame grabbing would add an additional 200mW power requirement to the proposed CPLD implementation. The power consumption for the Figure 5.24 implementation of the SLA algorithm is estimated to be between 350mW and 400mW.

5.7.3 Smart CMOS Camera Power Consumption

The power consumption of the Smart CMOS Camera is directly related to the level of illumination received by the device. This is a result of the quiescent current in the analogue section of NSIP structure being set to the line sense current and the necessity to have all the pixels in the array continuously biased at the full read potential. The analysis of the Smart CMOS Camera power consumption identified three separate

current draws that are made on the 5V V_{dd} supply. These are the array biasing and analogue processing current, the NSIP logic signal processing current, and the control logic current.

Consider the case of General Office illumination generating an average pixel current of 100nA then for the 100x100 array the total array current will be 1mA. Under these same conditions with the SLA width set to 3, the line current fed to each of the current mirror circuits will be 300nA. In the current mode circuit six separate V_{dd} to 0V channels exist. Thus the 100 mirror circuits will draw a total of 0.18mA from the device supply. The total illumination related current is thus 1.18mA when the average pixel current is set to 100nA.

In the NSIP discrete processing the bias current drawn by the input to the voltage comparators was set to 4μA by R_{ext} in Figure 5.20. Four of these comparators are used for each array readout line. A total of 100 readout lines are processed to give a 1.6mA current draw from the power supply. The power consumption of the switching inverters is given by equation (5.19) [101]. Where the switching frequency f_{SLA} is set to 1300Hz, as specified in Section 3.6. The inverter load capacitance C_L is 40fF. N_I is the total number of inverter drives. In the NSIP comparator section 500 inverter drives are activated at each array access cycle. In the NSIP circuit an additional 600 inverter drives are required to implement the output AND gates. Thus a total of 1100 inverter drives are activated for each array access in the Smart CMOS Camera. The power consumption of these inverters is given by equation (5.19) as 1.43μW. The comparator first stage bias current consumes 8mW, thus the NSIP switched inverter drive power consumption can be neglected.

$$P = N_I C_L V_{dd}^2 f_{SLA} \quad (5.19)$$

The design of the array pixel selection circuitry, and output addressing circuitry has not been included in the circuit descriptions of Chapter 5. However, from the array access and the device control requirements it has been estimated that a total of 3312 inverter drives are needed to allow the device to run of an external 2600Hz clock. The

inverter load capacitance is given as 40fF. Thus from equation (5.19) the array access and device control circuitry will consume 8.6 μ W.

The total estimated current draws and power consumption for the Smart CMOS Camera in the operational environments considered in Section 5.2 are given in Table 5.4. The power consumption of the inverter drives in the Smart Camera is 10 μ W, and this can be neglected. The current draw estimates for the Smart CMOS Camera supply were determined from a summation of the illumination related current drawn by the pixel array and analogue processing, and the current need to bias the first stage of the comparator circuits.

Operational Environment	Illuminance (lx)	Current Draw (mA)	Power Consumption (mW)
Storage Area	50	1.90	9.5
General Office	200	2.78	13.9
Assembly Work	500	4.55	22.8
Inspection	1500	10.45	52.2
Fine Detail Work	5000	31.10	155.5

Table 5.4 Power Consumption of 100x100 Smart CMOS Camera

5.7.4 Comparison of Power Consumption Estimates

Three separate methods of providing low level vision processing have been considered. In the use of close integration DSP devices the current draw of the TMS320C80 was specified as 1A to give power consumption in the order of 5 watts. This would provide a significant improvement on power consumption of the reported navigation systems [24-29] that employed PC and workstations to implement their low-level vision processing. Caution is needed in citing the DSP based solutions because of the limited connectivity in their processing structures.

It was noted in Section 3.5 that the SLA algorithm has a discrete implementation that can be readily implemented within current PC technology. It also lends itself to partial implementation within programmable logic as indicated in Section 5.7.2. The analysis of the of a CPLD implementation gave power consumption in the order of 400mW when the spatial resolution and framing rate were equivalent to those set for the Smart CMOS Camera. The CPLD implementation represents a significant advance on the DSP implementation. The improvements in power consumption are attributable to advances in programmable logic technology and by ensuring that the processor clock does not run faster than is necessary.

In Table 5.4 the 22.8mW of power consumed by the SLA CMOS implementation for an illumination level of 500lx represents a factor of 17 reduction in power consumption noted for the CPLD implementation. The power consumption of the CMOS implementation is dependent upon the level of environmental illumination. If it was necessary to limit the maximum power consumption to 22.8mW then iris control should be added to the Smart CMOS Camera to limit the level of illumination sensed by the imaging array. The results given in Section 5.6 demonstrate a limited quality of edge point detection for the current CMOS implementation. However, the extremely low power consumption afforded by this device does suggest that further research should be directed to the development if the Smart CMOS Camera.

5.8 Conclusion

The research presented in this Chapter demonstrated the integration of the SLA edge detection algorithm into a NSIP structure. It also demonstrated a random access pixel array suitable for use in conjunction with the NSIP. The array and NSIP structures were combined to form a test device for the Smart CMOS Camera. The test device was fabricated in the Mietec 2.4 μ m CMOS process. Results from the test device showed the detection of edge points and confirmed the successful integration of the SLA edge detection algorithm. The contrast sensitivity of the test device was considerably less than that required for successful implementation of the autonomous navigation problem examined in Section 3.6. Additional research will be necessary before the full potential of the Smart CMOS Camera can be realised.

The random access pixel array developed for the Smart CMOS Camera incorporated a gain BJT at each pixel site. A method of matching the BJT to the output current of the pixel was demonstrated. In the Mietec 2.4 μ m CMOS process current gains in excess of 120 were recorded in the BJT pixel structure for output currents ranging from 20nA to 600nA. A pixel 80x80 μ m layout, including the switching necessary to give two orthogonal array scans, was designed. Measurement established that this pixel layout gave sufficient output for an array of 100x100 pixel to be accessed with a frame rate of 6.5Hz.

The NSIP circuits employed a sub-threshold current mode implementation of the spatial derivatives required by the SLA detector. This current mode circuit was demonstrated to have a near constant contrast sensitivity response for photo currents ranging from 2nA to 2.7 μ A. A minimum contrast sensitivity of 0.06V/% contrast was reported for the 2nA inputs. This contrast sensitivity response allows the NSIP circuit to mimic the operation of biological retinas that respond to intensity contrasts. The near constancy of the response over a wide dynamic range simplified the design of analogue to discrete derivative conversions employed in the NSIP.

Results in Section 5.6 established that the Contrast Sensitive Circuit described and tested in Section 5.4 failed to provide a uniform response across the NSIP linear array processor. This lack of a uniform response was attributed to the device mismatch characteristics. These mismatch characteristics are at their worst when the circuit is operated in sub-threshold mode and when the devices are minimum sized. The circuit of Figure 5.15 was populated by minimum sized devices to maximise the frequency response and it was operated in sub-threshold mode so that the pixel currents could be directly processed in the NSIP structure.

The results of Figure 5.17 show the circuits useful frequency response extending up to 10KHz, thus a relaxation in the minimum size constraint can be tolerated. It has been reported [60-62] that the mismatch variance in CMOS devices is inversely proportional to the square root of the device area. Thus a redesign that increases the area of the current mode circuit devices will allow a reduction in the mismatch characteristics whilst maintaining the required pixel process rate of 1300Hz.

In the tested implementation of the NSIP circuit the spatial contrast circuits all shared the same layout, the mirrored devices all were set to the same orientation and the spacing between mirrored devices was minimised. This layout is accepted practice in the limitation of device mismatch. However as a result of the limited number of design fabrication and test cycle available there was no opportunity to find the orientation of the wafer striations that are a major contributor to device mismatch [19]. Thus a redesign needs to resolve the striation orientation either through foundry supplied data or through a fabrication and test cycle.

The analysis of power consumption given by the DSP implementation of low level vision processing and of the SLA implementations based on CMOS technology and CPLD technology established that the CMOS implementation exhibits the lowest power consumption figures. The 100x100 Smart CMOS Camera can be expected to consume less than one tenth of the power needed to implement the same spatial resolution and frame rate in a CPLD implementation. The CPLD implementation consumed one tenth of the power need to operate a TMS320C80 that implements low

level vision processing. The improvements in power consumption that can be achieved in the CMOS implementation of the SLA edge detection algorithm provide support to the arguments for continuation of research into this Smart Camera. In addition to the power consumption advantages that can be derived from the CMOS implementation, the low mass of the device in comparison to the CPLD and DSP implementations provides for the development of less massive autonomous systems.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

In this thesis, the design of a Smart CMOS Camera for use in autonomous navigation systems has been addressed. The research has focused on the development of a Near Sensor Image Processing (NSIP) structure that implements edge point detection at a camera frame rate of 6.5Hz. An algorithm specifically designed for incorporation in the NSIP structure was developed and tested through simulation. In order for this vision function to be integrated onto the same substrate as an image sensing array a new mixed signal structure referred to as the Scanned Layer Architecture (SLA) was developed.

In the review of autonomous systems in Chapter 2 it was established that an important limiting factor in the development of robotic units is the power consumption of the processors needed to implement low level vision tasks. The SLA NSIP reported in this thesis was designed to overcome this power limitation. The sensor was designed to minimise power consumption through the use of subthreshold CMOS circuits to achieve the task of edge point detection. The results given in Section 5.5 established that edge point detection could be realised through the SLA mixed signal processor. However device parameter variation across an array limited the quality of the edge results. Thus, the current NSIP design is not suitable for the stated aim of this research of providing the edge sets suitable for autonomous navigation processing.

The simulation of the SLA edge detector in Chapter 3 demonstrated that high quality edge point sets can be generated for natural images without the need to employ floating point arithmetic. The effectiveness of the SLA edge detector was further demonstrated through the quantitative analysis carried out in Chapter 4. The SLA simulation was extended to demonstrate how its 3-state edge point representation facilitated the extraction of line segments. The directional nature of the line segments gave rise to two important effects. In the first of these the directional information allowed the noise and texture related edge points to be removed from the image line

list. In the second, the directional information allowed co-linear lines to be grouped so that the uncertainty of geometrical model matches could be minimised.

The matching of lines extracted from images with geometrical models is a critical aspect of the implementation of vision based autonomous navigation. The processing requirements to implement wall following with the SLA edge point detection and line extraction algorithm were also analysed in Chapter 3. In this it was demonstrated that with current PC processor technology, wall following was feasible and a travelling pace of 1m/s could be sustained.

Analysis of the conflicting requirements of side view wall following and forward view target tracking revealed that if mechanical camera position shifts were to be avoided then the autonomous robot should be equipped with separate cameras for these two navigation tasks. The Smart CMOS Cameras that were the objective of this research would be well suited to this type of robot implementation because they have low mass and integrated processing that minimises their power consumption.

The SLA contrast sensitive circuit introduced in Chapter 5 is a significant departure from the neuromorphic temporal contrast processors reviewed in Chapter 2. The NSIP structure allows spatial contrast to be applied to the image data and, at a subsequent stage temporal displacements can be extracted. The SLA detector employed distributed processing structure to assign edge points to the image intensity profile. Sparse convolutions facilitated separate evaluations of the 1st and 2nd order spatial derivatives. Threshold circuits converted these derivatives into a discrete format. Logical operations applied to the discrete derivatives completed the edge point assignment.

The contrast sensitive circuit of Figure 5.15 exploits the inverse relationship between output resistance and output current of a MOSFET current mirror operated in the subthreshold mode. Matching the output resistances of two complementary mirrors that are connected to form a summing node allows a current contrast differential to be registered as a voltage output. No previous report of this current differential circuit

was found in the reviewed texts. The results detailed in Section 5.4 confirmed the low frequency relationship of the current contrast differential to the MOSFET Early Voltage. The results demonstrate that the current contrast differential varies by $\pm 20\%$ as the common mode current is varied over three orders of magnitude. When the differential currents are set by the outputs from current mode pixels a spatial contrast sensitive response is effected.

The spatial contrast processing of the SLA NSIP requires a random access current mode pixel array. In Section 5.3, the realisation of a pixel structure that was capable of supplying the spatial contrast circuit was researched. It was found that the parasitic PNP BJT, formed through a P+ diffusion on a N- well, would provide current gain of the order of 120. A design procedure was developed where the area of the P+ diffusion that formed the BJT emitter was set to give moderate bias conditions for the transistor at the expected pixel signal currents. This procedure enabled the formation of an image sensing array with pixels measuring $80 \times 80 \mu\text{m}$ and a fill factor of 50%.

Chapter 4 presented a new metric for the evaluation of edge detectors that reflects the specifications of a host vision system through the incorporation of scaling factors that set the metric's zero condition. This metric was referred to as the Edge Point Metric. This metric complies with the accepted practice of established metrics by assigning unity as the figure of merit of a perfect detector. However, it deviates from the accepted practice by assigning the figure of merit zero condition to a detector that fails to meet with the host vision systems edge point detection specifications. In established metrics [82,83] this zero condition is not specified and the metric function is limited to comparisons between edge detectors. The new metric allows a given edge detectors performance to be assessed against a systems specification.

The EPM zero figure merit condition is set by specifying the minimum levels of false positive and false negative returns that can be tolerated by the host vision system. A linear combination of the scaled levels of false returns is used to generate the EPM figure of merit. Prior to determining the levels of false positive and false negative

returns the new metric applies an algorithm that is designed to compensate for systematic shifts introduced by the detector under evaluation.

The compensation algorithm recovered line shifts of up to two pixels from the image ground truth locations. In addition, it identified line broadening edge points in the detector results. These displaced lines and line broadening returns were then excluded from the false return sets used to evaluate the figure of merit. In many vision systems these displacements and line broadening effects are insignificant to the systems performance. Thus, it was decided that the EPM method should evaluate these qualitative effects separately from the figure of merit. These qualitative effects are assessed through the probability of line displacement or line broadening in the edge detector results.

Although this research did not result in a functional Smart CMOS Camera there have been two significant advances made towards this goal. These were the design of a robust edge point detection algorithm that relied upon integer based arithmetic and the development of a contrast sensitive current mode differential circuit that mimicked the adaptive response of the bipolar cells found in biological retinas.

The measured results for the Smart Camera given in Chapter 5 were taken from two CMOS custom layout design, fabrication and test cycles. Further refinement of the Smart CMOS Camera structure will require additional design, fabrication and test cycles. It is evident from the low power consumption and low mass of smart processing structures created through the integration of mixed signal processing onto the image sensing substrate that significant advances can be made in respect of autonomous systems development.

6.1 Future Work

The restrictions imposed by the target CMOS environment for the SLA algorithm has given rise to a robust and efficient edge detection algorithm. The analysis of the DSP implementation requirements given in Chapter 3 established that PC processing could be employed to implement a navigation algorithm based on the SLA approach and incorporating geometric model matching as proposed in [15,37,38].

The PC based implementation was not pursued under the current research programme because the additional time and resources needed to follow this course, and the CMOS NSIP device held the prospect of significantly lower power consumption. However, it was recognised that valuable insights into autonomous navigation could be gained from creating a working robot that incorporated CCD or CMOS cameras and PC processing. The use of side view cameras to implement the wall following and separate forward view cameras to locate objectives should be investigated. Results are needed to determine the optimal spatial resolution and framing rate for the robot's camera. These camera functions need to be assessed with respect to positional uncertainty allowed in the robot's navigation processes.

The realisation of a low mass, low power consumption, smart camera that implements edge point detection as an integrated function remains an open problem. The navigation problem analysis carried out in this research demonstrated that this type of smart camera development is critical to the field of autonomous robots. The results from the SLA NSIP implementation, see Section 5.5, demonstrated that the random access pixel array and contrast sensitive circuits worked as predicted by the theoretical evaluations. However device parameter variations limited the practical implementation of a parallel set of 100 edge point detectors in a NSIP structure.

There are two possible courses of action that may be followed to seek a resolution to the problems of device parameter variation that limited the implementation of the SLA NSIP device. Firstly, analysis of the causes of the device parameter variation could be embarked upon under the auspices of a foundry that was interested in developing a CMOS process for mixed signal and analogue processing. Secondly, a

redesign of the current mirror circuit of Figure 5.8 should be embarked upon. In this redesign, the effects of the source to substrate voltage on device threshold voltage need to be considered to resolve the variation in the summing node potential noted in Section 5.6.

In the foundry linked research there is a need to resolve the causes of the striations noted by Pavasovic [19]. These striations present a severe restriction on the implementation of analogue processing that seeks to exploit the high component packing density afforded by CMOS fabrication processes. There is also a need to characterise through measurements the variance of the mismatch characteristics given by fabricated devices. This data can be used to extend the parameter set of the circuit simulation and thus the pre-fabrication analysis would reflect the mismatch characteristics of proposed camera structures. The measured mismatch characteristics can also be used to allow the SLA simulation to generate edge point results that are representative of a fabricated camera. Thus under algorithmic simulation a more robust edge detector could be developed.

The future of NSIP and Neuromorphic processing is closely linked to the key issue of device mismatch in the CMOS medium. Future contributors to this field need to either to improve the CMOS fabrication processing so that device mismatch is significantly reduced. Or to develop more robust and adaptive circuits that can tolerate the imperfections of the CMOS fabrication processes. The biological vision processes that we are attempting to replicate have been under development for more than 600 million years. The field of neuromorphic processing is 22 years old. Research in this field should be seen as evolutionary, it requires the exploration of new circuits and new algorithms as well as the refinement of existing neuromorphic and NSIP circuits. The failures of past circuits or algorithms should not discourage.

References

- [1] Attenborough, "Life on Earth," Chapter 2 Building Bodies, Collins 1979
- [2] Guyton, "Textbook of Medical Physiology," Saunders 1976.
- [3] Gail Erten, Fathi M. Salam, "Real time realisation of early visual perception," Computers & Electrical Engineering, pp 379-407, vol. 25, no. 5 1999
- [4] Duda RO, Hart P E, "Pattern Recognition and Scene Analysis," Wiley 1973
- [5] Barrow H G, Tenenbaum J M, "Recovering Intrinsic Scene Characteristics from Images," Computer Vision Systems, Ed: Hanson and Riesman, Academic press 1978
- [6] T. Uhlin and K. Johansson, "Autonomous mobile systems: A study of current research," Computational Vision and Active Perception Lab, The Royal Institute of Technology, Sweden, Tech. Rep. ISRN KTH/NA/P--96/03--SE, Jan. 1996.
- [7] Jordan D S, Holburn D, "CCD camera frame store and DSP processor Closely integrated digital image capture device for embedded applications," Proc. SPIE Vol. 2950, p. 196-207, Advanced Focal Plane Arrays and Electronic Cameras, Thierry M. Bernard; Ed. Dec 1996
- [8] D S Jordan and D M Holburn; "A closely integrated reconfigurable image capture system and its applications," Proceedings of the Sixth International Conference on Image Processing and its Applications (IPA97), Dublin, July 1997, pp. 156-160.

- [9] Bouvier G, Mhani, A., Sicard, G, "Contrast and motion-sensitive silicon retina," Proc. SPIE Vol. 2950, pp131-136, Advanced Focal Plane Arrays and Electronic Cameras, Dec 1996
- [10] Dallaire S, Tremblay M, Poussart D, "Smart-sensing VLSI architecture for the embedded extraction of dominant points along 2D contours," Proc. SPIE Vol. 2950, p. 294-305, Advanced Focal Plane Arrays and Electronic Cameras, Thierry M. Bernard; Ed., Dec 1996
- [11] J. Canny. "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6):679-698, November 1986
- [12] Sobel I E, "Camera Models and Machine Perception," PhD Thesis, Stanford University 1970.
- [13] Marr D, Hildreth E, "Theory of Edge Detection," Proceedings of the Royal Society of London, Series B, Volume 207, pp 187-217, 1980
- [14] Vittoz E A, "Analog VLSI Signal Processing: Why, Where and How?," Analog Integrated Circuits and Signal Processing, pp 27-44, July 1994
- [15] Beveridge J R, Riseman E, "Hybrid Weak-Perspective and Full-Perspective Matching", In IEEE CVPR. Champaign, IL, 1992, pages 432-438.
- [16] Förstner W, "10 Pros and Cons Against Performance Characterisation of Vision Algorithms," Workshop on Performance Characteristics of Vision Algorithms, Robin College, Cambridge, April 19, 1996, Edited by: Henrik I. Christensen, Aalborg, Wolfgang Förstner, Bonn and Claus B. Madsen, Aalborg

- [17] Mead C, "Neuromorphic electronic systems," IEEE Proceedings, Vol. 78, No. 10, pp. 1629-1636, 1990.
- [18] Mudra, R., R. Hahnloser and R.J. Douglas. "Neuromorphic Active Vision Used in Simple Navigation Behavior for a Robot," Proceedings of the 7th International Conference on Microelectronics for Neural Networks, pp32-36, Granada, Spain, April 7-9, 1999.
- [19] Pavasovic A, A.G. Andreou, and C.R. Westgate, "Characterization of sub-threshold MOS mismatch in transistors for VLSI systems," Journal of VLSI Signal Processing, No 6, pp. 75-84, June 1994.
- [20] A. Åström, J-E Eklund, Roberg Forchheimer, "Near-Sensor Image Processing Theory and Practice," SPIE Vol. 2950, pp242-253, 1996.
- [21] Holland, J.M., Martin, A., Smurlo, R.P., Everett, H.R., "The MDARAS Interior Platform," 22nd Annual Symposium and Exhibition, Association of Unmanned Vehicle Systems, AUVS '95, Washington, DC, 10-12 July, 1995.
- [22] Smurlo, R.P., Laird, R.T., Everett, H.R., Inderieden, R.S., Elaine, S., Jaffee, D.M., "MDARS Product Assessment System," Association of Unmanned Vehicle Systems, 22nd Annual Technical Symposium and Exhibition (AUVS '95), Washington, DC, July, 1995.
- [23] Ralf Möllerto, "Insect Visual Homing Strategies in a Robot with Analog Processing," Biological Cybernetics, special issue: Navigation in Biological and Artificial Systems
- [24] Alberto Broggi, Massimo Bertozzi, and Alessandra Fascioli, "Architectural Issues on Vision-based Automatic Vehicle Guidance: the Experience of the ARGO Project," Real-Time Imaging Journal, 6(4):313-324, August 2000

- [25] Massimo Bertozzi and Alberto Broggi, "Real-Time Lane and Obstacle Detection on the GOLD System," In Ichiro Masaky, editor, Proceedings IEEE Intelligent Vehicles'96, pages 213-218, Tokyo, Japan, September 18-20 1996. IEEE Computer Society,
- [26] Bertozzi M, Broggi A, "GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," IEEE Transactions on Image Processing, Vol 7, No 1, pp 62-81, January 1998,
- [27] Alberto Broggi, Stefano Nichele, Massimo Bertozzi, and Alessandra Fascioli, "Stereo Vision-based Vehicle Detection," In Proceedings IEEE IV-2000, Intelligent Vehicles Symposium, pages 39-44, Detroit, USA, 3-5 October 2000
- [28] S. Singh, B. Digney "Autonomous Cross-Country Navigation Using Stereo Vision," tech. report CMU-RI-TR-99-03, Robotics Institute, Carnegie Mellon University, January, 1999.
- [29] A. Yahja, S. Singh, and A. Stentz, "Recent Results in Path Planning for Mobile Robots Operating in Vast Outdoor Environments," Proc, 1998 Symposium on Image, Speech, Signal Processing and Robotics, September, 1998.
- [30] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. "Experiences with an interactive museum tour-guide robot", Technical Report CMU-CS-98-139, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, 1998.
- [31] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, J. Schulte, and D. Schulz. MINERVA: "A second-generation museum tour-guide robot". In Proceedings of the International Conference on Robotics and Automation (ICRA '99), 1999.

- [32] Joachim Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank E. Schneider, Jiannis Strikos, and Sebastian Thrun. "The mobile robot rhino," *AI Magazine*, 16(2):31--38, Summer 1995.
- [33] Thorsten Fröhlingshaus and Joachim Buhmann: "Real-Time Phase-Based Stereo for a Mobile Robot," *Proceedings of the First Euromicro Workshop on Advanced Mobile robots*. pp. 178-185, 1996.
- [34] Blaasvaer H, Pirjanian P, Christensen H I, "AMOR: An Autonomous Mobile Robot Navigation System," *IEEE, Int. Conference on Systems, Man, and Cybernetics 1994, Vol. 3*, pp. 2266-2271.
- [35] Pirjanian P, Christensen H I, "Hierarchical Control for Navigation Using Heterogeneous Models," *Modeling and Planning for Sensor Based Intelligent Robot Systems, Machine Perception and Artificial Intelligence series*, World Scientific 1995, Editors: H. Bunke, T. Kanade and H. Noltemeier, pp 344-361.
- [36] Davison A J, Murray D W, "Mobile Robot Localisation Using Active Vision," *Proceedings of the 5th European Conference on Computer Vision, Freiburg*, pp 809--825, 1998
- [37] Kosaka A, Pan J, "Purdue experiments in model-based vision for hallway navigation," *Proceedings of Workshop on Vision for Robots in IROS'95*, pp.87-96, 1995.
- [38] Kosaka A, Meng M, Kak AC, "Vision-guided mobile robot navigation using retroactive updating of position uncertainty," *Proceedings of 1993 IEEE International Conference on Robotics and Automation, Vol.2*, pp.1-7, 1993.

- [39] Y.Gavriley, A.Samarin, I.Bessarabov, A.Faure, E.Vasselin, J.L.Desbordes. "Active Vision for Adaptive Behavior of Autonomous Robot," Proc. of the Workshop on Robot Vision", Rostov-on-Don, May 1996
- [40] Vernon D, "Automated Visual Inspection and Robot Vision," Prentice Hall 1991
- [41] McCluney R, "Introduction to Radiometry and Photometry," Artech House 1994.
- [42] Datacube – "PCI Product Reference," <http://www.datacube.com/customer/maxpciplatforms.htm> (January 2000)
- [43] Texas Instruments, "Implementation of Image Processing Library for the TMS320C8x (MVP)," Texas Instruments Application Note BPRA059, 1997
- [44] R Dominguez-Castro, S Espejo, A Rodriguez-Vazquez, R A Carmona, P Foldsey, A Zarandy, P Szolgay, T Sziranyi, T Roska, "A 0.8-um CMOS Two-Dimensional Programmable Mixed-Signal Focal-Plane Array Processor with On-chip Binary Imaging and Instructions Storage," IEEE Journal of Solid State Circuits, Vol 32, pp 1013-1026, July 1997
- [45] S Espejo, A Rodriguez-Vazquez, R Dominguez-Castro JL Huertas E Sanchez-Sinencio. "Smart-Pixel cellular networks in analog current-mode CMOS technology," IEEE Journal of Solid State Circuits, Vol 29, pp 895-905, Aug. 1994
- [46] K.A. Boahen. "Retinomorphic Vision Systems i: Pixel Design," IEEE Int. Symp. On Circuits and Systems July 96
- [47] C. Mead & M.A. Mahowald, "A silicon model of early visual processing," Neural Networks, Vol. 1, pp. 91-97, 1988. 139, 153

- [48] M.A. Mahowald. "Silicon Retina with Adaptive Photo Detectors," SPIE Proc.. Visual Information Processing, Vol 1473, 1991
- [49] T. Delbruck and C.A. Mead, "Adaptive Photoreceptor Circuit with Wide Dynamic Range," Proc. of the International Circuits and Systems Meeting, pp 339-343, London, May 1994.
- [50] T. Delbruck and C.A. Mead, "Analog VLSI Phototransduction by Continuous-Time, Adaptive, Logarithmic Photoreceptor Circuits," CNS Memo 30, Computation and Neural Systems Department, California Institute of Technology, Pasadena CA 1994
- [51] T. Delbruck and C.A. Mead. "Time-derivative adaptive silicon photoreceptor array," SPIE Vol. 1541, Infrared Sensors: Detectors, Electronics, and Signal Processing. (pp. 92-99).
- [52] A.G. Andreou & K.A. Boahen, "A 48000 pixel, 590000 transistor silicon retina in current mode subthreshold CMOS," Proc.37th Midwest Symposium on Cir. And Sys., pp 97-102 1994
- [53] K.A. Boahen & A.G. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," Advances in Neural Information Processing 4, Vol. 4, pp. 762-772, 1992.
- [54] Joachim Buhmann, Martin Lades and Frank Eeckmann, "Illumination-Invariant Face Recognition with a contrast Sensitive Silicon Retina," Advances in Neural Information Processing Systems (NIPS) 6, Morgan Kaufmann Publishers, pp 769-776, 1994.
- [55] T Roska, L O Chua, "The CNN Universal Machine: An Analogic array computer," IEEE Trans. Circuits Systems II, vol 40,no.3, Mar 1993

- [56] A Bouzerdoum, A Moini, A Yakovleff, XT Nguyen, RE Bobner, K Eshraghian. "A Smart Visual Micro-Sensor," Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, pp 276-279, Oct. 1994.
- [57] A Moini, A Bouzerdoum, K Eshraghian, A Yakovleff, XT Nguyen, A Blanksby, R Beare, D Abbott, RE Bogner. "An Insect Vision-Based Motion Chip," IEEE Journal of Solid State Circuits, Vol32, pp 279-283, Feb. 1997.
- [58] A Yakovleff, D Abbott, XT Nguyen, K Esharghian, "Obstacle Avoidance and Motion-Induced Navigation," Proc. of Computer Architectures for Machine Perception Workshop (CAMPO95), pp384-3393, 18-20 September 1995
- [59] Indiverei Neuromorphic "Analog VLSI Sensor for Visual Tracking: Circuits and Application Examples," IEEE Transactions On Circuits and Systems-II: Analog and Digital Signal Processing, Vol 46, No11, November 99.
- [60] J. Bastos, M. Steyaert, B. Graindourze & W. Sansen, "Matching of MOS transistors with different layout styles," in Proc. IEEE Int. Conference on Microelectronic Test Structures, pp. 17-18, March 1996.
- [61] M. Steyaert, J. Bastos, R. Roovers, P. Kinget, W. Sansen, B. Graindourze, A. Pergot & Er. Janssens, "Threshold voltage mismatch in short-channel MOS transistors," Electronic Letters, Vol. 18, pp. 1546-1548, September 1994.
- [62] F. Forti & M.E. Wright, "Measurement of MOS current mismatch in the weak inversion region," IEEE Journal of Solid State Circuits, Vol. 29, No. 2, 138-142, 1994.
- [63] Y P. Tsividis, "Operation and Modeling of the MOS Transistor," McGraw-Hill 1987
- [64] ES Yang. "Microelectronic Devices," McGraw-hill 1988

- [65] J-E Eklund , C. Svensson , A. Åström, "VLSI Implementation of a Focal Plane Image Processor - A Realization of the Near-Sensor Image Processing Concept," IEEE Trans on VLSI Systems, vol 4, No 3, September 1996
- [66] Anders Åström , Robert Forchheimer , Jan-Erik Eklund, "Global Feature Extraction Operations for Near-Sensor Image Processing," IEEE Transactions on Image Processing, vol 5, No 1, January 1996
- [67] Anders Åström , Robert Forchheimer , Per-Erik Danielsson, "Intensity Mappings within the Context of Near-Sensor Image Processing," IEEE Transactions on Image Processing, 1998
- [68] IVP Integrated Vision Products AB, "MAPP2200 Product information," Ver2.1, 1998,
- [69] DA Martin, H-S Lee, I Masaki, "A Mixed-Signal array Processor with Early Vision Applications," IEEE Journal of Solid State Circuits, vol33, pp497-502,Mar 1998.
- [70] Ralph Etienne-Cummings,Viktor Gruev, Donghui."A High Density Focal-Plane Image Processing Array," (invited) Proc. Conf. Information Sciences and Systems, Baltimore, MD, 1999.
- [71] Ralph Etienne-Cummings, Viktor Gruev,Mohammed Adbel Ghani. "VLSI Implementation of Motion Centroid Localization for Autonomous Vehicles," Advances in Neural Information Processing Systems, Vol. 11, pp. 685-691, 1999.
- [72] Weeks A R, "Fundamentals of Electronic Image Processing," SPIE Optical Engineering Press, 1996.

- [73] M. Rowley and J.G. Harris. "An edge enhancement technique for analog VLSI vision chips," In IEEE International Conference on Neural Networks, pages 1000-1004, Washington, D.C., June 1996.
- [74] I. Farkas, R Miikkulainen,"Modeling the self-organization of directional selectivity in the primary visual cortex," Proceedings of IEE ICANN 99, pp 251-256, September 1999.
- [75] Zhaoping Li, "Neural dynamics in a recurrent network model of primary visual cortex," Proceedings of IEE ICANN 99, pp 280-285, September 1999.
- [76] Hajjar A, Chen T, "A VLSI Architecture for Real-Time Edge Linking," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 89-94, Vol. 21, No 1, January 1999.
- [77] Bergholm F, "Edge Focusing," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp726-741, Vol 9, No 6, November 1987
- [78] Nalwa V S, Binford T O, "On Detecting Edges," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 699-714, Vol 8, No 6, November 1986.
- [79] Elder J H, Zucker S W, "Local Scale Control for Edge Detection and Blur Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp699-716, Vol 20, No7 July 1998
- [80] S.M. Smith. "SUSAN - a new approach to low level image processing," Int. Journal of Computer Vision, 23(1), pp 45-78, May 1997.
- [81] Henrik I Christensen and Wolfgang Foerstener, "Performance Charecterisation of Computer Vision Algorithms. Machine Vision and Applications," Springer Verlag, Vol. 11, Nos. 5/6. March 1997. pp. 215--218.

- [82] W K Pratt, "Digital Image Processing," Wiley 1978
- [83] L Kitchen, A Rosenfeld, "Edge Evaluation Using Local Edge Coherence, IEEE Transactions on Systems, Man, and Cybernetics, Vol. Smc-11, No 9, pp 597-605, September 1981
- [84] Khvorostov PV, Braun M and Poon CS. "Edge quality metric for arbitrary 2D edges," Optical Engineering 35(11), 3222-6, 1996.
- [85] R. N. Strickland and D. Chang, "An Adaptable Edge Quality Metric," Proceedings of SPIE, vol. 1360, pp. 982-995, Bellingham, WA, USA, 1990.
- [86] P L Rosin, "Edges: Saliency Measures and Automatic Thresholding," Machine Vision and Applications, vol. 9, pp. 139-159, 1997.
- [87] Salotti M, Bellet F, Garbay C, "Evaluation of Edge Detectors: Critics and Proposal," Workshop on Performance Characteristics of Vision Algorithms, Cambridge, April 1996
- [88] T.B. Nguyen and D. Ziou, "Contextual and Non-Contextual Performance Evaluation of Edge Detectors," Vision Interface 99, Trois-Rivières, Canada, May 18-21, 1999.
- [89] M D Heath, S Sarker, T Sanocki, K W Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No 12, pp 1338-1359, December 1997
- [90] Dougherty S, Bowyer KW, "Objective Evaluation of Edge Detection Algorithms," Empirical Evaluation Techniques in Computer Vision, Ed: Bowyer, Philips, pp 172-187, IEEE Computer Society Press, 1988

- [91] Singh J, "Semiconductor Optoelectronics Physics and Technology," McGraw-Hill, 1995
- [92] Moini A, "Vision Chips," Kluwer Academic Publishers, Oct. 1999
- [93] Vandendriessche L, "Electrical Parameters 2.4 μ m CMOS," Mietec Alcatel, doc 13212, rev. 4, May 93.
- [94] Gray P R, Meyer R G, "Analysis and Design of Analog Integrated Circuits," Chapter on Models for Integrated-Circuit Active Devices", John Wiley & Sons, Inc. 1993.
- [95] Neamen D A, "Semiconductor Physics & Devices, Chapter on The Bipolar Transistor", Irwin Times Mirror Higher Education Group, Inc. 1997
- [96] Fluke 80 Digital Multimeter Specifications, Fluke 80 Users Manual 1995
- [97] Laker K R, Sansen W M C, "Design of Analog Integrated Circuits and Systems," McGraw-Hill, 1994
- [98] S. Shah & M.D. Levine. "A Computer Retina that Models the Primate Retina. SPIE Proceedings," Visual Information Processing III, vol. 2239 pg116- 1994
- [99] Mulder J, Van Der Woerd A C, Serdijn W A, Roermund A H M, "Application of the Back Gate in MOS weak Inversion Translinear Circuits", IEEE Transactions on Circuits and Systems, Vol42 No 11, pp 958-962, November 1995
- [100] Xilinx Inc, "XC95288XV High Performance CPLD" Advance Product Specification, DS050 (v2.0), January 29, 2001

- [101] Geiger R L, Allen P E, Strader N R, "VLSI Design Techniques for Analog and Digital Circuits", McGraw-Hill 1990.

Symbolic Terms

SLA Simulation

- $I(x,y)$ A image intensity profile sampled with row variable x and column variable y
- $d_{direction}^{order}(x,y)$ Analogue directional derivative operator applied to sample space with row variable x and column variable y . Superscript for the order of the derivative (1=1st and 2=2nd). Subscript for direction of the derivative (h =horizontal and v = vertical)
- $t_{direction}^{order}(x,y)$ Directional threshold operator applied to sample space with row variable x and column variable y . Superscript for the order of the derivative (1=1st and 2=2nd). Subscript for direction of the derivative (h =horizontal and v = vertical)
- $D_{direction}^{order}(x,y)$ Discrete directional derivative operator evaluated from the $d_{direction}^{order}(x,y)$ and $t_{direction}^{order}(x,y)$ operator outputs
- $EP_{direction}(x,y)$ Edge point assignment operator. Accepts inputs from a neighbourhood of $D_{direction}^{order}(x,y)$ operators to assign directional edge points. Subscript for direction of the derivative (h =horizontal and v = vertical)
- $Per_{1^{st}}$ The global percentage threshold applied to the 1st order derivative
- $Per_{2^{nd}}$ The global percentage threshold applied to the 2nd order derivative

Edge Point Metric

TP	True Positive Edge point assigned
TN	True Negative Edge point assigned
FP	False Positive Edge point assigned
DP	Displaced Positive Edge point assigned
DN	Displaced Negative Edge point assigned
WP	Wide Positive Edge point assigned
P(FN)	Condition Probability of a False Positive occurring
P(FP)	Condition Probability of a False Negative
P(DP)	Condition Probability of a Displaced Positive
P(WP)	Condition Probability of a Wide Positive

CMOS Implementation

K	Boltzmann's Constant
T	Temperture in Kelvin
q	electron charge
I_{ph}	Photo generated current given by a silicon detector
P_o	Incident light intensity and
R	Silicon reflection coefficient
E_p	Energy of incident photons
F	Fraction of minority carrier collected by photo detector
η_i	Internal Quantum Efficiency of photo detector
A	Area of the Vertical PNP emitter

W_B ,	Bipolar Transistor Base Width
W_E	Bipolar Transistor Emitter Width
W_{dE}	is the width of this depletion region
I_{pE}	the emitter current
I_{rg}	recombination current of emitter base depletion region
n_i	Intrinsic doping concentration
N_A	Acceptor doping concentration
N_D	Donor doping concentration
D_p	diffusion coefficient of holes
D_p	diffusion coefficient of electrons
τ_0	carrier lifetime
L_p	diffusion length of holes
R_T	transresistance of contrast sensitive circuit
R_O	MOSFET output resistance
V_{AW}	MOSFET equivalent of the BJT Early voltage
K_{AW}	MOSFET extracted device parameter for Early Voltage

Glossary

Adaptive Thresholding:

A threshold operation performed changing the threshold value based on local brightness characteristics of an image.

Bipolar Transistor

An active semiconductor device formed by two P-N junctions whose function is amplification of an electric current.

CCD Charge Coupled Device. A photo-sensitive image sensor implemented with large scale integration technology.

CMOS

Complementary Metal-Oxide Semiconductor. A MOS technology in which both P-channel and N-channel components are fabricated on the same die to provide integrated circuits that use less power than those made with other MOS (metal oxide semiconductor) or bipolar processes.

Contrast

The difference of light intensity between two adjacent regions in the image of an object.

Contrast Sensitivity

The contrast required to obtain a criterion response from a cell or a human subject as a function of spatial frequency. Falls off in sensitivity as the spatial frequency of the test pattern increases.

Cone Retina photoreceptor for day vision.

Convolution

Superimposing a $m \times n$ operator (usually a 3×3 or 5×5 mask) over an area of the image, multiplying the points together, summing the results to replace the original pixel with the new value. This operation is often performed on the entire image to enhance edges, features, remove noise and other filtering operations.

Convolution Kernel

The set of coefficient values that are used as weights for calculating the weighted average of the source neighbourhood for performing a convolution.

DSP Digital Signal Processing A processor used for high speed data manipulations of audio, video, graphical, or image information.

Depth of field

The range of an imaging system in which objects are in focus.

- DGC Detector to Ground-truth Comparison** This algorithm was developed to in this research to locate systematic displacements between the edge detectors
- Edge** A change in pixel values exceeding some threshold amount. Edges represent borders between regions on an object or in a scene.
- Edge Detector**
A process used to determine the true edge of an object.
- EPM Edge Point Metric.** A new metric developed in the course of this research. Designed to incorporate a minimum quality specification into the metric's figure of merit.
- Edge Sense**
An Edge Point generated by a detector is set to retain the sense of the 1st order derivative that gave rise to the edge by assigning the edge a 3-state value, no edge, positive edge or negative edge
- Frame**
A single picture, usually taken from a collection of images such as in a movie or video stream.
- Frame Grabber**
Computer card that samples and digitises analogue video signals so that the information may be processed, stored, or operated on by the computer. It is also called image acquisition or image capture board.
- Frame Rate**
The rate at which image frames are processed by a digital image processing system.
- Grey level**
A quantified measurement of image irradiance (brightness), or other pixel property typically in the range between pure white and black.
- Greyscale Image**
An image consisting of an array of pixels which can have more than two values. Typically, up to 256 levels (8 bits) are used for each pixel.
- Horizontal Cells**
Cells in the retina connected via gap junctions that mediate lateral information transfer over large distances.
- Image Scan**
A scanning pattern, generally from left to right while progressing from top to bottom of the imaging sensor.

Intensity

The relative brightness of a portion of the image or illumination source.

Low-Level Vision

A label applied to the vision processes need to convert the sampled intensity profile given by a camera into image primitives such as edge points or texture attributes at the pixel level.

Machine Vision

The use of devices for optical non-contact sensing to automatically receive and interpret an image of a real scene, in order to obtain information and/or control machines or processes.

Median Filter

A method of image smoothing which replaces each pixel value with the median greyscale value of its immediate neighbours.

Minimum Quality Specification

A specified limit for the performance of an edge detector used within a modular vision system.

MIPS Millions of Instructions per Second measure for computer processing speed.

NSIP Near Sensor Image Processing A image processing circuit that has been integrated on a sensor substrate. Designed to minimise power consumption and limit the mass of the vision system

Outer Plexiform Layers.

The layered structure of a retina that perform the low level vision functions of light sensing and spatial contrast enhancement

Parallel Processor

A hardware design using a number of processors so multiple pixels may be processed at the same time.

Passivation

The final, protective layer(s) of silicon nitride or silicon dioxide applied to a wafer.

Photodiode

A single photoelectric sensor element, either used stand-alone or a pixel site, part of a larger sensor array.

PCI bus

PCI local bus is a standard used in computers for high speed component-to-component connection.

Pipeline Processor

An image processor that passes streams of image data through a series of high speed specialized processing elements to process images.

Pixel Picture Element. The smallest distinguishable and resolvable area in an image. The discrete location of an individual photo-sensor in a solid state camera.

Pose Recovery

Refers to the process of finding the position and attitude of a robotic system within a known environment

Random Access

The ability to read out chosen lines or windows of information from an imager as needed.

Real Time Processing

In autonomous navigation, the ability of a system to perform a complete analysis and take action without halting the systems movement.

Resolution, Spatial

A direct function of pixel spacing. Pixel size relative to the sensors field of view.

Segmentation

Dividing an image into discrete objects and background.

SLA Scanned Layer Architecture

A term applied to the substrate layout structure required to implement parallel processing of the image data read out from an image sensing array

Sonar

Low frequency radiated acoustical waves just above human sound perception which are useful for the "illumination" of solid objects.

Spatial Filter

A filter that operates in the spatial domain as opposed to the frequency domain to accentuate or attenuates the appearance of the spatial details, for example the transitions of intensity in an image.

Spatial Resolution

The number of pixels in the horizontal and vertical dimensions used to represent a digital image.

Texture

The degree of smoothness of an object surface. Texture affects light reflection, and is made more visible by shadows formed by its vertical structures

Thresholding

The assigning of a binary value to a pixel based on whether its intensity falls below, or above a threshold value.

VLSI Very Large Scale Integration. Semiconductor fabrication technology that can create a density of between 1000 and 1,000,000 devices on each individual die.

Appendix A

Example SLA Edge Detection Results

The SLA edge detector that was optimised in Section 4.6 for the detection of narrow features as well as spread edges within the same image is tested on five images. These include indoor images that are representative of the scenes likely to be encountered by an autonomous navigation robot. The Lena and Clare images are accepted standard images for the field of vision processing. The standard images are processed with a SLA edge detector using the optimal parameter settings derived for the indoor image analysis. The standard image results allow for a general performance assessment to be made of the SLA edge detector. For each of the processed images the edge results are given in binary format. This binary format combines the 3-state directional edge sets generated by the SLA detector into a single image

The indoor images are all sampled at 768x576. They have varied illumination. In Image (a) the illumination is given by artificial ceiling lighting. In Image (b) the illumination is given by sunlight coming from the left and through the glass door. In Image (c) there are two sources of illumination, from the front of sunlight illuminates the room beyond the half open door and from above the artificial lighting illuminates the room that the image was taken from. The critical issues for autonomous navigation operation are the detection of the floor to wall boundaries, the identification of free floor space and the extraction of outlines associated with doors walls and furniture. It is evident from the results given that in all three images the SLA algorithm detects these critical features.

The Lena and Clare image results confirm the facility of the SLA detector to recover both fine detail and spread edges without recourse to the use of multiple scales in the detection process. Examine the fine detail that is detected at the corners of Lena's mouth and the upright strut behind Lena that is out of focus. The adaptive thresholds that are used in the SLA algorithm allow the folds in Clare's jacket to be resolved. The apparent limited resolution of the Clare image is attributable to its 256x256 size, in contrast the Lena image has a 512x512 sample space.



Image (a)
Artificial Ceiling Illumination

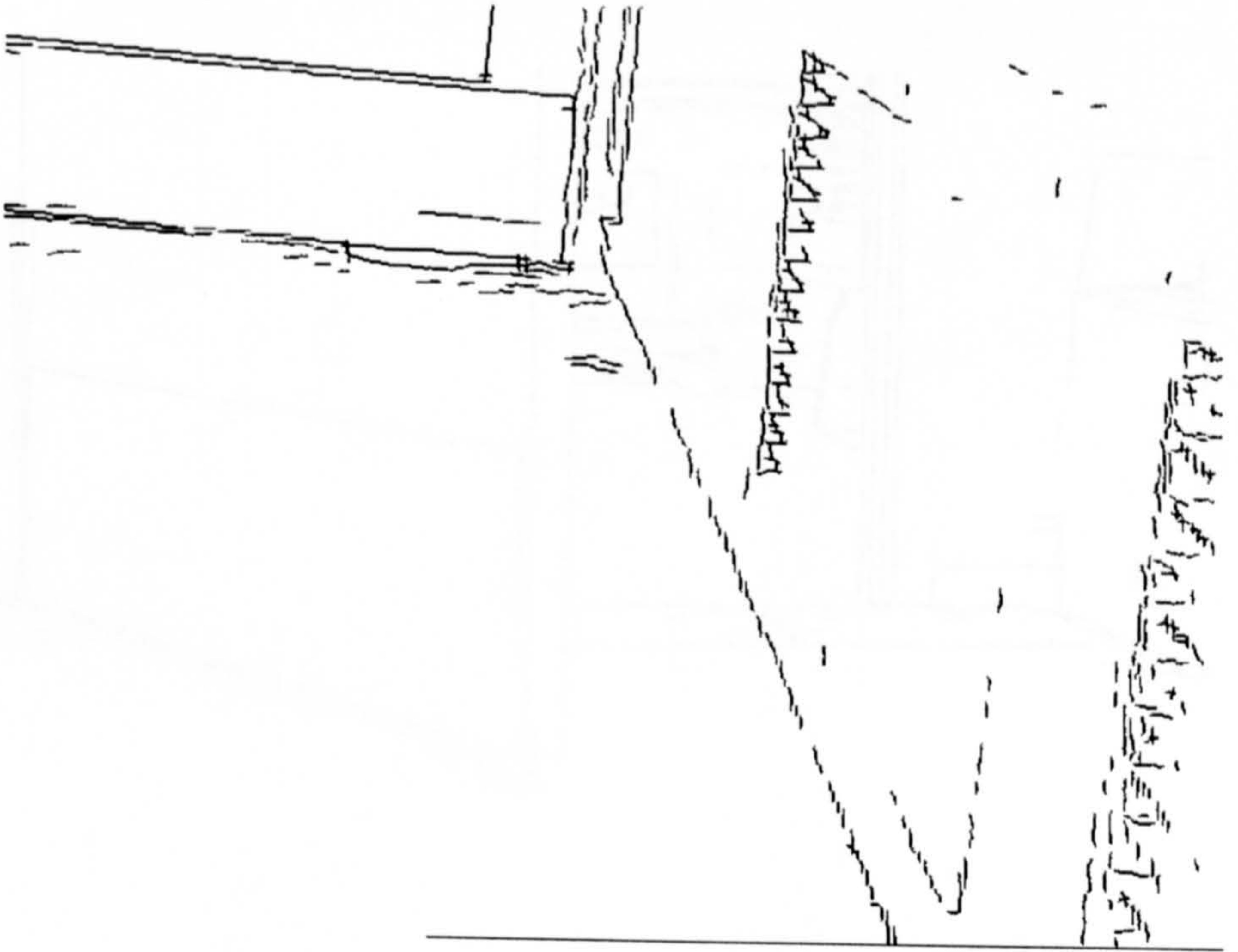


Image (b)
Sun-light Illumination

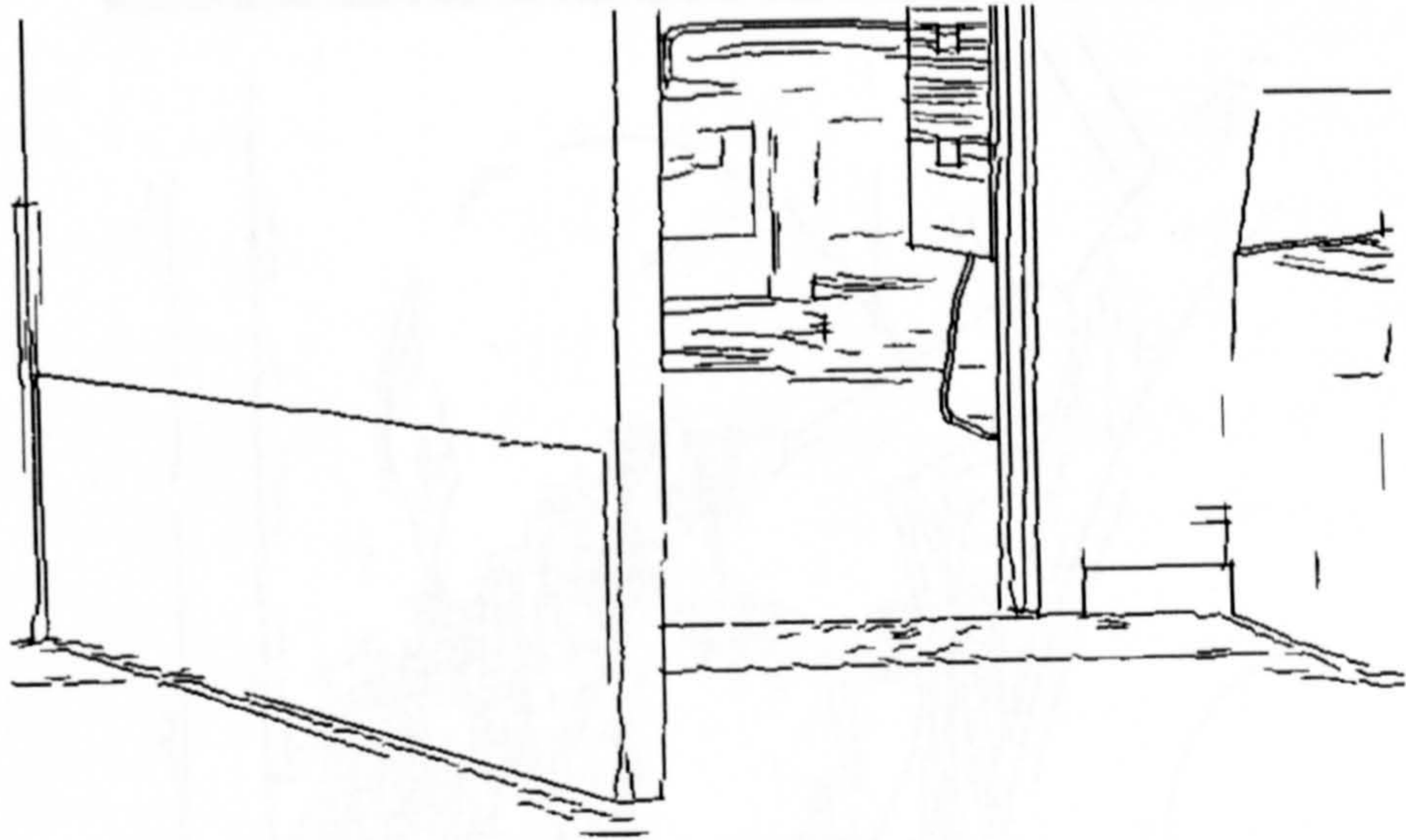


Image (c)

Sun-light and Artificial Illumination



Lena Image



Clare Image

Publications

- [1] Moorhead T.W.J., Binnie T.D., "Smart CMOS Camera for Machine Vision Applications," IPA 99 Conf. pp 865-869, July 99.
- [2] Moorhead T.W.J., Binnie T.D., "An Edge Point Metric for the Contextual Assessment of Detectors," Submitted to IEE Journal Vision, Image and Signal Processing April 2001.

SMART CMOS CAMERA FOR MACHINE VISION APPLICATIONS

Moorhead, T. W. J. & Binnie, T.D.

School of Engineering, Napier University, Scotland

Abstract. An edge detection algorithm designed for a custom CMOS hardware implementation is presented. The integration of this edge detector into an image-sensing chip has been evaluated through SPICE analysis and measurements on fabricated devices. The performance of this edge detector has been quantified through image processing simulation.

1 INTRODUCTION

Autonomous vision systems are limited in their application because of the need to employ high-speed processors to implement early vision tasks. The robotic systems that were reported by Murray et al (4) and Kosaka et al (2) demonstrate the viability of active vision but highlight operational limitations due to processor power requirements. In order to ameliorate these limitations, a low power smart CMOS camera that integrates the early vision tasks into an image sensing chip is proposed. This smart camera employs a novel processing architecture called Scanned Layer Architecture (SLA). In this architecture the early vision tasks are implemented through a distributed hybrid processor.

Edge points are one of the most common image primitives used in machine vision segmentation. These points are detected through analysis of the image spatial intensity gradients. The SLA vision chip detects edge points within an image through layered processing circuits. These layered circuits are placed adjacent to the image-sensing array. The image data is separately scanned through row and column layered processors. This architecture gives the SLA vision chip spatial resolution equivalent to that found in commercial CMOS cameras and provides efficient parallel processing of the sensed image.

The SLA edge detection algorithm (section 2) is a development of the Marr and Hildreth (3) edge detection scheme. The SLA edge detector employs directional derivatives and a distributed decision process. The SLA circuit (section 3) is designed for custom CMOS fabrication. The function of this circuit has been assessed through image processing simulation. Results of this simulation are presented in section 4. Comparisons are made between the SLA edge detection algorithm and the Canny (1) edge detection method.

2 SLA EDGE POINT DETECTION ALGORITHM

It was shown in (3) that edges points are located at maxima and minima in the 1st order spatial derivative of the image intensity profile. These points were also marked by zero crossings in a 2nd order spatial derivative of the intensity profile. A laplacian convolution was demonstrated to detect edge points. The noise susceptibility of the (3) detector was reduced by applying a gaussian filter to the intensity profile. This method returns a maximum gradient for all edges in the processed image. This is a computationally intensive process and is not practical for real time machine vision systems.

2.1 Directional Derivatives

The circuit based SLA edge detection algorithm was developed from analysis of the critical data paths in the (3) method. The SLA algorithm employs directional derivative operators to give an efficient circuit implementation of a 2nd order spatial derivative edge detector. In this algorithm, two pairs of directional derivative operators are required. These are 1st order directional operators (D_x^1 and D_y^1) and 2nd order directional operators (D_x^2 and D_y^2). The D_x operators are defined in equations 1 and 2.

$$D_x^1 = d/dx \quad (1)$$

$$D_x^2 = d^2/dx^2 \quad (2)$$

The circuit implementation of the SLA algorithm allows a row pixels to be processed simultaneously. In row 'k' of the intensity profile $I(x,y)$, the convolutions required for each pixel are given by equations 3 and 4. The variable 'n' signifies the pixel location within the processed row. The scan proceeds to the next row by incrementing the 'y' value in equations 3 and 4. The full algorithm is implemented in two separate scans of the processed image. In the first scan, the D_x convolution generates an edge map for the image, and in the second scan edges generated by the D_y convolution are added to this map.

$$D_x^1 \&I(x,y)|_{y=k} = I(x_{(n)}) - I(x_{(n-1)}) \quad (3)$$

$$D_x^2 \&I(x,y)|_{y=k} = 2\%I(x_{(n)}) - I(x_{(n-1)}) - I(x_{(n+1)}) \quad (4)$$

The presence of noise in captured images limits the performance of edge detection schemes. In the SLA algorithm the noise susceptibility is reduced through

extensions to the D operators. These extensions are realized by increasing the widths of the D operators. The convolutions for the D_x operators with widths of two are given in equations 5 and 6.

$$S2D_x^1 \& I(x,y)|_{y=k} = I(x_{(n+1)}) + I(x_{(n)}) - I(x_{(n-1)}) - I(x_{(n-2)}) \quad (5)$$

$$S2D_x^2 \& I(x,y)|_{y=k} = 2\%I(x_{(n)}) - I(x_{(n-2)}) - I(x_{(n+2)}) \quad (6)$$

The SLA algorithm has a minimum probability of detecting an edge, which is lying at 45° to the directional derivatives. A rotation of 45° will cause the spread of an edge to change by 41%. The effect of a 45° rotation on the probability of detecting an edge is measured in section 4.1.

2.2 Distributed Decision Process

The SLA algorithm employs a distributed decision process. In this process the D^1 magnitudes are compared to set thresholds to give a ($posD$, $negD$, $nullD$) three state output. The $nullD$ state signifies that there is no edge at this location otherwise the $posD$ and $negD$ states signify a possible edge and it's direction. The location of an edge within a group of $posD$ or $negD$ returns is marked by a zero crossing between adjacent D^2 outputs. The algorithm has two possible valid edge outputs. These are set by the edge direction, and are either a RisingEdge (RE) or FallingEdge (FE). The conditions for the RE and FE valid edges are given in equations 7 and 8.

$$RE = posD^1_{(n)} \& posD^2_{(n)} \& negD^2_{(n-1)} \quad (7)$$

$$FE = negD^1_{(n)} \& negD^2_{(n)} \& posD^2_{(n-1)} \quad (8)$$

If the D^2 zero crossing detector is presented with a ramp style edge nulls occur within the $D^2_{(n)}$ and $D^2_{(n-1)}$ markers. In order to ensure that this type of edge is detected, the algorithm employs extensions to the edge detection logic. The RE condition is OR'ed with the NulledRisingEdge (NRE) condition, where a null condition at $D^2_{(n)}$ is tested. A similar extension is used for FE .

$$NFE = posD^1_{(n)} \& posD^2_{(n+1)} \& nullD^2_{(n)} \& negD^2_{(n-1)} \quad (10)$$

The SLA algorithm provides the facility to apply thinning and smoothing to the streams of valid edges generated by the orthogonal scans. Boolean operators have been designed to remove isolated edges and straighten short line sections. The edge direction information supplied by the valid edge detectors is critical to the implementation of these functions.

Section 2.3 Edge Detector Performance Value

A quantitative measure for the performance of edge detectors has been developed. This is defined as the

detector Performance Value (PV). The PV is determined by numbers of false returns from the application an edge detection algorithm. The false returns are either False Positives (FP), or False Negatives (FN). If the edge detector is not affected by the applied noise there are no false returns, and PV is unity. When the level of false returns renders the segmentation results unusable for machine vision applications, the PV value is zero. The value of PV falls to zero when the probability of a false positive $P(FP)$ is 0.09 (one in eleven pixels is a FP), or when $P(FN)$ is 0.25 (one in four edges are missed).

$$PV = 1 - (4\%P(FN) + 11\%P(FN)) \quad (11)$$

3 SLA CIRCUIT IMPLEMENTATION

The SLA overview (Figure 1) illustrates the four layers of the row processor needed to implement edge point detection. The image data is processed a row at a time through this layered structure. In this, each accessed row generates a line of edge points, which are stored in the edge map. A second scan of the image through an identical column processor is used to complete the image edge detection.

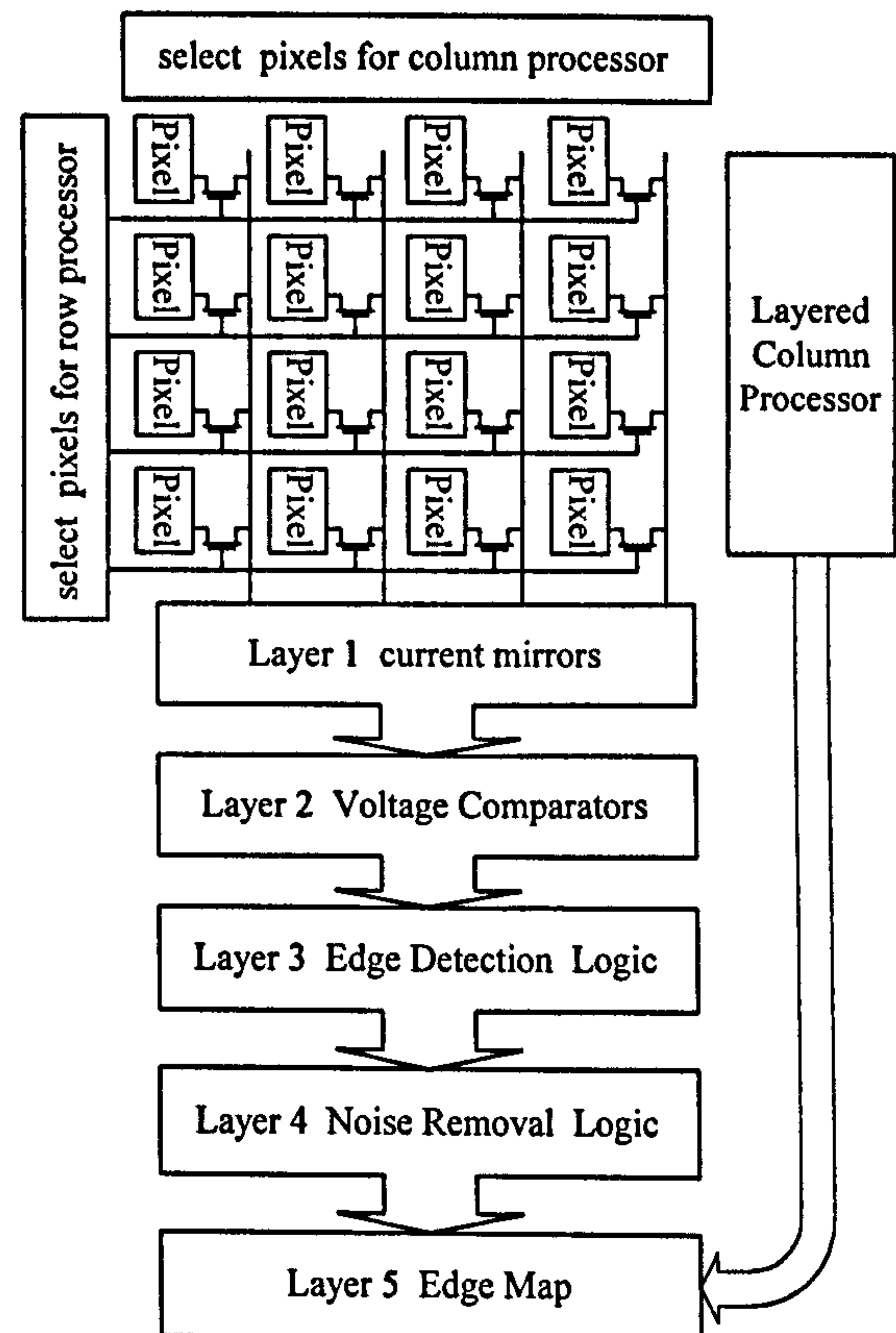


Figure 1. SLA Image Processor

In layer 1, the pixel photo currents are mirrored and summed to give the D_x^1 and D_x^2 analogue signals. A pair of D_x signals is generated for each pixel in the accessed row. In layer 2, voltage comparators convert the analogue D_x signals from each pixel to discrete three state outputs. In layer 3, logical circuits detect the

presence of valid edges within the D_x three state outputs. In layer 4, the streams of valid edge points are smoothed and isolated edge points removed. These processed edge points are then loaded into the edge map.

3.1 SLA Pixel

The analogue circuits of the SLA vision chip permit the wide dynamic range of silicon photo-detectors to be exploited. To facilitate this process the pixel array is set to operate in current mode. The pixel structure (Figure 2), is designed to provide a continuous current output that is directly related to the intensity of received light. In this structure, light is sensed by the N-well to substrate diode. A vertical PNP transistor formed by a central P+ diffusion provides current gain. Figure 2 includes an equivalent circuit for this structure.

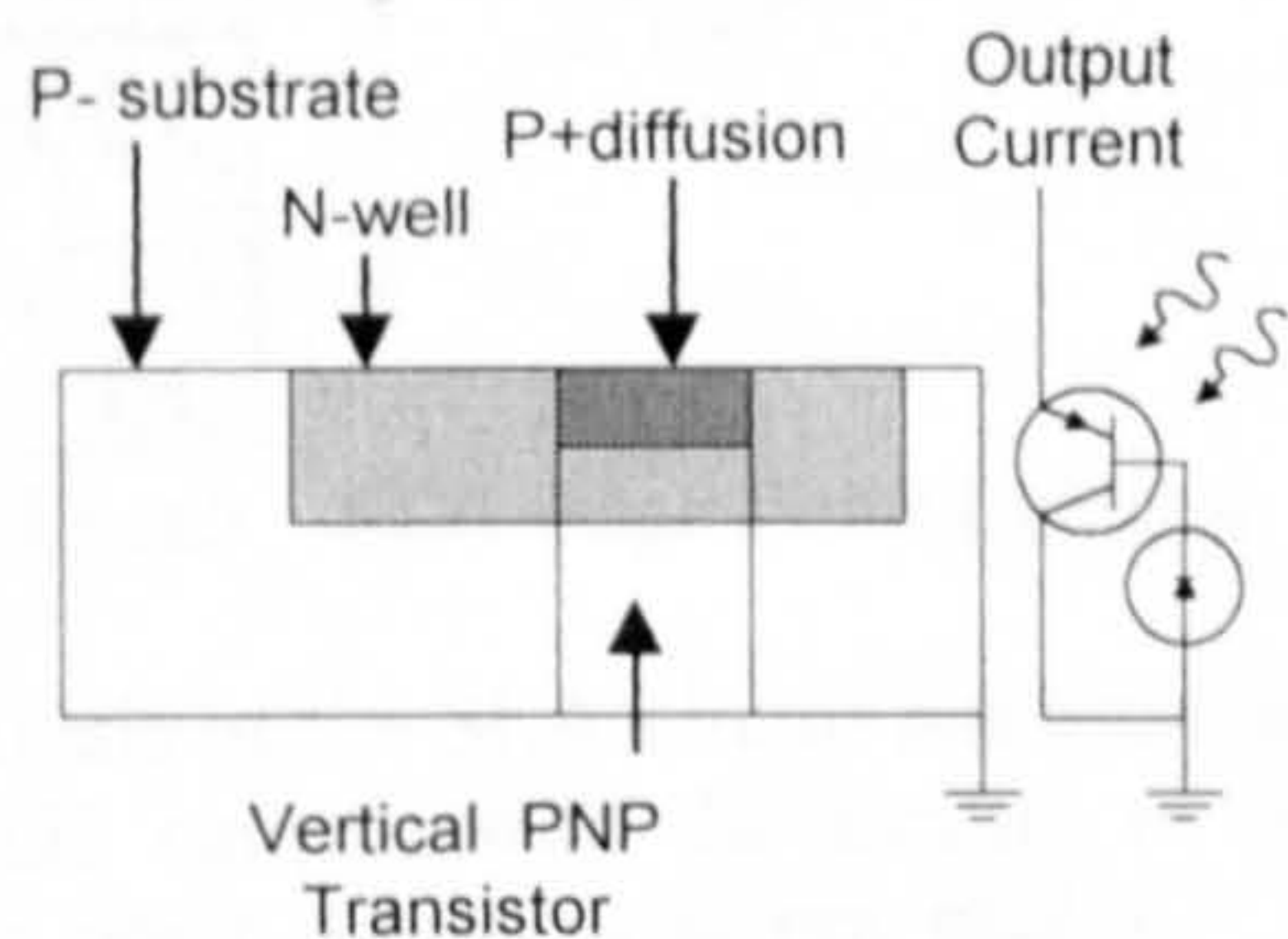


Figure 2 Current Mode Pixel

Measurements have been made on test pixels fabricated in the MIETEC N-well 2.4 μ m CMOS process. A 50x50 μ m N-well with a 5x5 μ m central P+ diffusion has been found to generate a current of 120nA under illumination of 500 Lux. The response of this pixel has been measured from 3Lux to 3000Lux. In this range, 100Lux corresponds to corridor lighting, 500Lux to office lighting, and 2000Lux corresponds to inspection lighting. A plot of the pixel current gain for the daylight visibility range is given in Figure 3. Measurements on switching devices in the 2.4 μ m technology established that a 10Hz framing rate is available for a 250x250 array under corridor illumination.

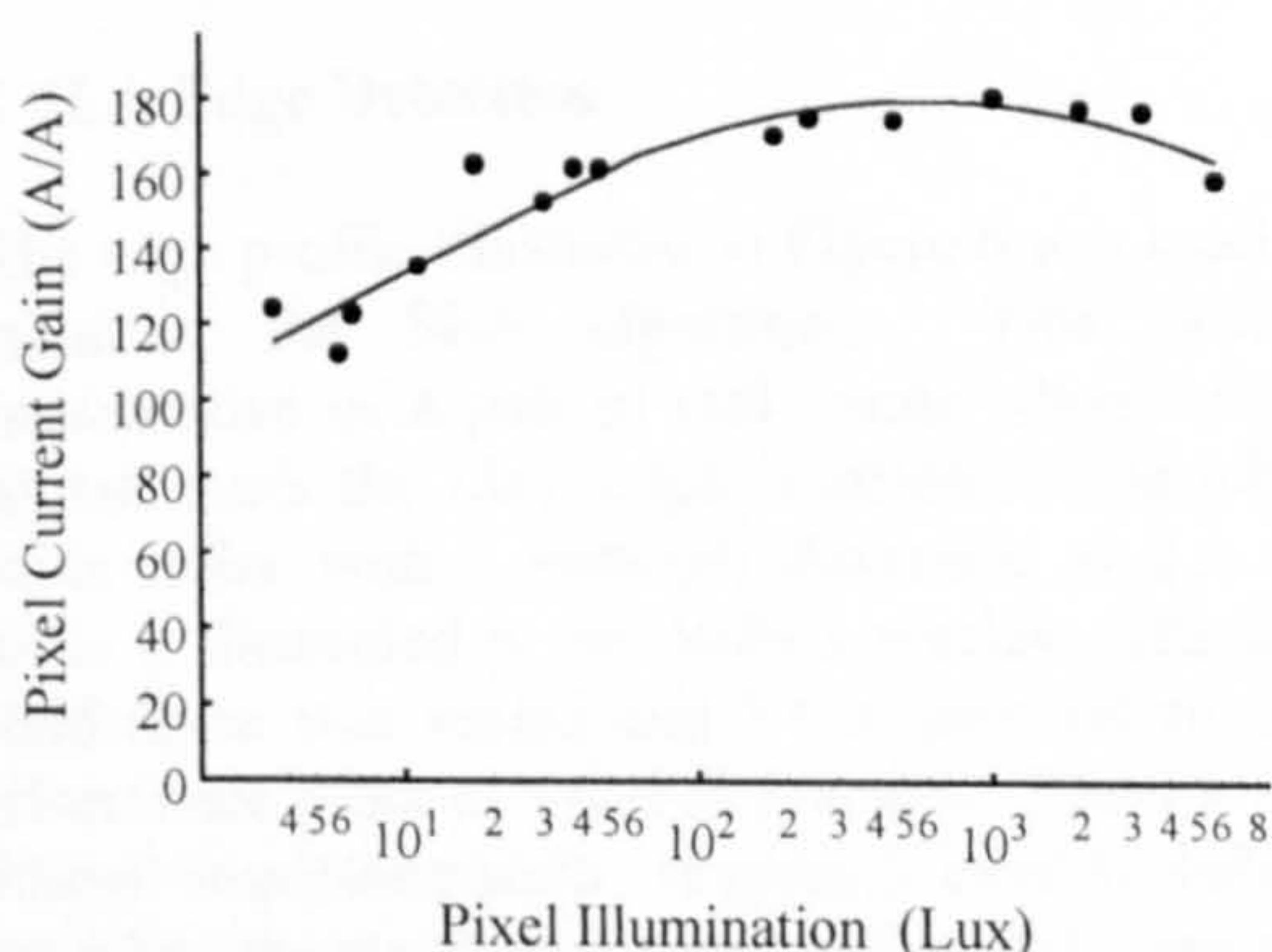


Figure3 Current Mode Pixel Gain

3.2 Layer 1: Current Mode Cell

The circuit of Figure 4 illustrates the processing cells used in layer 1. The purpose of this layer is to compute the $D^1 \& I(x,y)$ and $D^2 \& I(x,y)$ analogue signals defined in section 2.2. These analogue signals are realised as voltages at nodes formed when the positive and negative outputs from adjacent cells are linked.

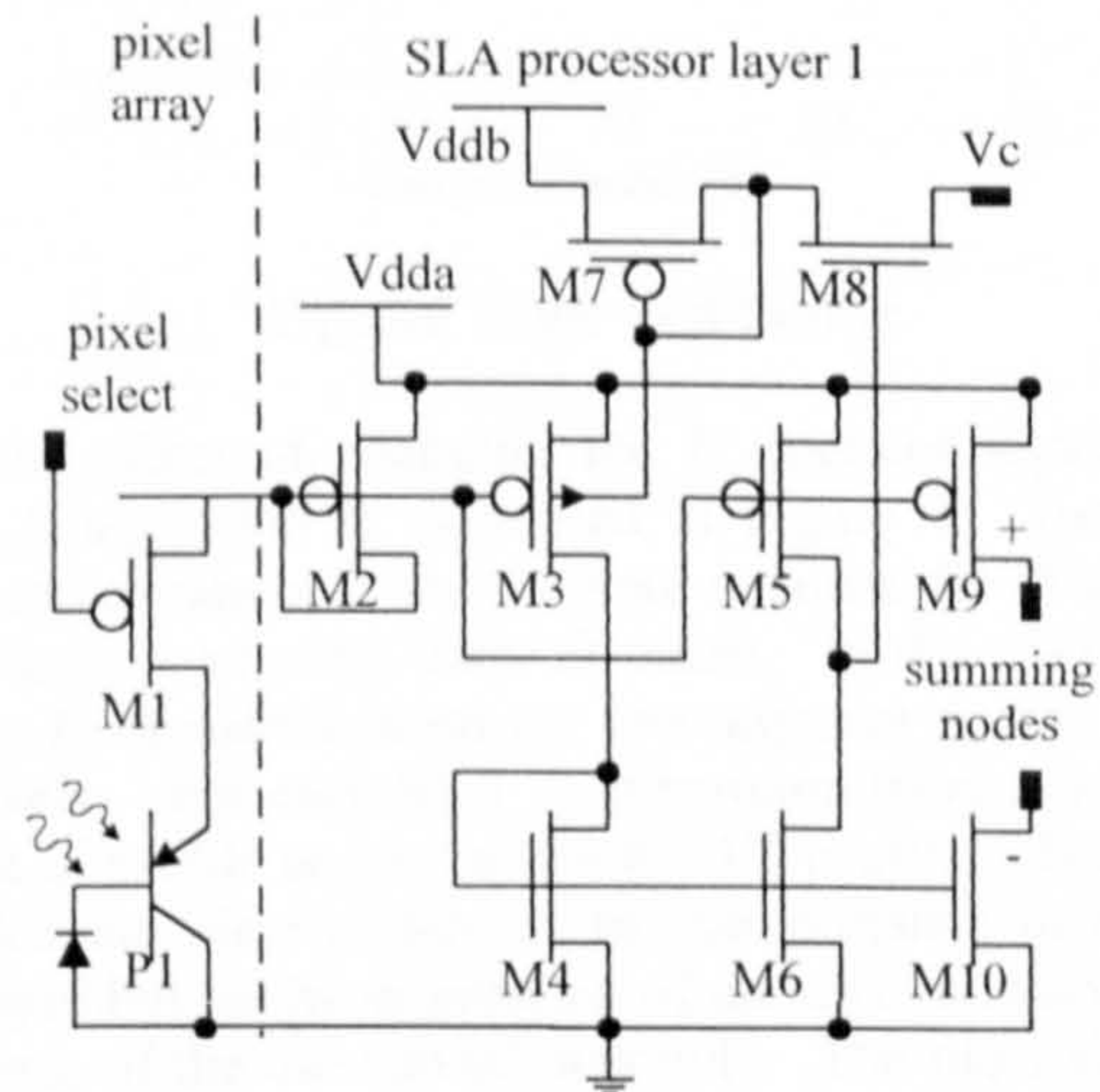


Figure 4 Layer 1: Current Mode Cell

The positive and negative outputs necessary for the $D \& I(x,y)$ analogue signals are supplied by multiple mirrors of the M9 and M10 devices shown in Figure 4. The conductance of M10 is matched to that of M9 through the M3-M8 circuit. A comparison between the external reference V_c and the M5/M6 divider potential sets the back-gate voltage of M3. This back-gate voltage modulates a mirror of the pixel current that is supplied to M4. The M4 current is mirrored in M6 to close the control loop. The pixel current is also mirrored in M5. The control loop maintains the M5/M6 divider voltage at $(V_c + V_{GS8})$. When this voltage is set to $V_{dda}/2$, M6 and M5 have equal conductance. The geometry of M6/M10 and M5/M9 are matched to give the equivalent output conductance. SPICE simulations have been used to confirm that this conductance match is maintained as the input pixel current is swept over the daylight visibility range of the 50x50 μ m pixel (1nA-2 μ A).

3.3 Three Layer Edge Detector

A schematic representation of the three-layer edge detection process is given in Figure 5. This illustrates the circuits and connections necessary for a minimal implementation of the SLA edge detection algorithm. The photo currents from four pixels are processed through layer 1 to yield the D^1 and D^2 analogue signals. In layer 2, voltage comparators provide a discrete conversion for these analogue signals. In layer 3 logic circuits detect the valid edges FE and RE (equations 7 and 8) within the discrete outputs. SPICE analysis has

been used to confirm that this circuit implements SLA edge detection algorithm.

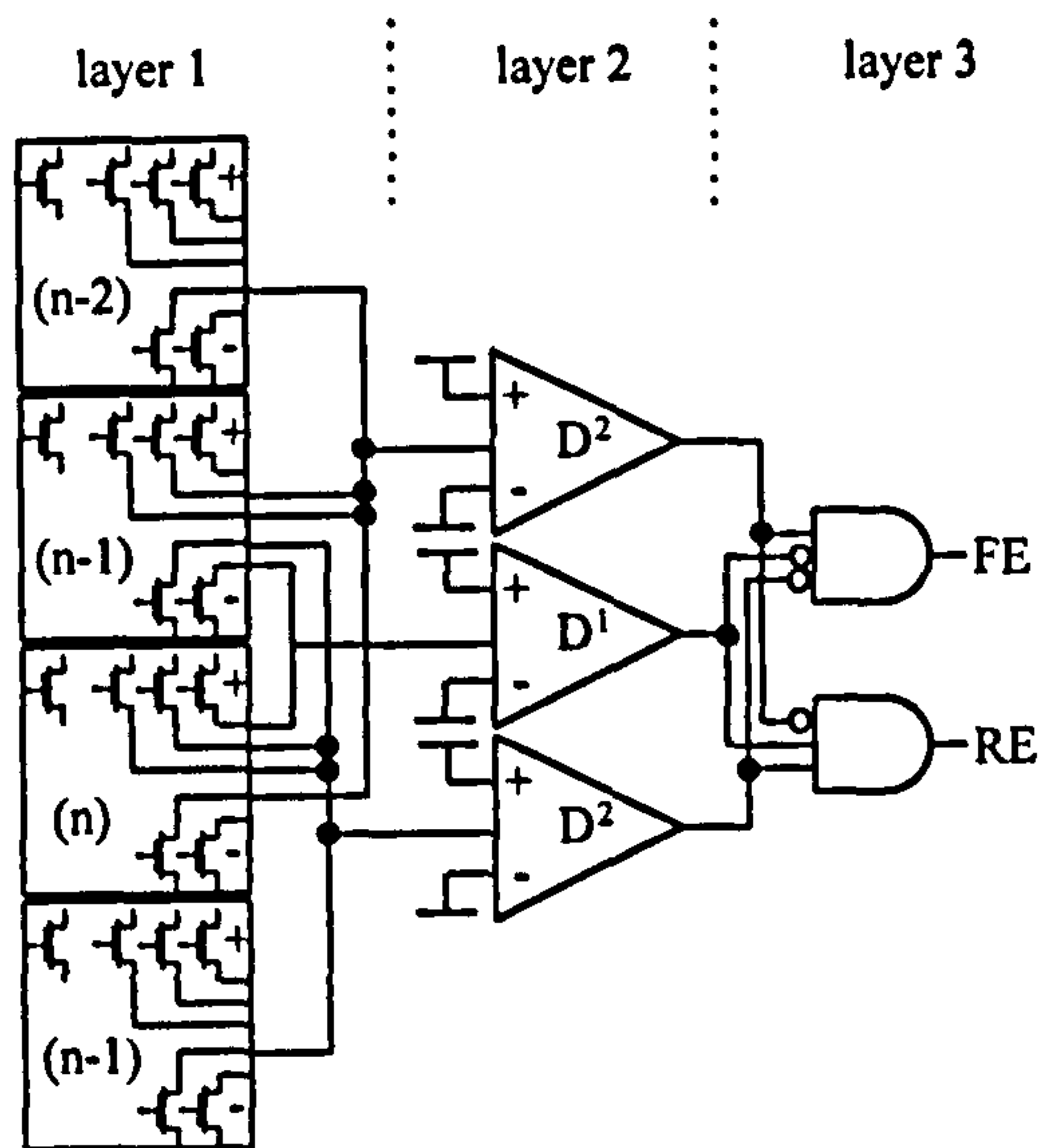


Figure 5 SLA Edge Point Detector.

Analysis of the SLA circuit has established that a smart CMOS camera with the facility to provide real time edge point detection is practicable. The pixel design and linear processing of the SLA first layer gave this smart camera the facility to detect edges within an image where the contrast levels range over three orders of magnitude. In comparison, a machine vision system that processes sampled image data would require a 10-bit digital conversion of pixel intensities to detect edges over three orders of magnitude.

4 IMAGE PROCESSING SIMULATION

Image processing simulation has been used to assess the SLA algorithm and compare its performance to Canny edge detector distributed by Parker (5). The code used in the simulation reflected the functional limitations of low-complexity SLA circuits described in section 3. The simulation also reflected the addressing limitations of the substrate-based processor.

4.1 SLA Edge Detection

The edge profile illustrated in Figure 6 was used in the evaluation the SLA algorithm. This profile is representative of a pair of real image edges. Gradient maxima mark the ideal edge locations. The effect of adding noise with a standard deviation of 2.3 to this profile is illustrated in the dashed overlay. The level of added noise was varied and PV's recorded to provide performance plots of Figures 7 and 9. The PV for the minimal implementation (Figure 5 circuit) reduced to zero when the standard deviation of added noise was set to 0.5 .

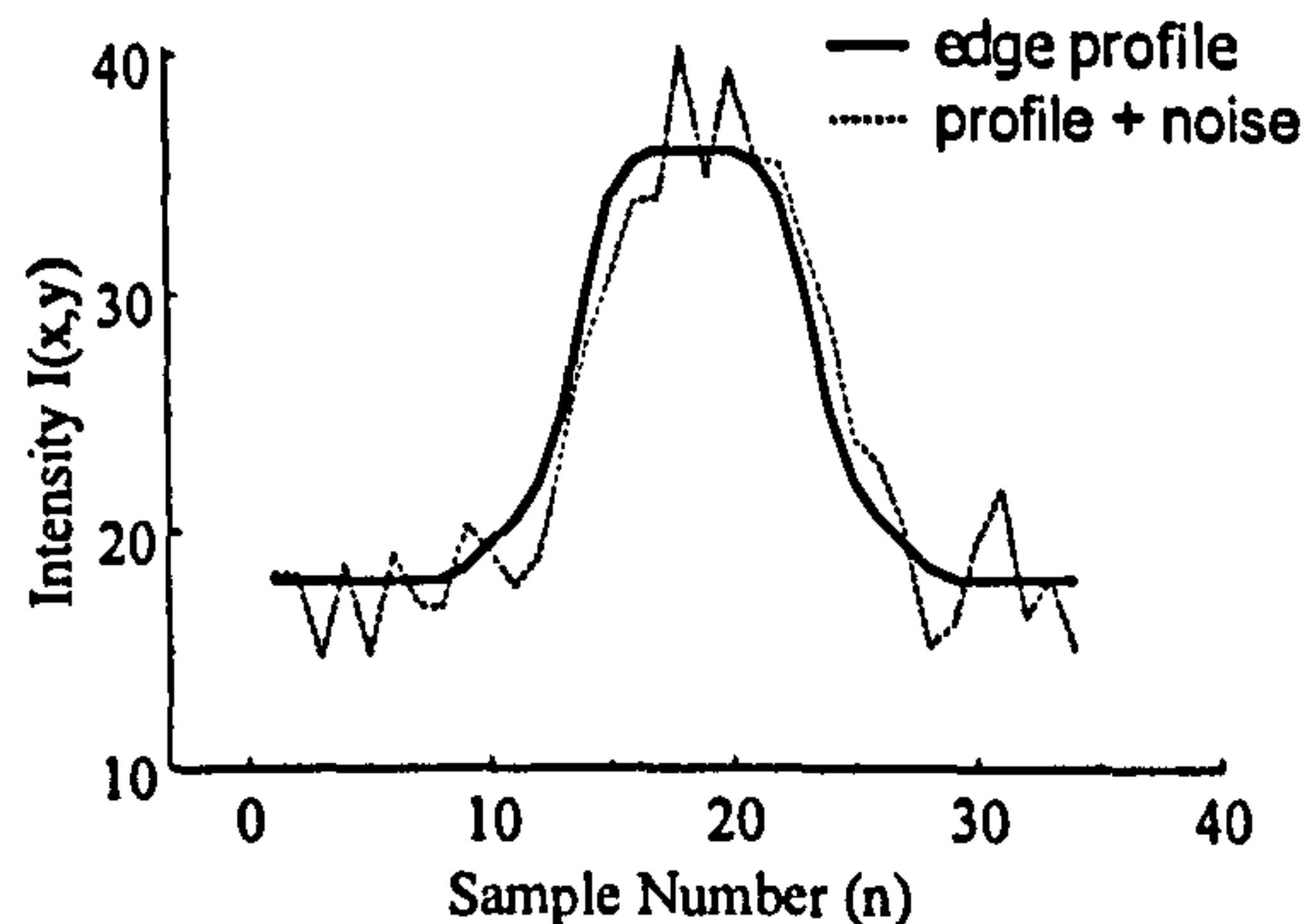


Figure 6 Edge Test Profile

The effect of changing the D operator width in the SLA algorithm is illustrated in Figure 7. The "S2D" plots (equations 5 and 6), were obtained for D operators acting on adjacent pairs of pixels. In the "S4D" plots, the D operators acted on two adjacent groups of four pixels. For each SLA configuration there is a plot for the 0° profile and a plot for the 45° profile. These plots illustrate the variation in the performance of the SLA algorithm as the orientation of an edge is rotated in the plane of the directional operators. The plots also show that extending the width of the D operators reduces the noise susceptibility.

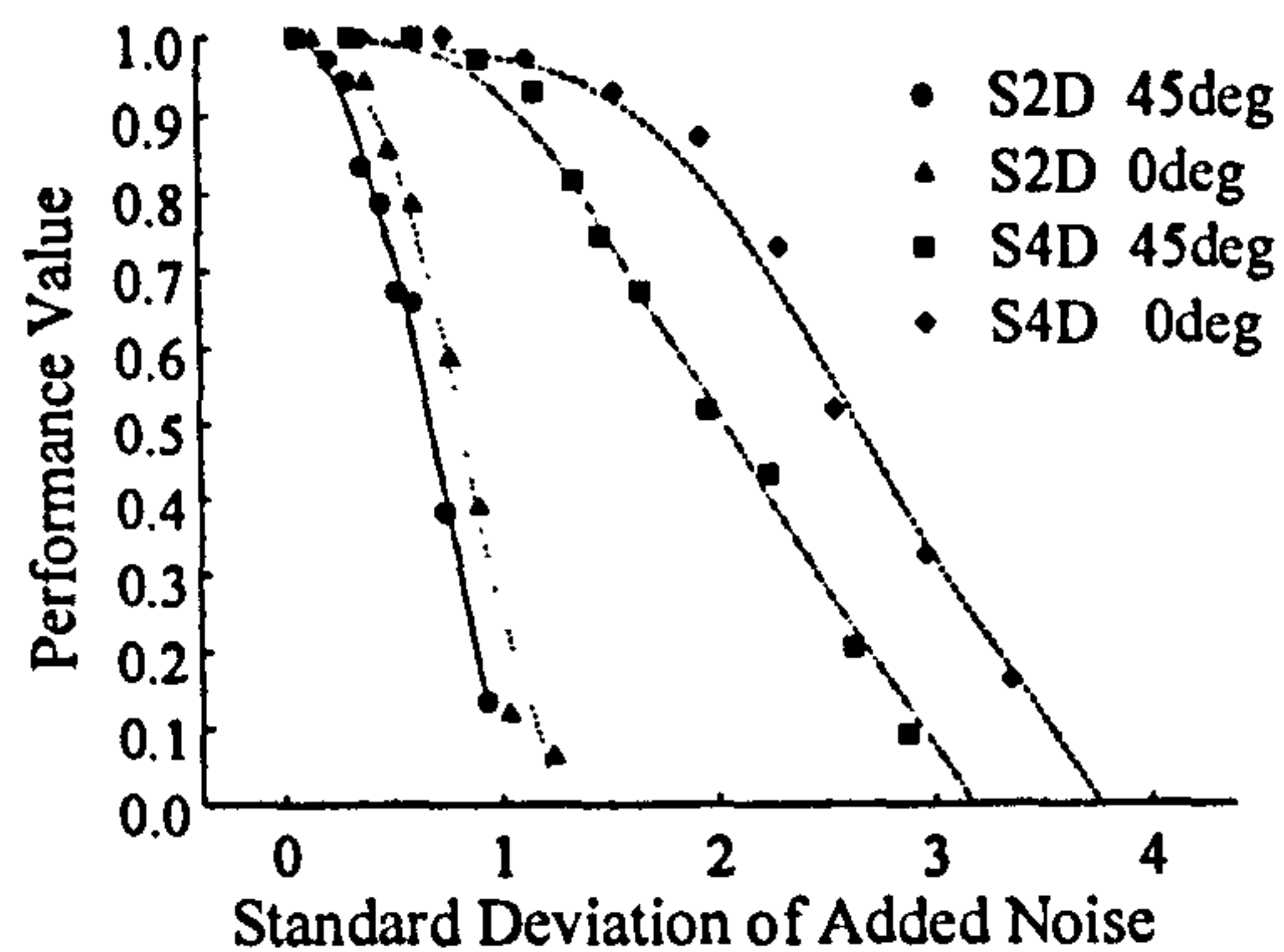


Figure 7 Comparison of SLA Widths Two and Four

4.2 Canny SLA Comparison

The SLA algorithm has been compared to the Canny edge detector (5). These comparisons have been made through performance value tests on synthetic images and through segmentation results from captured images. In the comparison tests the Canny detector employed a sigma of 1.0 and 7×7 convolution masks. The SLA detector employed 8×1 convolution masks. The Canny thresholds were set to a high of 60 and a low of 30. The SLA detector employed a null band of ! 20 for the D^1 operator, and null band of a !5 of the D^2 operator. The Canny detector required a floating-point processor. The SLA algorithm employed integer additions, subtractions and logical functions.

The segmentation results from the "Clare" image given in Figure 8 provide a subjective comparison of the SLA and Canny edge point detectors. These results show that both algorithms detect hard and soft edges within the image and reject noise. The SLA edge detector gives better corner definition, considering the jacket lapels the Canny detector gives a more rounded return. The finer grain of the SLA algorithm also provides a more detailed segmentation of the eyes.



Figure8 Segmentation Results for the Clare Image

Results from a set of performance tests carried out on a synthetic image are illustrated in Figure 9. These results demonstrate that the SLA and Canny detectors have equivalent responses. The synthetic image had a series of steps (height of 64). The fall-off in PV for both detectors was due to false positives. At a SNR of 11.8, both detectors properly detected the image steps.

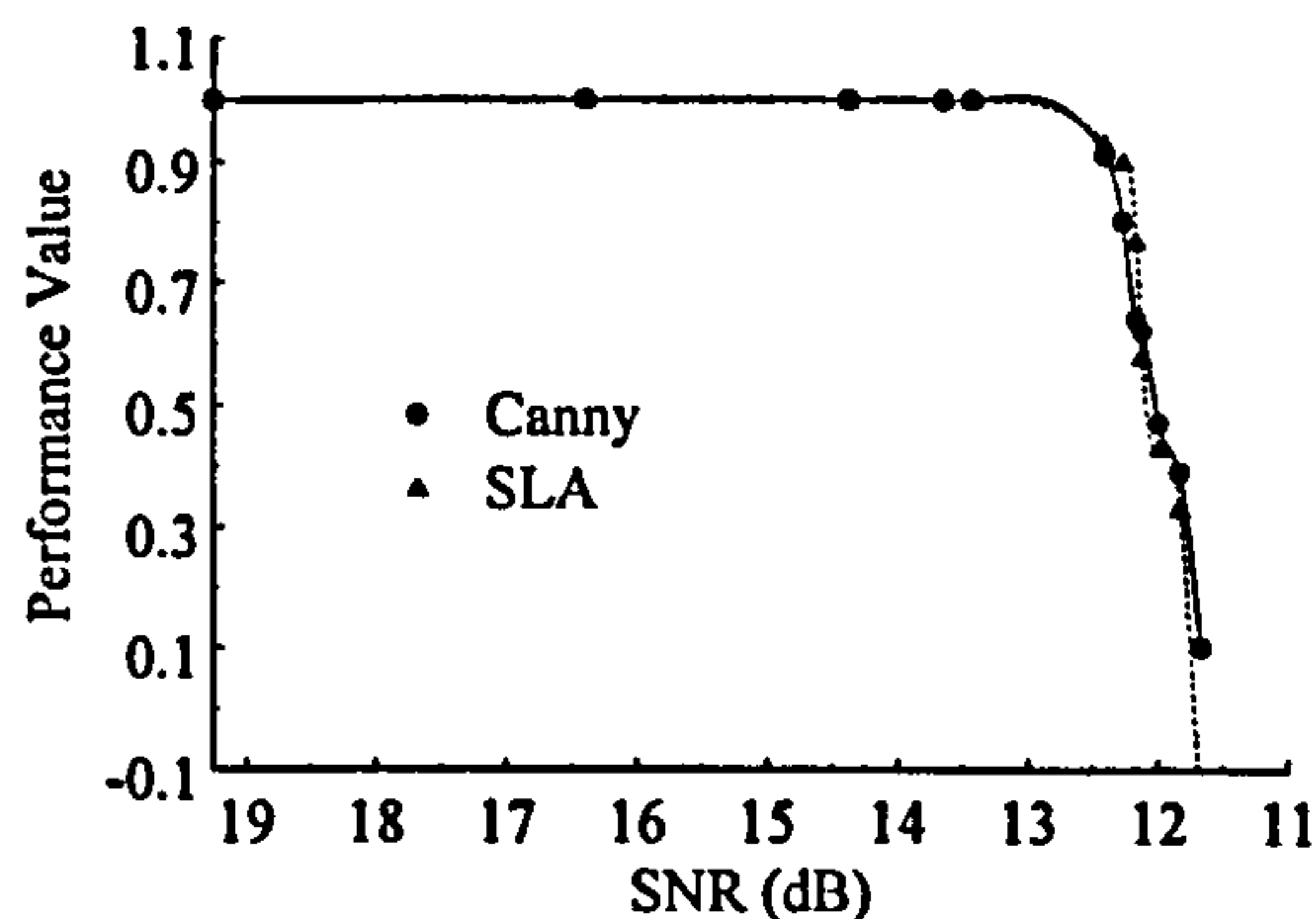


Figure9 Synthetic Image Segmentation

5 CONCLUSION

The proposal for a Smart CMOS Camera has been evaluated through measurements on fabricated CMOS devices, SPICE analysis and results from image processing simulation. An edge detection algorithm designed for CMOS circuit implementation has been developed. The algorithm has been realised through a Scanned Layer Architecture, which provided efficient parallel processing of the image data. This architecture allowed the smart camera to have a spatial resolution similar to that found in commercial CMOS cameras.

The simulation results have established that the SLA algorithm has a performance similar to the computationally intensive Canny edge detection

method. The low power required by the SLA CMOS circuit, coupled with its high quality edge detection provides a solution to the problem of primitive extraction for autonomous vision systems. Future work will include the fabrication of sensing array with an orthogonal scan and layered processing circuits.

6 REFERENCES

- 1 Canny J, 1986. "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8.6,679-698
- 2 Kosaka A, Juiyao P, 1995, "Purdue Experiments in Model-Based Vision for Hallway Navigation", Proc. of Workshop on Vision for Robots in IROS'95 Conf., Pittsburgh, PA, 87-96.
- 3 Marr D, Hildreth E, 1980, "Theory of Edge Detection", Proc. R.Soc. London, 207, 187-217
- 4 Murray D, Bradshaw K, McLauchlan P, Reid I, Sharkey P, 1995, "Driving Saccade to Pursuit using Image Motion", Journal of Computer Vision, 16, 3, 205-228.
- 5 Parker J, 1997, "Algorithms for Image Processing and Computer Vision", John Wiley & Sons, New York, NY:
- 6 Pratt W, 1991, "Digital Image Processing 2nd Ed.", John Wiley & Sons, New York, NY
- 7 Staunton R, 1998, "Edge operator error estimation incorporating measurements of CCD TV camera transfer function", IEE Proc.- Vis. Image Signal Process., 145, No. 3.
- 8 Xu Li-Qun, Machin D, Sheppard P, 1998, "A Novel Approach to Real-time Non-intrusive Gaze Finding", British Machine Vision Conference 98, 428-437. New York, NY:
- 9 Delbruck. T. (1993). Silicon retina with Correlation-Based, Velocity-Tuned Pixels. IEEE Transactions on Neural Networks, Vol. 4, No. 3, pp. 529--541, May
- 10 Dominguez-Castro, *et al.* (1997) Mixed-Signal Focal-Plane Array Processor, IEEE JSSC Vol. 32 No7.

An Edge Point Metric for the Contextual Assessment of Detectors

T.W.J. Moorhead, T.D. Binnie, School of Engineering, Napier University, Edinburgh

Abstract

The optimisation and analysis of edge detectors are critical factors in the successful implementation of vision systems. Existing metrics for appraising edge detector performance are not comprehensive. We report a new metric, referred to as the Edge Point Metric (EPM), that measures the performance of a detector within the context of vision system's operation. In this a minimum quality specification is determined for detectors considered for use within a vision system. An autonomous navigation system is used to demonstrate the EPM assessment method. The metric employs a new ground-truth comparison algorithm that classifies the detector results as true segmentation, distorted segmentation and segmentation errors. The EPM evaluates the performance of the detector through a scaled summation of the segmentation errors. Receiver Operator Characteristic curves are used to choose optimal detector parameters. Results from an optimised detector are used to compare the operation of the EPM, Pratt and Kitchen Rosenfeld metrics.

1 Introduction

We present a new metric for the quantitative evaluation of edge detectors. This metric is referred to as the Edge Point Metric (EPM). Existing metrics [1-6,7] provide a figure of merit that ranges from zero to unity. A perfect detector is assigned a unity rating, but the significance of a zero rating is not well defined. In the EPM figure of merit the zero condition is set to Förstner's minimum quality specification [8], thus a detector can be assessed within the context of the requirements of a vision system [9]. In our research the EPM metric was used to assess the suitability of edge detectors for use within the vision system illustrated in Figure 1. This system was designed to implement the wall following task required for autonomous indoor navigation [10,11]. The critical floor to wall boundaries are identified through the use of Beveridge's Local Search Algorithm that match extracted image lines with a geometric model [12]. Uncertainty assessment determined that the maximum omission rate of edge points within the extracted lines should not exceed 1:6 pixels. Analysis of representative room and corridor images established that it was necessary to attain this omission rate with a step edge profile Signal to Noise Ratio (SNR) of 6dB or less.



Figure 1 Vision Based Navigation System

Förstner [8] expressed the edge detector quality requirement through equation (1). In this equation q_0 represents the edge detector's minimum quality specification. Parameter r represents the result given by the algorithm a tested using data d and tuning parameters t . The measured quality value $q(r)$ is assumed to increase with improving quality, $q(r)$ is evaluated and q_0 is specified for each quality attribute of the detector.

$$q(r|d,a,t) \geq q_0 \quad (1)$$

In our analysis of edge detectors, for use in the vision based navigation system, the test data was provided by a synthetic image from which graphically drawn hairline outlines were used to give a ground truth image. Gaussian noise with a zero mean was then added to the synthetic image to produce a set of ten test images where the SNR varied from 22dB to 2.3dB.

The EPM figure of merit detailed in Section 2 provides a metric representation of Förstner's minimum quality specification [8]. The figure of merit is evaluated from a scaled linear summation of the probabilities of false positive and false negative returns generated by the tested detector. The scaling of the summation is set to register a merit value of zero or less when the detectors results are equal to or worse than the systems minimum quality specification. The false negative probability $P(FN)$ is defined as the probability of a false negative occurring within the set of valid edge point sites. The false positive probability $P(FP)$ is defined as the probability of a false positive occurring within the set of valid non-edge point sites [7]. The sets of valid and non-valid edge point sites and the false return occupancy of these sets are determined through the application of heuristics contained within a Detector to Ground-truth Comparison algorithm (DGC), that was developed for use with the EPM method. Furthermore the DGC algorithm provides for the assessment of the detectors susceptibility to line broadening and the displacement of detected edge points [7]. In Section 3 Receiver Operator Characteristic (ROC) curves are used to select optimal edge strength thresholds for the SUSAN [13] and Sobel detectors. In Section 4 edge point results generated by the optimised SUSAN detector are used to compare the operation of the EPM with the Pratt and Kitchen Rosenfeld metrics [1,4].

2 Edge Point Metric

2.1 Detector to Ground-truth Comparison (DGC) Algorithm

Edge point detectors allocate edge points through a series of discrete kernel based operations. These operations can give rise to systematic displacements of the detected edge points and cause line broadening. We viewed these displacement and broadening effects as short-form distortions within the segmentation results. In the context of the Figure 1 vision system, localised segmentation distortions of 1 or 2 pixels do not limit the performance of the system. The reported metrics [2,4,6,7] in their figure of merit evaluations attribute a relatively low significance to short-form segmentation distortions.

The EPM method utilises the DGC algorithm to detect the presence of displacement or line broadening pixels within the detector results and hence can distinguish between true segmentation results, segmentation distortions and segmentation errors. An overview of the hierarchical structure of the DGC algorithm is given in Figure 2. For a given Intensity profile ($I_{(x,y)}$) the DGC algorithm takes two input image sets. An Edge Point set ($EP_{(x,y)}$) generated by the application of an edge detector to the intensity profile, and a Ground Truth set ($GT_{(x,y)}$) that marks the valid edge points in the intensity profile. In each set an edge point is marked by a '1' and a non-edge point is marked by a '0'. The DGC algorithm employs three heuristic phases to classify each pixel within the edge map into one of seven states. These states are: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Displaced Positive (DP), Displaced Negative (DN) and Wide Positive (WP). The *DP*, *DN*, and *WP* states account for the segmentation distortions within the detectors results, the *FP* and *FN* states account for the segmentation errors and the *TP* and *TN* states account for the true segmentation results.

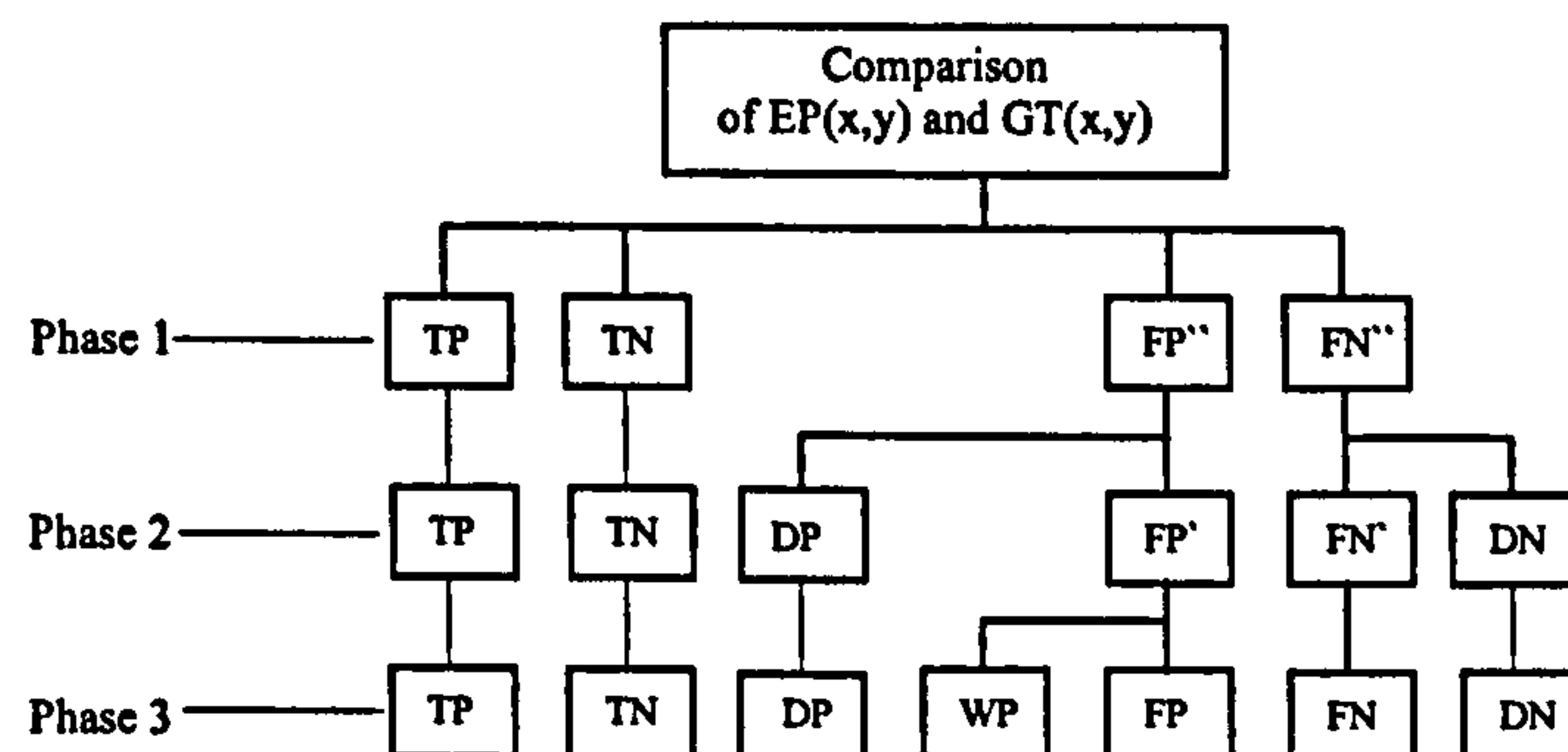


Figure 2 DGC Algorithm Decision Hierarchy

In Phase 1 the DGC algorithm processes the $EP_{(x,y)}$ and $GT_{(x,y)}$ image sets using the heuristic given by Table 1 to generate an interim image $Map1_{(x,y)}$ populated by TP, TN, FP, FN states. In the second and third phases of the DGC algorithm spatial heuristics, that employ 5x5 convolution kernel illustrated in Figure 3(a), locate displacement and line broadening states within the FP and FN results. The centre pixel of this kernel is termed pixel Central (pC). The results of the spatial heuristic tests are loaded into the pC location.

$EP_{(x,y)}$	$GT_{(x,y)}$	$Map1_{(x,y)}$
1	1	TP
0	0	TN
0	1	FN
1	0	FP

Table 1 Phase 1 Heuristic Assignment

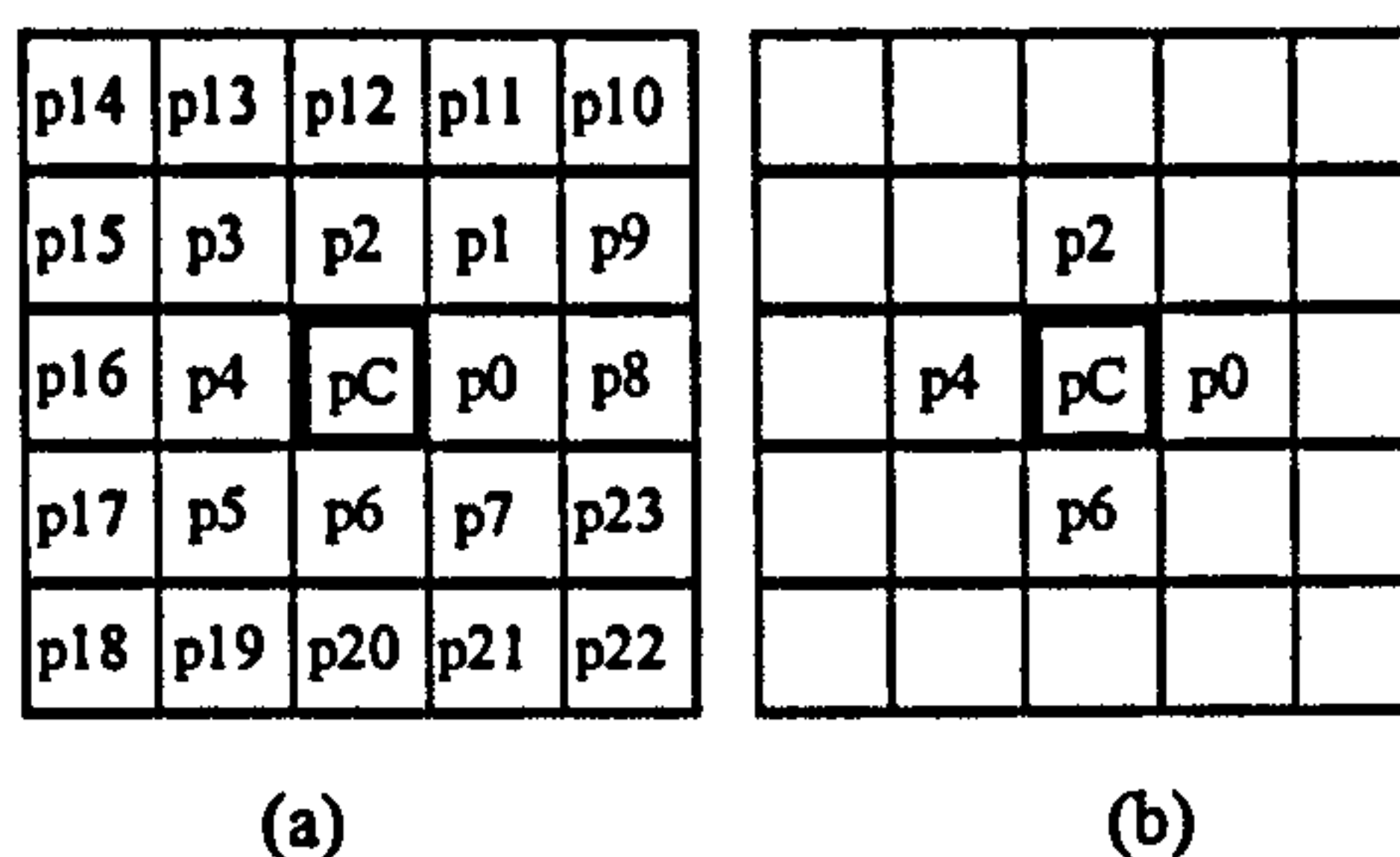


Figure 3 (a) Full DGC convolution kernel, (b) kernel for 4-connected single displacement heuristic

In Phase 2 of the DGC algorithm the $Map1_{(x,y)}$ image is processed using heuristics that discriminate DP states in the FP returns and the DN states in the FN returns. Single displacements are tested for in 45° intervals around pC and double displacements are tested for in 90° intervals around pC [7]. Phase 2 of the DGC algorithm creates a $Map2_{(x,y)}$ image that is populated by TP, TN, FP, FN, DP, DN states; the TP, TN states are unchanged from the $Map1_{(x,y)}$ image. The tests for a single displacement at $0^\circ, 90^\circ, 180^\circ, 270^\circ$, in Phase 2 are given by equation (2) for DP reassignment, and by equation (3) for DN reassignment. The pixels used in these heuristics are noted in the four-connected kernel of Figure 3(b).

$$\text{if } \left\{ (pC = FP) \text{ and } \left\{ \begin{array}{l} (p0 = FN) \\ \text{or } (p2 = FN) \\ \text{or } (p4 = FN) \\ \text{or } (p6 = FN) \end{array} \right\} \right\} \text{ then } \{pC = DP\} \quad (2)$$

$$\text{if } \left\{ (pC = FN) \text{ and } \left\{ \begin{array}{l} (p0 = FP) \\ \text{or } (p2 = FP) \\ \text{or } (p4 = FP) \\ \text{or } (p6 = FP) \end{array} \right\} \right\} \text{ then } \{pC = DN\} \quad (3)$$

The kernels used to test for a single diagonal displacement at 45° , 135° , 225° and 315° are illustrated in Figure 4. We determined that in order for a diagonal displacement to be assigned it is necessary to test for the occurrence of two adjacent pixel shifts on the test diagonal. This double displacement criteria associates the diagonal reassignments with a systematic detector response. The heuristics that checks for a double pixel shift on the 45° diagonal are given by equation (4) for DP reassignment, and by equation (5) for DN reassignment. The component terms of equations (4) and (5) are rotated in 90° increments to give the three additional sets of heuristics that check for the diagonal displacements of 135° , 225° and 315° .

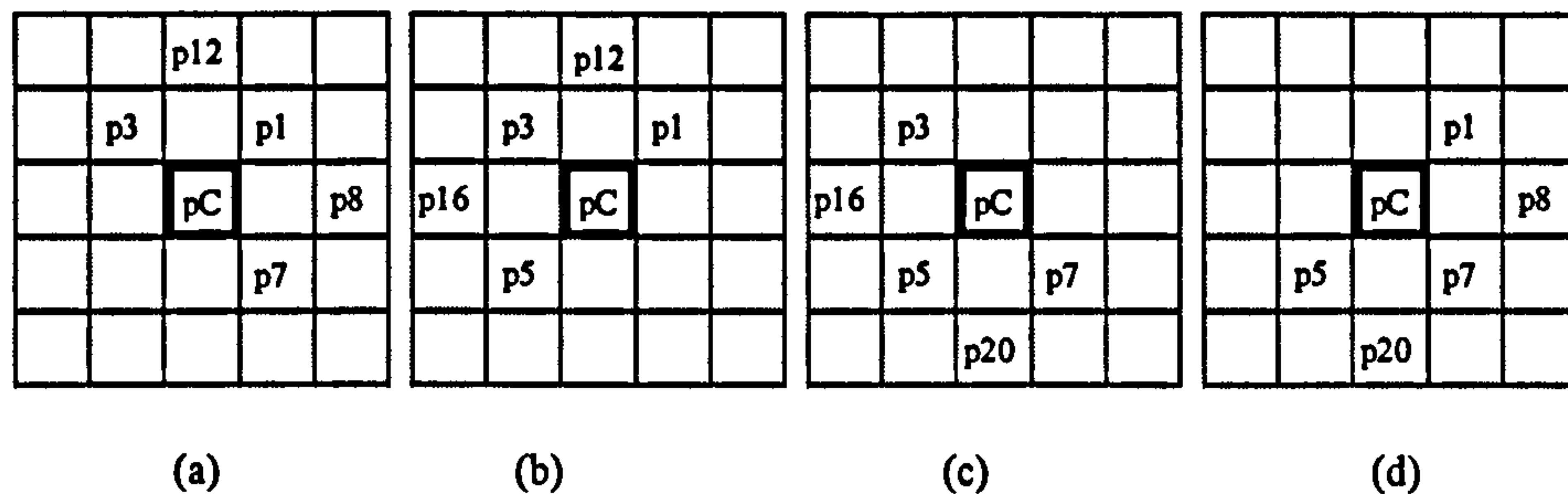


Figure 4 Phase 2 diagonal displacement kernels, (a) 45° heuristic, (b) 135° heuristic, (c) 225° heuristic, (d) 315° heuristic

$$\text{if } \left\{ \left\{ \begin{array}{l} (pC = FP) \\ \text{and } (p1 = FN) \end{array} \right\} \text{ and } \left\{ \begin{array}{l} (p3 = FP) \\ \text{or } (p7 = FP) \end{array} \right\} \text{ and } \left\{ \begin{array}{l} (p12 = FN) \\ \text{or } (p8 = FN) \end{array} \right\} \right\} \text{ then } \{pC = DP\} \quad (4)$$

$$\text{if } \left\{ \left\{ \begin{array}{l} (pC = FN) \\ \text{and } (p1 = FP) \end{array} \right\} \text{ and } \left\{ \begin{array}{l} (p3 = FN) \\ \text{or } (p7 = FN) \end{array} \right\} \text{ and } \left\{ \begin{array}{l} (p12 = FP) \\ \text{or } (p8 = FP) \end{array} \right\} \right\} \text{ then } \{pC = DN\} \quad (5)$$

The kernels used to test for a double displacement at 0° , 90° , 180° and 270° are illustrated in Figure 5. Similar to the diagonal displacement assignment it is necessary for two adjacent pixel shifts to occur in the test

direction. Additionally the double displacement heuristics test for a *TN* separation of the displaced pixels. The *TN* separation test ensures that noise related edges that track normal to the true image outlines are registered as segmentation errors. The heuristics that check for double displacement in the 0° direction are given by equation (6) for *DP* reassignment, and by equation (7) for *DN* reassignment. The component terms of equations (6) and (7) are rotated at 90° intervals to give the three additional sets of heuristics that check for the displacements of 90° , 180° and 270° .

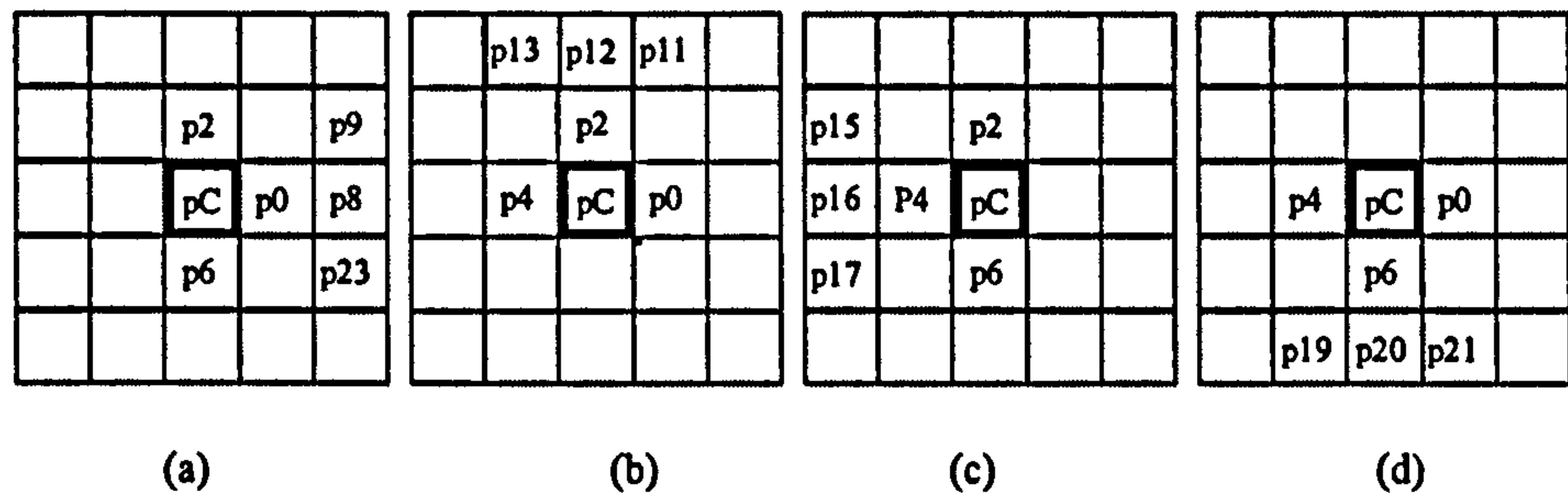


Figure 5 Phase 2 double displacement kernels, (a) 0° heuristic, (b) 90° heuristic, (c) 180° heuristic, (d)

270° heuristic

$$\text{if } \left\{ \left[\begin{array}{l} (pC = FP) \\ \text{and}(pO = TN) \\ \text{and}(p8 = FN) \end{array} \right] \text{and} \left[\begin{array}{l} (p2 = FP) \\ \text{or}(p6 = FP) \end{array} \right] \text{and} \left[\begin{array}{l} (p9 = FN) \\ \text{or}(p23 = FN) \end{array} \right] \right\} \text{then } \{pC = DP\} \quad (6)$$

$$\text{if } \left\{ \left[\begin{array}{l} (pC = FN) \\ \text{and}(pO = TN) \\ \text{and}(p8 = FP) \end{array} \right] \text{and} \left[\begin{array}{l} (p2 = FN) \\ \text{or}(p6 = FN) \end{array} \right] \text{and} \left[\begin{array}{l} (p9 = FP) \\ \text{or}(p23 = FP) \end{array} \right] \right\} \text{then } \{pC = DN\} \quad (7)$$

In Phase 3 of the DGC algorithm the *FP* returns of $Map2_{(x,y)}$ are tested to see if they can be designated as width modulation pixels. In this Phase an *FP* return that increases the width of the detected line is reallocated to the *WP* state. In Phase 3 the DGC algorithm uses a single heuristic test. The convolution kernel for this test is illustrated in Figure 6. This heuristic applies the eight-connected test of equation (8), to check for line broadening pixels. The *TP*, *TN*, *DP*, *FN* and *DN* returns in $Map2_{(x,y)}$ are not changed by the operation of this heuristic, these plus the *FP* and *WP* results from the third heuristic phase are loaded into $Map3_{(x,y)}$.

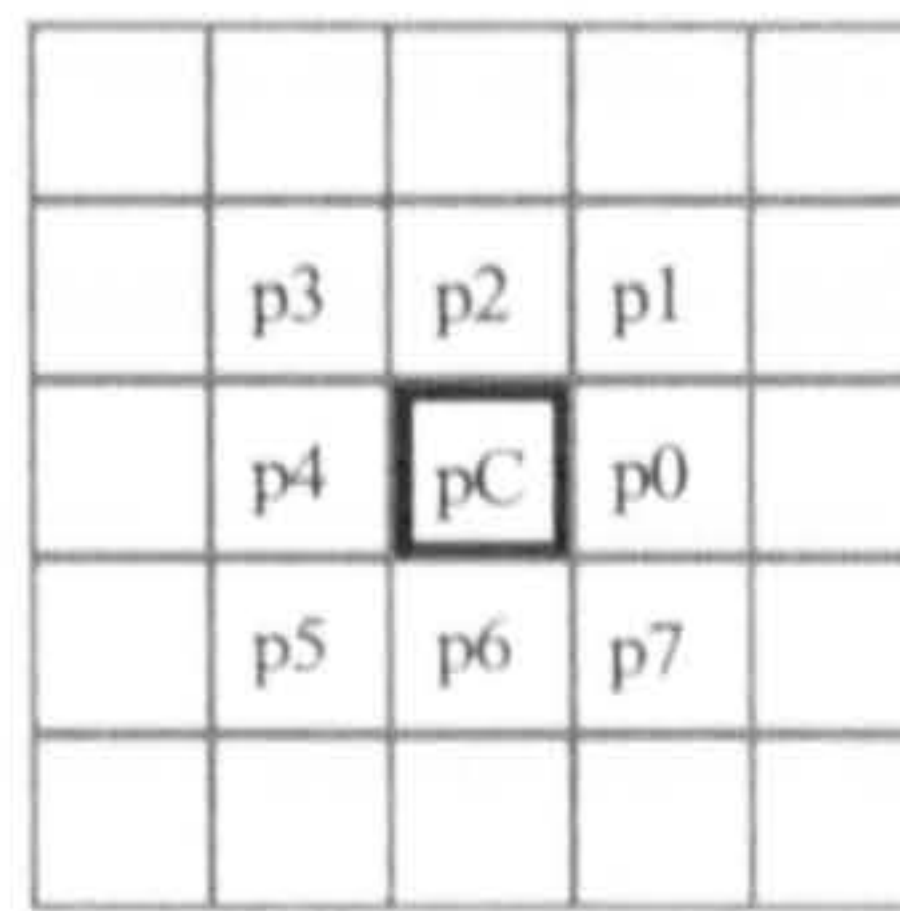


Figure 6 Kernel for Phase 3 Test

$$\text{if } \left\{ (pC = FP) \text{ and } \left[\begin{array}{l} (p0 = TP) \text{ or } (p0 = DP) \\ \text{or } (p1 = TP) \text{ or } (p1 = DP) \\ \text{or } (p2 = TP) \text{ or } (p2 = DP) \\ \text{or } (p3 = TP) \text{ or } (p3 = DP) \\ \text{or } (p4 = TP) \text{ or } (p4 = DP) \\ \text{or } (p5 = TP) \text{ or } (p5 = DP) \\ \text{or } (p6 = TP) \text{ or } (p6 = DP) \\ \text{or } (p7 = TP) \text{ or } (p7 = DP) \\ \text{or } (p8 = TP) \text{ or } (p8 = DP) \end{array} \right] \right\} \text{ then } \{pC = WP\} \quad (8)$$

2.2 DGC Operation

The operation of the DGC algorithm is illustrated in Figure 7, where the dashed line that crosses the 5x6 pixel-grid, marks the hairline separation of the two regions of differing intensity. The pixels with bold outlines mark the ground-truth pixels for this intensity discontinuity. The grey filled pixels in Figures 7(a), (b) and (c) give example detector results with systematic shifts, line broadening and false returns. The DGC allocation of states is given by the labels assigned to the Figure 7 pixels. The *TN* assignment labels have been omitted.

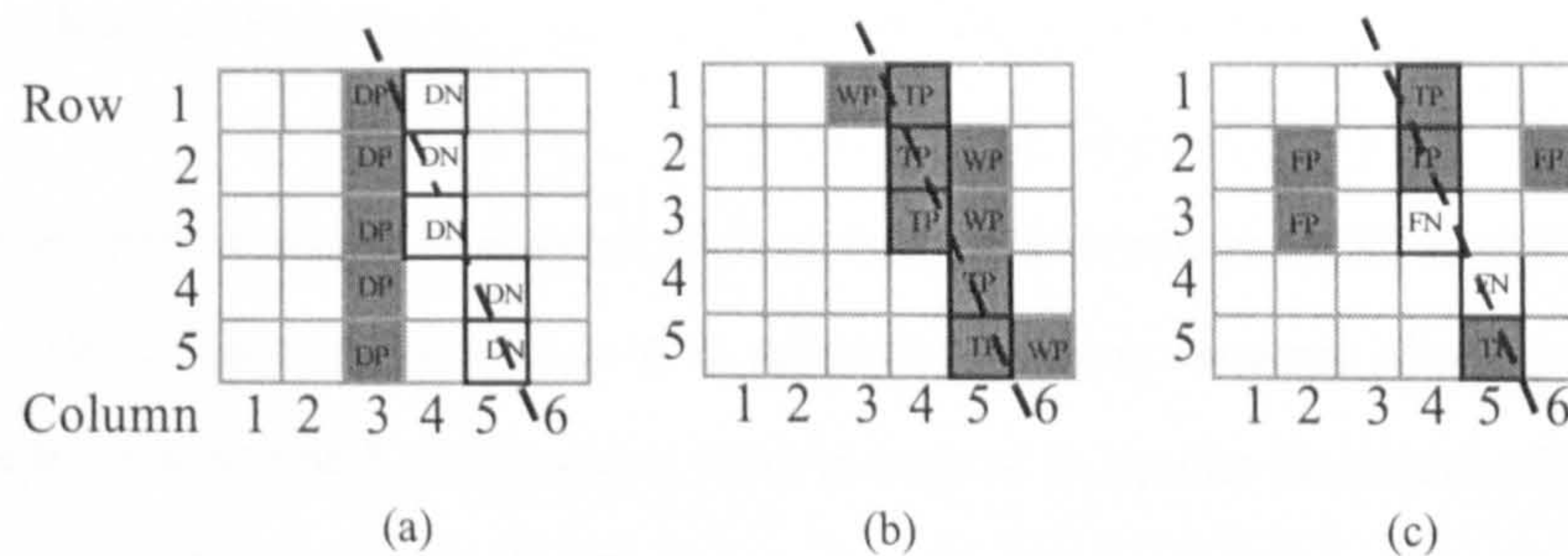


Figure 7 DGC Algorithm example results, (a) systematic shift, (b) line broadening, (c) false returns.

In Figure 7(a) the detector's results are shifted to the left of the ground-truth pixels. Single and double shifted pixels are reassigned to the *DP* and *DN* states. Figure 7(b) illustrates results from a detector that generates

line-broadening pixels. The *FP* and *FN* returns in Figure 7(c) indicate errors in the image segmentation. These segmentation errors contribute to a reduction in the usefulness of the edge detector. In contrast the detector results of Figure 7(a) and (b) have no *FN* or *FP* returns remaining after the reallocation phases, and these give complete segmentations that contain distortions of the ground truth results. The quality of these complete segmentations may be assessed against the degree of displacement and line broadening.

In the assessment of a detector for use in the autonomous problem outlined in Figure 1, the limited displacements registered as *DP* results by the DGC algorithm do not materially effect the facility of the vision system to locate and follow the floor to wall boundaries. The *WP* results generated by the DGC algorithm were removed from the detector results through the use of a thinning algorithm. The EPM figure of merit evaluation for the autonomous navigation system of Figure 1 was based upon the *FN* and *FP* return probabilities taken from the DGC results.

2.3 EPM Figure of Merit

Analysis of edge detector results established that the extraction of valid line segments was limited through the clustering of *FN* and *FP* returns. It was found that the *FP* returns would cluster to form false line segments and the *FN* returns clustered to form extended breaks in the detected outlines. The *FP* returns have the facility to connect with eight adjacent pixels whereas the *FN* returns are normally limited to connecting with two adjacent pixels. Thus under clustering the *FP* returns are four times more likely to link and form an error segment than the *FN* returns.

The EPM figure merit evaluated by equation (9) is given by a linear summation of the $P(FP)$ and $P(FN)$ probabilities. The S_1 scaling factor is set to '6' to reflect the maximum frequency of allowed missed edge pixels, given as 1:6 in Section 1. The S_2 scaling factor is set to '4' to equalise the relative effects of the two types of error probability under clustering. The linear summation of equation (9) registers zero or less when the less when the probability values for $P(FN)$ and $P(FP)$ are 0.167 and 0.042 respectively. The probabilities $P(FP)$ and $P(FN)$ are calculated from the DGC results as given in equations (10) and (11). The *TP*, *TN*, *FP*, *FN*, *DP* and *DN totals* are found by accumulating the number of pixels in each of these states in the $Map3_{(x,y)}$ image. If there are no *FN* or *FP* segmentation errors in the detectors results then the EPM figure of merit

given by equation (9) will be unity. The incorporation of the minimum quality specification [8] into the figure of merit evaluation allows the unity to zero range of the metric to register the degree of conformity of the tested detector with the system's requirements. Contextual assessment can be made through the use of representative captured images for which ground-truth sets exist or through synthetic images in which the SNR is varied across the system's operational range. In the case of representative image evaluation the tested detector must register an EPM value greater than zero to conform to the system requirements. In the case of a synthetic image tests the detector must register an EPM value greater than zero at all SNR's above the specified SNR limit for the system.

$$EPM = 1 - S_1(P(FN) + S_2P(FP)) \quad (9)$$

$$P(FN) = \frac{FN_{total}}{TP_{total} + FN_{total} + DP_{total}} \quad (10)$$

$$P(FP) = \frac{FP_{total}}{TN_{total} + FP_{total} + DN_{total}} \quad (11)$$

The importance of the DGC algorithm is illustrated in the evaluation of the false return probabilities. The *DN* returns are removed from the *Map1*_(x,y) *FN* results, because under the heuristic tests they have been allocated as uncovered *TP*'s with a local *DP* match. In the *Map3*_(x,y) image the number of valid edge point sites is then given by the sum of the *TP*, *FN* and *DP* totals, and this sum provides the normalisation of the *P(FN)* evaluation. The *FP* returns that have been identified by the DGC heuristics as *DP* or *WP*, are removed from the *Map1*_(x,y) *FP* results and the number of possible non-edge pixels in the *Map3*_(x,y) image is given by the sum of the *TN*, *FP* and *DN* totals. This sum of the valid non-edge point sites provides the normalisation of the *P(FP)* evaluation. The *WP* returns extend the space occupied by the valid edge points, but they do not increase the number of valid edge points thus the *WP* total is excluded from the *P(FN)* and *P(FP)* calculations.

The metrics reported by [2,6,7] incorporate the susceptibility of the tested detector to edge point displacement and line broadening into their figure of merit calculations. In the DGC results the susceptibilities of the detector to displacement and broadening are given as the probability evaluations of equations (12) and (13). The displacement susceptibility is given by the probability *P(DP)*, defined as the

probability of a DP return occurring within the set of valid edge point sites. The broadening susceptibility is given by the probability $P(WP)$, defined as the probability of a WP return occurring within the union of the valid edge point and wide positive sets. If the system specification requires the inclusion of these effects, then it is necessary to specify the maximum allowed frequencies of displaced edge points and broadening edge points to extend the linear summation of equation (9).

$$P(DP) = \frac{DP_{total}}{TP_{total} + FN_{total} + DP_{total}} \quad (12)$$

$$P(WP) = \frac{WP_{total}}{TP_{total} + FN_{total} + DP_{total} + WP_{total}} \quad (13)$$

3 ROC curves.

Receiver Operator Characteristic (ROC) curves provide an effective means of analysing the response of a detector with respect to tuning parameters and thus allow for the selection of optimal parameters for a given application [14]. Figures 8(a) and 8(b) give examples of ROC $P(FN):P(FP)$ curves that facilitate the selection of the edge strength parameter for the Sobel and SUSAN detectors. The log:log format of the Figure 8 curves ensures that the low probability ranges which are important to the function of the edge detector are adequately displayed. It is the norm in ROC curves to plot $P(TP):P(FP)$, however we replace $P(TP)$ with $P(FN)$, equivalent to $(1-P(TP))$, to give an inversion of the ROC characteristic. This inversion allows for the plotting of the EPM zero condition onto the ROC graph and thus contextual assessments can be made within the optimisation process.

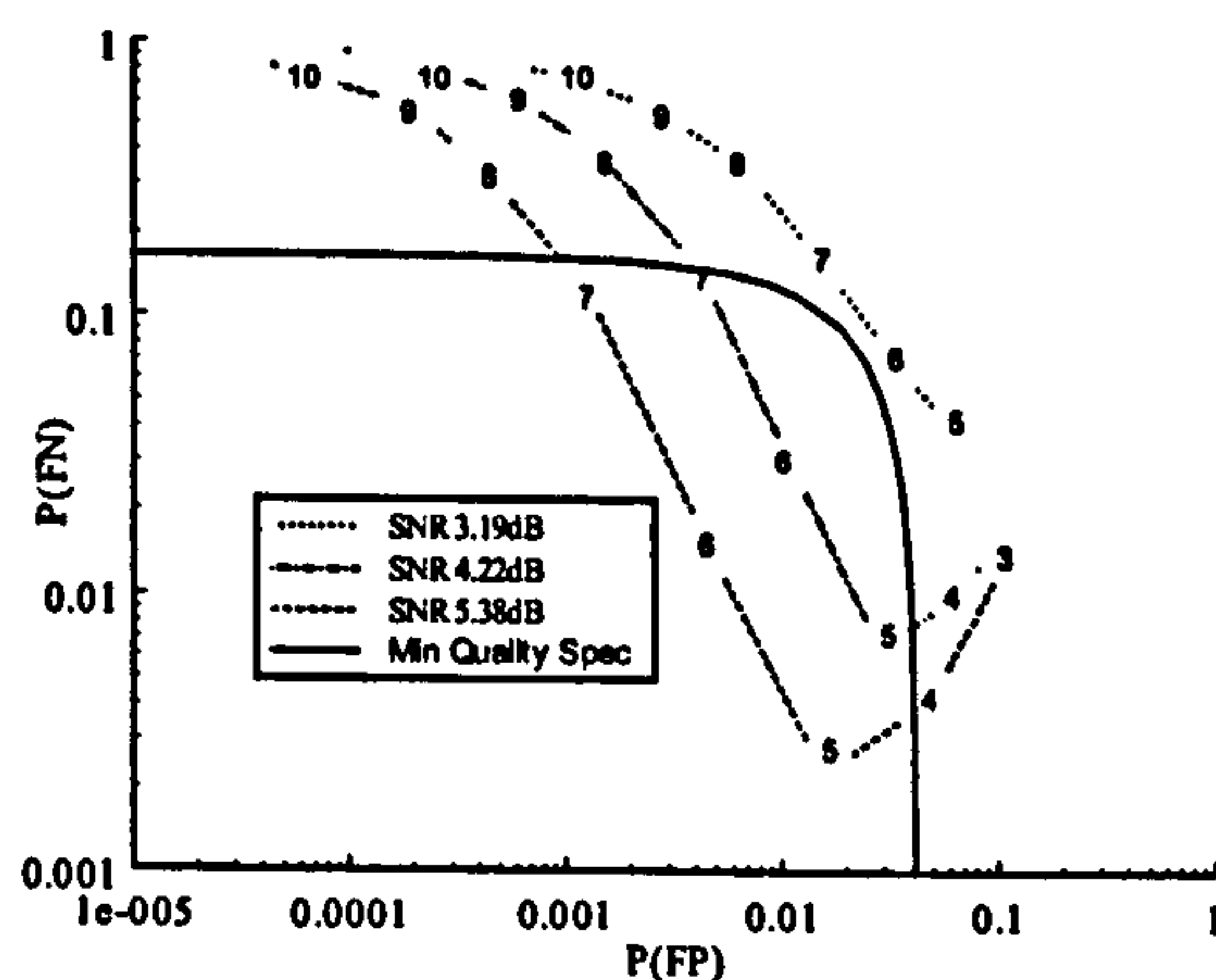


Figure 8(a) ROC Curves SUSAN Detector Edge Strength

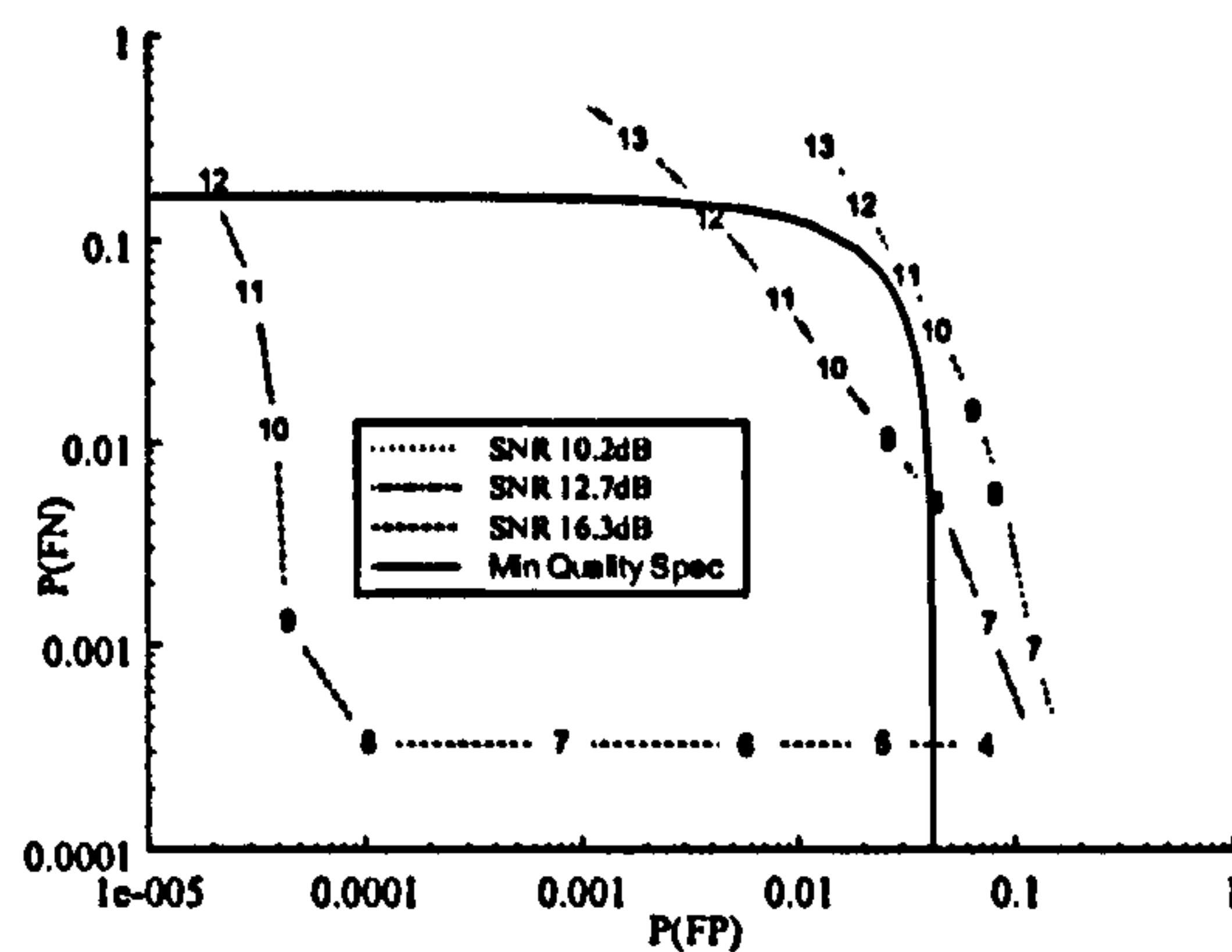


Figure 8 (b) ROC Curves Sobel Detector Edge Strength

Figure 8(a) plots the $P(FP)$ and the $P(FN)$ results given by equations (10) and (11) for the SUSAN detector on test images with SNR's of 5.38dB, 4.22dB and 3.19dB, while the edge strength threshold varied from '3' to '10' grey levels. The test edge strength values are inset on the SNR plots. Figure 8(b) plots the $P(FN)$ and the $P(FP)$ for the Sobel detector on test images with SNR's of 16.3dB, 12.7dB and 10.2dB, while the edge strength threshold varied from '4' to '13' grey levels. The ROC curves of Figure 8 incorporate the EPM zero condition that sets the minimum quality specification for the detector [8]. The unbroken trace that meets the $P(FN)$ axis at 0.167 and the $P(FP)$ axis at 0.042 denotes the systems minimum quality specification. A detector result that occurs within the area enclosed by the minimum quality specification and the $P(FN)$ and $P(FP)$ axis is then known to comply with the vision system specifications.

The SUSAN detector complies with the vision system specification at SNR's as low as 4dB. If the SUSAN edge strength tuning parameter is set to '6' then the error margin is maximised. In the Sobel detector the error margin is maximised by setting the edge strength parameter to '10'. At SNR's of 10dB or less the Sobel detector fails to comply with the vision system edge point specification. In Section 1, it was stated that the detector was required to meet the minimum quality specification at a SNR of 6dB or less. Thus the Sobel detector was unsuitable for use in the researched autonomous navigation problem. The ROC assessment can be repeated for the optimisation of other detector tuning parameters. These parameters include the low pass

spatial filters used to pre-process the image intensity profiles and the hysteresis limits for edge following procedures.

4 Metric Results and Comparisons

In order for the comparisons to be drawn between the EPM, Pratt and Kitchen Rosenfeld metrics the synthetic test image was populated with vertical bars, this was to comply with the limitations of the Kitchen Rosenfeld metric. The performance results generated by the metrics are illustrated in Figure 9 for the SUSAN detector with the threshold strength set to '6'. At high SNR levels all metrics register near unity results indicating a good detector performance. At an SNR of 8dB the metrics roll-off, the EPM value intersects the metric axis at 3.5dB. Thus the EPM assessment method establishes that the SUSAN detector complies with the specifications for the autonomous vision system. In contrast the Pratt and Kitchen-Rosenfeld metrics do not carry any information in respect of a systems specification.

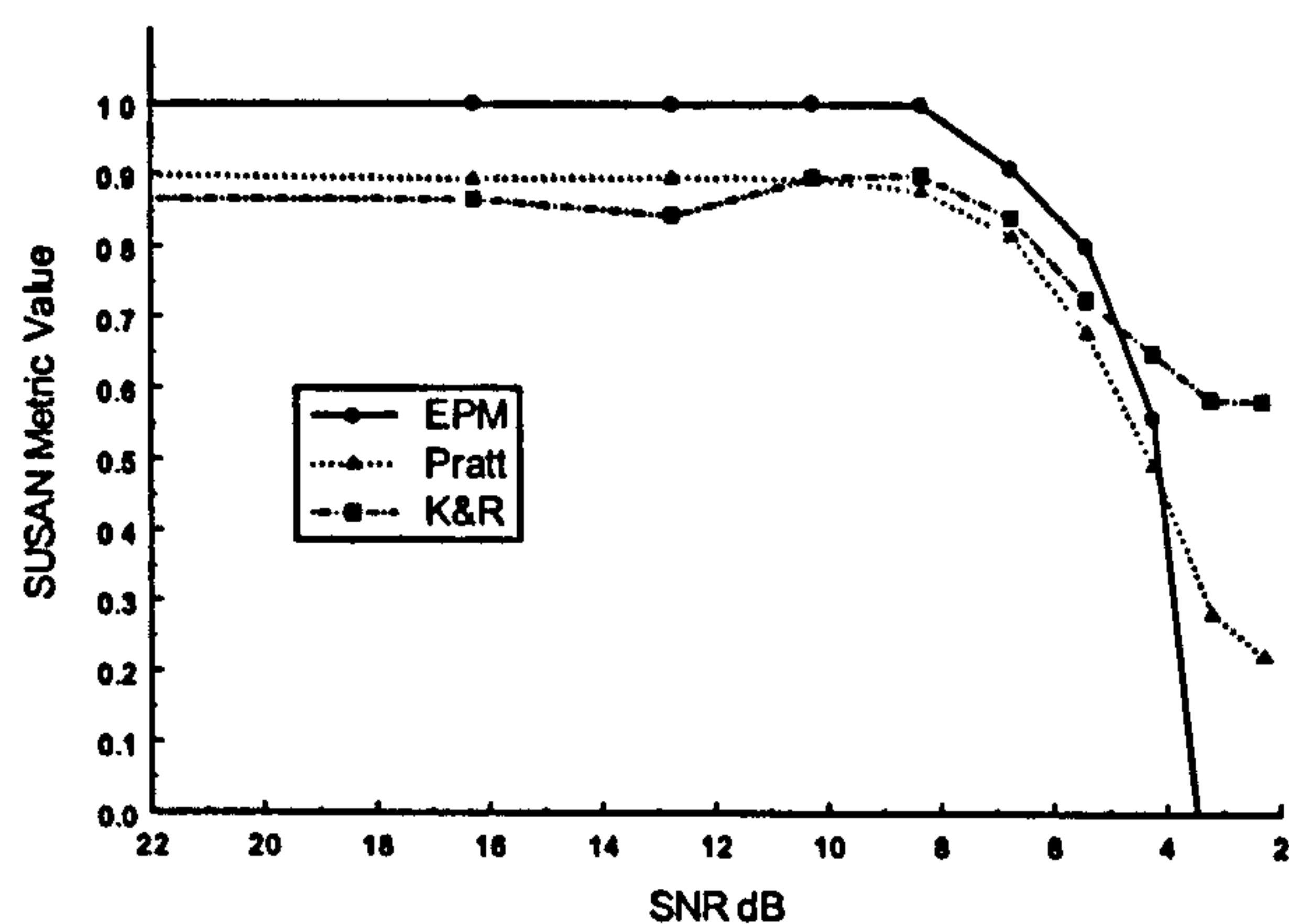


Figure 9 SUSAN Metric Results for the Vertical Bar Test Images

The EPM probability $P(WP)$ was found to vary from a minimum of 0.002 at a SNR of 22dB to a maximum of 0.09 at 2.3dB. The SUSAN detector has an integral edge thinning function that gives rise to this low level of width modulation. The EPM probability $P(DP)$ was found to register 0.5 ± 0.1 across the test range in the SUSAN detector results. This displacement probability indicates that 1-in-2 edge points are displaced in the detector results. These systematic displacement are limited to a maximum edge point shift of two pixels. For the autonomous navigation system that was the subject of our research the systematic displacement and line

broadening effects do not limit the system performance, thus it was appropriate to exclude these from the detector's merit value evaluation.

5 Conclusion

It was recognised by Förstner [8] that as a result of variation in system requirements the detector's minimum quality specification needs to be determined separately for each vision system. In the EPM metric we provide a frame work for the generation of this minimum quality specification. This method is dependent upon a ground-truth set existing for the systems test data. The DGC algorithm compared the detector and ground-truth results to classify the detector results true segmentation, distorted segmentation and segmentation errors. Four performance probabilities were identified within the detector's results. These were the probabilities of a false positive, a false negative, a displaced positive and a broadening positive result. The detector was then given a merit value based on a scaled linear summation of selected performance probabilities. This merit value ranged from unity to zero for a detector that complied with the system's specification. The scaling of this summation reflected the host system's allowed maximum frequency of occurrence of the selected performance measures.

The EPM figure of merit used in the development of the Figure 1 autonomous navigation system was given by a scaled summation of the probabilities of false positive and a false negative occurring within the detector's results. The scaling factors were set to give a figure of merit of zero when the frequency of occurrence of a false positive reached 1:24, or when the frequency of occurrence of a false negative reached 1:6. It was demonstrated that the zero merit rating could be incorporated into ROC curves to facilitate the choice of optimal detector parameters. It was shown that the EPM figure of merit would agree with the Pratt and Kitchen-Rosenfeld [1,4] metrics, as their figures of merit follow the same general curves for a given set of detector results.

References

- [1] Kitchen L, Rosenfeld A, *Edge Evaluation Using Local Edge Coherence*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. Smc-11, No 9, pp 597-605, September 1981
- [2] Khvorostov P V, Braun M and Poon C S. "Edge quality metric for arbitrary 2D edges", Optical Engineering 35(11), 3222-3226, Nov. 1996.
- [3] Peli M, Mala D, "A study of Edge Detection Algorithms", Computer Graphics and Image Processing, Vol 20, pp 1-21, 1982
- [4] Pratt W K, "Digital Image Processing" 2nd Ed, Wiley-Interscience, 1991.
- [5] Rosin P L, "Edges: saliency measures and automatic thresholding", Machine Vision and Applications, vol. 9, pp. 139-159, 1997.
- [6] Strickland R N. and Chang D, "An Adaptable Edge Quality Metric," Proceedings of SPIE, vol. 1360, pp. 982-995, Bellingham, WA, USA, 1990.
- [7] Venkatesh S, Kitchen L, "Edge Evaluation Using Necessary Components", CVGIP: Graphical Models and Image Processing, Vol 54, pp 23-30, January 1992.
- [8] Förstner W, "10 Pros and Cons Against Performance Characterisation of Vision Algorithms", Workshop on Performance Characteristics of Vision Algorithms, Robin College, Cambridge, April 19, 1996, Edited by: Christensen H I, Förstner W, Madsen C B
- [9] Nguyen T B, Ziou D, "Contextual and Non-Contextual Performance of Edge Detectors", Vision Interface '99, Trois-Rivières, Canada, pp 82-89, May 1999.
- [10] Kosaka A, Pan J , "Purdue experiments in model-based vision for hallway navigation," Proceedings of Workshop on Vision for Robots in IROS'95, pp.87-96, 1995.
- [11] Blaasvaer H, Pirjanian P, Christensen H I, "AMOR: An Autonomous Mobile Robot Navigation System", IEEE, Int. Conference on Systems, Man, and Cybernetics 1994, Vol. 3, pp. 2266-2271.
- [12] Beveridge J R, Riseman E, "Hybrid Weak-Perspective and Full-Perspective Matching", In IEEE CVPR. Champaign, IL, 1992, pages 432-438.

- [13] Smith S M, *SUSAN - a new approach to low level image processing*. Internal Technical Report TR95SMS1, Defence Research Agency, Chobham Lane, Chertsey, Surrey, UK, 1995. Available at www.fmrib.ox.ac.uk/~steve for downloading.
- [14] Dougherty S, Bowyer K W, "Objective Evaluation of Edge Detectors Using a Formally Defined Framework",