New Architecture Paradigms for Analog VLSI Chips

Ichiro Masaki, L. Richard Carley *, Steven Decker, Berthold K. P. Horn, Hae-Seung Lee, David A. Martin, Charles G. Sodini, John L. Wyatt, Jr.

Massachusetts Institute of Technology
*** Carnegie Mellon University**

Abstract — This paper proposes hypothetical guidelines for identifying applications which are more suitable for analog VLSI (Very Large Scale Integration) chip implementations rather than conventional digital approaches. Significant merits are realized when new architecture paradigms become available by replacing conventional digital circuits with analog ones. There are two major reasons to make new architectures possible: the smaller silicon area and different basic functions of analog circuitry. Even with conventional algorithms, the smaller silicon area of analog circuitry can convert off-line-type algorithms to on-the-fly-type algorithms in some applications. This aspect is discussed through a case study on array processors for stereo visual processing. The other reason for possible new architectures is that analog circuitry has different basic functions than digital systems. Some examples of the analog-based basic functions, such as a smoothing function with resistive grids, are discussed in this paper. The hypothetical guidelines proposed in this paper suggest that the analog VLSI implementation is suitable where the applications can take advantage of these new architecture paradigms.

Introduction

Analog VLSI vision chips are gathering increased interest. Readers will find an overview of this research field in [wyatt 92], [mead 89], and [koch 91]. The features of the analog VLSI chips include small silicon area, low power consumption, and low mass-production cost. Analog VLSI chip research is still in its infancy, and there are no guidelines so far to determine in what cases analog chips are more suitable than digital implementations. The purpose of this paper is to make preliminary comparisons between analog and digital vision chips by using a case study approach. Significant merits of analog VLSI chips are appreciated when analog implementations lead us to new system architecture paradigms which are different from conventional digital-based ones. These paradigm changes could happen because of two reasons. The first reason is that the silicon area sizes of analog implementations are smaller than comparable digital systems. This scale difference not only scales down conventional systems in size, but also opens a new horizon for different architectures which were impractical with conventional digital approaches. The second reason is the difference in basic arithmetic/logic functions. The basic digital functions include addition and subtraction which are based on bit operations. Analog systems have different basic functions which are directly based on the natural characteristics of resistors, capacitors, and transistors. Resistive grids, for example, generate a smoothing function naturally. The differences in basic functions between the analog and digital chips lead to different algorithms and different system architecture.

All the analog vision chips developed so far are dedicated to particular visual tasks and do not have any flexibility. The inflexibility, along with a long development time, was considered as a significant drawback of the analog vision chips. This issue is addressed in this paper by proposing a programmable hybrid analog/digital array processor chip. A digital memory unit of each processing element in the array processor chip is programmable and controls the data flow and the operation instruction at each processing element, while an arithmetic/logic unit of each processing element is significantly smaller than conventional digital implementation.

In this paper, we first consider possible architecture-paradigm changes caused by the silicon area size differences. Array processors for stereo vision are chosen as a case to study differences in silicon area sizes between analog and digital schemes for implementing conventional basic functions, such as addition, subtraction, multiplication, division, etc. Section 1 describes the stereo vision algorithm. Implementation of the stereo algorithm into programmable array processors is discussed in Section 2. Analog and digital circuitry for realizing the array processors is considered in Section 3. Section 4 addresses the first aspect of the comparisons — that is, the difference between the analog and digital implementations for conventional vision algorithms. It is necessary to consider various issues described in Sections 1 through 4 for comparing the analog and digital versions because the comparison results depend on all of those issues and cannot be independent from the total system configurations.

0-8186-6492-4\95 $4.00 © 1995 IEEE

The second aspect of the analog vs. digital comparisons is what new algorithms become available by using analog circuits instead of digital ones. This aspect is addressed by discussing some examples of new algorithms in Section 5. Section 6 is for some hypothetical guidelines deduced from the previous sections. Closing remarks are in Section 7.

1. Stereo vision algorithm

A stereo vision system measures distances to objects from differences of multiple images which are acquired by TV cameras located at different positions. A distant or near object is projected onto multiple camera image planes with a small or large disparity, respectively. The disparity is defined as a difference of object image positions relative to image frames among multiple images.

For the analog vs. digital comparison case study, the first case is defined as "a programmable processor for a trinocular stereo vision processing task." We choose "programmable" processors over "dedicated" application-specific processors to give a high degree of generality to the case study results. Stereo vision is chosen for two reasons: a large application field and a reasonable mix of early and intermediate vision tasks. The potential applications of stereo vision include three dimensional sensors for intelligent vehicles and robots. The adaptive cruise control system, for example, controls the vehicle speed automatically on highways depending on the traffic conditions in front. The automobile equipped with this controller cruises at a maximum speed set by the driver if the stereo vision system does not find any slow-going vehicles in the near front area. The vehicle speed is automatically reduced when necessary, based on the three dimensional positional information generated by the stereo vision system, to keep a safe distance to the vehicle in front.

In this paper, we classify various visual processing tasks into three categories: early-vision, intermediate-vision, and late-vision processing tasks. The early-vision process transforms a single image into another single image. Image smoothing and edge detection, as examples of the early-vision processing tasks, convert intensity images to smoothed intensity images and to edge images, respectively. The late-vision processing tasks take symbolic-format data both as the inputs and outputs. Anything between the early-vision and the late-vision are considered as the intermediate-vision. Examples of the intermediate-vision include stereo correlation for distance measurement and direct template image matching for locating and/or classifying objects. We limit the scope of this paper to tasks which have direct relationships with image-format data, although the application fields of analog VLSI chips are not limited to visual processing. The stereo vision algorithm which will be considered in this paper consists of two phases: edge calculation and stereo correlation phases.

Trinocular vision was chosen over binocular vision for the case study. The binocular system measures distances to objects from positional offsets, or disparities, of two images which are acquired by right and left TV cameras located separately. A significant problem of the binocular stereo vision algorithm is a correspondence problem. For example, in Figure 1.1, it is impossible to determine whether objects are located at P1 and P2 or P3 and P4 only from the right and left images. Algorithms to solve the correspondence problem are still research issues. The trinocular vision system reduces the correspondence problem in a straight way. It uses a third center camera for evaluating possible correspondences between the right and left images. With the example in Figure 1.1, the object positions are determined as P1 and P2 from the center image without any sophisticated algorithms. We choose a trinocular stereo system for the case study because this mature straight-forward algorithm allows us to concentrate on the implementation aspect.

Figure 1.2. shows our stereo algorithm. Three cameras are located on a single horizontal line. In the first step, positive vertical edges are calculated. The *positive* and *negative* edges are defined as edges at which the intensity level increases or decreases significantly from the left to the right, respectively. A relatively large kernel (7-pixel-high by 5-pixel-wide) eliminates the need for an image smoothing process which is required before the edge calculation process in many applications. The kernel is a vertical rectangle for high sensitivity to vertical straight edge lines. The edge strength values at pixel P(*) are defined as follows:

If "right" > "left"	positive_edge_sterength = "right" / "left"	(1.1)
Otherwise	positive_edge_strength = zero	(1.2)

where,
"right" and "left" are convolution values for Pixel P(*) with convolution kernels shown in Figures 1.3.

In the second part of Step 1, stereo correlation, the corresponding horizontal pixel rows in the three positive edge images are correlated with each other for all possible horizontal positional offsets. In this shift-and-correlate process,

the right and left images are shifted by "s" pixels horizontally to the right and left, respectively, while the center image stays at the same position. The pixel-level correlation values are defined as the sum of products of three corresponding edge strength values in the window as described below:

$$COR_p(n,m,s) = \sum_{w=-k}^{+k} L_p(n, m+s+w) \times C_p(n, m+w) \times R_p(n, m-s+w) \quad (1.3)$$

where $COR_p(n, m, s)$: Correlation value at Pixel(n,m) with Shift(s) with positive edge images
$L_p(n, m+s+w)$: Positive edge strength value at Pixel (n, m+s+w) in left image
$C_p(n, m+w)$: Positive edge strength value at Pixel (n, m+w) in center image
$R_p(n, m-s+w)$: Positive edge strength value at Pixel (n, m-s+w) in right image
K: Half of the window width around location C (n,m). Typical "K" value is 2 for a 5-pixel-wide and 1-pixel-high window.

All the calculated pixel-level correlation values are outputs from this system for post-processing. Post-processing includes the detection of optimal "s" in a sub-pixel mode by interpolating the pixel-level correlation values. Thresholding operations are also included in the post-processing to restrict attention to actual positive edge locating in the image and thereby to eliminate noise. Steps 3 in Figure 1.2 repeats Steps 1 for negative edges as follows:

If "left" > "right" negative_edge_strength = "left" / "right" (1.4)
Otherwise negative_edge_strength = zero (1.5)

$$COR_n(n,m,s) = \sum_{w=-k}^{+k} L_n(n, m+s+w) \times C_n(n, m+w) \times R_n(n, m-s+w) \quad (1.6)$$

where all the notations are similar to Equations (1.1) - (1.3) except that Equations (1.4) - (1.6) are for negative edge images.

Two depth maps are calculated independently in the post-processing: one from positive edge images and the other from negative ones. These two depth maps are integrated into a single map at the end of the post-processing. Practicality of the trinocular stereo vision concept described above is known in experiments within a reasonable depth range.

This task cannot be carried out by a single off-the-shelf microcomputer due to the large demand of computational power. For example, typical required specifications for adaptive cruise control systems for automobiles are described in Section 4. They require 150 million mathematical operations, such as additions and multiplications, per second. If the clock speed is 30 MHz and each mathematical operation, including data address control, takes two machine clocks on average, the speed of the single-microprocessor-based system turns out to be one order of magnitude slower than the required speed. It is not practical to use ten microprocessors to compensate for the slow speed, because automobile applications need low-cost, low-power-consumption, and physically-small systems. We will, therefore, consider developing array processors in the following sections to meet the application requirements.

2. Implementation of stereo vision with array processor

This section describes implementation of the stereo algorithm, including both digital and analog implementations, by using array processor architecture. A total system consists of the following three stages:

(1) image acquisition stage
(2) edge calculation stage, and
(3) stereo correlation stage

In the first stage, the right, center, and left images are acquired simultaneously and are stored in on-chip shift registers. The intensity image data are sent to the second stage for edge calculations. The right, center, and left edge images are transferred to the stereo correlation stage with some time offsets which generate necessary positional horizontal shifts among three edge images. The stereo correlation values with various shifting values are calculated in the third stage. This total system configuration is considered as an on-the-fly type because the data goes through the system from the input port to the output port without being stored in a tentative value memory unit. A typical counter system configuration is called an off-line type, in which the data are transferred between the processing units and the tentative memory units several times.

With this example case, we assume that identical array processor chips are used both for the edge calculation and stereo correlation with different programs. The array processors which we consider in this paper are MIMD (Multi-Instruction Multi-Data) type, where each processing element carries out a different instruction independently. SIMD (Single-Instruction Multi-Data) machines require less silicon areas, but cannot deliver enough flexibility required for the stereo application.

As described in Figure 1.3 and Equations (1.1) and (1.2), the edge detection operation consists of the following three procedures: calculation of "right", calculation of "left", and positive edge strength calculation represented by Equations (1.1) or (1.2). All these three procedures are carried out in the (19 x 6) array described in Figure 2.2. The top and bottom rows of the combined kernel shown in Figure 1.3 include five weights (1, 2, 0, 2, 1), respectively. These five weights are mapped on the fourth row from the top in the array processor shown in Figure 2.2. The weights of "1" and "0" are omitted and, therefore, the first, third, and fifth columns in the fourth row in Figure 2.2 are left blank. The second and fourth column processing elements store the weights of "2" in their analog data memory units. Circles in Figure 2.2 represent analog memory units. Each processing unit has a single analog memory unit, and the units which are not used are omitted in the figure. The top row in Figure 2.1 includes five intensity values. These values are stored in five analog memory units in the third row in Figure 2.2. The multiplication of Weight (-3, -1) in Figure 1.3 and Pixel (-3, -1) in Figure 2.1 is carried out by the arithmetic unit of Processing Element (3, 2) in Figure 2.2. The locations are represented in a form of (row, column), and the top left corner has the smallest row and column values. The top left location in Figures 1.3 and 2.1 is (-3, -2) to make the center location of the 7-by-5 region (0, 0). In Figure 2.2, the top left location is (1, 1) and the bottom right location is (19, 6). The addition of Pixel (-3, -2) multiplied by Weight (-3, -2) and Pixel (-3, -1) multiplied by Weight (-3, -1) is carried out by Processing Element (2, 2). In Figure 2.2, square boxes represent arithmetic units. Processing Elements (2, 2) and (3, 2) have an addition symbol and a multiplication symbol in the boxes because these elements carry out addition and multiplication, respectively. The bottom row, Row (+3), in Figure 2.1 is stored in the fifth row from the top in Figure 2.2. In a similar fashion, the weights at Rows (-2) and (+2) in Figure 1.3 are stored in the 9th row in Figure 2.2. The "right" and "left" values are output from Processing Elements (2, 6) and (2, 3), respectively. Equations (1.1) and (1.2) are carried out by Processing Element (1, 6) which has a "/" symbol in the box.

The stereo correlation requires (7 x 150) processing elements where the window range is +/- 2 pixels and the minimum overlap width among three images is 150 pixels. Figure 2.3 shows a simplified stereo correlation with the array processor. The right, center, and left edge images are stored in analog memory units in the first, second, and third row of the array processor shown in Figure 2.3. These edge images are either positive or negative edges, and we assume that the right and left images are already shifted properly. Five multiplication values are output from five processing elements in the third row from the top in Figure 2.3. Each output represents the product of three corresponding pixels in the right, center, and left images. The window size is assumed to be +/- 1 for a 1-by-3 window in this figure. Processing Elements (4, 3), (5, 4), and (6, 5) output the correlation values in the window for three different pixel positions in the center image, where Processing Element (n, m) indicates the element in the n-th row from the top and m-th column from the left.

Each processing element consists of the following four units as shown in Figure 2.4.

(1) Arithmetic unit
(2) Data flow control switching network
(3) Data storage
(4) Program memory
(5) Sample and hold

The arithmetic unit has ten functions listed in Table 2.1.

Table 2.1. List of Functions

Numbers	Functions		
1	C	= A + B	if (A+B)<Max.
		= Max.	otherwise
2	C	= (A + B) / 2	if ((A+B)/2) < Max.
		= Max.	otherwise
3	C	= A - B	if A>B
		= 0	otherwise
4	C	= (A / Max.) x (B / Max.) x Max.	
5	C	= (A / Max.) x (B / Max.) x 10 x Max.	
			if {(A / Max.) x (B / Max.) x 10} < 1
		= Max.	otherwise
6	C	= (A / Max.) x (B / Max.) x 100 x Max.	
			if {(A / Max.) x (B / Max.) x 100} < 1
		= Max.	otherwise
7	C	= A / B	if (A / B) < Max.
		= Max.	otherwise
8	C	= (A / B) x 10	if ((A / B) x 10) < Max.
		= Max.	otherwise
9	C	= (A / B) x 100	if ((A / B) x 100) < Max.
		= Max.	otherwise
10	C	= Max.	if A > B
		= 0	otherwise

All the functions are carried out in a fixed point fashion, and several functions, such as Functions # 2, 5, 6, 8, and 9, include scaling operations. The data flow control switching unit is connected to six bi-directional data paths with four neighboring processing elements: north, east, west, and south processing elements. The unit also has two input paths: one from the arithmetic unit and the other from the data storage. Two output paths go to the arithmetic unit. The directional consistency is only one restriction on the data flow; input paths can be connected to output paths and/or bi-directional paths, but not to any input paths. One path can be connected to multiple paths. The data storage unit stores one analog data and the contents of the unit can be shifted from the left to the right with dedicated clock signals. Some of the data storage units are used for storing intensity pixel data, weight values for edge detection, or edge values for stereo matching. The program memory unit stores two kinds of control signals: arithmetic-control signals to chose the functions to be carried out by the arithmetic unit and data-flow-control signals to set up the data flow control switching network. The sample and hold function is included in the arithmetic unit for pipeline operations.

3. Hybrid and digital implementations of array processor

We are developing two versions for the array processor specified in the previous section: a hybrid analog/digital version and a digital version. The hybrid analog/digital version consists of both analog and digital circuits. All the arithmetic operations are carried out in an analog current form. The calculated values are transferred from processing elements to other elements in an analog form as CMOS transistor gate voltage. The digital memory unit controls the data flow network switches and also choose an appropriate operation for each arithmetic unit. The approximate parameters, obtained from our ongoing design effort, are shown in Table 3.1. Although the feasibility study of the implementation is still underway, this concept design is used for a case study in this paper.

Table 3.1. Approximate Parameters of Hybrid Analog/Digital Array Processor Chip

Chip size	10mm x 10mm
Fabrication Technology	2 micron
Size of each processing element	10^5 square micron
Number of processing element in single chip	10^3 (32x32)
Clock speed	10 MHz
Accuracy	+/- 1%

Similar array processor architecture can be implemented using digital circuits. Possible implementation formats include bit-serial and bit-parallel schemes. Although the bit-serial architecture is slower compared to the bit-parallel system, it meets the required speed specified by the adaptive cruise application for automobiles. A problem with the bit-serial architecture, however, is its large memory requirement for the micro-program. Division and multiplication operations especially require large memory capacities. It is not a problem if the array processor chip works in a SIMD fashion, because we need only a single micro-program memory for all the processing elements included in the chip and the micro-instructions can be broadcasted. The array processor chip which is being considered in this paper, however, operates in a MIMD fashion, because the SIMD scheme is not flexible enough for this type of application. The MIMD scheme requires micro-program memory for each processing element, and therefore it is not practical to implement the MIMD concept using a bit-serial scheme for automotive applications which require low cost, low power consumption, and low mass-production cost. The approximate parameters which we are considering for the bit-parallel digital MIMD array processor chip are shown in Table 3.2, and the comparisons between the analog and digital implementations are discussed in Section 4.

Table 3.2. Approximate Parameters of Digital Array Processor Chip

Chip size	10mm x 10mm
Fabrication Technology	1.2 micron
Size of each processing element	10^6 square micron
Number of processing element in single chip	10^2 (10x10)
Clock speed	100 MHz
Accuracy	+/- 1% (6 bits)

4. Analog vs. digital comparisons

This section compares the analog and digital VLSI implementations described in the previous section. In Section 3, we assumed different fabrication technologies, 2 micron for the analog version and 1.2 micron for the digital version, because usually more advanced technologies are available for digital VLSI chips. The difference in available technologies, however, is mainly caused by market incentives not by technical reasons. The digital market is much larger and therefore the fabrication technology for digital is advanced faster. We accept this technology difference in this paper, although it may be better to assume the same technology for long term comparisons of analog vs. digital.

Section 3 showed that the analog implementation is one order of magnitude smaller, even with the less advanced fabrication technology, while its clock speed is one order of magnitude slower compared to the digital implementation. Now the issue is what would be the criteria to compare analog vs. digital VLSI implementations. The comparisons would be easy if we could use criteria which are independent from applications. For example, the product values of the speed and the silicon area are independent from the applications. The comparison results, however, depend on applications heavily, and therefore application-independent criteria cannot be used. Lets assume the application specifications listed in Table 4.1 for more detailed comparisons. These specifications were developed with the adaptive cruise control applications for automobiles in mind [masaki 93], [masaki 91].

Table 4.1. Application Specifications

Image Size	250-pixel-wide x 125-pixel-high
Shifting Range	50-pixel
Required Frame Rate	10 Hz

If we take the on-the-fly configuration shown in Figure 2.1, the number of required processing elements is independent from the speed of each processing element as long as their speed is high enough for the application. In other words, the digital implementation is one order of magnitude larger because the size of the digital processing element is one order of magnitude larger compared to the analog implementation. The speed advantage of the digital version does not affect the comparison results and it only makes idling time longer.

In order to take advantage of the high speed of the digital VLSI implementation, the whole processing task must be divided into ten pieces and these pieces can be carried out serially in an off-line fashion. With this system configuration, the number of processing elements is reduced by one order of magnitude and the product of the number of processing elements and the processing speed of each processing element becomes equal to that of the analog version. This digital system should deliver the same performance compared to the analog system if the other conditions stay the same. The other conditions, however, do not stay the same because the off-line-type scheme causes three problems described below and the cost/performance of the digital system turns out to be inferior to that of the analog VLSI systems. The problems with the off-line-type scheme are

(1) necessity of storing tentative values
(2) bottlenecks between processors and tentative value memories, and
(3) overhead of control including data addressing.

Lets consider more details of each issue. In the edge calculation stage, for example, we need an additional memory unit with the digital implementation for tentatively storing the intensity image data because the edge cannot be calculated in an on-the-fly fashion as can be done with the analog system. In addition, a tentative value storage memory unit is necessary with the digital system because the number of processing elements is not large enough for carrying out the whole convolution with the 7x5 kernel without dividing it into several processing tasks. These image memory and tentative value memory units alone need 825 K bits. These memory units are too big to be implemented in the same chip with the processors, and therefore the bottleneck problem arises as the second issue. The speed of 100MHz in Table 3.2 is for the internal communication within the chip and the inter-chip communication speed is significantly slower. The third issue is the control overhead. The data addressing, for example, becomes more complex when we divide the task into small pieces. Additional overhead makes the system larger and slower. In this particular example, therefore, analog VLSI implementation should be more efficient than the digital scheme because of the difference in the total system architecture. The on-the-fly architecture with the analog implementation makes a significant difference in terms of the cost/performance compared to the off-line-type architecture for digital chips.

On the other hand, the analog VLSI implementation has its limitations. A common limitation of analog chips is their accuracy. The accuracy of the analog systems is limited mainly by transistor mismatches if we assume conventional fabrication technologies which are being used for digital chips. We do not consider expensive technologies, such as individual laser trimming, in this paper because our current major interests are low-cost systems to be mass-produced. There are two ways to avoid the accuracy problem. With the first way, we can consider various application techniques which do not require high accuracy for vision processors. Two examples are discussed below:

(1) Since the major cause of errors with the analog systems comes from transistor mismatches, we get less error among operations which are carried out by the same transistor. For example, Figure 4.1 shows a simplified binocular stereo matching system. Three pixels, #n through #(n+2), in the left image are correlated to the right image with disparities of 0 through 2 pixels. The best disparity for the left image pixel #n is detected by finding the peak value at Processing Cell #n. In other words, the accuracy of this particular application depends only on the accuracy within each processing cell. The errors among different cells, caused by transistor mismatches, do not influence the results at all. This is an example of increasing the "effective" accuracy of the analog systems by mapping the required operations on the processor hardware appropriately.

(2) The required accuracy of the vision processor depends heavily on applications. Lets consider off-line and on-line systems for vision-based robots as an example. The off-line vision-based robot measures the location of the stationary object only once, in an off-line fashion, before moving the robot arm to grab the object. The location of the object is sent to the robot-arm-controller and the arm accesses the object without any on-line visual feedback. The on-line vision-based robot, in contrast, moves the arm to the object by measuring the positional relationship between the arm and the object continuously. With this example, the vision system for on-line control requires higher processing speed but less accuracy compared to the off-line scheme. It is necessary to consider the required accuracy for the vision processor from a total system point of view.

The other way to address the limited accuracy of analog systems is to increase the system accuracy by extending the system architecture to a multi-valued discrete scheme which consists of multiple accuracy-layers. It was reported [kawahito 88], for example, that the performance of a 32x32-bit multiplier using multi-valued logic is comparable to that of the fastest binary multiplier.

5. New architecture paradigms

In the above sections, we assumed conventional algorithms for comparing analog and digital VLSI implementations. The algorithm developments depend, unconsciously or consciously, on the basic arithmetic/logic functions available with particular hardware in mind. The basic functions with the digital systems include addition and subtraction based on logical bit operations. The basic functions of analog systems are different because they are based on the behavior characteristics of resistors, capacitors, and transistors. The differences in the basic functions naturally lead to differences in algorithms and system architecture in many cases. This section will discuss some examples of new architecture paradigms which become available because of the basic functions of the analog VLSI systems.

5.1 Resistive grids

A well known basic function with analog VLSI chips is a two-dimensional smoothing function using resistive grids. With a simplified one-dimensional model in Figure 5.1, for example, the original intensity data at pixels are given as D(n-1), D(n), D(n+1), etc. The smoothed data are given as V(n-1), V(n), V(n+1), etc. The smoothing function width depends on approximately $(1 / \sqrt{RG})$ where G is the data conductance and R is the network resistance. This basic function is useful for various visual tasks, such as image smoothing, edge detection, optical flow, and binocular stereo tasks [bair 91], [knight 83].

The "silicon retina" analog chip, for example, resembles a Laplacian filter function by calculating differences between the smoothed intensity image and the original intensity image at every pixel for calculating spatial and/or temporal derivatives of the original intensity image. The smoothed intensity image is calculated by resistive grids [mead 89]. Another application of the resistive grids is an analog VLSI chip which detects the position and orientation of a single object on a uniform background. The chip calculates the first and second moments of the object's spatial intensity distribution using uniform and quadratic lines connected to two-dimensional resistive grids. The centroid, which indicates the position, is given by the normalized first moment, and the axis of least inertia derived from the moment values indicates the orientation of the object. Another feature of this chip is that it takes a continuous-time processing style without clock signals while conventional digital chips take a discrete-time scheme. The operation of this chip averages the currents, through many portions of the grid, reducing the errors to well below the mismatch between individual resistors. In experimental tests, this chip determined the angle of a nearly round object within a worst-case error of +/- 2 degrees out of a 360-degree rotation, and could operate at the remarkable speed of 5,000 frames/second [standley 91].

The resistive grid scheme can be generalized by replacing the horizontal resistors by a small network of resistors and constraint boxes to perform sensor fusion tasks [harris 91], [seidel 94]. In general, linear reciprocal resistive networks have behavior that automatically satisfies Maxwell's minimum heat theorem. Nonlinear reciprocal networks satisfy a more general extremum principle by minimizing network content or cocontent. Thus resistive networks are a natural implementation of vision algorithms that are posed in terms of minimization of penalty function [wyatt 92], [harris 89], [yu 92].

With these applications, sometimes both the image acquisition part and the processing part can be integrated into a single chip. This integration is called focal-plane processing and will be discussed in more details in sub-section 5.4. Practical implementations of resistive grids can also be realized by using switched capacitors or charge buckets. The next sub-section describes some examples of usage of the charge buckets [seidel 94], [umminger 92].

5.2 Charge splitting and mixing

With charge splitting and mixing approaches, amounts of electric charge trapped in potential wells represent values. The potential wells are formed using CCD (Charge Coupled Device) technology. Operations among the charge values are carried out by splitting the charge from a single well into multiple wells and/or merging charge from multiple wells into a single well.

Figure 5.2 shows image smoothing/segmentation operations with nine original intensity pixel data stored in nine CCD wells that serve as pixel-nodes. The pixel-nodes are connected via a series of eleven charge buckets. The charge values at the pixel-nodes are averaged through charge splitting and mixing operations. The segmentation circuits prohibit smoothing where the intensity difference between neighboring pixels exceeds the threshold value in order to preserve sharp edges [keast 92].

5.3 Dynamic control within image integration period

With conventional CCD cameras, electric charge is accumulated at a constant rate, for static images, at each pixel during the accumulation time. Conventional digital vision systems take accumulated charge values, after the completion of the accumulation, as the input data which represent image intensity, and do not control the accumulation process itself. A dynamic sensitivity imaging chip limits the maximum charge in each well by draining excessive charge over a time-dependent barrier which is installed between the common drain and each pixel well. The total charge in each well cannot exceed the height of the barrier at any time during the accumulation period. The barrier goes higher gradually in each accumulation period as shown in Figure 5.3. This accumulation-control suppresses the imager sensitivity for bright light and extends the imager's intensity dynamic range. These type of functions become available *only* through analog implementations since they necessary occur prior to A/D conversion [decker 93].

5.4 Focal plane processing

Focal-plane processing schemes involve integration of an image acquisition function and an image processing function into a single chip. They are normally available only with analog processing systems because they avoid expending area for A/D conversion and because the analog processors are physically small. The small silicon size is realized because of two factors: analog systems do not need to convert analog sensor outputs into digital signals, and the analog processors themselves are smaller than digital systems. The focal-plane processing opens a new horizon for different architecture paradigms. A dynamic intensity imaging chip in the previous sub-section is an example. Resistive-grids-based systems in Sub-section 5.1 can be implemented with the focal-plane processing scheme if the reduced fill factor of the image acquisition cells is acceptable.

Another example is an analog depth-sensor chip which carries out an optical slit algorithm for three dimensional visual measurements [gruss 91]. With this system, a light projector emits a slit pattern over the object area. The slit line is deformed by the three dimensional shape of the surface in the object area. This deformation is observed by an imaging device, and the three dimensional surface is reconstructed using triangulation. With a conventional paradigm, multiple images are acquired with different optical slit line positions. The new system measures the timing of the peak of the reflected light at each pixel while the optical slit is scanning the object area. The peak timing is measured by analog circuits which are integrated with each pixel. This scheme became possible by reducing the silicon area size for the peak-timing-detection unit by using analog circuits instead of digital ones. It is robust to transistor mismatches between cells because each peak timing is measured within a single cell without interfering with other cells. In experimental tests this system has measured scene depth to about 0.1% accuracy (+/- 0.5 mm at a depth of 0.5 m) operating at a speed of 1,000 frames/second [gruss 91].

6. Guidelines for selecting analog or digital implementations

The major theme of this paper is "In what cases do analog VLSI implementations deliver better performances than digital systems?" We chose a case study approach because it is difficult to answer this question generally. It is, however, worthwhile to propose hypothetical guidelines in this section. The analog VLSI implementations can deliver better cost/performance when they can make architecture paradigm changes. The paradigm changes can happen because of the following three reasons: (1) small silicon area, (2) different basic arithmetic/logic functions, and (3) flexible A/D conversion point. Each of them is discussed below in more detail:

6.1 Small silicon area

The case study on the adder, subtractor, multiplier, divider, and data flow controller in Section 3 suggests that analog implementations require less silicon area than the digital versions for many functions. This hypothesis is for both wiring areas and processing cell areas in VLSI chips. Less silicon area implies less power consumption and less mass-production costs. The small silicon area of analog circuitry can make architecture paradigm changes which lead us to significant cost/performance improvements.

The MIMD array processor discussed in Section 3, for example, became practical for small scale applications only with the analog VLSI implementation. Massively parallel MIMD array processors were conventionally limited only to large scale applications because their system sizes were large with digital implementations. For smaller scale applications, digital SIMD array processors with bit serial operations were more popular because of their smaller silicon sizes. Application fields for MIMD and SIMD machines are quite different, and it is impossible to emulate the MIMD capabilities practically with SIMD systems. A feature of small silicon area associated with analog implementations extends the application areas of massively parallel MIMD processors to smaller scale systems. In many cases, n units of processing elements do not simply increase the processing capability of a single processing element by a factor of n. Sometimes a high degree of parallelism makes possible architecture paradigm changes. With the example in Section 3, a high degree of parallelism made on-the-fly-type processing available and eliminated tentative value memory and complex control. These eliminations contribute in solving the bottleneck problems between processors and memories. It is necessary to consider bottlenecks in both the analog-based and digital-based architecture from a total system point of view. The first hypothetical guideline is to compare possible total system architectures for analog- and digital-implementation in identifying appropriate implementation formats for particular applications.

6.2 Different basic functions

Basic operational functions with digital systems include addition and subtraction based on bit operations. The basic arithmetic/logic operations with analog VLSI implementations, in contrast, are based on the natural characteristics of analog components, such as resistors, capacitors, and transistors. For example, the logarithmic current-to-voltage relationship of bipolar and subthreshold MOS transistors combined with the obvious exponential voltage-to-current relationship, converts multiplication and division to addition and subtraction which can be performed directly on voltage or currents [gilbert 76]. Even trigonometric functions can be generated directly from a relatively small number of transistors [gilbert 82].

Section 5 discussed some examples of new architecture paradigms which are practical only with analog VLSI implementations. When an application is given, it is important to think about appropriate algorithms both for analog and digital implementations based on the available basic functions with the analog and digital VLSI implementations. The different basic functions may lead to different algorithms, and the difference in algorithms may cause significant difference in cost/performance. The second hypothetical guideline is to go back to the algorithm level for comparing analog and digital implementations. It is not reasonable to assume the identical architecture for different implementation forms which have different basic functions.

6.3 Flexible A/D conversion point

In many cases, sensors provide analog signals to digital processors and analog-to-digital conversion is carried out as a part of the sensor-to-processor interface. Analog processors can move this conversion point to a later stage to reduce the bandwidth for the A/D conversion and subsequent processing. Sometimes the conversion can be eliminated when the processors are connected to analog actuators.

With the optical slit system in Section 5.4, analog makes sensor-and-processor integration possible because the analog-to-digital conversion point was moved to a later stage and the size of the peak timing detection unit was reduced by using analog circuits. In this case, rather than carrying out sample and A/D conversion operations roughly 1,000 times for each sensor during a sweep of the slit, the analog VLSI implementation finds the slit based directly on the analog sensor output and A/D converts only the time of the peak.

The integration of sensors and processors does not simply mean that the sensors and the processors are laid out in the same chip. The integration has a potential of opening a new horizon for different algorithms. The third hypothetical guideline is to consider where to optimally place the analog-to-digital conversion point from a total system point of view.

7. Closing remarks

Research on analog VLSI vision chips started, in its history, with dedicated ASIC (Application Specific Integrated Circuits) chips such as the systems discussed in Section 5. In the first stage of the research, which was devoted to ASICs, two possible drawbacks were considered for analog VLSI chips: inflexibility and limited accuracy. In the following research stage, more flexible programmable machines, such as the hybrid analog/digital array processor, are being developed as discussed in Section 4. Now the remaining possible restriction for analog VLSI chips is their limited accuracy. Section 4 includes some ideas to get around this limitation.

Most significant merits from analog VLSI chip approaches are expected when new system architecture paradigms become available by replacing conventional digital implementations with analog ones. Some possible architecture paradigms were discussed in Section 5. Although we took visual processing tasks as the cases in this paper, potential application fields for analog VLSI chips will not be limited to the vision applications. An example of other application fields include connectionist algorithms. Neural nets type algorithms, for example, match to massively parallel processing, and small silicon size of analog processing elements would be suitable for massively parallel machines. As another potential application field, analog VLSI approaches may be useful for systems which include analog sensors. With these applications, analog data processing may be able to move or eliminate the analog-to-digital conversion points.

Acknowledgments

This work has been supported by NSF and ARPA under Contract No. MIP 9117724. Special thanks to all the members of the analog VLSI systems group at Massachusetts Institute of Technology.

References

[bair 91] W. Bair and C. Koch, "Real-Time Motion detection Using Analog VLSI Zero-Crossing Chip," *Proc. Visual Information Processing: From Neurons to Chips,* SPIE - The Int'l Soc. Optical Engineering, Bellingham, WA, 1991, pp. 59-65.

[decker 93] Personal correspondence with Steven Decker

[gilbert 76] B.Gilbert, "New Analog Multiplier Opens Way to Powerful Function Synthesis," *Microelectronics,* Vol. 8, No. 1, 1976, pp. 26-36.

[gilbert 82] B. Gilbert, "A Monolithic Microsystems for Analog Synthesis of Trigonometric Functions and Their Inverses," *IEEE J. Solid-State Circuits,* Vol. SC-17, No. 6, Dec. 1982, pp. 1179-1191.

[gruss 91] A. Gruss, L.R. Carley, and T.Kanade, "Integrated Sensor and Rangefinding Analog Signal Processor," *IEEE J. Solid-State Circuits,* Vol. SC-26, No. 3, Mar. 1991, pp. 184-191.

[harris 89] J.G. Harris, et al., "Resitive Fuses: Analog Hardware for Detecting Discontinuities in Early Vision," in *Analog VLSI Implementations of Neural Systems*, Kluwer, Norwell, Mass., 1989.

[harris 91] J.G. Harris, "Analog Models for Early Vision," PhD thesis, California Institute of Technology, Pasadena, Calif., 1991.

[kawahito 88] S. Kawahito, et al., "A 32x32-bit Multiplier Using Multiple-Valued MOS Current-Mode Circuits," *IEEE J. Solid-State Circuits,* Feb. 1988, pp. 124-132.

[keast 92] C.L. Keast and C.G. Sodini, "An Integrated Image Acquisition, Smoothing and Segmentation Focal Plane Processor," *VLSI Circuit Symposium,* 1992.

[knight 83] T.F. Knight, "Design of an Integrated Optical Sensor with On-Chip Preprocessing," PhD. thesis, MIT, 1983.

[koch 91] C. Koch, Implementing Early Vision Algorithms in Analog Hardware," *Proc. Visual Information Processing: From Neurons to Chips,* SPIE - The Int'l Soc. for Optical Engineering, Bellingham, WA, 1991, pp. 2-16.

[masaki 91] I. Masaki, ed., *Vision-based Vehicle Guidance*, Springer-Verlag, New York, N.Y., 1991.

[masaki 93] I. Masaki, ed., *Proc. IEEE Int'l Symposium on Intelligent Vehicles*, 1993.

[mead 89] C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, Mass., 1989.

[seidel 94] M.N. Seidel, "Switched-Capacitor networks for Image Processing: Analysis, Synthesis, Response-Bounding, and Implementation," Sc.D. thesis, MIT, 1994.

[standley 91] D. Standley and B.K.P. Horn, *Proc. Visual Information Processing: From Neurons to Chips,* SPIE - The Int'l Soc. for Optical Engineering, Bellingham, WA, 1991, pp. 194-201.

[umminger 92] C.B. Umminger and C.G. Sodini, "Switched Capacitor Networks for Focal Plane Processing Systems," *IEEE Trans. Circuits and Systems for Video Technology,* Vol. 2, No. 4, Dec. 1992, pp. 392-400.

[wyatt 92] J.L. Wyatt, Jr., et al., Analog VLSI Systems for Image Acquisition and Fast Early Vision Processing," *Int'l J. Computer Vision,* Vol. 8, No. 3, 1992, pp. 217-230.

[yu 92] P.C. Yu, et al., "CMOS Resistive Fuse for Image Smoothing and Segmentation," *IEEE J. Solid-State Circuits,* Vol. 27, No. 4, Apr. 1992.

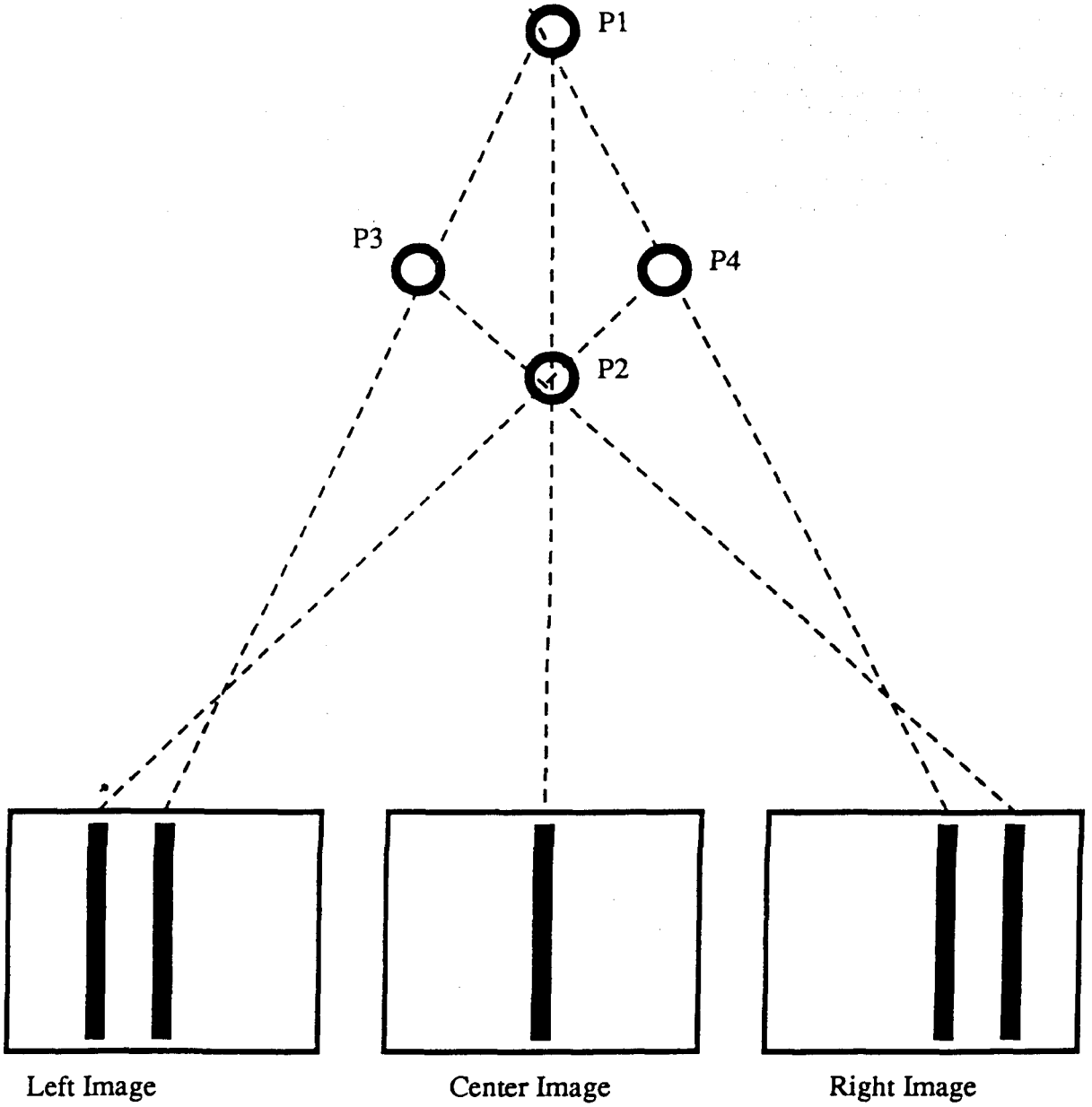


Figure 1.1 Correspondence problem resolved by trinocular imaging

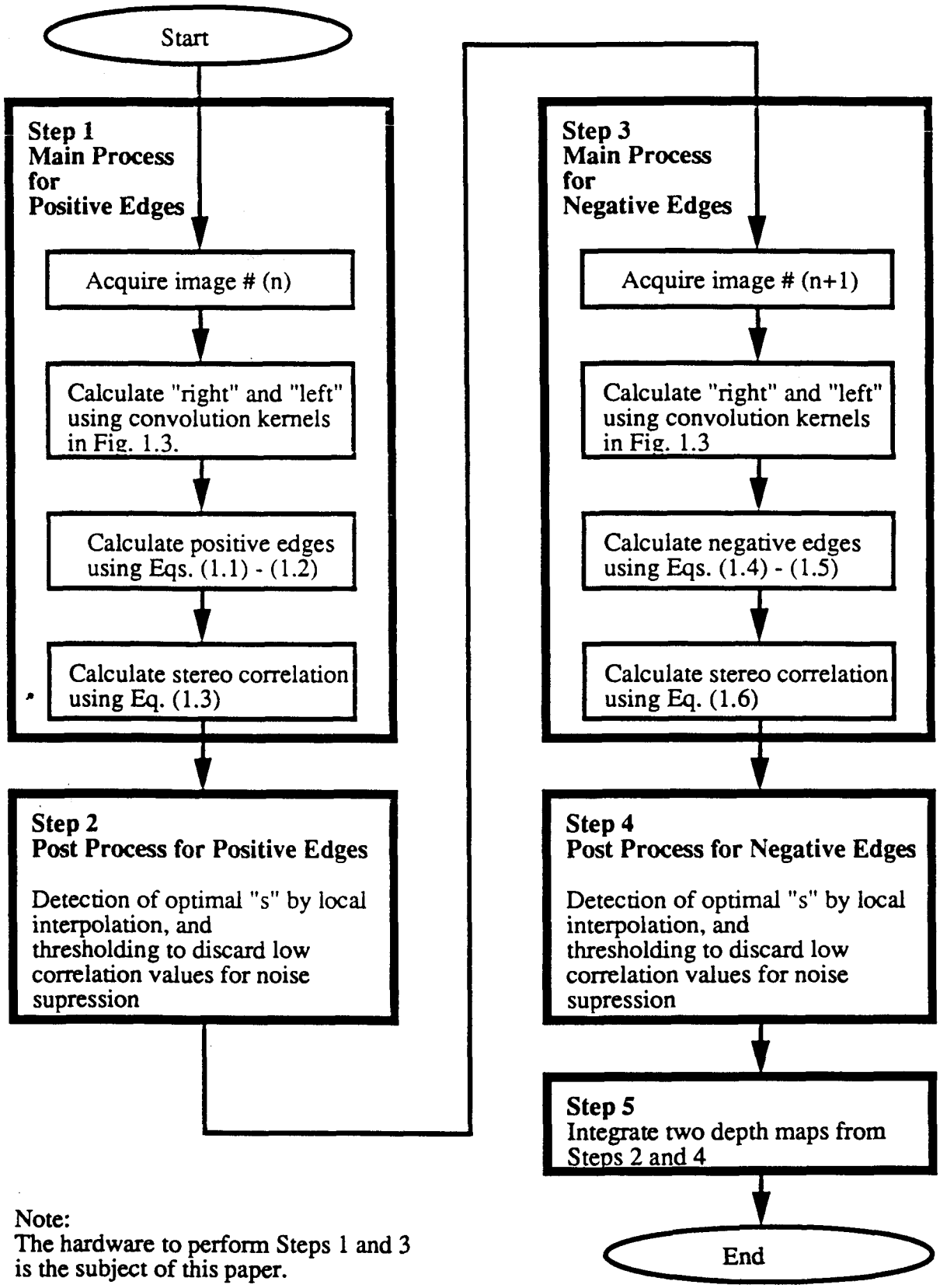


Figure 1. 2 Stereo algorithm

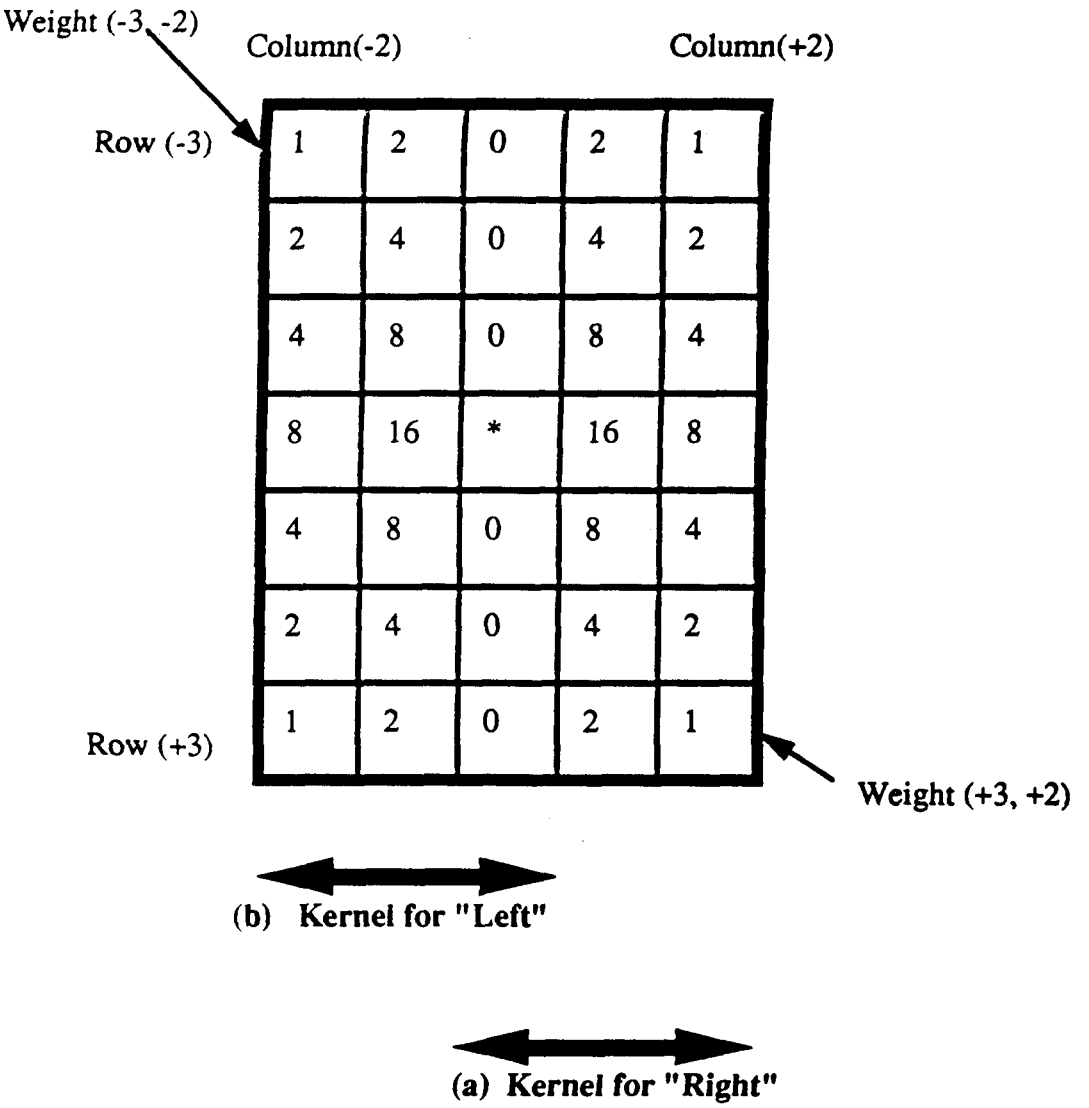


Figure 1. 3 Kernels for Edge Detection

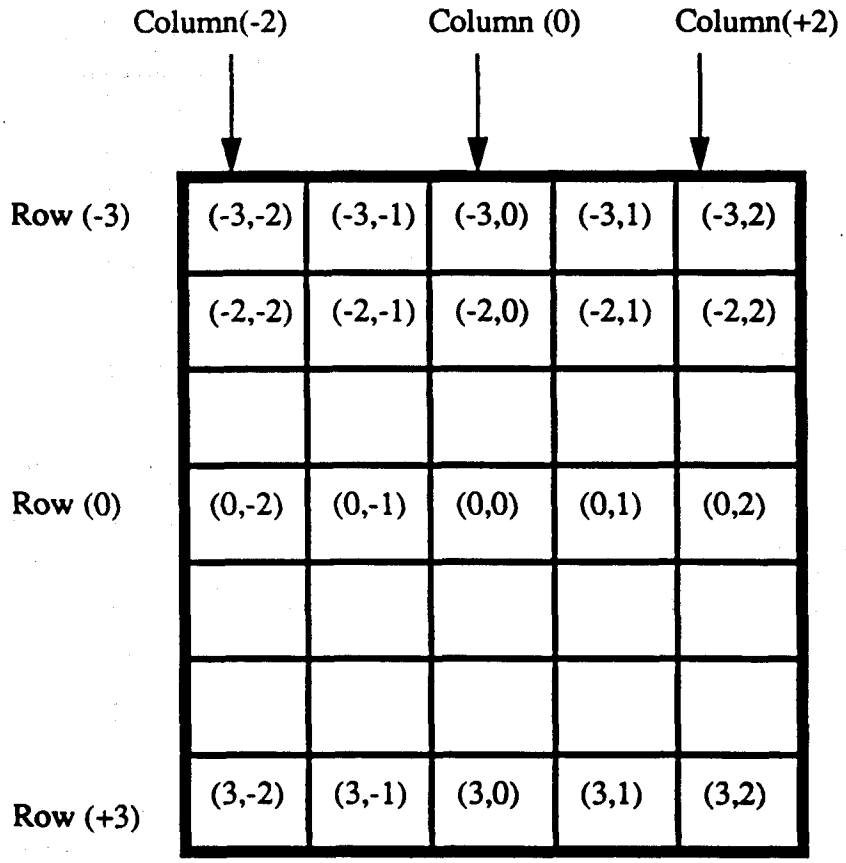


Figure 2.1 Indexed Portion of Intensity Image for Edge Detection

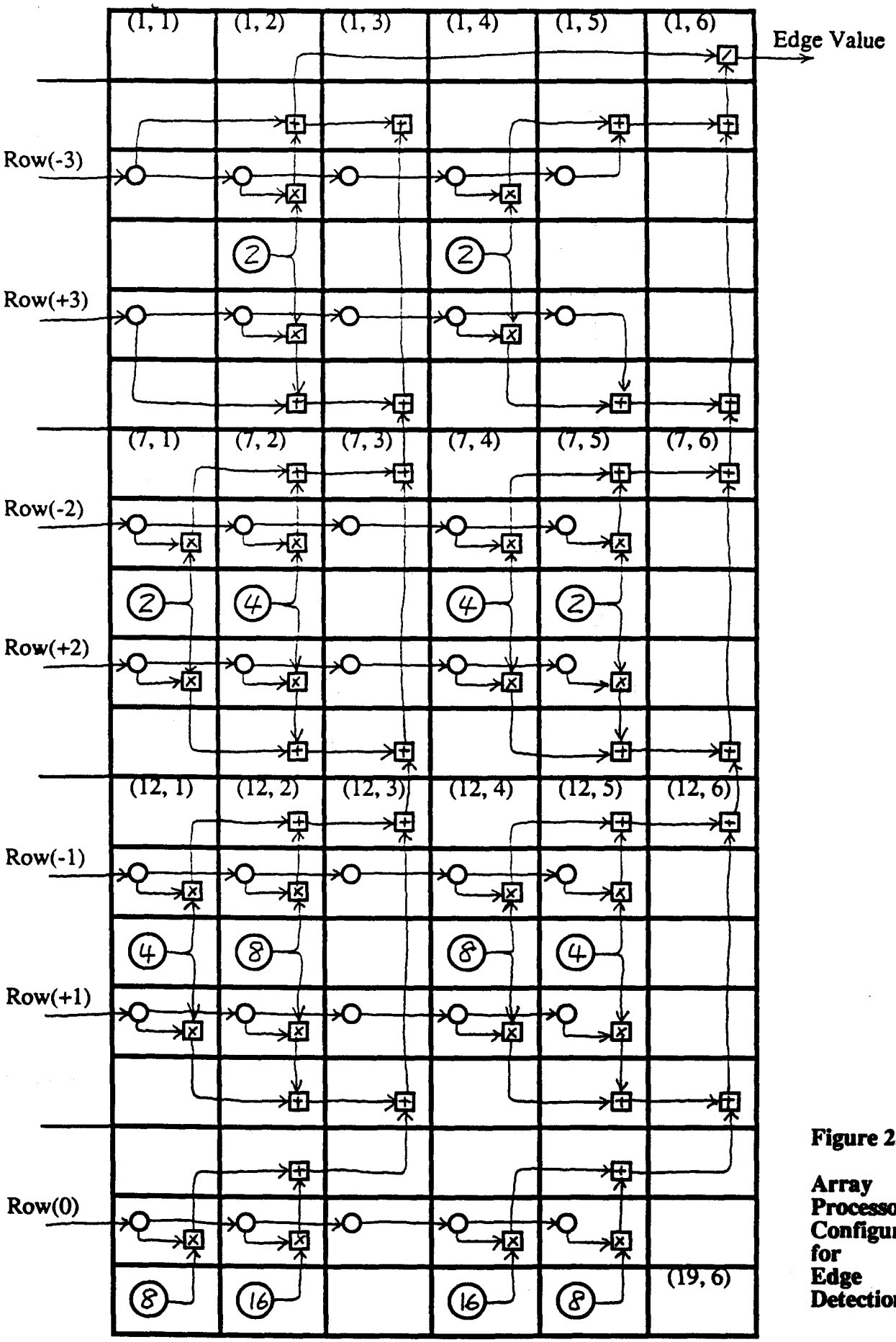


Figure 2.2

Array Processor Configured for Edge Detection

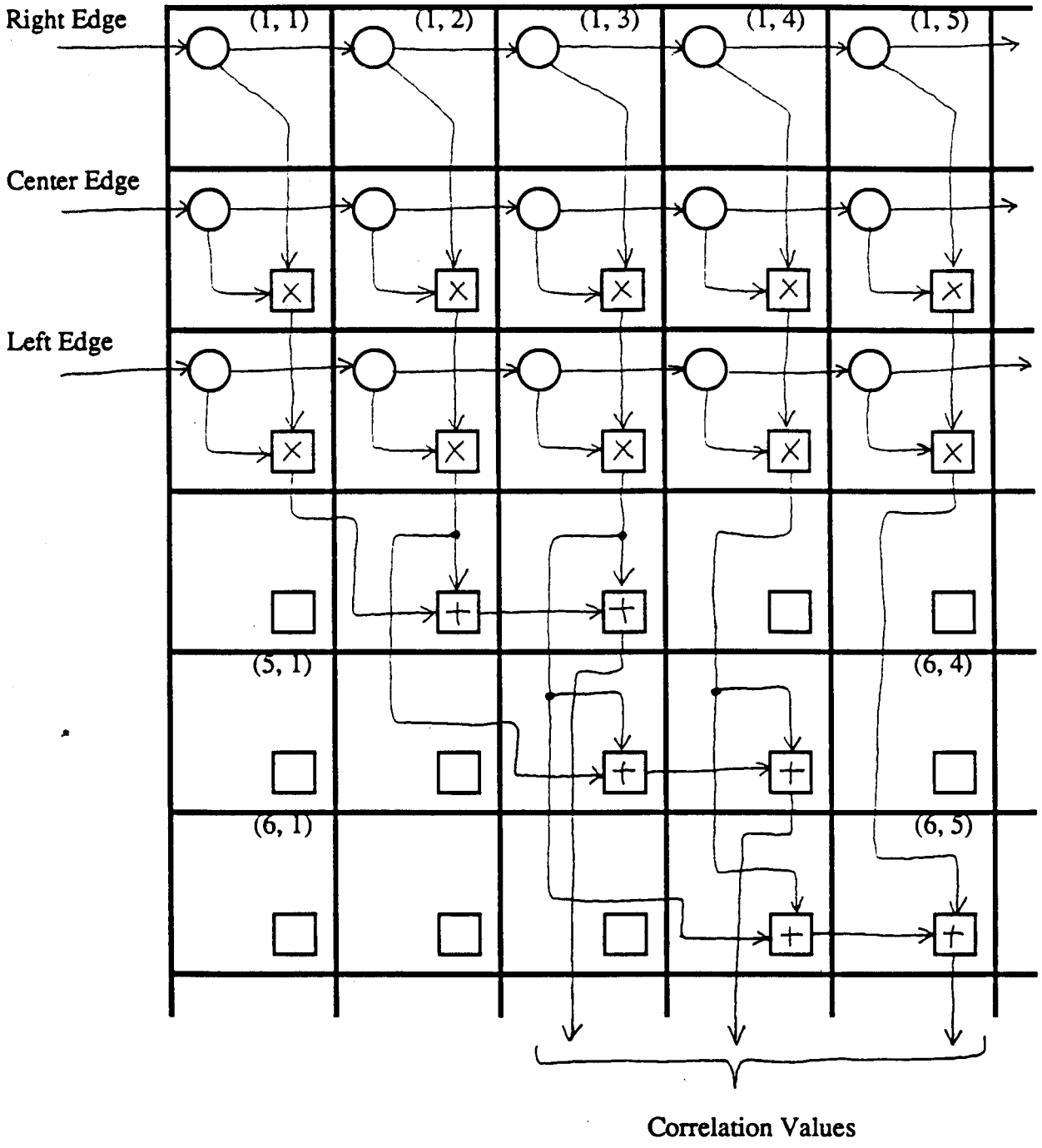


Figure 2.3 Array Processor Configured for Stereo Correlation

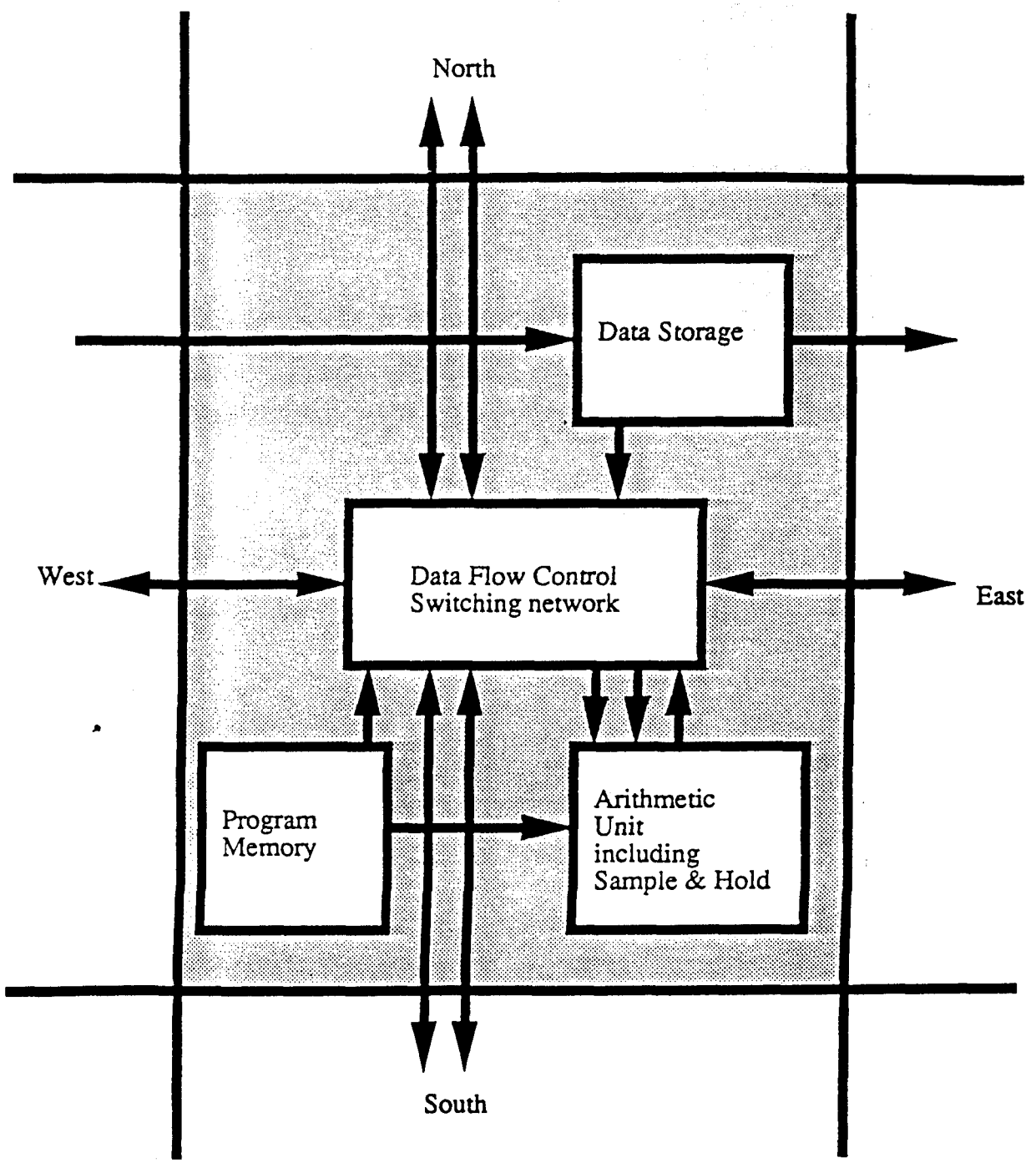


Figure 2. 4 Array Processor Element

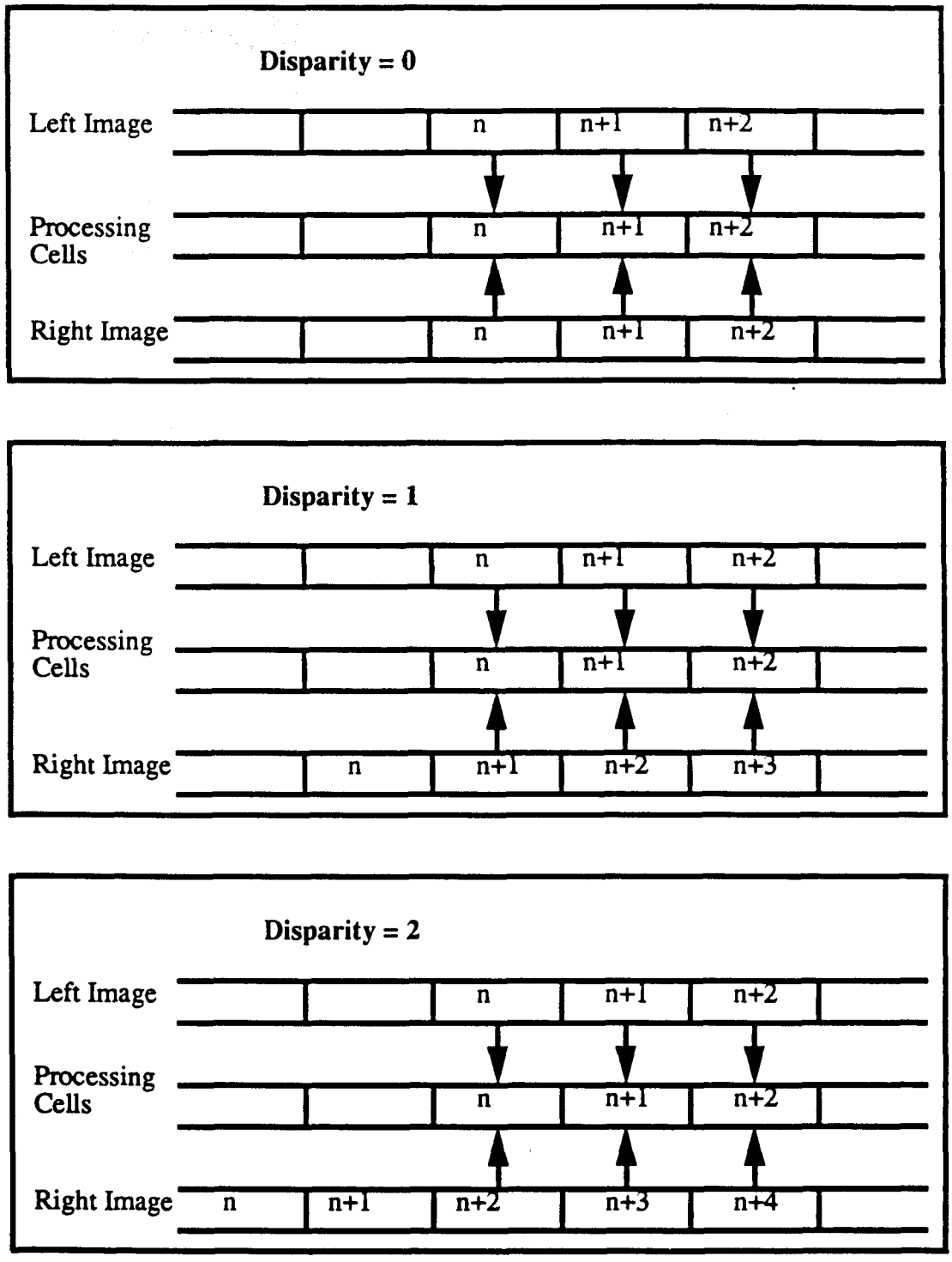


Figure 4. 1 Binocular Stereo

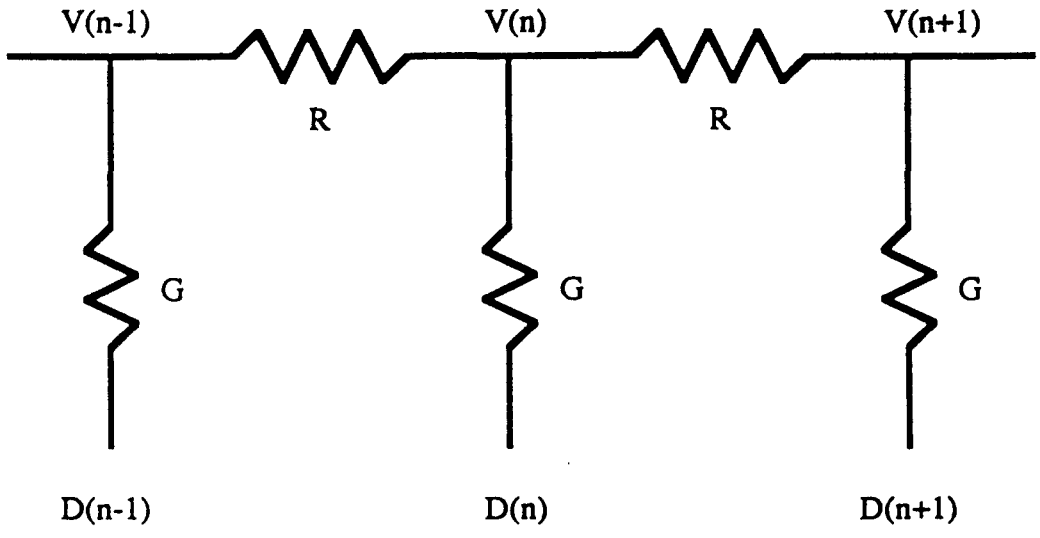


Figure 5. 1 1-D Version of Resistive Grid, with Input and Output Encoded as Voltages

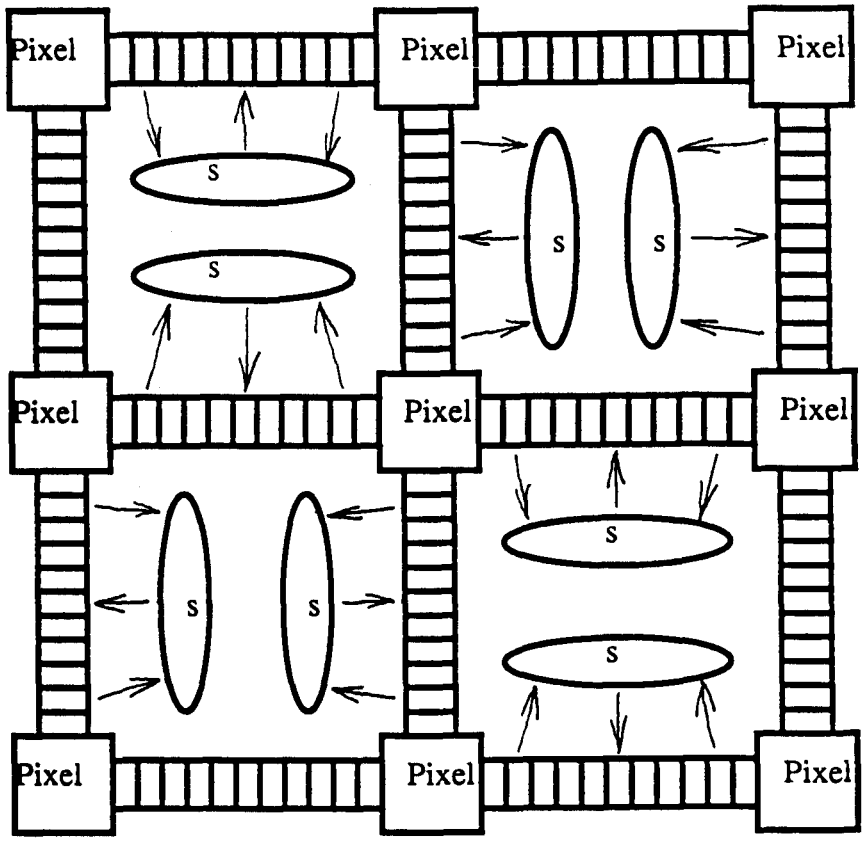


Figure 5. 2 CCD Smoothing and Segmentation Architecture

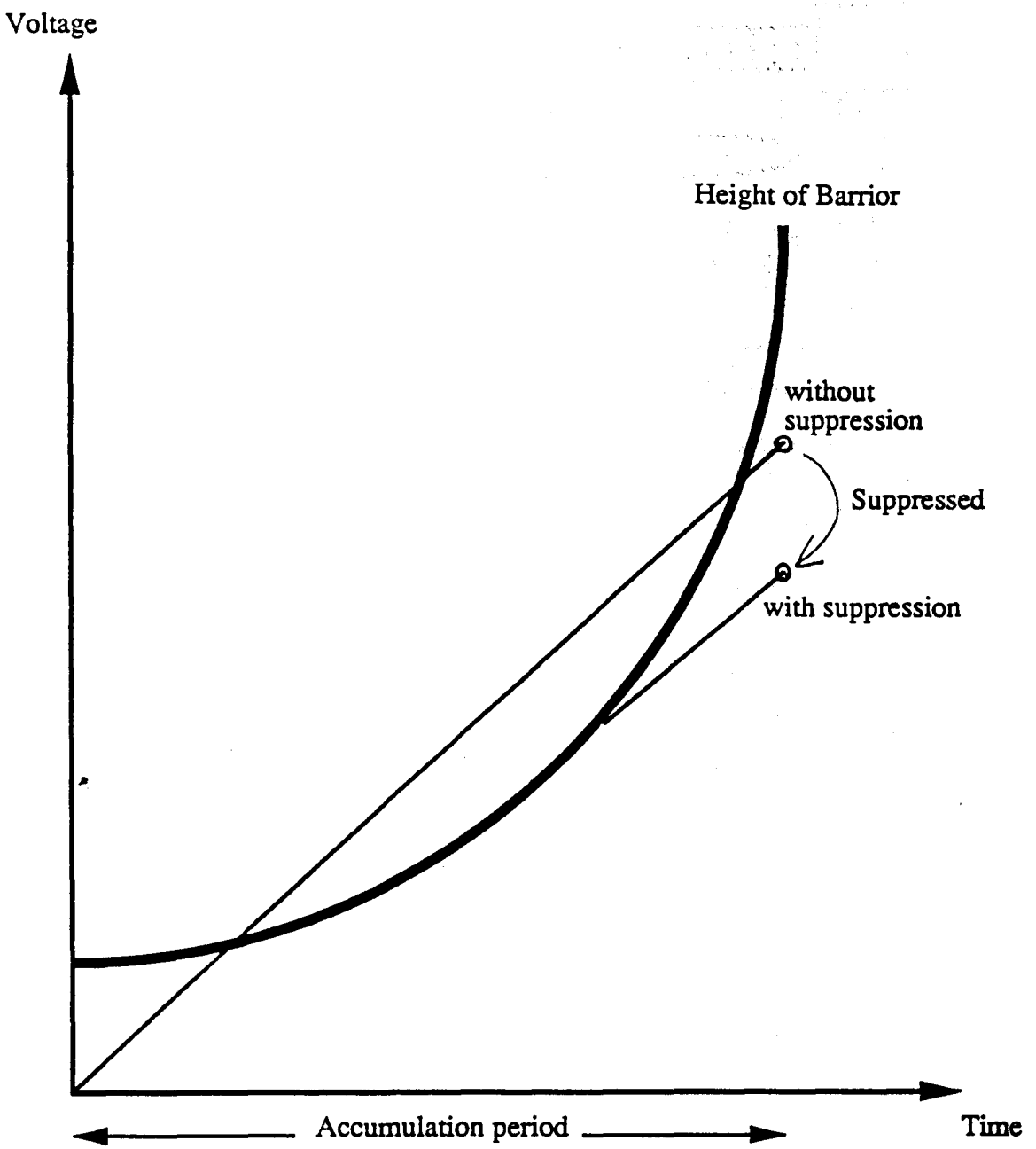
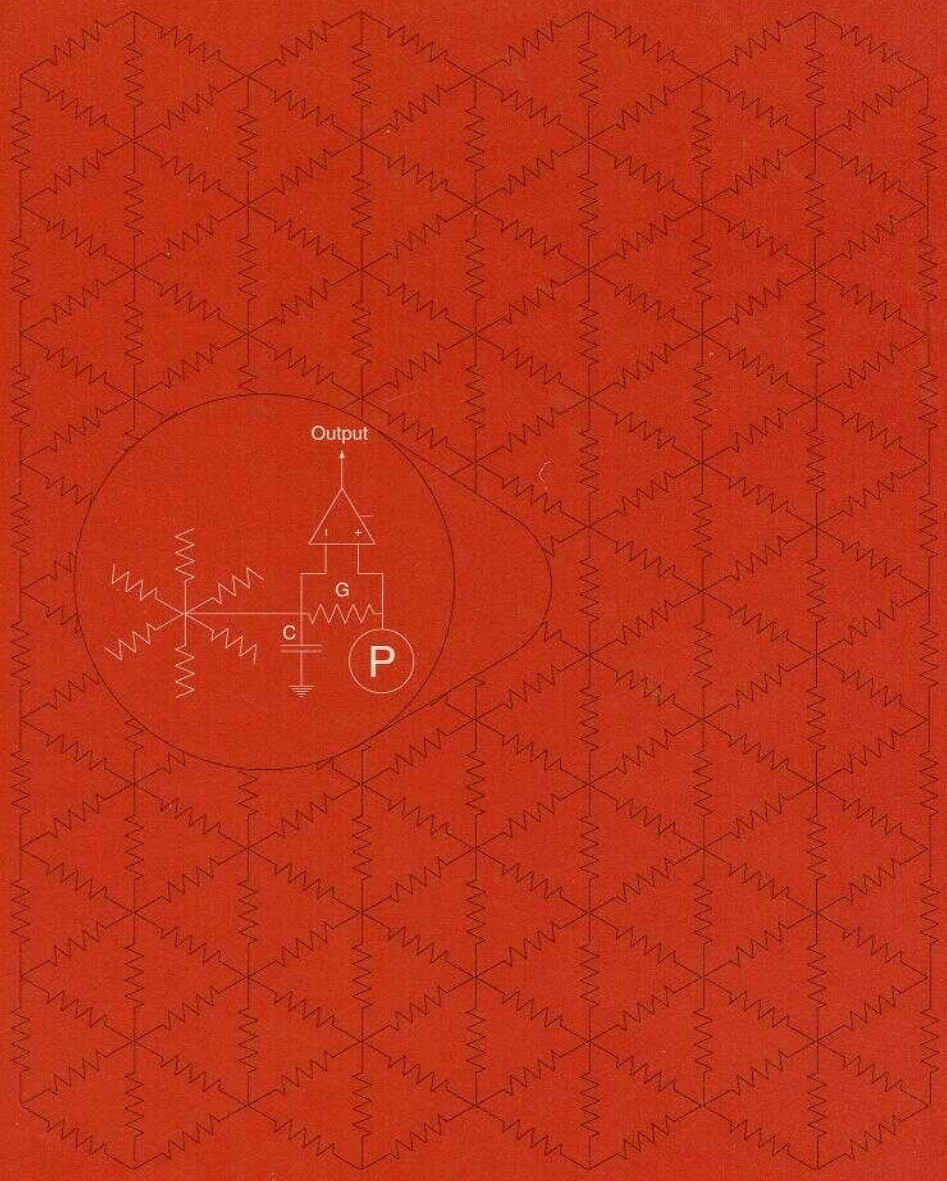


Figure 5. 3 Dynamic Sensitivity CCD Imager

VISION CHIPS
Implementing Vision Algorithms with Analog VLSI Circuits

Christof Koch and Hua Li



IEEE Computer Society Press

The Institute of Electrical and Electronics Engineers, Inc.