Analog Integrated Circuits and Signal Processing, 23, 211–226, 2000
© 2000 Kluwer Academic Publishers. Manufactured in The Netherlands.

An Analog VLSI, Scale Invariant Method for Edge Detection

D. M. WILSON
Department of Electrical Engineering, University of Washington, Seattle, Washington 98195-2500 USA
Email: wilson@ee.washington.edu

Received October 19, 1998; Revised May 28, 1999; Accepted June 25, 1999

Abstract. A simple technique for detecting adjustable contrast in a visual scene is presented. The circuit elements can be used to detect contrast in any array of sensors or processing elements where spatial relationships among neighboring elements define contrast or the presence of an edge. This technique eliminates the need for a differential pair, thereby allowing more than two inputs to be compared for contrast in a single processing step. The circuit elements first smooth erroneous edges in the array through the use of a resistive network, then, the mean (scaled by an adjustable amount) of a pixel and its neighbors is compared to the harmonic mean of the same pixels to detect the presence of contrast within the pixel neighborhood. Comparison between the mean and harmonic mean allows the detection of contrast to be scale-invariant as long as the transistors remain in subthreshold operation. This circuit offers the massively parallel processing inherent to focal plane processing within an 18% fill factor in a 2 μm process, 6.8 μW typical power dissipation per element, and 0.67 ms response time at low power subthreshold operation. Results for a proof of concept, 8×8 array of pixels with light inputs, as well as a purely electronic input, 4×4 array are presented.

Key Words: analog VLSI, focal plane processing, photo detectors

1. Introduction

The image segmentation process typically begins with some form of pre-processing of image contrast followed by an elementary edge detection operation. In an array of raw photodetector input, the edge is often defined as a certain change in intensity between neighboring pixels. If this raw photodetector input is first pre-processed, edge detection may correspond to contrast associated with motion, velocity, texture, or similar second order features. If an array gathers input from electronic signals or other non-light based signal, edge detection is still a valuable initial processing task. In auditory processing, edge detection can correspond to distinguishing fundamental auditory components from their harmonics, formant frequencies, or similar components of a speech or sound signal. In this paper, we discuss a method for edge detection using electronic input and light input mode, however, the edge detection method can be used in any application where moderate to high contrast edges are of interest. This edge detection method is scale invariant, producing a constant edge profile even in the presence of significant changes in background illumination.

Edge detection is a fundamental by-product of image segmentation and is usually proceeded, in hardware or in software, by some form of image correction or enhancement. In preparation for detecting edges, contrast of interest in the image is often enhanced while contrast generated from noise, photodetector defects, non-uniform illumination or similar irregularity is downplayed. Neighborhood averaging is the simplest form of image pre-enhancement, where each pixel value is replaced with the average of itself and a pre-specified set of neighboring values. While averaging can remove erroneous edges induced by noise over time or over space, it can also blur areas that contain edges of interest. A number of weighted averaging techniques, such as the Savitsky and Golay fitting procedure, resolve this issue by maintaining areas of sharp contrast while blurring regions of medium to low contrast. Similar filters, such as median and other forms of rank filtering, can preserve contrasts important to image interpretation while blurring or eliminating contrast generated by image noise [1].

Once pre-enhanced or pre-processed, the image is ready for segmentation, where edges are determined and the image is divided into unique regions of interest and activity. As the first step in segmentation, images may be eroded, dilated, or subject to a similar window, compute, and replace policy for merging related objects or separating spatially close objects that are not related. Most image segmentation schemes perform edge detection based on some form of one-dimensional (black and white) or multi-dimensional (color) thresholding process. Powerful computer systems may take a more wholistic approach to image segmentation using techniques such as split-and-merge and region growing to segment the image as a whole rather than a sum of parts. In the more conventional schemes, the image is thresholded to a binary image and then an XOR window used to determine the presence of edges [1].

In general, edge detection on the focal plane is restricted to very simple edge detection algorithms so that resolution is nor prohibitively sacrificed in the imager itself. Focal plane processing allows part of the signal processing electronics to be placed local to each photodetector in an array of pixels. This type of architecture enables massively parallel processing of the image to be performed by the simultaneous operation of all the focal plane processing elements. Typically, focal plane processing elements have been analog to reduce total power dissipated on the focal plane and to minimize electronic noise introduced by circuitry on the focal plane. Focal plane processing has been successfully demonstrated for edge detection [2–7], as well as other tasks such as image pre-enhancement [8–11], motion detection [12–19], orientation estimation [20], centroid detection [21], and skeletonization [22,23]. In order to preserve a useful resolution in these arrays, effective focal plane processing has traditionally been restricted to small transistor count tasks.

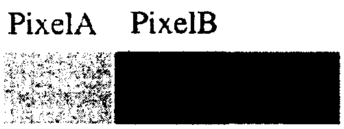
Focal plane processing methods for edge detection have typically been restricted to pre-enhancement of relevant edges in the image followed by the thresholding of the image to a binary equivalent to facilitate binary edge computation. Pre-enhancement of edges has been accomplished using various smoothing techniques [5] which have been successfully implemented in hardware using a relatively small amount of silicon real-estate. This stage of processing attenuates edges of low magnitude and high spatial frequency more than those of higher magnitude and lower spatial frequency, the resistive network has the effect of decreasing spatial noise in the image for subsequent processing. Outputs of the smoothed pixel corresponding to neighboring pixels are typically compared and thresholded based on a constant value to determine the presence of an edge. Smoothing using these techniques has the unavoidable consequence of decreasing signal strength and reducing signal to noise ratio.

Once pre-enhanced, edges have been detected on the focal plane by applying the difference operator between two neighboring pixels and OR'ing these results with other difference operations in the pixel neighborhood. Other edge detection operations involving the weighted evaluation of differences between pixels and their neighbors, including two dimensional Laplacian masks and compass gradient masks, have been too complex to implement explicitly on the focal plane.

We present a technique for edge detection that pre-processes contrast and computes edges on the focal plane with scale invariance to changing background levels and fluctuations. While we use a resistive network for initial smoothing, subsequent circuits replenish the gain lost in initial smoothing with an edge detection threshold that can be varied across a wide range. The image is first smoothed using a horizontal resistive network similar to those previously demonstrated by Mead [8]. Then, the harmonic mean of a pre-defined window is computed and compared to its mean value, shifted by an adjustable amount, to detect an edge. The smoothing requires six transistors per pixel, and the edge detection 14 transistors per pixel. The comparison of mean (offset by an adjustable amount) and harmonic mean offers automatic gain control (scale invariance) and adjustable edge detection thresholds.

2. Edge Detection Analysis

We compare our edge detection technique here to the simple thresholding technique discussed in the introduction for binary image processing. We choose this simple technique for comparison, because it can compete with our mean comparison technique in terms of space and power consumed in a focal plane implementation. The simplest, binary edge detection technique is to threshold the difference between a pixel and its neighbor (an analog to digital conversion operation) and to OR the result (a binary operation) with differences between the same pixel and other neighbors. The neighbors to a pixel are defined in a predetermined window. For example, in a one dimensional array, the neighbor may be defined as the pixel to the right (B) of the central pixel (A). Each pixel is compared to its rightmost neighbor (B), thresholded, and the results indicates the presence of an edge or lack thereof in binary form:

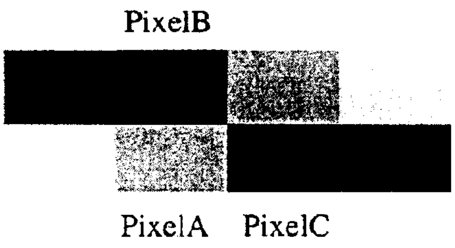


$$Edge_A = \begin{pmatrix} 0 \text{ if } (|A-B|>K) \\ 1 \text{ if } (|A-B| \leq K) \end{pmatrix}$$

In this example, the edge is active low. The definition of an edge in one dimension using the conventional difference operator is then

$$|A-B|>K \qquad (1)$$

where K is a constant. In binary processing of a two dimensional array, the difference operator is applied between the central pixel and all its neighbors. A neighborhood can be defined as all of the nearest neighbors or some subset of the nearest neighbors. For example, a neighborhood and associated edges might be defined as follows around the pixel A:



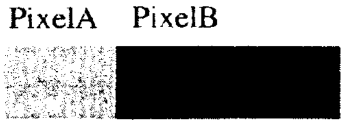
$$Edge_A = 0 \text{ if } (|A-B|>K)$$

OR

$$Edge_A = 0 \text{ if } (|A-C|>K)$$

where each pixel's neighborhood is defined as the pixel above and the pixel to the right of it and the edge is active low.

The technique we propose here for edge detection compares the arithmetic mean to the (scaled) harmonic mean (AMHM) in a neighborhood of pixels. Using the same neighborhoods as for the conventional difference operator examples discussed above, this mean comparison method is expressed in one dimension as follows.

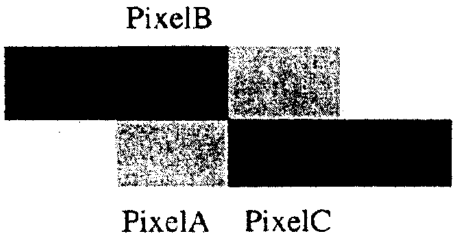


$$Edge_A = \begin{pmatrix} 0 \text{ if } \left(\frac{A+B}{2} > \frac{K}{\frac{1}{A}+\frac{1}{B}}\right) \\ 1 \text{ if } \left(\frac{A+B}{2} \leq \frac{K}{\frac{1}{A}+\frac{1}{B}}\right) \end{pmatrix}$$

where K is a constant and the edge is active low. An edge is then present when

$$\frac{A+B}{2} > \frac{K}{\frac{1}{A}+\frac{1}{B}} \qquad (2)$$

This definition can be extended to a two-dimensional neighborhood in a straightforward manner



$$Edge_A = \begin{pmatrix} 0 \text{ if } \left(\frac{A+B+C}{3} > \frac{K}{\frac{1}{A}+\frac{1}{B}+\frac{1}{C}}\right) \\ 1 \text{ if } \left(\frac{A+B+C}{3} < \frac{K}{\frac{1}{A}+\frac{1}{B}+\frac{1}{C}}\right) \end{pmatrix}$$

where the pixel neighborhood is defined in the same way as for the conventional binary edge detection procedure discussed previously. The definition of an edge in two dimensions is simply:

$$\frac{A+B+C}{3} > \frac{K}{\frac{1}{A}-\frac{1}{B}+\frac{1}{C}} \qquad (3)$$

Both the difference and AMHM operators are preprocessing steps in the segmentation of an image. We demonstrate their operation on grayscale images whose inputs are representative of light intensity, however, either method can be extended to color images or to other preprocessed versions of the image based on texture, hue, motion, saturation or similar characteristics. The AMHM method is an alternative approach to the segmentation step, but produces the same binary outputs as conventional difference operators, and, as a result, can be integrated with a variety of digital image techniques.

The mean comparison (AMHM) method offers two advantages over the conventional binary difference operator for edge detection.

- scale invariance as the background illumination scales up or down. The AMHM method detects the same number of edges while the conventional operator produces more edges as background illumination increases (for a particular scaling constant K).
- multiple edge sensitivity. The boolean OR operation in conventional edge detection gives equal weight to an edge whether it is the only edge in the pixel neighborhood or one of multiple edges. The AMHM method combines information from both edges, so that two low contrast edges in the pixel neighborhood can produce an edge, as well as a single high contrast edge. This weighted contrast sensitivity is more similar to grayscale image processing methods rather than binary image processing techniques.

To demonstrate scale invariance for the one dimensional case, we multiple all pixels by a scaling constant X and calculate the resulting revised threshold for edge detection in terms of the original threshold (K). The binary edge detection operator, after scaling, is simply

$$|XA - XB| > K \tag{4}$$

which results in a new detection threshold of $\frac{K}{X}$. To examine the impact of scaling on the AMHM method, we first convert the one-dimensional edge operator of (2) to an expression that contains the difference operator $(A\text{-}B)$

$$\frac{|A - B|}{\sqrt{AB}} > K' \text{ where } K' = \sqrt{2K - 4} \tag{5}$$

After scaling all pixels in the neighborhood by X, this expression becomes:

$$\frac{|XA - XB|}{\sqrt{XAXB}} > K' \text{ which is equivalent to } \frac{|A - B|}{\sqrt{AB}} > K' \tag{6}$$

In the one dimensional case, it is clear that the first difference operator, when scaled, reduces the threshold for edge detection by the scaling factor X while the AMHM method results in no difference in the edge detection threshold (scale invariant). The two dimensional case is more complex since, using the binary edge detection method, an edge is determined by a boolean (OR) combination of several one dimensional cases while the two dimensional case for the AMHM method evaluates the entire pixel neighborhood in a single operation, an approach that is more similar to grayscale digital image processing and masking techniques. To compare the effect of scaling by a constant (X) on each of the two cases, involves breaking down the possible scenarios for edge detection as follows

- Case 1 single edge, edges occur at AB or at AC border, $A\text{-}B$ or $A\text{-}C$ is negative.
- Case 2 single edge, edges occur at AB or at AC border, $A\text{-}B$ or $A\text{-}C$ is positive.
- Case 3 multiple edges at AB and at AC border, $A\text{-}B$ and $A\text{-}C$ both positive or both negative.
- Case 4 multiple edges at AB and at AC border, $A\text{-}B$ negative and $A\text{-}C$ positive or vice versa.

For all four cases, the conventional binary edge operator produces the same results. Scaling all three pixels in the neighborhood up by a constant X is equivalent to scaling the edge detection threshold down by this same constant, the lack of scale invariance is identical to that of the one dimensional neighborhood. To examine the impact of scaling on the AMHM method in two dimensions, we convert each edge scenario in cases 1 through 4 (with some simplifications) to an equivalent change in the edge detection threshold.

- Case 1 single edge, $A < < B$ and $A = C$

$$\frac{A + B + C}{3} > \frac{K}{\frac{1}{A} + \frac{1}{B} + \frac{1}{C}} \tag{7}$$

reduces to

$$\frac{B}{3} > \frac{K}{\frac{1}{A} + \frac{1}{A}} \text{ which is equivalent to } \frac{|A - B|}{A} > K'$$
$$\text{where } K' = \frac{3K}{2} - 1 \tag{8}$$

- Case 2 single edge, $A \gg B$ and $A = C$

$$\frac{A + A}{3} > \frac{K}{\frac{1}{B}} \text{ which is equivalent to } \frac{|A - B|}{B} > K'$$
$$\text{where } K' = \frac{3K}{2} - 1 \qquad (9$$

- Case 3 multiple edges. $A \ll B$. $A \ll C$ and $B = C$ (similar to case where $A \gg B$, $A \gg C$ and $B = C$

since, using the
dge is determined
of several one
dimensional case
the entire pixel
, an approach that
image processing
ıare the effect of
of the two cases,
ble scenarios for

r at AB or at AC

r at AB or at AC

at AC border, A-B
gative.
at AC border, A-B
e versa.
ional binary edge
. Scaling all three
a constant X is
etection threshold
ıe lack of scale
one dimensional
ıact of scaling on
sions, we convert
ugh 4 (with some
ıange in the edge

$A = C$

(7)

$$\frac{|A-B|}{A} > K'$$

(8)

$= C$

$$\frac{|A-B|}{B} > K'$$

(9)

$A \ll C$ and $B = C$
$\gg C$ and $B = C)$

$$\frac{B+B}{3} > \frac{K}{\frac{1}{A}} \text{ which is equivalent to } \frac{|A-B|}{A} > K'$$

$$\text{where } K' = \frac{3K}{2} - 1 \qquad (10)$$

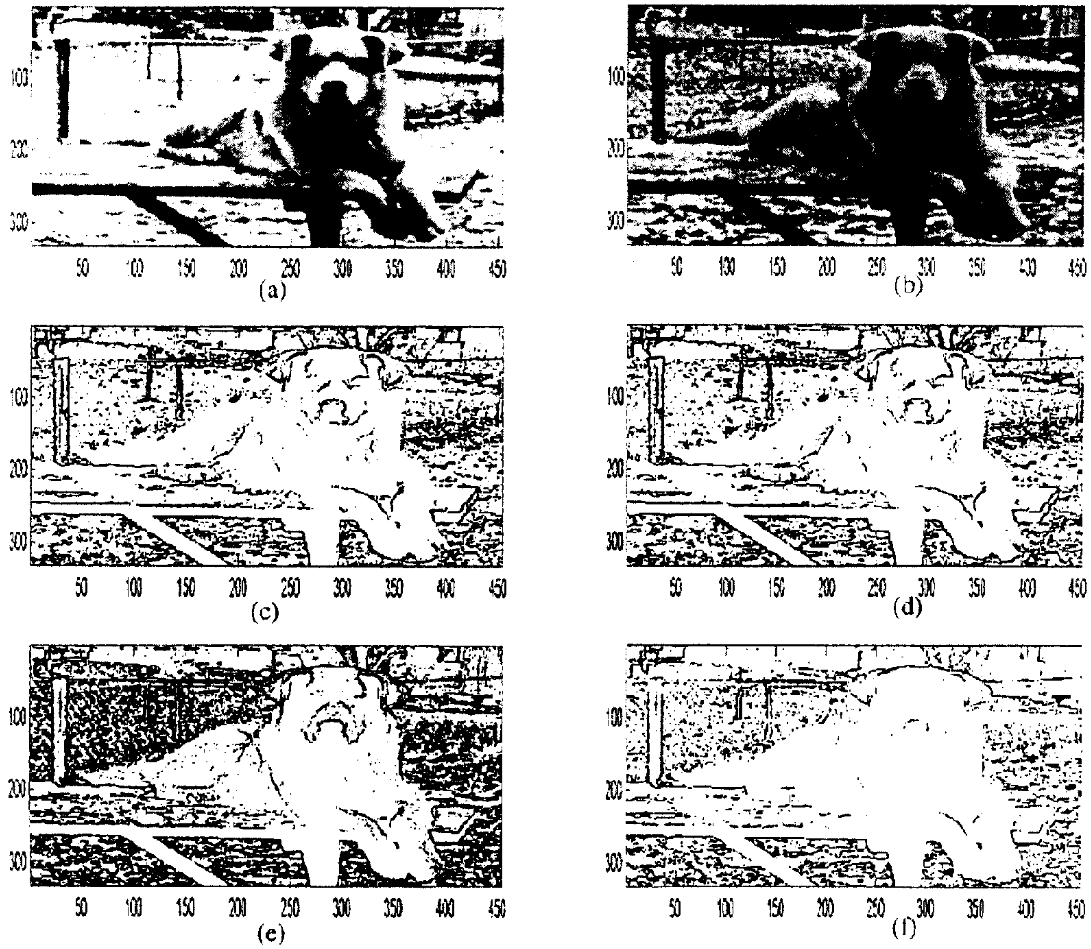
- Case 4 multiple edges, B ≪ A ≪ C (similar to C ≪ A ≪ B)

$$\frac{C}{3} > \frac{K}{\frac{1}{B}} \text{ which is equivalent to } \frac{|B-C|}{B} > K'$$

$$\text{where } K' = 3K - 1 \qquad (11)$$

It is obvious that, in all four cases above, scale invariance is achieved because multiplication by a constant X results in X dropping out of the left side of the equation in the same way as in the one-dimensional pixel neighborhood. This scale invariance of AMHM method over the difference operator for edge detection and image segmentation is shown using software simulation in Fig. 1. The original image (Trooper, the dog) is scaled to two different levels of background illumination (Fig. 1(a) and Fig. 1(b)). Edges computed using AMHM are the same for the two levels of background illumination (Fig. 1(c) and Fig. 1(d)), using a difference operator in binary image processing, however, the number of computed edges decreases significantly from a bright background Fig. 1(e) to a dim background Fig. 1(f).

It is interesting to note that the edge detection



(a)
(b)
(c)
(d)
(e)
(f)

Fig. 1. Scale invariance in edge detection. Conventional edge detection produces more edges as the background illumination increases, while the AMHM comparison scheme maintains the same edges. The original image in (b) is the image in (a) scaled (multiplied) by 0.5. Images in (c) and (d) are computing by comparing the scaled mean to the harmonic mean. Images in (e) and (f) are computing by comparing the differences in neighboring pixels to a fixed threshold value. The noise in the image, as demonstrated by a large number of edges in the grass and snow in the background, can be smoothed out using presmoothing techniques suitable for hardware implementation.

threshold for Cases 1 through 3 where a single edge or like edges are considered remains the same at $\frac{3K}{2} - 1$ but the edge detection threshold $(3K - 1)$ for Case 4, where two unlike edges are present, is higher. Using the difference operator, the threshold remains the same, regardless of whether single edges, multiple like edges, or multiple unlike edges affect a particular pixel. In any of these cases, the difference between a pixel and its neighbor has to exceed a constant threshold K at least once in all possible relationships between a pixel and its assigned neighborhood. Multiple edges (corners) are not given additional emphasis using the difference operator. However, in the AMHM method, multiple unlike edges demonstrate a lower detection threshold than multiple like edges or single edges. This behavior is more akin to grayscale, digital image processing techniques where a weighting mask (e.g. Laplacian or compass gradient) is convolved with the original image before thresholding.

This result is best observed visually in Fig. 2. Fig. 2(a) and Fig. 2(c) demonstrate the invariance of edge threshold location using the difference operator to detect all types of edges; an edge is present when a point on the solid lines exceeds that on the dotted lines. Fig. 2(b) and Fig. 2(d) demonstrate the varying threshold for multiple like or single edge detection thresholds as compared to the lower threshold for multiple unlike edges. An edge is present when the solid line (scaled arithmetic mean) exceeds that of the dotted line (harmonic mean). The edge detection threshold is the intersection of the two curves. When the edge *(A-C)* is negative and the edge *(A-B)* is positive (unlike edges), as illustrated in point A on Fig. 2(d), the edge detection threshold is significantly lower than

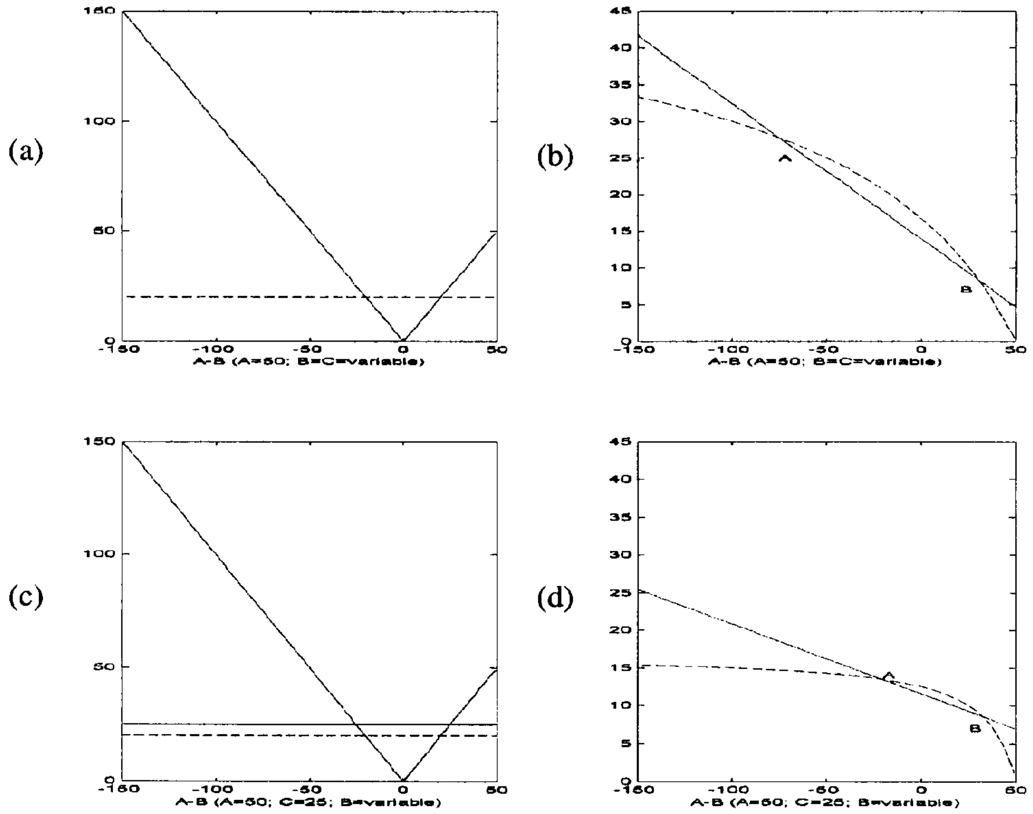


Fig. 2. Edge detection thresholds for various scenarios. The difference operator is used in (a) and (c) to set a constant detection threshold (intersection of solid and dotted curves), regardless of the nature of edges present in the pixel neighborhood. The AMHM method, however, in (b) and (d) demonstrates that the detection threshold is lower for multiple unlike edges than for single edges or multiple edges of the same type.

when

present

The

provide

multipl

neighb

gained

single

increas

provide

magnif

(Fig. 1)

around

contras

operato

toward

3. Ci

The edg

three m

of Fig.

Fig. 3. E

Results o

the differ

compute

looks onl

when multiple, like edges or a single edge are present (Point A in Fig. 2(b)).

The threshold variation demonstrated in Fig. 2 provides better sensitivity to edges in situations where multiple contrast levels are present in a pixel neighborhood. This enhanced sensitivity is not gained at the expense of decreasing the threshold for single edge detection threshold which in turn, would increase the noise in the segmented image. Fig. 3 provides an example of this enhanced sensitivity by magnifying the edges detected around Trooper's nose (Fig. 1). The AMHM method detects the lower edges around the interface between nose and mouth, where contrast is low, but edges are multiple. The difference operator method detects only the high contrast edges toward the top of the nose.

3. Circuit Description

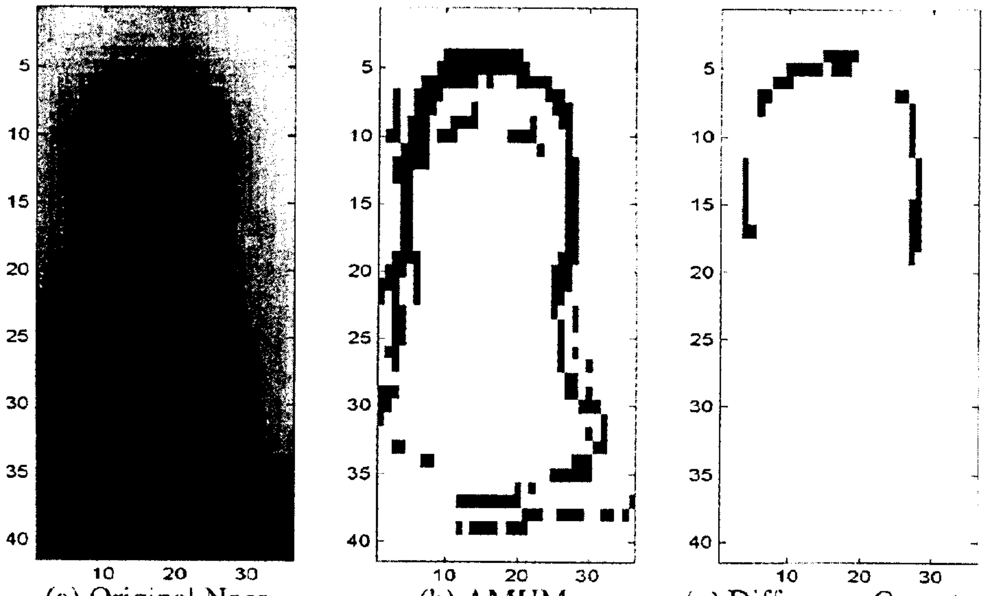
The edge detection scheme described here consists of three major stages as described in the block diagram of Fig. 4. Stage 1 smoothes the image to reduce noise in the input array and the erroneous detection of edges. Stage 2 calculates the harmonic mean and mean of the smoothed inputs within the neighborhood window, and Stage 3 compares the harmonic mean with the mean offset by an adjustable amount to detect edges. The edge detection threshold is set in Stage 3.

The first stage smoothes the image to eliminate erroneous edges caused by spatial noise in the image. Smoothing is performed using a resistive network arranged in a grid pattern. The output voltage at each pixel or processing element is spread to neighbors in the x and y directions where the effect of a pixel on its neighbors decreases exponentially with increasing distance away from the pixel.

The second stage calculates the mean of the inputs (smoothed phototransistor currents in the case of image processing) in the selected neighborhood window as follows

$$I_{mean} = \frac{I_{m,n} + I_{m,n-1} + I_{m-1,n}}{3} \qquad (12)$$

and the harmonic mean as follows.



(a) Original Nose (b) AMHM (c) Difference Operator

Fig. 3. Edge detection in low contrast, multiple edge pixels. The original image in (a) is a magnification of Trooper's nose from Fig. 1. Results of edge detection using the mean comparison (AMHM) method are shown in (b) and the results of standard edge detection using the difference operator are shown in (c). Since the AMHM method looks at both neighboring pixels simultaneously, it is better able to compute an edge in pixels that have low contrast with their neighbors but have multiple edges with their neighbors. The difference operator looks only at one difference between the pixel and its neighbors at a time and is not able to combine effects of these differences.

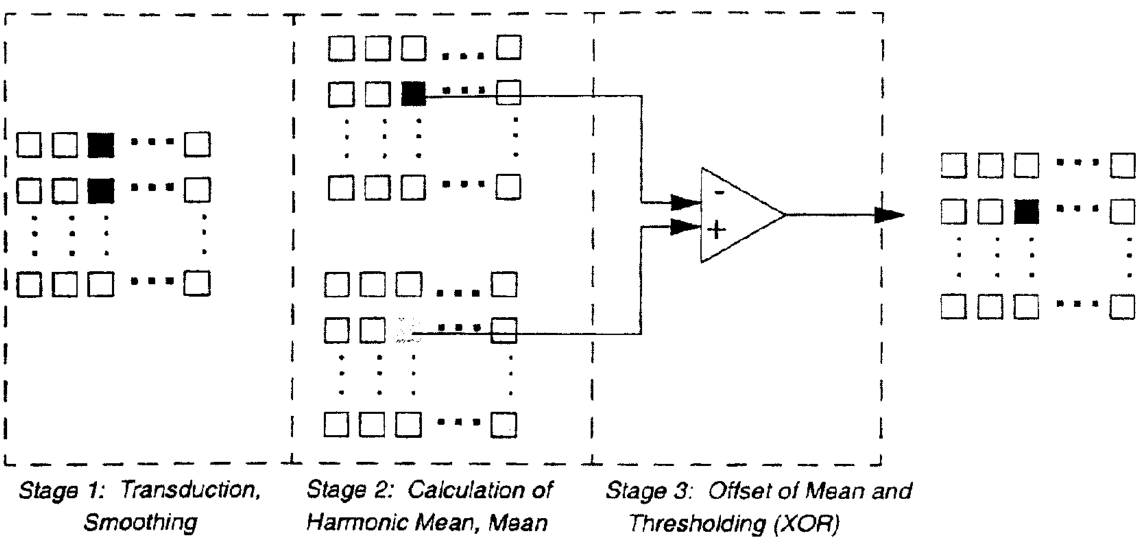

-
+
Stage 1: Transduction, Smoothing
Stage 2: Calculation of Harmonic Mean, Mean
Stage 3: Offset of Mean and Thresholding (XOR)

Fig. 4. Block diagram of image segmentation process. Stage 1 contains the array of sensory input and the resistive smoothing network. In stage 2 of the processing, the harmonic mean and mean of each pixel and its neighbor on top and to the left is calculated. In the last stage, the mean is offset by an adjustable, controllable amount and compared to the harmonic mean to establish the presence of an edge.

$$\frac{1}{I_{hmean}} = \frac{1}{I_{m,n}} + \frac{1}{I_{m,n-1}} + \frac{1}{I_{m-1n}} \tag{13}$$

The neighborhood window used here consists of each pixel $(I_{m,n})$, its neighbor above $(I_{m-1,n})$, and its neighbor to the left $(I_{m,n-1})$. The neighborhood window, although set in this configuration during fabrication, can be altered in fabrication to another geometry or through the use of fuses, can be programmed after fabrication to a variety of desired geometries. The expression for the harmonic mean in equation (13) does indicate the possibility of division by zero, however, in circuit implementations of this relationship, the dark currents are always non-zero, eliminating the potential for division by zero.

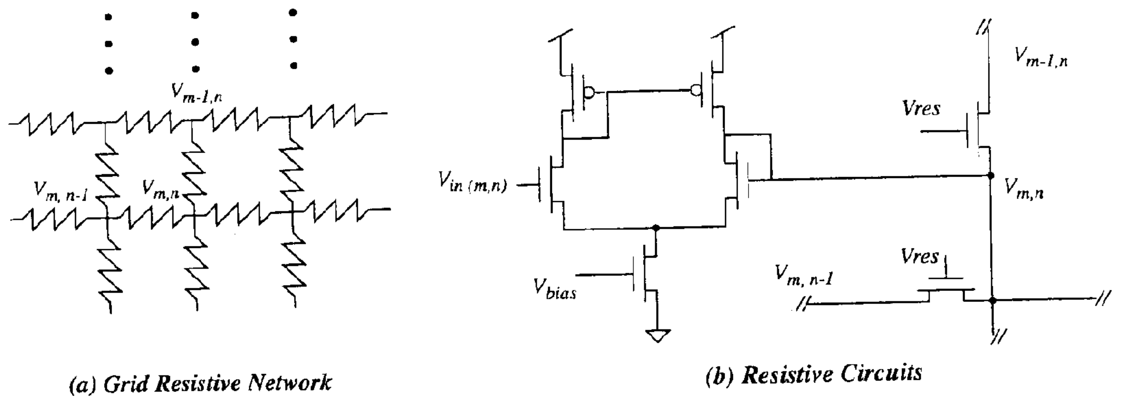
The last stage of processing compares the harmonic mean to the mean, where the mean can be shifted to adjust the edge detection threshold to a range of desired values.

$$PixelOut_{m,n} = \begin{cases} 0 & \text{if} \quad I_{hmean} < \frac{I_{mean}}{K} \\ 1 & \text{if} \quad I_{hmean} \geq \frac{I_{mean}}{K} \end{cases} \tag{14}$$

Comparison between the harmonic mean and the mean current within a neighborhood allows the edge threshold to remain constant across a range of input currents, background illuminations and local contrasts. The parameter K allows the magnitude of an edge to be defined in real time and adjusted to suit the current image or application.

3.1. Stage 1: Minimization of Erroneous Edges

Most sensory images are prone to noise across the spatial dimensions of a sensor array. Phototransistor and photodiode arrays are no exception. Images, as well, contain spatial noise that is often and easily misinterpreted as edges during image segmentation. The resistive network has been used in several implementations of analog VLSI retinas and focal plane processing to reduce the impact of spatial noise in subsequent image processing steps. The resistive network is implemented here in a grid network using transistors as resistors to smooth each input down the horizontal and vertical dimensions of the array (Fig. 5). The smoothing effect of each input across the resistive network from its point of origin has been well documented in the literature [8]. Used as a preprocessing tool here, the resistive network attenuates edges of high spatial frequency while maintaining the amplitude of edges of low spatial frequency. In this way, erroneous or localized edges are minimized so that they remain unrecognized in subsequent processing, and are effectively disregarded as noise.



(a) Grid Resistive Network

(b) Resistive Circuits

Fig. 5. Resistive network for smoothing sensory input array. Transconductance amplifiers and transistors implemented as nonlinear resistors are used to smooth the effect of each input away from the point of origin for that input.

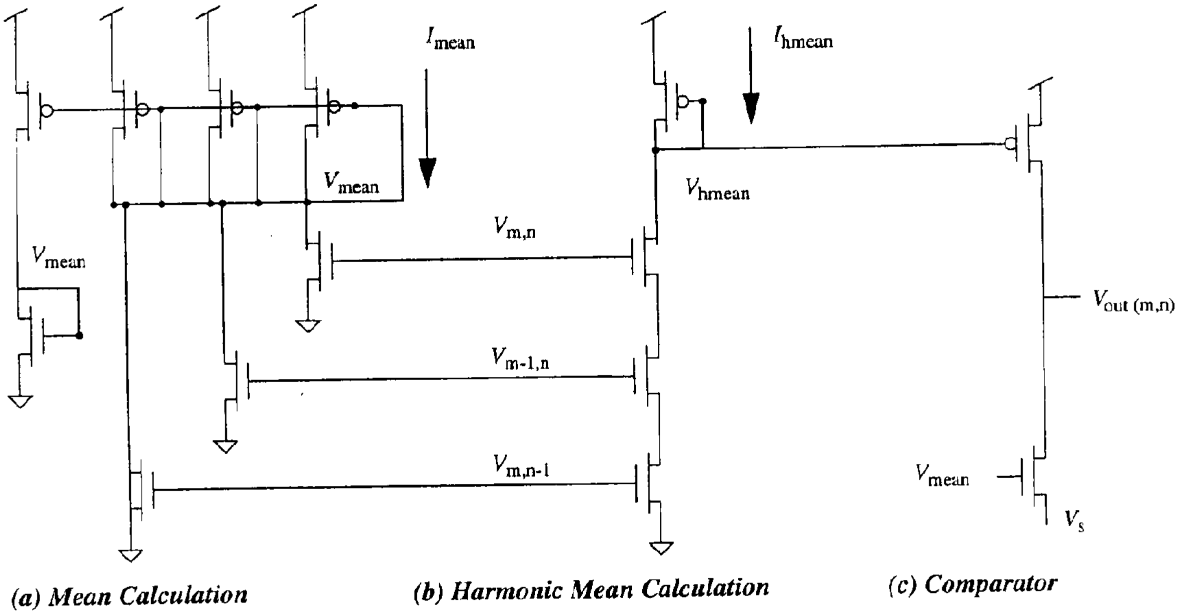
3.2. *Stage 2: Calculation of Mean Metrics*

In the next stage of processing, the inputs are prepared for the thresholding process that follows. Two mean metrics are calculated the mean and the harmonic mean. Circuits for finding the mean and harmonic mean are shown in Fig. 6. The mean output current, I_{mean}, is found by taking KCL at the summing node V_{mean} as follows:

$$I_{mean} + I_{mean} + I_{mean} = I_{m,n} + I_{m,n-1} + I_{m-1,n} \tag{15}$$

which leads to the result in equation (12).

The harmonic mean circuit Fig. 6(b) is a harmonic



(a) Mean Calculation

(b) Harmonic Mean Calculation

(c) Comparator

Fig. 6. Mean and harmonic mean circuits. Shown are circuits to generate the (a) mean current and (b) harmonic mean current of each pixel and its neighbor above and to the left. In order to compare the mean and harmonic mean of the three inputs and to obtain a non-zero edge detection threshold, the mean is offset (shifted downward) by a fixed amount in (c) the comparator stage. The output $V_{out(m,n)}$ is active low and indicates an edge that exceeds a preset threshold controlled by V_s.

Table 1. Characteristics of the edge detection circuits.

Parameter	Description
Technology	2 μm n-well, Orbit
Chip Size	2.3 × 2.3 mm^2
Package	40 pin DIP
Number of Pixels	4 × 4 (electronic input), 8 × 8 (light input)
Photoreceptor Size	2,500 μm^2
Cell Size	14,000 μm^2
Fill Factor	18%
Circuit Response Time	0.67 ms at 0.65 V nominal input (5 V supply)
Circuit Power Dissipation	6.8 μW per processing element (5 V supply)

correlator previously used for a variety of applications from detecting bumps for image processing applications [24] to neural oscillators for robust motor control [25]. Assuming the transistors operate in the subthreshold region, we apply the following exponential voltage-current relationship [8]:

$$I = I_o \exp\left(\frac{\kappa V_g}{\varphi_t}\right)[\exp((-V_s) - \exp(-V_d))] \qquad (16)$$

where I_o is a lumped fabrication constant, κ is the back gate effect, and φ_t is the thermal voltage. Assuming the exponential terms for the drain voltages in each transistor are negligible, some algebra yields the harmonic mean relationship described in (13). Realistically, the subthreshold parameter κ varies among transistors and with the source voltage of these transistors, so that this equation can only approximate the behavior of the harmonic mean circuit.

3.3. *Stage 3: Thresholding*

After the mean metrics are calculated, scale invariant thresholding is now performed as shown in Fig. 6(c). The mean is first scaled by a fixed amount according to the following relationship:

$$I_{mean(\text{scaled})} = \frac{I_{mean}}{I_o \exp\left(\frac{V_s}{\varphi_t}\right)} \qquad (17)$$

The mean current is scaled by an adjustable amount so that it intersects the harmonic mean curve and establishes a non-zero edge detection threshold (refer to Fig. 2). The output of the processing element is active low and is scanned out using a standard decoding, random access arrangement.

4. Experimental Results

A prototype, 4 × 4 array of edge detection elements with electronic inputs and an 8 × 8 array of focal plane processing elements with phototransistor inputs has been fabricated in a 2.0 μm, CMOS n-well process. Characteristics of the chip are summarized in Table 1 and components of the chip are shown in Fig. 7. The fill factor for the prototyped active pixels is 18%, but can be increased to a 40% fill factor (including a 10% reduction in pixel size with optimized layout of guard rings) using a 0.5 μm feature size or smaller for resolutions up to 200 × 200 on a 1 cm^2 die. In this section, we describe experimental results for the edge detection circuits in response to both electronic and light inputs.

Detailed response characteristics are obtained through tests of the 4 × 4 array containing only circuits and no photodetectors. Important to contrast or edge detection applications are effects on detection threshold of smoothing (spreading resistance in the resistive network), the mean offset factor (controlled by V_s), the power supply (V_{dd}), and the location of the edge in the neighborhood window. The 8 × 8 array of elements with photodetector inputs is also tested to confirm the functionality of the edge selection and the fundamental behavior characterized in the electronic input array.

The electronic input array has been tested using an automated data acquisition set-up based on Labview software and National instruments data acquisition hardware. Test transistors from each array have been used to experimentally characterize MOSFET parameters for each chip tested. The imaging array has been tested by projecting images of interest onto the focal plane using a 12 mm, manual focus, manual iris lens.

4.1. *Effect of Contrast Enhancement (Smoothing V_{res})*

Linear averaging of neighboring pixels tends to blur all edges in an image, reducing contrast among adjacent pixels uniformly. The resistive network used here allows the image to be smoothed in such a way

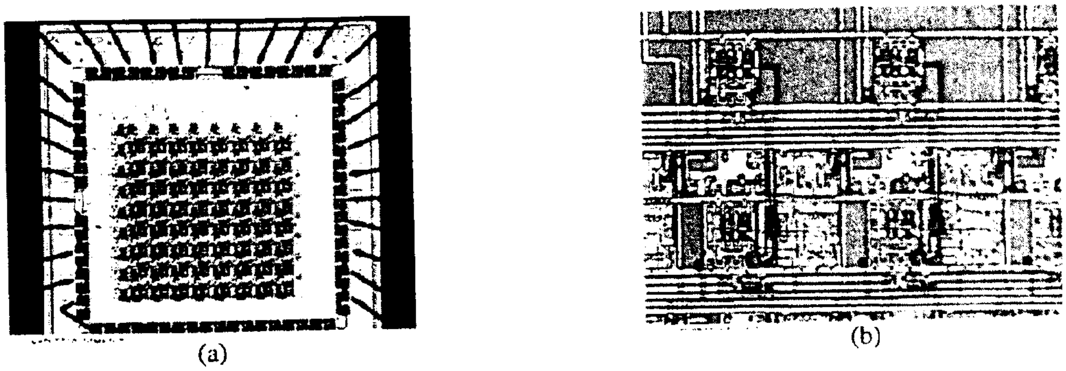


Fig. 7. Layout of edge detection elements. Shown above is (a) the 8 × 8 array of edge detection elements with phototransistor inputs and (b) a pair of pixels with focal plane processing and a column decoder element. The size of the active pixel can be significantly decreased with the use of a smaller feature size process.

that more distant pixels have less of an impact on a particular pixel than pixels in the immediate or local neighborhood. This type of smoothing enables sharp, long edges and high spatial contrast to be preserved while image and texture noise are smoothed or blurred out of the image. As the degree of smoothing is increased, the edge detection threshold should increase, because the number of edges smoothed to a value beneath the detection threshold increases with the degree of smoothing. Experimental results are evaluated for both the array with electronic inputs and the array with photodetector inputs. Nominal values for the threshold adjustment parameter V_s and V_{dd} are chosen at 5 V and 0.1 V respectively. The impact of smoothing for an edge located at all three possible locations in the neighborhood element is shown in Fig. 8.

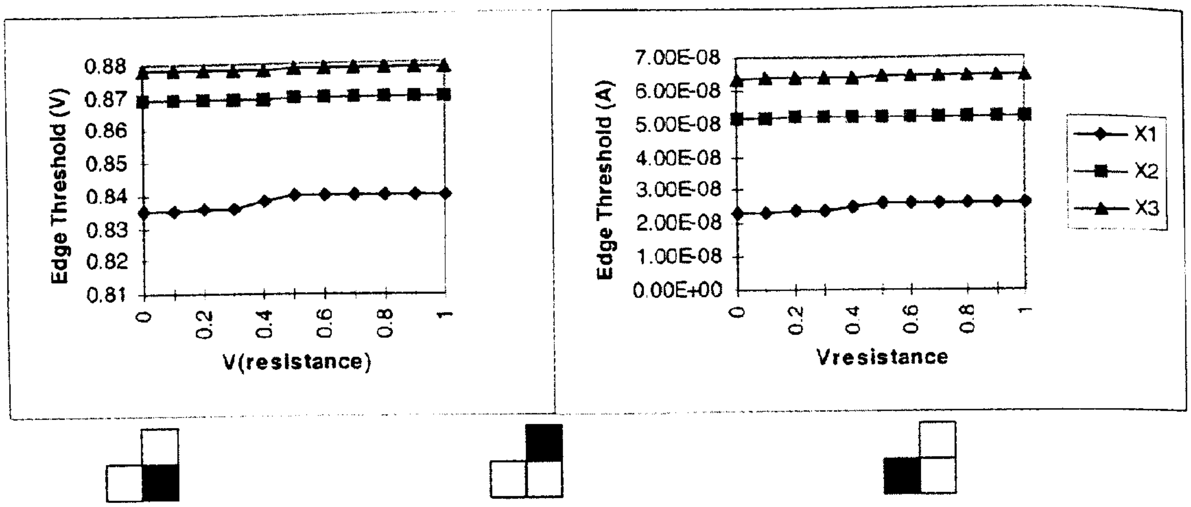
As expected, as smoothing (V_{res}) increases, the edge detection threshold increases. The same edge or contrast appears smaller after smoothing, causing the edge detection threshold to appear as if it has increased. Differences in results due to the position of the "dark" pixel in the neighborhood window are due to the degree of smoothing in horizontal and vertical directions. All pixels at column m or greater are low, while all pixels at column m-1 or smaller are high. As a result, when X1 is the low pixel in the neighborhood window, smoothing occurs in both vertical and horizontal directions, significantly reducing the edge sharpness and lowering the detection threshold. Similarly, when X3 is the low pixel in the neighborhood window, smoothing takes place only in one horizontal direction, maintaining the edge sharpness and increasing the detection threshold. A similar argument may be made for X2, whose edge detection threshold lies between that for X1 and X3.

Smoothing effects are limited in this small array, because even in the center of the array, an input can spread no more than two pixels before it reaches the perimeter of the array. As the array size increases, this effect will be more pronounced, allowing the degree of smoothing to more substantially affect the effective edge detection threshold when other parameters remain constant.

The same fundamental effects of smoothing are observed in the phototransistor array as in the electronic input array. For the same nominal parameters as illustrated in Fig. 8, image results are shown for varying degrees of smoothing in Fig. 9. The input image Fig. 9(a) consists of two point sources of light on a dark background, where one light source is larger and brighter than the second. With no smoothing ($V_{res} = 0$), edges associated with both light sources are detected Fig. 9(b), with smoothing ($V_{res} = 0.5$), however, edges on only the brighter light source are detected, as the dimmer light sources is smoothed into the dark background and fails to meet the edge detection threshold Fig. 9(c).

4.2. *Effect of Adjustable Mean Scaling* (V_s)

Recall that the source voltage V_s controls where the intersection between mean and harmonic mean of the three currents in the neighborhood window occurs. Decreasing the voltage V_s shifts the mean curve (line) upward, thereby decreasing the intersection point of mean and harmonic mean and the edge detection

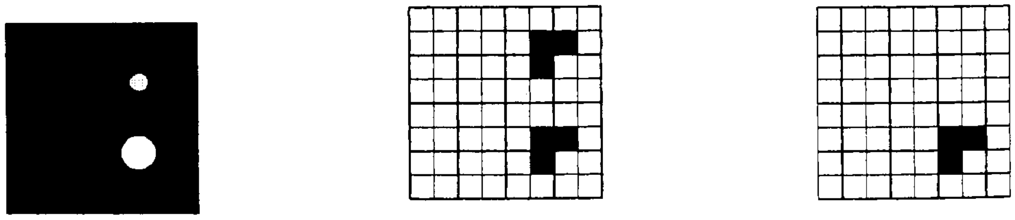


X1 = "Dark" pixel placed as shown X2 = "Dark" pixel placed as shown X3 = "Dark" pixel placed as shown

Fig. 8. Effect of smoothing on edge detection threshold (electrical input array). Shown above is the impact of smoothing (V_{res}) on the edge detection threshold of the array. The edge detection threshold is defined as the difference between "dark" and "light" pixels required to indicate an edge. For this test, all pixels to the right of the windowing element are a nominal high value (equal to an order of magnitude less than the two stationary inputs) and all pixels at and to the left of the element $V_{m,n}$ are at a nominal low value (equal to an order of magnitude lower current than the two stationary inputs). The change in threshold characteristics is dependent on the location of the "dark" pixel in the neighborhood window, due to variations in smoothing caused by the proximity of each location to the other two pixels in the windowing element and to the "high" left and "low" right sides of the array.

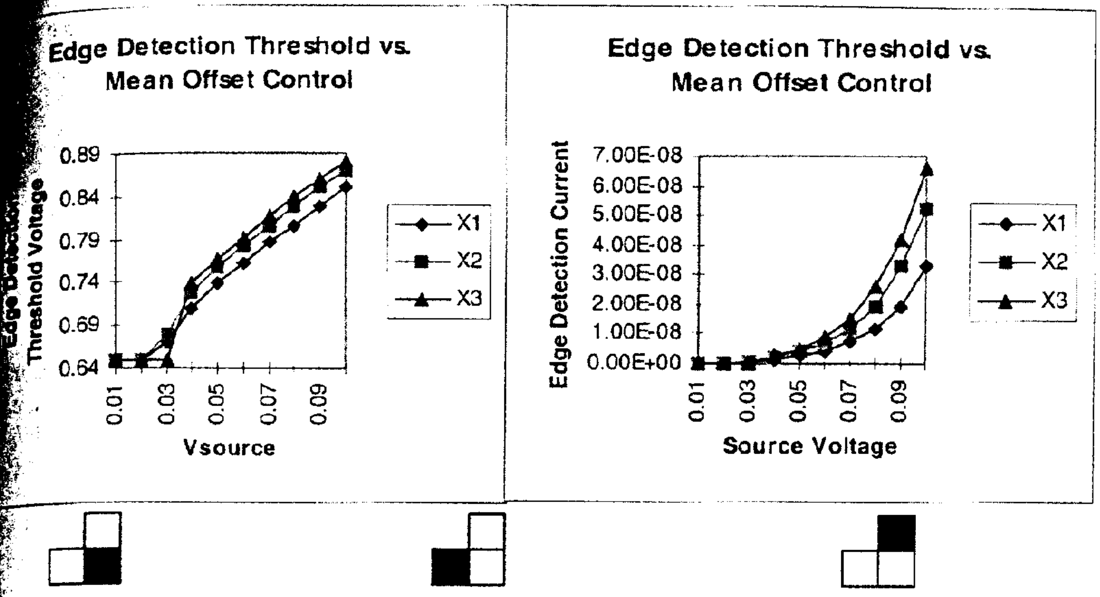
threshold for each windowing element. The mean scaling factor is an exponential function of the voltage V_s, because this factor scales the mean current rather than the mean voltage.

After the harmonic mean saturates, it can be considered relatively insensitive to changes in the inputs as compared to the mean. After the intersection point of harmonic mean and offset mean surpasses the knee of the harmonic mean curve, the edge threshold current increases exponentially with increasing V_s, while the edge detection voltage increases linearly as a result of the exponential relationship between voltage and current in subthreshold operation. This behavior is verified in Fig. 10 for nominal values of V_{dd} and V_{res} (5 V and 0.5 respectively) where the edge detection threshold becomes a linear function of the source voltage V_s after the intersection of mean and harmonic mean surpasses the knee of the harmonic mean curve. Likewise, the edge threshold current is an exponential function of the control voltage V_s beyond



(a) Original Image (b) Output, $V_{\text{res}} = 0$ (c) Output, $V_{\text{res}} = 0.5$

Fig. 9. Effect of smoothing on edge detection threshold (phototransistor input array). Shown above is the thresholded response of the focal plane processing array for varying values of the spreading resistance V_{res}. As expected, as the smoothing increases with V_{res}, the number of edges detected decreases due to the increased edge detection threshold as described in Fig. 8. The (a) input image consists simply of two point sources of light on a dark background. Outputs for (b) and (c) are active low, indicating edges present in the dark blocks.

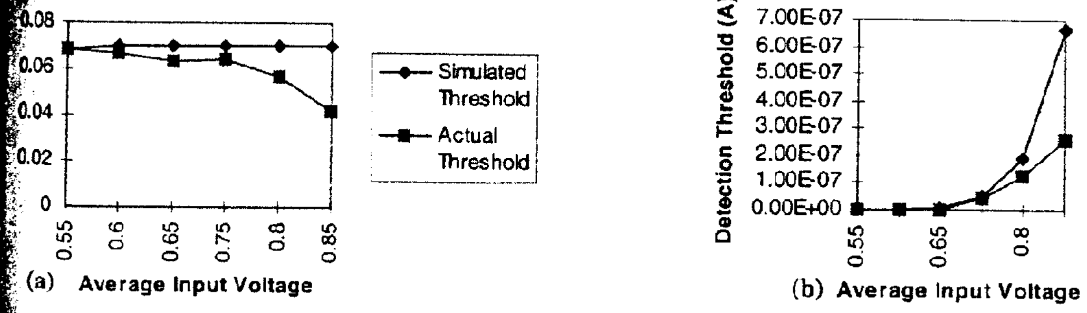


) "Dark" Pixel at $V_{n,m}$ (b) "Dark" Pixel at $V_{n,\,m-1}$ (c) "Dark" Pixel at $V_{n-1,\,m}$

. Effect of mean offset on edge detection threshold (electrical input array). Shown above is the impact of mean offset (V_s) on the etection threshold of the array. The edge detection threshold is defined as the difference between "dark" and "light" pixels required cate an edge. For this test, all pixels at column m or greater outside the windowing element are light while all pixels at column m-1 aller outside the windowing element are "dark".

nee of the harmonic mean curve. Differences in haracteristic curves of Fig. 10 are a result of the tion of the edge in the neighborhood window.

n order for this configuration to be useful in edge ection, it is important that the edge detection shold as a percentage of background (common de) levels remain relatively constant over variams in the inputs. Mathematically, the intersection point of harmonic mean and mean should remain constant over changes in the input range, since the mean is offset by a percentage of its original value. These results are confirmed in the electronic input array of edge detection elements as shown in Fig. 11. Simulated results are calculated, assuming a smoothing effect of 70% (each input in the windowing element loses 30% of its value to smoothing in the



11. Effect of changes in common mode input range on edge detection threshold. Shown above are (a) voltage threshold and (b) ent threshold results for simulated and actual comparisons of the offset mean and harmonic mean. The average input voltage across the re array is shown on the x-axis. The X2 and X3 voltages (refer to Fig. 10) inside the windowing elements for each test are 100mV less n the average array voltage, the center voltage in the windowing element (X1) is swept upward from the X2/X3 value until the detection put for that window becomes active (low).

surrounding pixels) for nominal values of $V_{res} = 0.5$V and a bias of 2.0 V on the transconductance amplifier connecting each input to the resistive network. Inputs on the lefthand (LHS) and righthand (RHS) side of the array are held at 100mV below and 100mV above the average input voltage respectively. Inputs in the windowing element (X1, X2, and X3 as indicated in Fig. 10) are held at the lower value, corresponding to the LHS of the array; the input located at position X1 is increased until an edge is detected at X1. As the RHS (''high'' side) of the array and subsequently the array average moves into above threshold, the simulated and actual results diverge due to the breakdown of the exponential current-voltage relationship upon which simulated results are based. The detection threshold is defined as the current or voltage present at X1 that is required to cause the output at X1 go low (active), indicating the presence of an edge. Nominal values of parameters for this series of tests are $V_{source} = 0.1$ and $V_{dd} = 5.0$V. Across the subthreshold range of operation tested, the detection threshold varies only 7% until inputs begin the transition between subthreshold and above threshold operation.

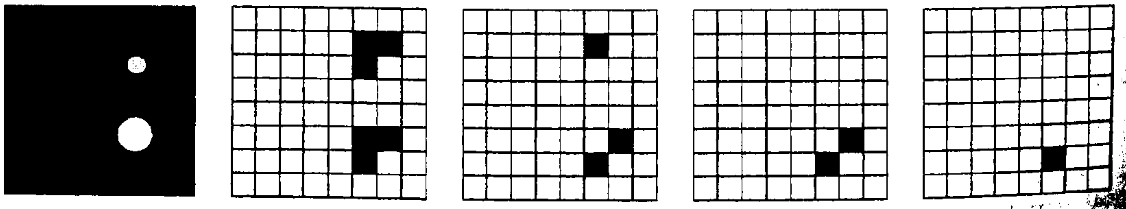
Similar behavior is observed as a function of the mean offset and background levels in the performance of the 8 × 8 phototransistor input array as shown in Fig. 12. The original image Fig. 12(a) consists of two point sources of light, identical to the image used to evaluate the effect of smoothing (Fig. 9). At high values of V_s Fig. 12(b), all edges in both objects are detected, and at low values of V_s Fig. 12(d), only the highest contrast edges in the brightest object are detected. In a higher resolution array, partial edges of objects are less likely to appear as the sensitivity of the edge detection threshold to the source voltage V_s decreases with higher array sizes.

4.3. *Effect of Power Supply Variations* (V_{dd})

For the portable applications targeted in this research, it is important for the edge detection circuits described here to maintain similar behavior as the supply voltage varies across its effective operating range. As shown in Fig. 13, the edge detection threshold varies only about 0.5% per volt of power supply variation, a difference that is insignificant in the detection of edges in the phototransistor input array (Fig. 14). The small dependence of circuit performance on power supply is a direct result of the second-order impact of V_{ds} on the subthreshold current-voltage equations for PFETs. Resilience to power supply fluctuations is well suited to battery powered applications where the supply voltage typically varies a volt or more from the fully charged to low charge modes.

5. Conclusions

We have designed, fabricated, and tested an 8 × 8 light input and a 4 × 4 electronic input array of scale invariant, analog VLSI elements for detecting edges in an L-shaped pixel neighborhood. The threshold for edge detection can be effectively controlled by varying the degree of smoothing across the array or through a constant offset parameter. The elements demonstrate fast response time for subthreshold operation and low power dissipation. Size and fill



(a) Original Image (b) $V_s = 0.1462$ (c) $V_s = 0.1427$ (d) $V_s = 0.1414$ (e) $V_s = 0.1362$

Fig. 12. Effect of mean offset on edge threshold (phototransistor input array). Shown above is the thresholded response of the [illegible] processing array for varying values of the mean offset, source voltage V_s. As this control voltage increases, the edge detection [illegible] decreases from infinity allowing more edges to be detected. As the control voltage decreases, only the highest contr[illegible] [illegible] detected.

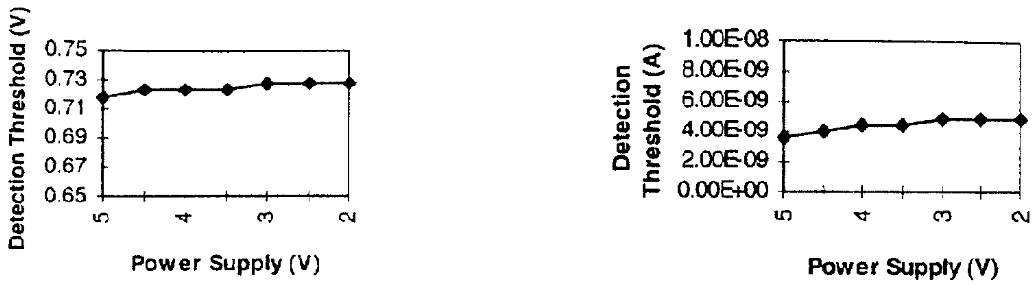
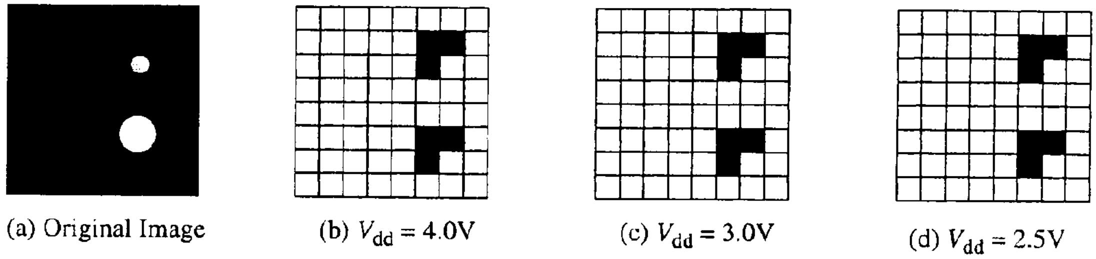


Fig. 13 Effect of V_{dd} on edge detection threshold (phototransistor array). Results are shown for the edge detection threshold expressed as a (a) voltage and (b) current for nominal values of V_{res} (0.5 V) and V_s (0.1 V) and all inputs but one held at 0.65 V. A 0.5%/V variation in threshold occurs across a range of power supply voltages (V_{dd}) from 5 to 2 V.


(a) Original Image

(b) $V_{\rm dd} = 4.0$V

(c) $V_{\rm dd} = 3.0$V

(d) $V_{\rm dd} = 2.5$V

Fig. 14 Effect of V_{dd} on edge detection threshold (phototransistor input array).

factors for the prototype elements are less than optimal and can be improved easily by using a smaller feature size process to enable a 200 × 200 array of these pixels to be fabricated on a cm^2 die.

The elements presented here are compact, low power (6.8 μW per element) and fast (0.67 msec system response time). Applications for these focal plane processing arrays are as a preprocessing tool for more complex image processing systems and as simple segmentation schemes for less complex image processing tasks such as single feature tracking and basic manufacturing inspection tasks. The obvious next steps of this work is to evaluate the AMHM elements on a larger imager, up to 200 × 200 for projection of complex images and evaluation of subsequent edge detection.

References

1. J. C. Russ, *Image Processing Handbook*. CRC Press, Boca Raton, FL, Chapters 3, 4, 5, 6.
2. J. G. Harris, "Continuous time segmentation networks." *Visual Information Processing from Neurons to Chips*, SPIE, 1473, pp. 161–171, 1991.
3. W. Bair and C. Koch, "An analog VLSI chip for finding edges from zero crossings." *Neural Information Processing Systems* 3, pp. 399–405, 1991.
4. J. G. Harris, C. Koch, and J. Luo, "A two-dimensional analog VLSI circuit for detecting discontinuities in early vision." *Science* 248, pp. 1209–1211, 1990.
5. M. D. Rowley and J. G. Harris, "A comparison of three one-dimensional edge detection architectures for analog VLSI vision systems." *IEEE International Symposium on Circuits and Systems*, Hong Kong, 9–12, 1997, pp. 1840–1843.
6. T. Ando, K. Yamamoto, and K. Sawada, "Contrast edge detection using differential input technique of a CCD." *Electronics Letters* 28(3), 2190–2191, 1992.
7. J. G. Harris, C. Koch, and J. Luo, "Resistive fuses analog hardware for detecting discontinuities in early vision." In *Analog VLSI Implementation of Neural Systems*, pp. 27–55 Kluwer, London, 1989.
8. C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley Reading, MA, Chapter 3, 1989.
9. W. Bair, and C. Koch, "Real-time motion detection using an analog VLSI zero-crossing chip." *Proceedings of the SPIE* 1473, pp. 59–65, 1991.
10. M. Rowley and J. Harris, "An edge enhancement technique for analog VLSI early vision applications." *IEEE International Conference on Neural Networks* Washington DC, pp. 1000–1004, 1996.
11. T. Delbruck and C. A. Mead, "Photoreceptor circuit with wide dynamic range." *International Circuits and Systems Conference*. London, England, 1994.

12. Andreou, A. G., K. Strohbehn, and R. E. Jenkins, "Silicon retina for motion computation." *Proc IEEE Intl Symp Circuits and Systems*, Singapore, pp. 1373–1376, 1991.
13. T. Delbruck, "Silicon retina with correlation-based velocity-tuned pixels." *IEEE Trans on Neural Networks* 4(3), 529–541, 1993.
14. J. Kramer, R. Sarpeshkar, and C. Koch, "Analog VLSI motion discontinuity detectors for image segmentation." *IEEE Symposium on Circuits and Systems*, Atlanta, GA, pp. 620–623, 1996.
15. R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "A focal plane visual motion measurement sensor." *IEEE Trans on Circuits and Systems. I, Fundamental theory and applications* 44(1), pp. 55, 1997.
16. C. Koch, A. Moore, W. Bair, T. Horiuchi, B. Bishofberger, and J. Lazarro, "Computing motion using analog VLSI vision chips. An experimental comparison among four approaches." *Proc IEEE Workshop on Visual Motion*, pp. 312–324, 1991.
17. J. Tanner, and C. Mead, "An integrated optical motion sensor." *VLSI Signal Processing II* IEEE Press, New York, pp. 59–76, 1986.
18. A. Momi, A. Bouzerdoum, K. Eshraghian, A. Yakovleff, X. T. Nguyen, A. Blanksby, R. Beare, D. Abbott, and R. E. Bogner, "An insect vision-based motion chip." *IEEE J Solid State Circuits* 32(2), pp. 279–284, 1997.
19. R. Sarpeshkar, W. Bair, and C. Koch, "Visual motion computation in analog VLSI using pulses." *Advances in Neural Information Processing Systems* 5, pp. 781–788, 1993.
20. D. Standley, "An object position and orientation. IC with embedded imager." *IEEE Journal of Solid State Circuits* 26(12), 1991.
21. S. DeWeerth, "Analog VLSI circuits for stimulus localization and centroid computation." *International Journal of Computer Vision* 8, pp. 191–202, 1992.
22. T. G. Morris, D. M. Wilson, and S. P. DeWeerth, "Analog VLSI circuits for manufacturing inspection." *1995 Advanced Research in VLSI* IEEE Computer Society Press, Los Alamitos, California, pp. 241–255, 1995.
23. T. G. Morris and S. P. DeWeerth, "Analog VLSI arrays for morphological image processing." *ASAP* San Francisco, California, pp. 32–142, 1994.
24. T. Delbruck, "Bump circuits for computing similarity and dissimilarity of analog voltages.", *Proceedings of International Joint Conference on Neural Networks* Seattle, WA, July 8–12, pp. I-475–I-479, 1991.
25. B. C. Harris and S. P. DeWeerth, "Analog encoding circuits for digital CMOS neural oscillator." *ISCAS '95*, Seattle, Washington, 1995.

Denise Michelle Wilson was born in Chicago, Illinois, U.S. in 1966. She received her Ph.D. and M.S. in Electrical Engineering from the Georgia Institute of Technology in 1995 and 1989 respectively and has a B.S. in Mechanical Engineering from Stanford University (1988). She is currently an assistant professor at the University of Washington (Seattle, WA, U.S.) in the department of electrical engineering and worked previously at the University of Kentucky in a similar position from 1996–1998. Her research interests focus on the development of signal processing architectures, array platforms and other infrastructure for visual, auditory and chemical sensing microsystems.

Professor Wilson has also worked for several years for Applied Materials, a semiconductor capital equipment supplier from 1990–1992. She has been a member of IEEE since 1989.