

A Novel Technique for Image-Velocity Computation

Chu Phoon Chong, *Member, IEEE*, C. Andre T. Salama, *Fellow, IEEE*, and Kenneth C. Smith

Abstract—A novel technique for image-velocity computation is described. Using one set of intersecting delay lines that are physically perpendicular to each other, the two perpendicular components of image velocity can be computed separately if the image area is known. To compute the image velocity without knowledge of the image area, two sets of intersecting delay lines are required. This technique is suitable for analog VLSI implementation.

I. INTRODUCTION

DIGITAL image processing generally requires a large amount of high-speed hardware, which, in many cases, is highly specialized [1]. Image-velocity computation using conventional digital-signal processing (DSP) techniques is slow even if state-of-the-art high-speed digital hardware is used. Conventional DSP techniques require the digitization of two image frames. The digitized data of at least one frame must be stored in RAM before any processing can be performed. This requires the use of both a high-speed A/D converter and a RAM. Moreover, the difficulty of integrating an A/D converter with each pixel leads to sequential image digitization. This severely limits the parallel processing capability of the associated digital processor and, in most cases, leads to distortion of the moving image. Specialized digital hardware is used to implement algorithms that compute optical flow in the image. Such algorithms are highly computationally intensive [2], [3]. Thus, it is very difficult and, in most cases, impossible to achieve real-time image-velocity computation.

Focal-plane image processing using analog VLSI systems has been shown to be a viable alternative [4]–[6]. Although the pixel densities of imagers with built-in focal-plane image-processing capabilities (smart imagers, for short) are not as high as those of conventional CCD imagers, the preprocessing functions performed can significantly reduce the amount of computation of the associated postprocessor. In some cases, the smart imager can operate in a stand-alone mode. This possibility leads to new applications of smart imagers where additional computing and memory resources are not available.

Analog VLSI systems employing focal-plane image-velocity computation have been described earlier by Mead

[3]. These systems generate a velocity-space map of a moving image. However, the transistor count per pixel is greater than 60 and the pixel density is low. Moreover, voltage multipliers, which are difficult to implement in CMOS without a large input offset voltage, are required.

In this paper, a new image-velocity computation technique that is well-suited for incorporation into smart imagers using analog VLSI techniques is introduced. The detailed analysis described has served as the foundation for a successful silicon implementation using CMOS technology. The design and implementation of the imager are described in an accompanying paper [7].

II. DELAY-TIME-BASED IMAGE-VELOCITY COMPUTATION

A novel image-velocity computation technique is proposed here, based on multiple signals propagating on parallel delay lines. A delay line consists of a series of individual delay elements connected in series to slow the speed of propagation of a signal in the line. In order to control the speed of propagation of the signal in the delay line, the delay elements must all have precisely controlled delay. Ideally, all signals should leave one end and arrive at the other end of the delay lines simultaneously.

When a moving image is projected onto a set of intersecting delay lines as shown in Fig. 1, some signals will cross the image boundary and enter the region inside the image boundary. The length of time that a signal stays inside the image boundary depends on the relative velocity of the signal with respect to the image. Consider the positive x -direction delay lines only. If the lengths of time during which all signals stay inside the image boundary are summed, the total time is (see Appendix A)

$$T_{1x} = \frac{A_{IM}}{\mu(V_s - V_x)} \quad V_s > V_x \quad (1)$$

where A_{IM} is the area of the image, μ is the distance between two pixels, V_s is the signal velocity, and V_x is the x component of the image velocity V_{IM} . Since T_{1x} is not a function of V_y (the y component of the image velocity V_{IM}), the x component of V_{IM} can be computed if T_{1x} can be measured and all parameters (A_{IM} , μ , and V_s) are known.

Similarly, for the positive y -direction delay lines, the sum of the lengths of time all signals propagating on the

Manuscript received April 10, 1990; revised February 29, 1992. Paper was recommended by Associate Editor Hua Lee.

The authors are with the Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada M5S 1A4.
IEEE Log Number 9201706.

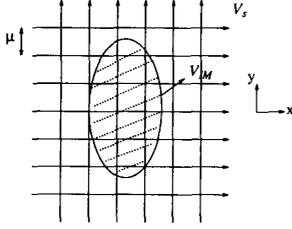


Fig. 1. A moving elliptical image is projected onto a set of intersecting delay lines.

delay lines stay inside the image boundary is

$$T_{1y} = \frac{A_{IM}}{\mu(V_s - V_y)} \quad V_s > V_y \quad (2)$$

Thus, if T_{1x} and T_{1y} can be measured, the image velocity V_{IM} can be computed.

Using (1) and (2) to compute V_{IM} requires knowledge of A_{IM} . If A_{IM} is not known, two sets of intersecting delay lines are required. One set consists of positive x -direction and positive y -direction delay lines and the other set consists of negative x - and negative y -direction delay lines as shown in Fig. 2. The expression for T_1 associated with the negative x - and negative y -direction delay lines are, respectively,

$$T_{1xN} = \frac{A_{IM}}{\mu(V_x + V_x)} \quad (3)$$

and

$$T_{1yN} = \frac{A_{IM}}{\mu(V_s + V_y)}. \quad (4)$$

The subscripts xN and yN denote the negative x and negative y directions of the delay lines, respectively.

Combining (1)–(4) results in

$$V_x = \frac{(T_{1x} - T_{1xN})V_s}{T_{1x} + T_{1xN}} \quad (5)$$

and

$$V_y = \frac{(T_{1y} - T_{1yN})V_s}{T_{1y} + T_{1yN}}. \quad (6)$$

Thus, to compute V_x and V_y , only T_{1x} , T_{1xN} , T_{1y} , T_{1yN} , and V_s must be known.

III. IMAGE-AREA QUANTIZATION AND COMPUTATIONAL ERROR

Our analysis so far has assumed that T_{1y} is a continuous function of A_{IM} . However, the resolution of the image is finite and the image area “seen” by the imager is a quantized value of the actual image area. The quantized image area is denoted by A_{IMq} .

If only one set of intersecting delay lines is used, the computational error $E_r(V_x)$ due to image-area quantiza-

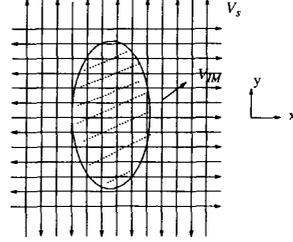


Fig. 2. Two sets of intersecting delay lines are used to compute the image velocity.

tion is

$$E_r(V_x) = \left(\frac{V_s}{V_x} - 1 \right) \left(1 - \frac{A_{IMq}}{A_{IM}} \right). \quad (7)$$

Equation (7) shows that for low values of V_x , $E_r(V_x)$ is large if $A_{IMq} \neq A_{IM}$. Thus, to compute the velocity of an image that moves slowly, either an imager with a very high pixel density may be used, so that $A_{IMq} \approx A_{IM}$, or the signal velocity V_s must be set to a low value resulting in an increase in the period of computation.

If two sets of intersecting delay lines are used, the quantized image area “seen” by the positive x -direction delay lines may not be equal to that “seen” by the negative x -direction delay lines. Let A_{IMq}^+ and A_{IMq}^- denote the quantized image areas “seen” by the positive x -direction and the negative x -direction delay lines, respectively. The computation error $E_r(V_x)$ for this case is

$$E_r(V_x) = \frac{[1 - (A_{IMq}^-/A_{IMq}^+)] [1 - (V_s/V_x)^2]}{[1 - (A_{IMq}^-/A_{IMq}^+)] + [1 + (A_{IMq}^-/A_{IMq}^+)] [(V_s/V_x)]} \quad (8)$$

It follows that if $A_{IMq}^- = A_{IMq}^+$, $E_r(V_x)$ is zero even if V_x is low. Comparing (7) and (8) leads to the following conclusions: if one set of intersecting delay lines is used, the computational error is a function of image-area quantization. However, if two sets of intersecting delay lines are used, the computational error is a function of only the mismatch of the image-area quantizations.

The quantized image area can be expressed as

$$A_{IMq} = \left(1 - \frac{V_x}{V_s} \right) \cdot \sum_{A_{IM}} \left[\mu^2 Q \left(\frac{d_j^i}{\mu \left\{ [(V_y/V_s)]^2 + [1 - (V_x/V_s)]^2 \right\}^{1/2}} \right) \right] \quad (9)$$

where d_j^i is the length of the j th delay line covered by the image. The operator $Q[x]$ gives the quantized value of x . The summation sign $\sum_{A_{IM}}$ indicates that only delay lines intersecting the image are considered.

The image-area quantization error is a function of the orientation of the image. When the shape of the image is irregular and/or $V_y \neq 0$, the quantized image area is very

difficult to calculate without using numerical methods. However, it is possible to estimate the limits of the quantization error in the image area. Consider the case of an image with arbitrary shape shown in Fig. 3. When the image is projected onto a vertical plane to form a line, the longest and the shortest line projections are denoted by l_M and l_m , respectively. The angles of the longest and the shortest lines with respect to the y axis are θ_M and θ_m , respectively. Assuming that each delay line produces a worst-case image-area quantization error of μ^2 (the unit area), the quantization error in A_{IM} , $E_q(A_{IMq}) = (A_{IM} - A_{IMq})/A_{IM}$ lies within the limits

$$\frac{l_m \mu}{A_{IM} \cos \theta_m} < E_q(A_{IMq}) < \frac{l_M \mu}{A_{IM} \cos \theta_M}. \quad (10)$$

As a special case, consider a rectangular image with dimensions l by w oriented in a direction such that its edges are parallel to the x and y axes, as shown in Fig. 4. The velocity of the image $V_{IM} = V_x$ and $V_y = 0$. In this case the apparent width of the image is w and the apparent length of the image is $l/(1 - V_x/V_s)$. The quantized apparent image area is

$$A'_{IMq} = \mu^2 Q\left[\frac{w}{\mu}\right] Q\left\{\frac{l}{\mu[1 - (V_x/V_s)]}\right\} \quad (11)$$

It follows that the quantized image area is

$$A_{IMq} = \mu^2 \left(1 - \frac{V_x}{V_s}\right) Q\left[\frac{w}{\mu}\right] Q\left\{\frac{1}{\mu[1 - (V_x/V_s)]}\right\}. \quad (12)$$

If $V_y \neq 0$, then (10) can be used to estimate the limits of the image-area quantization error. For this special case, (10) becomes

$$\frac{\mu}{l} < E_q(A_{IMq}) < \frac{\mu}{l} \left[1 + \left(\frac{l}{w}\right)^2\right]. \quad (13)$$

Combining (7) and (13) leads to

$$\left(\frac{V_s}{V_x} - 1\right) \frac{\mu}{l} < E_r(V_x) < \left(\frac{V_s}{V_x} - 1\right) \frac{\mu}{l} \left[1 + \left(\frac{l}{w}\right)^2\right]. \quad (14)$$

Thus, if the pixel density is high and an appropriate value of V_s is chosen, the computational error due to image-area quantization is low.

IV. MEASUREMENT OF T_1 USING LIGHT-INTENSITY-MODULATED DELAY LINES

There are several techniques available for the measurement of T_1 . Each implies a different pixel density and a different accuracy of measurement. The technique to be described here is based on a delay element whose delay (one of two possible values) is selected (or modulated) by the intensity of light falling onto the pixel. The delay element has a delay denoted by τ_1 inside the image boundary, and a delay τ_0 outside the image boundary (in the case under consideration, $\tau_1 \gg \tau_0$).

If the image is traveling in the x direction at a velocity

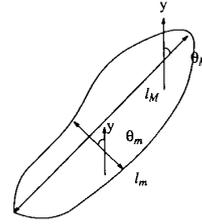


Fig. 3. An image with an arbitrary shape oriented in a particular direction.

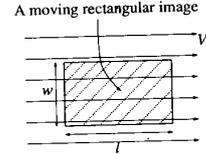


Fig. 4. A moving rectangular image is projected onto the delay lines.

V_x , the sum of the delays of all delay lines T_2 can be expressed as

$$T_{2x} = \frac{A_T}{\mu^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}}{(\mu^2/\tau_1)[1 - (V_x\tau_1/\mu)]} \quad (15)$$

where A_T is the total area of the imager and the subscript x appended to T_2 indicates the direction of the delay lines.

Similarly,

$$T_{2y} = \frac{A_T}{\mu^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}}{(\mu^2/\tau_1)(1 - (V_y\tau_1/\mu))}. \quad (16)$$

If two sets of intersecting delay lines are used, V_x and V_y can be expressed as

$$V_x = \frac{(T_{2x} - T_{2xN})(\mu/\tau_1)}{T_{2x} + T_{2xN} - (2A_T/\mu^2/\tau_0)} \quad (17)$$

and

$$V_y = \frac{(T_{2y} - T_{2yN})(\mu/\tau_1)}{T_{2y} + T_{2yN} - (2A_T/\mu^2/\tau_0)} \quad (18)$$

where the subscript N appended to T_2 indicates the negative (x or y) direction of the delay lines.

The main disadvantage of using light-intensity-modulated delay lines to measure T_1 is the "bending" of the delay line at the image boundary due to different velocities of propagation inside and outside the image boundary. (The frame of reference is the image itself.) This situation resembles that of light crossing the boundary between air and glass. If the angles of incidence are not equal for all signals, as in the case of a square image moving on the imager shown in Fig. 5, the sum of the

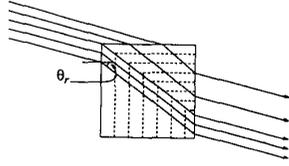


Fig. 5. The refraction of signals crossing the boundary of a square image.

delays of x -direction delay lines can be expressed as

$$T_{2x} = \frac{A_T}{\mu^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}}{(\mu^2/\tau_1)[1 - (V_x\tau_1/\mu)]} + \epsilon_T \quad (19)$$

where ϵ_T is an error term given by

$$\epsilon_T = - \left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}[(V_y\tau_1/\mu)]}{2(\mu^2/\tau_1)[1 - (V_x\tau_0/\mu)][1 - (V_x\tau_1/\mu)]^2} \quad (20)$$

on the assumption that the angle of refraction θ_r is in the range $45^\circ < \theta_r < 90^\circ$.

Neglecting ϵ_T , the approximate value of V_x , V_{xe} is given by

$$V_{xe} = \frac{\mu}{\tau_1} \left(1 - \frac{[1 - (\tau_0/\tau_1)]A_{IM}}{(\mu^2/\tau_1)\{T_{2x} - [A_T/(\mu^2/\tau_0)]\}}\right) \approx V_x \quad (21)$$

If $V_x, V_y \ll \mu/\tau_1$, the error in V_{xe} due to signal refraction is strongly a function of V_y/V_x and is only weakly dependent on the absolute value of V_x . For example, using $\mu = 127.5 \mu\text{m}$, $A_T = 14.44 * 10^6 \mu\text{m}^2$, $A_{IM} = 1 * 10^6 \mu\text{m}^2$, $\mu/\tau_0 = 12.75 * 10^6 \mu\text{m/s}$ and $\mu/\tau_1 = 25.56 * 10^3 \mu\text{m/s}$, plots of the errors in V_{xe} , as a function of V_y/V_x and V_x are shown in Figs. 6 and 7, respectively. From the plots, the error in V_{xe} is approximately $V_y/2V_x$. However, the error in V_{xe} becomes significantly greater than $V_y/2V_x$ if V_x approaches μ/τ_1 .

The fact that the errors in V_{xe} and V_{ye} are $V_y/2V_x$ and $V_x/2V_y$, respectively, leads to an error-correction scheme based on the following equations:

$$V_{xec} = \frac{4}{3}V_{xe} \left(1 + \frac{V_{ye}}{2V_{xe}}\right) \quad (22)$$

and

$$V_{yec} = \frac{4}{3}V_{ye} \left(1 + \frac{V_{xe}}{2V_{ye}}\right) \quad (23)$$

where V_{xec} and V_{yec} are the corrected values of V_{xe} and V_{ye} . After error correction, the errors in V_{xe} and V_{ye} , as functions of V_y/V_x , are shown in Figs. 8 and 9, respectively. After error correction has been performed, the error in V_{xe} is reduced from over 100% to less than 7%.

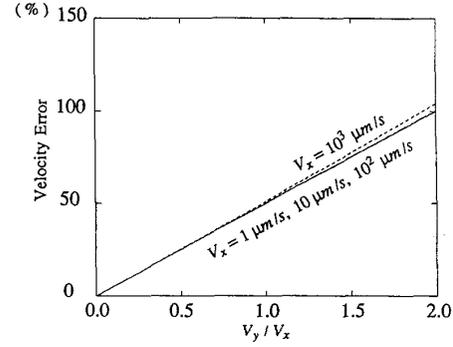


Fig. 6. The velocity error versus V_y/V_x . The value of V_x varies from 1 to $10^3 \mu\text{m/s}$.

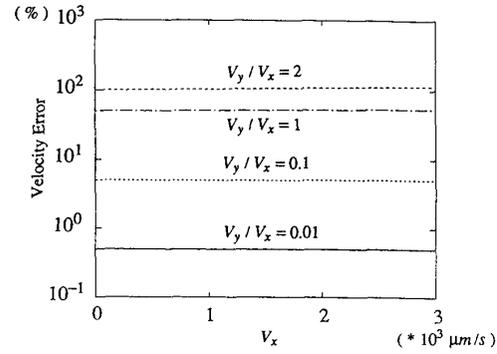


Fig. 7. The velocity error as a function of V_x , for values of V_y/V_x ranging from 0.01 to 2.

Note that the error in V_{xe} increases as V_y/V_x increases. Similarly, the error in V_{ye} increases as V_x/V_y increases, or conversely, as V_y/V_x decreases, as shown in Fig. 9. The error in V_{ye} is less than 5% if V_y/V_x is greater than 0.5.

If two sets of intersecting delay lines are used, V_{xe} and V_{ye} are given by

$$V_{xe} = \frac{(T_{2x} - T_{2xN})(\mu/\tau_1)}{T_{2x} + T_{2xN} - (2A_T/\mu^2/\tau_0)} \quad (24)$$

and

$$V_{ye} = \frac{(T_{2y} - T_{2yN})(\mu/\tau_1)}{T_{2y} + T_{2yN} - (2A_T/\mu^2/\tau_0)} \quad (25)$$

The errors in V_{xe} and V_{ye} due to refraction of signals are small provided that $V_x\tau_1/\mu, V_y\tau_1/\mu \ll 1$. A plot of the errors in V_{xe} and V_{ye} versus V_y/V_x is shown in Fig. 10. As indicated, the errors in V_{xe} and V_{ye} are less than 5% if V_x and V_y are less than $10^3 \mu\text{m/s}$. For large values of V_x and V_y , the errors in V_{xe} and V_{ye} can be as large as 25% (for $V_x = 5 * 10^3 \mu\text{m/s}$ and $V_y = 10^4 \mu\text{m/s}$). However, an error correction process to be now described may reduce the error significantly:

If $(V_x\tau_1/\mu)^2 \ll 1$, the relationship between V_x and V_{xe}

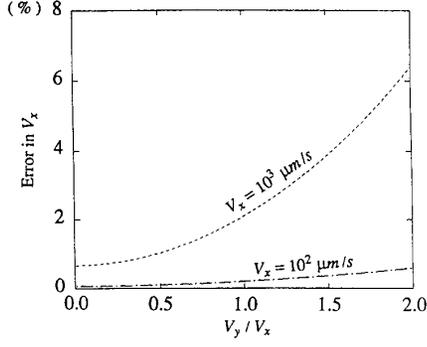
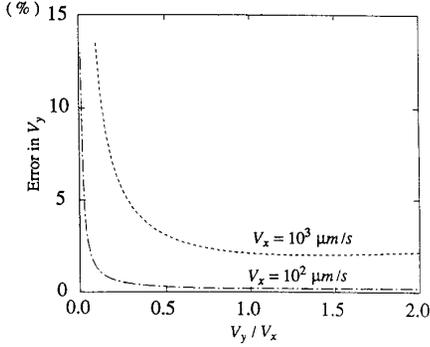
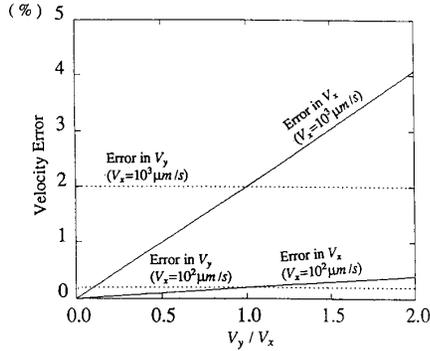

 Fig. 8. The error in computed V_x after error correction is performed.

 Fig. 9. The error in computed V_y after error correction is performed.


Fig. 10. The error in velocity obtained using (24) and (25).

can be approximated by [see Appendix B]

$$V_{xe} \approx 2V_x \left[\frac{1 - (V_y \tau_1 / \mu)}{2 - (V_y \tau_1 / \mu)} \right] \quad (26)$$

The corrected value of V_{xe} , denoted by V_{xec} , can be expressed as

$$V_{xec} = \frac{V_{xe} [2 - (V_{ye} \tau_1 / \mu)]}{2 [1 - (V_{ye} \tau_1 / \mu)]} \quad (27)$$

Note that (27) is obtained by substituting V_{xec} and V_{yec} for V_x and V_y in (26), respectively.

Similarly,

$$V_{yec} = \frac{V_{ye} [2 - (V_{xe} \tau_1 / \mu)]}{2 [1 - (V_{xe} \tau_1 / \mu)]} \quad (28)$$

Using (27) the error for the case of $V_x = 5 \times 10^3 \mu\text{m/s}$ and $V_y = 10^4 \mu\text{m/s}$ is reduced from 24% to less than 5%.

V. THE EFFECT OF A DELAY-LINE MISMATCH ON THE ACCURACY OF VELOCITY MEASUREMENT

If each delay line has a different signal velocity, the apparent length of each delay line under the image d'_i can be expressed as

$$d'_i = \frac{d_i}{1 - (V_x \tau_{1e} / \mu)} \quad (29)$$

where τ_{1e} is the average delay of the delay elements in the delay line.

Using (29), the apparent area of the image can be expressed as

$$A'_{IM} = \frac{A_{IM}}{1 - (V_x \tau_1 / \mu)} + \mu \sum_{A_{IM}} (\epsilon_d) \quad (30)$$

where

$$\epsilon_d = d_i \left\{ \frac{(V_x / \mu)(\tau_{1e}^j - \tau_1)}{[1 - (V_x \tau_1 / \mu)][1 - (V_x \tau_{1e}^j / \mu)]} \right\} \quad (31)$$

and τ_{1e}^j is the average delay of the delay elements on the j th delay line inside the image area A_{IM} .

Close examination of (31) shows that the mismatch contribution in a delay line to the error in A'_{IM} is proportional to the product term $d_i(\tau_{1e}^j - \tau_1)$. If the process-parameter variation is highly random, the longer the length of delay line under the image, the closer the value of τ_{1e}^j to its statistical mean τ_1 . Therefore, an increase in d_i is offset by a decrease in $(\tau_{1e}^j - \tau_1)$. However, if the process-parameter variation is systematic (which leads to the grouping of delay elements of a particular delay into well-defined regions), a delay line with a long d_i might contribute significantly to the error in A'_{IM} . In short, to achieve high accuracy in velocity computation using the imager proposed here, the systematic process-parameter variation should be reduced as much as possible.

Using (30), the computed velocity can be expressed as

$$V_x = \frac{(T_{2x} - T_{2xN})(\mu / \tau_1)}{T_{2x} + T_{2xN} - (2A_T / \mu^2 / \tau_0)} + \epsilon_v \quad (32)$$

where the error term ϵ_v is given by

$$\epsilon_v = \frac{-2[1 - (\tau_0 / \tau_1)] \sum_{A_{IM}} (\epsilon_d)}{T_{2x} + T_{2xN} - (2A_T / \mu^2 / \tau_0)} \quad (33)$$

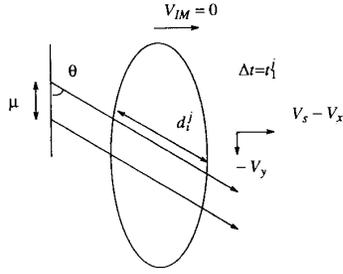


Fig. 11. A moving elliptical image is projected onto the delay lines. The frame of reference is the image itself.

If $\tau_0 \ll \tau_1$, $V_x \tau_1 / \mu \ll 1$ and $\tau_{1e}^j = \tau_{1e}$ for all values of j , (33) reduces to

$$\epsilon_v = V_x \left(1 - \frac{\tau_{1e}}{\tau_1} \right). \quad (34)$$

Thus, the percentage error in the computed velocity is equal to the percentage mismatch of the average delays of the delay lines.

VI. CONCLUSION

A novel technique for image-velocity computation has been described, along with a detailed analysis of its limitations. In actual implementations using currently available technology, the largest source of computational error is expected to be delay-line mismatch. The analysis described in this paper has served as the basis for a silicon implementation of an imager with built-in image-velocity computation capability, as described in an accompanying paper.

APPENDIX A

Consider the case of a moving elliptical image projected onto the x -direction delay lines shown in Fig. 11. The area of the image is A_{IM} . The frame of reference is the image itself. A signal propagating on the j th delay line stays inside the image boundary for a time interval of t_i^j . Let d_i^j be the length of the part of the j th delay line that is inside the image boundary, then

$$t_i^j = \frac{d_i^j}{(V_y^2 + (V_s - V_x)^2)^{1/2}}. \quad (A1)$$

Multiplying both the numerator and the denominator by $\mu \sin \theta$ leads to

$$t_i^j = \frac{d_i^j \mu \sin \theta}{\mu (V_s - V_x)} \quad (A2)$$

where θ is the angle between the resulting signal velocity and the y axis.

The sum of all possible values of t_i^j , which is denoted by T_{1x} , is given by

$$T_{1x} = \sum_{A_{IM}} t_i^j = \frac{\sum_{A_{IM}} (d_i^j \mu \sin \theta)}{\mu (V_s - V_x)} \quad (A3)$$

where the summation sign indicates that only signals that propagate on delay lines that intersect the image boundary are considered.

If μ is small enough,

$$\sum_{A_{IM}} (d_i^j \mu \sin \theta) = A_{IM}. \quad (A4)$$

Combining (A3) and (A4) leads to

$$T_{1x} = \frac{A_{IM}}{\mu (V_s - V_x)}. \quad (A5)$$

APPENDIX B

If $\tau_0 \ll \tau_1$ and $\mu / \tau_0 \gg V_x$, the error in T_{2x} due to signal refraction can be approximated by

$$\epsilon_T \approx - \frac{A_{IM} [(V_y \tau_1 / \mu)]}{2(\mu^2 / \tau_1) [1 - (V_x \tau_1 / \mu)]^2}. \quad (B1)$$

Using (16),

$$T_{2x} = \frac{A_T}{\mu^2 / \tau_0} + \frac{A_{IM}}{(\mu^2 / \tau_2) [1 - (V_x \tau_1 / \mu)]} - \frac{A_{IM} [(V_y \tau_1 / \mu)]}{2(\mu^2 / \tau_1) [1 - (V_x \tau_1 / \mu)]^2} \quad (B2)$$

Substituting (B2) and the similar expression to T_{2xN} into (24), leads to

$$V_{xe} = 2V_x \left(\frac{\{1 - [(V_x \tau_1 / \mu)]^2\} - [(V_y \tau_1 / \mu)]}{2\{1 - [(V_x \tau_1 / \mu)]^2\} - (V_y \tau_1 / \mu)} \cdot \{1 + [(V_x \tau_1 / \mu)]^2\} \right). \quad (B3)$$

If $(V_x \tau_1 / \mu)^2 \ll 1$, (B3) reduces to

$$V_{xe} \approx 2V_x \left[\frac{1 - (V_y \tau_1 / \mu)}{2 - (V_y \tau_1 / \mu)} \right]. \quad (B4)$$

REFERENCES

- [1] I. N. Parker, "VLSI architecture," in *VLSI Image Processing*, R. J. Offien, Ed. London: Collins, 1985.
- [2] H. Buxton and J. Wijek, "Towards computer visions," in *VLSI Image Processing*, R. J. Offien, Ed. London: Collins, 1985.
- [3] C. A. Mead, *Analog VLSI and Neural Systems*. MA: Reading, Addison-Wesley, 1989, Chapter 14.
- [4] M. A. Sivilotti, M. A. Mahowald, and C. A. Mead, "Real-time visual computations using analog CMOS processing arrays," *Stanford Conf. on Advanced Research in VLSI*. Cambridge, MA: The MIT Press, 1987, pp. 295-312.
- [5] S. P. DeWeerth and C. A. Mead, "A two-dimensional visual tracking array," *Fifth MIT Conf. Advanced Research in VLSI*. Cambridge, MA: The MIT Press, 1988, pp. 259-275.
- [6] C. B. Umminger and S. P. DeWeerth, "Implementing gradient following in analog VLSI," *Decennial Caltech Conf. on VLSI*. Cambridge, MA: The MIT Press, 1989, pp. 195-208.
- [7] C. P. Chong, C. A. T. Salama, and K. C. Smith, "An imager with built-in image velocity computation capability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 3, pp. 306-312, Sept. 1992.

Chu Phoon Chong (M'91), C. Andre T. Salama (F'87), and Kenneth C. Smith, for photographs and biographies, see p. 312 of this issue of this TRANSACTIONS.