ON THE ACCURACY OF IMAGE VELOCITY COMPUTATION USING DELAY LINES

Chu Phoon Chong, C. Andre T. Salama and Kenneth C. Smith
Department of Electrical Engineering
University of Toronto
Toronto, Ontario
Canada M5S-1A4

Abstract

This paper describes the accuracy and limitations of image velocity computational technique based on delay lines. The major computational errors encountered so far are due to image-area quantization, delay-line mismatches and signal refraction at the image boundary.

I. Introduction

The human capacity to move around without colliding with either stationary or moving objects is largely a result of the parallel image processing performed by the central nervous systems aided by preprocessing performed by the retina. In the future, robots that move autonomously will require a similar capacity.

Image processing in artificial vision systems is generally performed using digital-signal processing (DSP) techniques [1]. To perform an image-velocity computation using DSP technique requires that at least one image frame be digitized and stored in a high speed RAM first before processing can begin. Additional memory is also required to store any intermediate result produced. The algorithms used are computationally intensive and require the use of high-speed real-time processors [2,3]. Thus, using DSP technique usually leads to high power dissipation and bulky system integration.

Alternative analog systems without the particular shortcomings associated with DSP systems have been reported recently [4,5]. While an analog system in which a processoring element is integrated within each pixel does not have as high a pixel density as that of a simpler CCD imager, the preprocessing performed by analog preprocessing can greatly reduce the computational intensity of an associated digital post processor.

In this paper, the accuracy and the limitations of the image-velocity computation technique using delay lines as described in [5] are discussed.

II. Delay-line based image-velocity computation

A set of delay lines running in the positive x-direction is represented in Fig. 1. A signal, injected into each delay line at the left, propagates on the delay line to the right at a well-controlled velocity denoted by V_s. If a moving image of area A_{IM} is projected onto the set of delay lines, the sum of the time intervals during which signals on all lines stay inside the image boundary, T_{1x}, has been shown to be [5]

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

where μ is the distance of separation between two adjacent delay lines, and V_x is the x-component of the image velocity V_{IM}.

If two sets of delay lines are used (one running in the positive x-direction and the other, the negative x-direction, as shown in Fig. 2), the x-component of the image velocity can be computed using

$$V_x = \frac{(T_{1x} - T_{1xN}) V_s}{T_{1x} + T_{1xN}} \qquad (2)$$

where T_{1xN} is the sum of the time intervals during which signals on all negative x-direction lines stay inside the image boundary. Thus, if T_{1x} and T_{1xN} can be measured, image velocity can be computed using two sets of delay lines without knowledge of either the orientation, the shape, or the area of the image.

III. Image-area quantization and computational error

In practice, one important source of computational error result from image-area quantization. Since the pixel density of the imager incorporating the image-velocity computation technique is finite, the image area as "seen" by the imager is a quantized value of A_{IM}, denoted by A_{IMq}. A_{IMq} is a function of the shape and the orientation of the image. For a given shape, the image-area quantization error is lowest for the case when the image is oriented in a direction such that the number of delay lines covered is minimum.

If only one set of delay lines is used (as in Fig. 1), the computational error of V_x, $E_r (V_x)$, is found to be

$$E_r (V_x) = \left[\frac{V_s}{V_x} - 1 \right] \left[1 - \frac{A_{IMq}}{A_{IM}} \right] \qquad (3)$$

Thus, as the image velocity approaches the signal velocity, the computational error due to image-area quantization decreases. To compute the velocity of a slow moving image, either an imager with high pixel density should be used, or the signal velocity should be reduced. However, reducing the signal velocity increases the computational period.

When two sets of delay lines are used (as in Fig. 2), $E_r (V_x)$ is given by

$$E_r (V_x) = \frac{\left[1 - \frac{A_{IMqN}}{A_{IMq}} \right] \left[1 - \left[\frac{V_s}{V_x} \right]^2 \right]}{\left[1 - \frac{A_{IMqN}}{A^{+}_{IMq}} \right] + \left[1 + \frac{A_{IMqN}}{A_{IMq}} \right] \left[\frac{V_s}{V_x} \right]} \qquad (4)$$

where A_{IMqN} and A_{IMq} are the quantized image areas "seen" by the negative x-direction and positive x-direction delay lines respectively. Unlike the case when a single set of delay lines is used, the computational error with two sets of delay lines is not a function of the absolute value of the image-area quantization. Rather, it is a function of the mismatch in quantized image areas "seen" by the two sets of the delay lines.

IV. Light-intensity modulated delay line and signal refraction

There are several different methods available for the measurement of T_{1x} and T_{1xN}. The simplest one requires different velocities for the signals propagating inside and outside the image boundary. For such a scheme, a light-intensity modulated delay line can be used in place of a simple delay line [5].

0-7803-0593-0/92 $3.00 1992 IEEE

If the signal velocity inside and outside the image boundary are μ / τ_1 and μ / τ_0 respectively, the sum of the delays of all delay lines running in the positive x-direction, T_{2x}, is

$$T_{2x} = \frac{A_T}{\mu^2 / \tau_0} + \left[1 - \frac{\tau_0}{\tau_1} \right] \frac{A_{IM}}{\frac{\mu^2}{\tau_1} \left[1 - \frac{V_x \tau_1}{\mu} \right]} \quad ; \; V_y = 0 \tag{5}$$

where A_T is the total area of the delay lines.

Using equation (5) and similar expression for T_{2xN}, leads to

$$V_x = \frac{\left[T_{2x} - T_{2xN} \right] \frac{\mu}{\tau_1}}{T_{2x} + T_{2xN} - \frac{2 A_T}{\mu^2 / \tau_0}} \tag{6}$$

Since the delay of an individual delay line can be measured, T_{2x} and T_{2xN} can be known, in which case, V_x can be computed using equation (6) without knowledge of either the area, orientation or shape of the image. However, a problem arises with light-intensity modulated delay lines. It is signal refraction at the image boundary when $V_y \neq 0$ as shown in Fig. 3.

If the angles of incidence are not equal for all signals, as in the case of the square image shown in Fig. 3, equation (5) becomes

$$T_{2x} = \frac{A_T}{\mu^2 / \tau_0} + \left[1 - \frac{\tau_0}{\tau_1} \right] \frac{A_{IM}}{\frac{\mu^2}{\tau_1} \left[1 - \frac{V_x \tau_1}{\mu} \right]} + \varepsilon_T \tag{7}$$

where ε_T is an error term given by

$$\varepsilon_T = - \left[1 - \frac{\tau_0}{\tau_1} \right] \frac{A_{IM} \left[\frac{V_y \tau_1}{\mu} \right]}{2 \frac{\mu^2}{\tau_1} \left[1 - \frac{V_x \tau_0}{\mu} \right] \left[1 - \frac{V_x \tau_1}{\mu} \right]^2} \tag{8}$$

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

Using $\mu = 127.5\ \mu m$, $A_T = 14.44\ mm^2$, $A_{IM} = 1\ mm^2$, $\mu / \tau_0 = 12.75\ m/s$ and $\mu / \tau_1 = 25.56\ mm/s$, plots of the computational error as functions of V_y / V_x and the absolute value of V_x are shown in Figs. 4 and 5. From these two graphs, it is clear that the computational error in V_x is strongly dependent on V_y / V_x, but is only weekly dependent on the absolute value of V_x. If $V_x \ll \mu / \tau_1$, the computational error in V_x is approximately equal to 0.5 V_y / V_x. This fact may be used in practice to correct the computed value of V_x.

If two sets of delay lines are used, the computational error due to signal refraction is as plotted in Fig. 6. As shown, the computational error is less than 5% if both components of the image velocity are less than $10^3\ \mu m/s$. As the image velocity increases, the computational error increases. For example, the computational error increases to 24% when V_y is $10^4\ \mu m/s$.

To eliminate the computational error due to signal refraction requires a new method of measuring T_{1x} and T_{1xN} which uses a constant signal velocity.

V. Computational error due to delay-line mismatches

If signals propagate on different delay lines at different velocities, equation (6) becomes

$$V_x = \frac{\left[T_{2x} - T_{2xN} \right] \frac{\mu}{\tau_1}}{T_{2x} + T_{2xN} - \frac{2 A_T}{\mu^2 / \tau_0}} + \varepsilon_v \tag{9}$$

where

$$\varepsilon_v = \frac{-2 \left[1 - \frac{\tau_0}{\tau_1} \right]}{T_{2x} + T_{2xN} - \frac{2 A_T}{\mu^2 / \tau_0}} \sum_{A_{IM}} \left\{ d_i^j \frac{\frac{V_x}{\mu} \left[\tau^j_{1e} - \tau_1 \right]}{\left[1 - \frac{V_x \tau_1}{\mu} \right] \left[1 - \frac{V_x \tau^j_{1e}}{\mu} \right]} \right\} \tag{10}$$

where $\sum_{A_{IM}}$ denotes only delay lines covered by the image are considered in the summation. d_i^j is the length of j^{th} delay line covered by the image during a cycle of computation. τ^j_{1e} is the average delay of the delay elements covered by the image on the j^{th} delay line. In this special case, τ_1 is the global average of the delays of all elements on the imager.

Close examination of equation (10) shows that the computational error of V_x due to delay-line mismatch is a function of d_i^j ($\tau^j_{1e} - \tau_1$). The higher the value of d_i^j, the larger the number of delay elements covered by the image. If the process-parameter variation is highly random, a larger number of delay elements implies the value of τ^j_{1e} is closer to its statistical mean τ_1. Thus, an increase in d_i^j is offset by a decrease in ($\tau^j_{1e} - \tau_1$). However, if process-parameter variation is highly systematic (which leads to the grouping of delay elements of a particular delay into well-defined regions), a delay line with long d_i might contribute significantly to the computational error.

If $\tau_0 \gg \tau_1$, $V_x \tau_1 / \mu \ll 1$ and $\tau^j_{1e} = \tau_{1e}$ for all values of j, equation (10) reduces to

$$\varepsilon_v = V_x \left[1 - \frac{\tau_{1e}}{\tau_1} \right] \tag{11}$$

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

VI. Experimental results

An imager with built-in image-velocity computation capability has been implemented using 3μ-CMOS technology. The micrograph of the imager is shown in Fig. 7. The array size is 30 by 30 pixels and the pixel density is 61 pixels/ mm^2. A detailed description of the imager was reported elsewhere [6]. Only the accuracy of computation achieved currently will be described here.

For cases where the y-component of the image velocity is zero, the velocity error as a function of V_x / V_s is plotted as shown in Fig. 8. According to equations (7) and (8), if $V_y = 0$, the computational error due to signal refraction is zero. Thus, the sources of computational error are only image-area quantization and delay-line mismatch. According to equation (4), the computational error due to image-area quantization for these cases varies from 0% to 10%. However, as the image velocity approaches the signal velocity, the computational accuracy is highly sensitive to delay line mismatch. This fact is confirmed by close examination of equations (9) and (10). For values of $V_x / V_s > 0.7$, computational error due to delay-line mismatch accounts for greater than 90% of the total computational error.

For cases where the y-component of the image velocity is $10^4\ \mu m/s$, the velocity error as a function of V_x / V_s is plotted as shown in Fig. 9. The computational error for these cases is significantly increased due to signal refraction at the image boundary. The theoretical lower bounds plotted in the same figure are calculated using infinite pixel density. Error correction based on the fact that the computational error is approximately 0.5 V_y / V_x if V_y, $V_x \ll V_s$, was performed and the resulting error is also plotted in Fig. 9.

From the plots shown in Figs. 8 and 9, we may conclude that (*i*) computational error due to image-area quantization is limited to less than 10% if $V_x / V_s > 0.4$, (*ii*) computational error due to delay line mismatch becomes significant if $V_x / V_s > 0.7$, and (*iii*) if $V_x \approx V_y$ computational error is dominantly due to signal refraction at the image boundary. To lower the computational error for cases when

$V_y \neq 0$, the pixel density must be increased significantly so that the range of V_x can be extended below 0.4 V_s without increasing the quantization error significantly.

VII. Conclusions

The accuracy of image-velocity computation using delay lines is described. Three major sources of computational error are identified, namely, image-area quantization, signal refraction and delay-line mismatch. Computational errors obtained experimentally are found to be generally less than 10% if the y-component of the image velocity is zero. However, if $V_y \approx V_x$, computational error due to signal refraction becomes quite large.

Appendix A: The derivation of equation (1)

Consider the case of a moving elliptical image projected onto the x-direction delay lines shown in Fig. A.1. 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 length of time t_1^j, which is given by

$$t_1^j = \frac{d_1^j}{(V_y^2 + (V_s - V_x)^2)^{1/2}} = \frac{d_1^j \mu SIN \theta}{\mu (V_s - V_x)} \; ; \; V_x < V_s \quad \text{(A.1)}$$

where θ is the angle between the resulting signal velocity and the y-axis, and d_i^j is the length of the part of j^{th} delay line inside the image boundary. The sum of all possible values of t_1^j, which is denoted by T_{1x}, is given by

$$T_{1x} = \sum_{A_{IM}} t_1^j = \frac{\sum_{A_{IM}} (d_i^j \mu SIN \theta)}{\mu (V_s - V_x)} \quad \text{(A.2)}$$

where the summation sign indicates that the only signals considered are that propagate on delay lines that intersect the image boundary. If μ is small enough, $\sum_{A_{IM}} (d_i^j \mu SIN \theta) = A_{IM}$, thus

$$T_{1x} = \frac{A_{IM}}{\mu (V_s - V_x)} \quad \text{(A.3)}$$

References

[1] R.J. Offen, Editor: "VLSI image processing," Collins, London, U.K., 1985.

[2] H. Buxton and J. Wiejak: "Towards computer visions," *ibid.*

[3] C.A. Mead: "Analog VLSI and neural systems," Chapter 14, Addison-Wesley, Reading, MA, 1989.

[4] T. Horiuchi, J. Lazzaro, A. Moore and C. Koch: "A delay-line based motion detection chip," to appear in Advances in Neural Information Processing Systems, Vol. 3, Morgan Kaufmann Publishers.

[5] C.P. Chong: "Real-time image processing using analog VLSI," Proc. IEEE Workshop on Visual Signal Processings and Communications, Taiwan, pp. 169-172, 1991.

[6] C.P. Chong, C.A.T. Salama and K.C. Smith: "A novel technique for image-velocity computation," Accepted for publication in IEEE Trans. on Circuits and Systems for Video Technology.

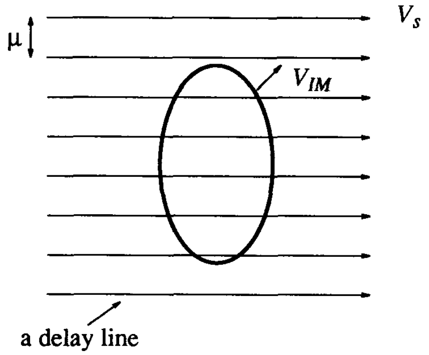

Fig. 1: Using one set of delay lines in image-velocity computation.

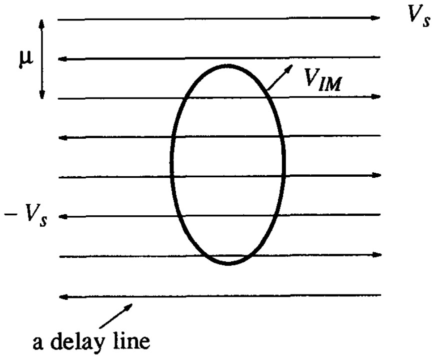

Fig. 2: Using two sets of delay lines in image-velocity computation.

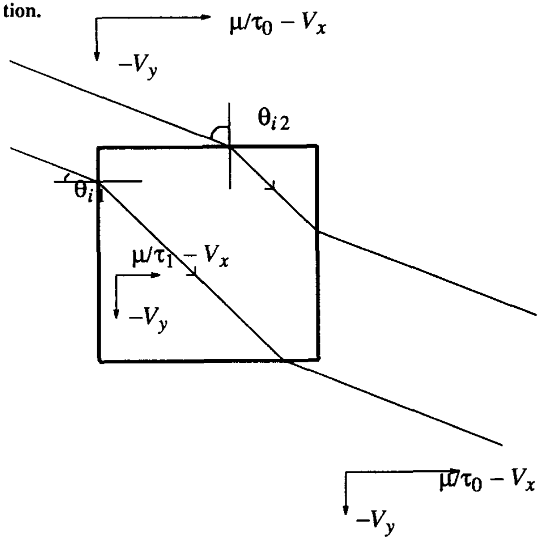

Fig. 3: Signal refractions at the image boundary.

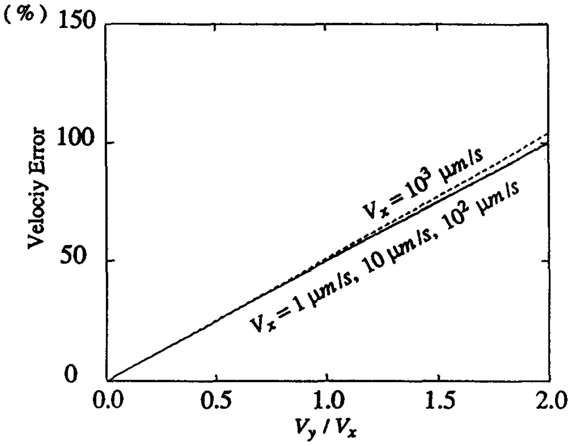

Fig. 4: The velocity error versus V_y / V_x. The value of V_x varies from 1 μm/s to 10^3 μm/s.

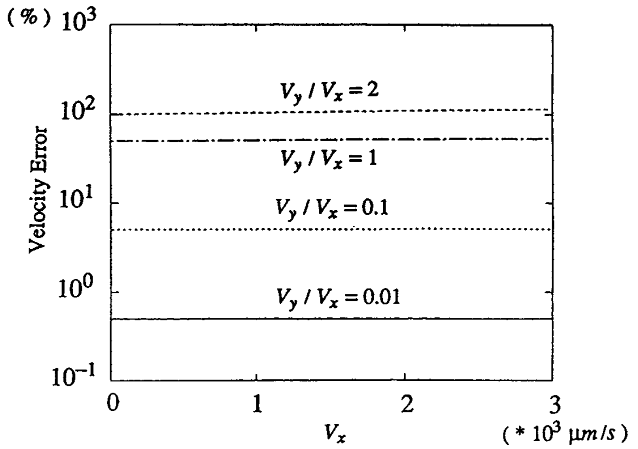

Fig. 5: The velocity error as a function of V_x, for values of V_y / V_x ranging from 0.01 to 2.

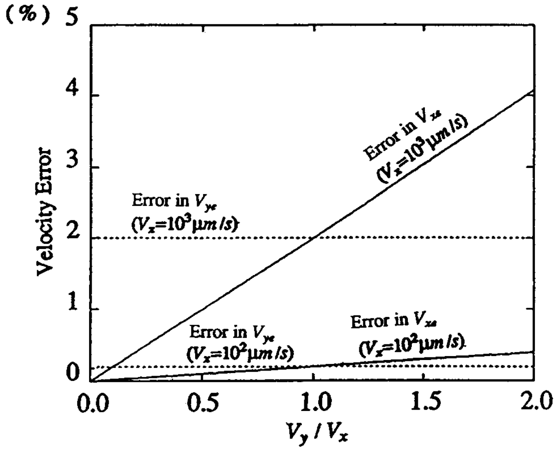

Fig. 6: The error in velocity obtained using equation (6) and the similar equation for V_y.

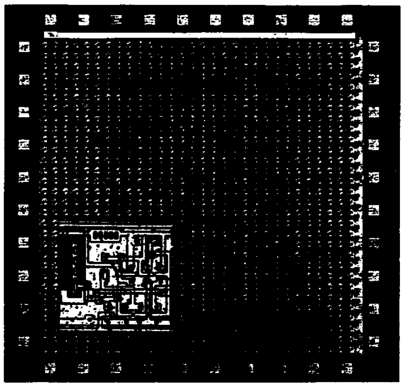

Fig. 7: The micrograph of the image-velocity computing imager.

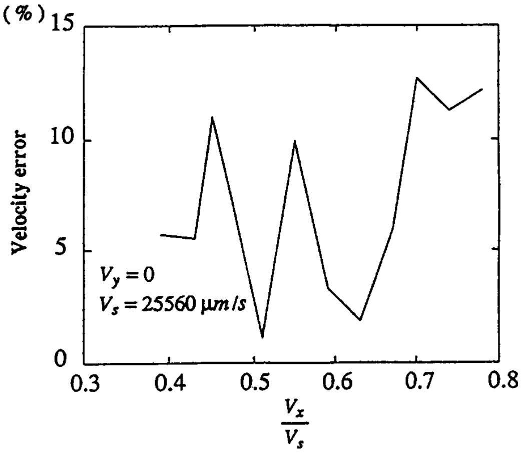

Fig. 8: The computational error in V_x when $V_y = 0$

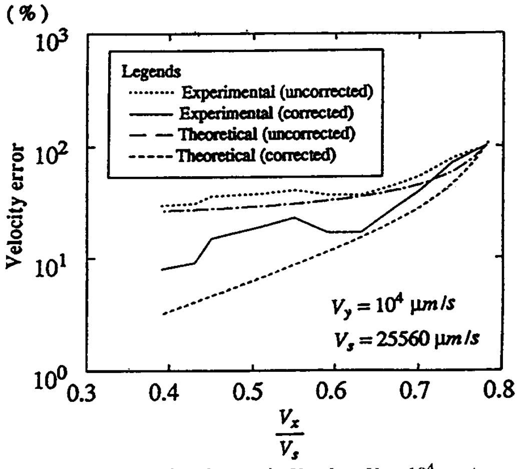

Fig. 9: The computational error in V_x when $V_y = 10^4$ $\mu m/s$.

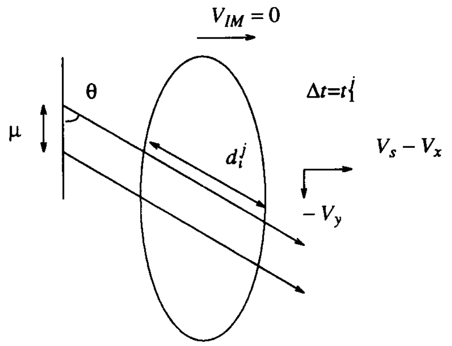

Fig. A.1: A moving elliptical image is projected onto the delay lines. The frame of reference is the image itself.