A Comparison of Three One-Dimensional Edge Detection Architectures for Analog VLSI Vision Systems

Matthew D. Rowley, Member, IEEE, and John G. Harris, Member, IEEE

Abstract: A comparison is made between three architectural models used for edge detection in analog VLSI early vision systems. In analog VLSI computational networks, signal strength is a paramount issue due to the need to overcome circuit limitations such as offsets, noise, and finite gain. Therefore algorithms mapped into silicon networks must take full advantage of available signal strengths to maximize signal-to-noise ratios. It will be shown that a discrete Differenced Gaussian algorithm retains a greater amount of the available signal than algorithms using thresholded zero-crossings from the Difference of Gaussian (DoG) or the Laplacian of Gaussian (LoG) functions.

I. INTRODUCTION

The resolution and performance of analog VLSI computational circuits is limited by the achievable signal-to-noise ratio [1]. Therefore it is not only crucial to make use of all the signal magnitude possible to maximize this ratio, but signal losses due to filtering or inefficient architectural realizations must be minimized. Feature information becomes more difficult to retain as practical implementation issues such as offsets, noise, and finite gain are considered since larger noise figures require greater amounts of filtering. In this discussion offsets, noise, and finite gain will be lumped into a single parameter and referred to simply as noise. The algorithm, then, and its corresponding silicon realization must retain the available signal strength while minimizing noise contributions.

There have been several analog realizations reported in the literature which perform such computations as motion or velocity estimation based on the localization and movement of edges within an image. Survey discussions of many of these applications are covered in the literature [1], [2], [3]. In these systems, small signal-to-noise ratios result in poor performance when operated in low contrast environments.

The remainder of this paper will compare the computational processes of three different early vision architectures. Signal representations chosen for each architecture are unique up to the signal labeled S_i. Processing from this point on is functionally identical and therefore the equations representing these processing steps are identical and can be described by

$$O_i = \begin{cases} 0 \text{ if } D_i < V_{th} \\ 1 \text{ if } D_i > V_{th} \end{cases} \quad (1)$$

where O_i is the discrete, binary output signal indicating edge locations, D_i is the absolute value of the derivative of the signal S_i represented by

$$D_i = |S_i - S_{i+1}| \quad (2)$$

and V_{th} is the threshold voltage. Zero crossings are used in these architectures to guarantee single pixel wide edges. The crucial computation, however, is thresholding which allows for the separation of edge signals from noise. In these architectures Gaussian functions will be used to approximate the symmetric, decaying exponential responses of the HRes [4] filtering networks due to the ease with which the processes can be represented mathematically [1].

Section II will address an architecture which computes the thresholded zero-crossings of the Laplacian of Gaussian function for edge detection. Section III will address an architecture which is based on isolating the thresholded zero-crossings from a Difference of Gaussians function [3] to determine edge locations. Section IV will address an architecture based on a discrete Differenced Gaussian function for edge detection. Section V will compare the results of each algorithm. Finally, section VI will present out conclusions.

II. LAPLACIAN OF GAUSSIAN ALGORITHM

The first architecture which will be examined implements a Laplacian of Gaussian function. Figure 1 depicts the one-dimensional architectural implementation where logarithmic photoreceptor outputs are buffered through transamps (voltage-to-current converters) onto a resistive network implemented by HRes circuits [4]. The characteristic length of the resistive network is independently controlled by an off-chip bias voltage. It can be shown that the systems' step response can be described by

$$D_i = \frac{A}{2} \left[erf\left(\frac{\sqrt{2}x}{2\sigma}\right) - 2erf\left(\frac{\sqrt{2}(x-1)}{2\sigma}\right) + erf\left(\frac{\sqrt{2}(x-2)}{2\sigma}\right) \right] (3)$$

where σ is the characteristic length, A is the input signal magnitude, and x is the spatial position. Locally differenced responses, D_i, of the LoG architecture to a 20 mV step input are shown in Figure 2 where the family of curves is generated by varying the filtering. There are three points of interest within Figure 2.

---

1. Simulations using the actual exponential filter response characteristics have shown similar results to using Gaussians.
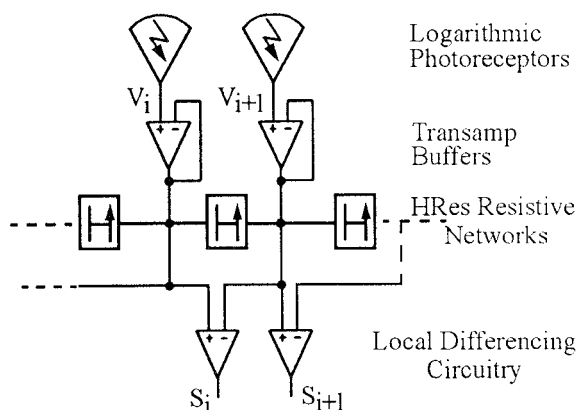
Figure 1. LoG Architecture. The architecture depicted computes the discrete second derivative of a Gaussian function. This computation is done by the difference of differences circuitry which approximates a one-dimensional Laplacian of Gaussian function.

First, the signal response decreases more rapidly as σ increases than in the architectures discussed in sections III and IV. This results in a smaller signal-to-noise ratio and ultimately in a system which does not respond well in low contrast environments. Second, the maximum attainable signal difference for an ideal (noiseless) system is A which occurs when $\sigma = 0$ but for this example the peak signal is approximately 6 mV and occurs when $\sigma = 1$. Third, the optimal σ will increase for all architectures in practical implementations since additional noise signals must be filtered out.

One detractor of using this architecture is the computational complexity involved in implementing it in a silicon network. The additional circuitry would result in greater offsets and consequently lower S/N ratios, reduced resolution, and greater power consumption than in the two subsequent architectures.
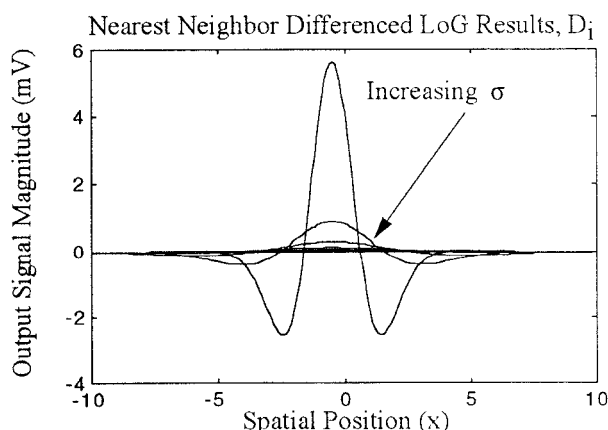

Figure 2. LoG Response. In this figure, nearest neighbor differenced results from the LoG architecture are shown. Notice the rapid loss of signal as the filtering increases from $\sigma = 1,2..16$.
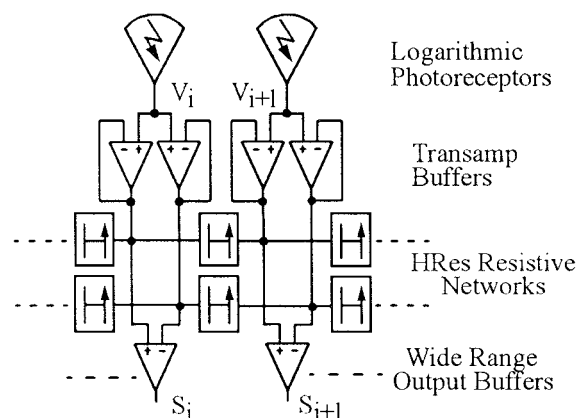

Figure 3. DoG Architecture. Shown here are two adjacent photo-processing cells from the zero-crossing chip. Photoreceptor outputs are buffered, filtered, and then a DoG operation is performed. Edge signals are determined by thresholding locally differenced outputs.

III. DIFFERENCE OF GAUSSIAN ALGORITHM

The second algorithm [3] uses thresholded zero-crossings from a Difference of Gaussians function for localizing edges. Figure 3 shows two cells from the one dimensional array. Logarithmic photoreceptor outputs are buffered through transamps onto two separate HRes networks. The characteristic length [4] of each network is independently controlled by off-chip bias voltages. Therefore a DoG operation is performed by the wide range output buffers. Edges are subsequently computed from the wide-range transamp outputs by magnitude thresholding nearest-neighbor differences as shown in equation (1).

It can be shown that the step response of an ideal (noiseless) realization of the system shown in Figure 3 is of the form

$$S_i = \frac{A}{2}\left[\text{erf}\left(\frac{\sqrt{2}x}{2\sigma_{F1}}\right) - \text{erf}\left(\frac{\sqrt{2}x}{2\sigma_{F2}}\right)\right] \qquad (4)$$

where A denotes the magnitude of the input step signal, σ_{F1} and σ_{F2} denote the characteristic lengths of the respective filtering functions, and x represents the spatial position.

Figure 4 shows a plot of the locally differenced values from equation (4) when a step input of A = 20 mV is applied to a system with filtering functions having characteristic lengths $\sigma_{F2} = 1.6\sigma_{F1}$ and $\sigma_{F1} = 1,2..16$. The factor 1.6 has been chosen to best approximate a Gaussian response [5]. The arrow indicates the change in response as σ_{F1} is spanned. As can be seen in Figure 4, the maximum attainable signal difference is approximately 3 mV when $\sigma_{F1} = 1$. The maximum achievable signal difference between adjacent points for an ideal system of infinite length is the entire input signal magnitude, A. This case occurs when
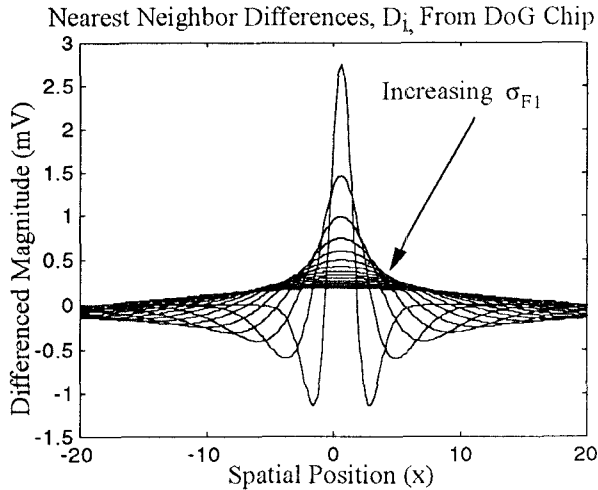
Figure 4. Shown here is a plot of the discrete nearest neighbor differences of the functions resulting from the DoG chip when $\sigma_{F2} = 1.6\sigma_{F1}$. The peak nearest neighbor signal difference is approximately 3 mV.
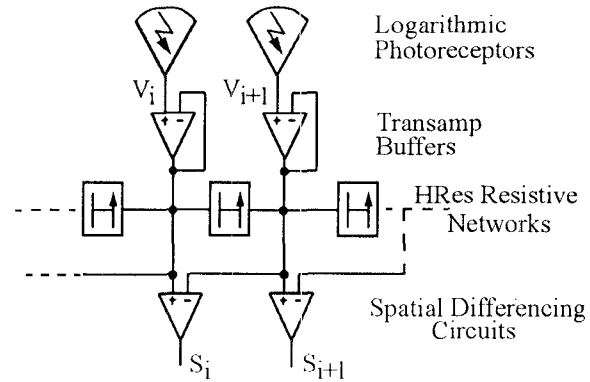

Figure 5. DG Architecture. Two processing cells of a chip employing a Locally Differenced Gaussian computation for edge localization.

there is no filtering in one resistive network, resulting in a sharp transition at the step input, and an infinite amount of filtering in the second network, where a dc average of half the input signal magnitude, A/2, is obtained from the step input ($\sigma_{F2} \neq 1.6\sigma_{F1}$). As in the previous architecture, the complexity of this design will result in large offsets which lead to small S/N ratios.

An interesting comparison between the DoG and LoG responses is that the LoG response in Figure 2 is more spatially compact than the DoG response shown in Figure 4 for the filtering constants chosen. The reason for this is that an optimal fit between the DoG approximation and the LoG response has not been performed since the goal of this discussion is to determine which architecture retains the greatest amount of available signal when realistic filtering requirements are considered.

tion is introduced into the system and the characteristic length is varied from 1 to 16. As can be seen the peak output occurs at $\sigma = 1$ and has an approximate value of 8 mV. In this algorithm the maximum signal difference between adjacent points for an ideal infinite length system is A just as in the previous systems. This condition occurs when $\sigma = 0$ and the entire step input transition is contained between adjacent points.

Several notes on the DG algorithm. First, this technique can produce thick edges since a simple magnitude comparison is being performed. The thicker edges can result in reduced resolution and require additional processing to perform edge thinning. Second is that this architecture is the least complex of the three which will result in smaller offsets and better S/N ratios. Lastly, a chip implementing this architecture has been fabricated and is currently under test.

V. COMPARISON

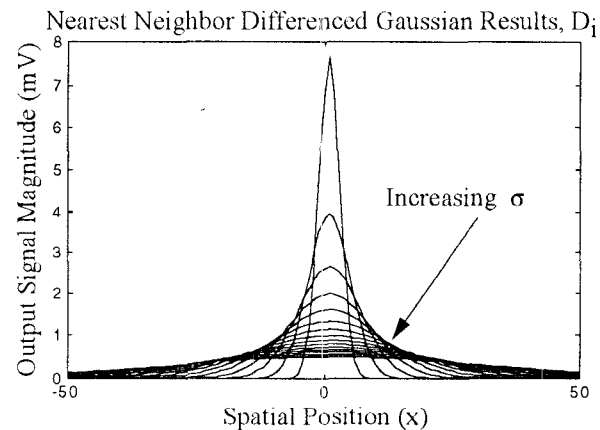As was discussed previously, the maximum recovered signal for each algorithm in an ideal situation is

IV. DIFFERENCED GAUSSIAN ALGORITHM

The last algorithm uses localized differences from a single filtered version of the input to isolate edge locations as depicted in Figure 5. Again two processing cells composed of logarithmic photoreceptors, buffers, a single filtering network, and the edge detection circuitry are shown. In this implementation, absolute value circuits are employed as the local differencing circuits. It can be shown [6] that the expression representing the ideal system response to a step input signal is

$$S_i = \frac{A}{2}\left[\text{erf}\left(\frac{\sqrt{2}x}{2\sigma}\right) - \text{erf}\left(\frac{\sqrt{2}x - 1}{2\sigma}\right)\right] \quad (5)$$

where A is the input signal magnitude, σ is the characteristic length, and x is the spatial position. Figure 6 shows the differenced results when a 20 mV step func-

Figure 6. DG Results. Nearest neighbor differencing results showing the peak signal magnitude as σ is varied from 1 to 16.
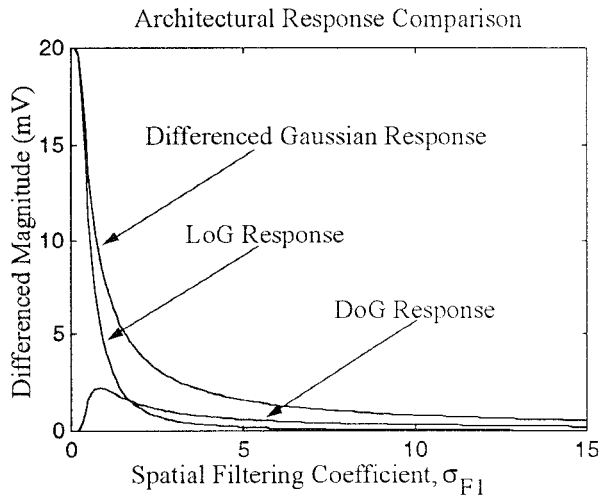
Architectural Response Comparison

Figure 7. This plot shows the peak signal magnitudes between adjacent points for all the algorithms. For the DoG algorithm, $\sigma_{F2} = 1.6\sigma_{F1}$ has been assumed. Therefore the filtering coefficient plotted along the x-axis refers to σ_{F1}.

A. The improvement in the signal-to-noise ratio, however, is dependent upon the amount of filtering that is applied to each system. In the DoG algorithm it has been shown that the proper ratio between the filtering constants to approximate the second derivative of a Gaussian is 1.6 [5]. Figure 7 shows a comparison of the results obtained from all three algorithms when a step input signal 20 mV is applied. The filtering coefficients used are $\sigma_{F2} = 1.6\sigma_{F1} = 1.6\sigma_{DG} = 1.6\sigma_{LOG}$ where σ_{DG} and σ_{LOG} represent the filtering constants used in the Differenced Gaussian and Laplacian of Gaussian algorithms respectively. The peak signal obtained for the DoG algorithm occurs at $\sigma_{F1} = .7$ and has the value 2.25 mV while the Differenced Gaussian and LoG algorithms yield a signal difference of 10.4 mV at that same amount of filtering.

As the amount of filtering increases, all functions tend towards zero but for characteristic lengths between 1 and 10 the Differenced Gaussian algorithm provides superior signal retention characteristics. The LoG algorithm yields similar signal retention characteristics to the DG algorithm at very low filtering constants but quickly loses this capability as the filtering increases. Since the DG algorithm operates simply on thresholded signal differences, it retains a greater amount of the available signal compared to the other algorithms which attempt to localize zero-crossing signals which have a slope greater than some predetermined threshold. One consideration when evaluating these results is that extremely low or high filtering constants are not practical for VLSI implementations. Filtering constants below 1 really perform no filtering at all while filtering constants above 15 or 20 spread edge signals over an extremely large number of processing cells making edge

detection difficult.

From these results one can see that the Differenced Gaussian algorithm makes better use of the available signal strength than the other methods. Therefore, one can conclude that it is intrinsically easier to localize a change in magnitude than it is to localize a change in slope in analog VLSI networks. In addition, the DG algorithm has a simpler computational structure which results in a more compact, lower power realization with reduced offsets and a better S/N ratio.

One technique which can be used to improve the performance of all three architectures in low contrast or noisy environments is Non-Nearest Neighbor Differencing [6]. Essentially this technique increases the spatial sampling distance used in the differencing computations to retain greater amounts of the available signal.

VI. CONCLUSIONS

This paper has compared three architectures used for computing edge locations in one-dimensional analog VLSI networks. It is shown that using a Differenced Gaussian algorithm is superior to a DoG or LoG implementation in silicon networks due to its ability to retain greater signal magnitudes. Signal retention is essential in order to overcome noise sources in the analog computational circuits. The DG algorithm also is the least complex to implement since it has the fewest computational elements which also has the added benefit of reducing overall system size and offsets.

VII. REFERENCES

[1] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, "Analog VLSI Architectures for Motion Processing: From Fundamental Limits to System Applications," in *Proceedings of the IEEE*, vol. 84, no. 7, pp. 969-987, July 1996.

[2] C. Koch and H. Li, Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits, IEEE Computer Society Press, 1995.

[3] W. Bair and C. Koch, "Real-time Motion Detection Using an Analog VLSI Zero-Crossing Chip," in SPIE vol. 1473, *Visual Information Processing: From Neurons to Chips*, pp. 59-65, 1991.

[4] C. Mead, "Aggregating Signals," in Analog VLSI and Neural Systems, by C. Mead, Reading: Addison-Wesley, chp. 7, 1989.

[5] Marr and Hildreth, "Theory of Edge Detection," in *Proc. Royal Society of London B*, vol. 207, pp. 187-217, 1980.

[6] M. Rowley and J. Harris, "An Edge Enhancement Technique for Analog VLSI Early Vision Applications," in *IEEE International Conference on Neural Networks*, pages 1000-1004, Washington, D.C., June 1996.