Analog VLSI Architecture for Computing Heading Direction

Giacomo Indiveri, Jörg Kramer and Christof Koch
Computation and Neural Systems Program
CALTECH, Pasadena, CA 91125, U.S.A.
giacomo@klab.caltech.edu

Abstract— **We describe a parallel analog VLSI architecture that integrates optical flow data obtained from arrays of elementary velocity sensors to estimate the direction of heading for pure translational motion sequences. We performed software simulations at different levels of abstraction to evaluate the most important qualitative properties of the optical flow field, determine the best functional operators for the implementation of the architecture and define the numerical values of some of the free parameters present in the hardware implementation. Furthermore we describe the properties of the circuits used in the hardware implementation and the functionality of the elementary motion sensors used to extract the optical flow field.**

Measurements from the chips developed and software simulation results are provided to show the validity of the followed approach.

I. INTRODUCTION

Real-time vehicle navigation tasks have proven to be computationally intensive and extremely complex in terms of resources used. Moreover, mobile applications of navigation systems place severe constraints on their size, power consumption, shock resistance and manufacturing cost. To overcome these problems, characteristic of most industrial applications of vehicle navigation systems, an attractive approach is to implement them as parallel image processing architectures, using analog VLSI circuits. Since analog circuits are limited by low precision in the values of the state variables, we target applications for vehicle navigation that make use mainly of qualitative features of the optical flow. To best exploit such qualitative properties, to provide a proper representation containing only the important parameters for successive computations, and to extract various features required for passive navigation such as direction of heading, we focused our research on image processing architectures that make use of motion sensing elements based on local interactions among neighboring pixels.

In this article we describe an approach for computing the direction of heading for pure *translational* motion, defined as the spatial location on the image plane from which all the optical flow vectors diverge. We thereby take advantage of the fact that for vehicle navigation it is possible to compensate for the *rotational* component of motion using lateral accelerometer measurements from other sensors often already present on the vehicle.

We tackled the problem by first analyzing the computational properties of the optical flow for images obtained from outdoor moving vehicles, and the possible sources of errors that could arise from noise in the input images (*e.g.* due to bad lighting conditions or extreme speeds) or noise in the state variables (*e.g.* due to device mismatch) in a hardware implementation. As a second step we designed and simulated a parallel hardware architecture robust to noise in the inputs and in the state variables. The computational operators have been implemented using analog circuits, starting with an array of local velocity sensing elements, whose performance has been evaluated.

II. SOFTWARE SIMULATIONS

Software simulations were performed on sequences of images obtained from a camera with a 64×64 pixel silicon retina placed on a moving truck (courtesy of B. Mathur at Rockwell Corporation). The use of real world images of roads and cars obtained from an analog VLSI device, rather than sequences of artificial images, is essential for specifying the architectures that have the best characteristics in response to vehicle navigation types of data. An image from a typical navigation scene is shown in Fig. 1.



Fig. 1. Image obtained from a camera mounted on a truck moving on a straight road. On the left there is a shadow of a tree approaching the camera.

A. Optical flow

To reduce the computational cost of the simulations and to concentrate on system-level issues we used a standard procedure for the optical flow extraction, rather than simulating in detail the operation of the elementary motion sensors here described. Specifically, we implemented an algorithm based on the *image brightness constancy equation* [1] [2]: for natural motion scenes, if $I(x, y, t)$ represents the

image brightness at time t and position (x, y), we assume that

$$\frac{d}{dt}I(x, y, t) = 0 \quad (1)$$

in subregions of the image. By writing out the temporal and spatial derivatives, Eq. (1) can be rewritten as

$$\nabla I \cdot \overline{v} + \frac{\partial}{\partial t}I = 0 \quad (2)$$

and the velocity vector $\overline{v}$ can be computed as:

$$\overline{v} = -(\nabla I^t \cdot \nabla I)^{-1} \nabla I^t \frac{\partial}{\partial t} I$$

for each subregion where the square matrix $(\nabla I^t \cdot \nabla I)$ is invertible. This particular algorithm provides results that are qualitatively similar to those measured from the elementary motion sensor circuits: the data is sparse and is reliable only for areas in the image with high contrast. The optical flow field associated to the image of Fig. 1 is shown in Fig. 2.

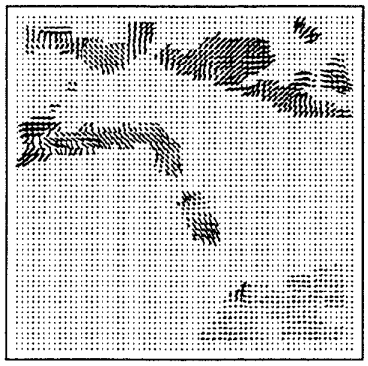

Fig. 2. Optical flow field for straight road sequence as determined with a standard machine-vision algorithm (see text for details).

B. Heading direction

Despite the sparseness of the data, having defined the types of images to be analyzed and the types of processing to perform, we can use *a priori* information to integrate selectively such data and determine qualitative properties of the optical flow. Specifically, for the application domain considered, we are interested in detecting changes in the direction of heading only along the horizontal axis. It is then sufficient to consider arrays of 1-D velocity sensors neglecting the vertical component of the optical flow vectors. Considering only the horizontal components of the optical flow vectors, with an ideal stimulus, the heading direction will correspond to the point where the vectors change direction. Then, if we assign positive values to vectors pointing in one direction and negative values to vectors pointing in the opposite direction, the heading direction position will correspond to the point closest to the *zero-crossing* in the array. Unfortunately, even in the case of pure translational motion, real image sequences may generate erroneous optical flow vectors that give rise to spurious zero-crossings. The problem then is to detect, among all zero-crossings present, the one that corresponds to the true heading direction. Assuming that the heading direction changes smoothly, integrating in time and analyzing the spatio-temporal characteristics of the curve obtained from the optical flow values it is possible to detect the correct zero-crossing. To verify the validity of our assumptions we performed software simulations at different levels of abstraction. The high-level abstraction simulations were used to determine the principal computational primitives for a system robust to noise in the input data, while the low-level abstraction simulations were used to determine the fine structure of the architecture so that the system would be robust to noise arising from device mismatch and offsets in a hardware implementation.

C. High-level abstraction simulations

High-level simulations were performed on the 2-D images from the straight-road sequence. The algorithm used in the simulations employs the following operators: saturating non-linearity on the horizontal components of the optical flow vectors (only the sign of the vector is needed); spatial smoothing on the resulting values; evaluation of the steepness of the zero-crossings present in the array and finally selection of the zero-crossing with maximum steepness. The zero-crossing with maximum steepness will be selected only if its position is in the neighborhood of the previously selected zero-crossing. This ensures that the computed heading direction position will shift smoothly with time. Fig. 3 shows a simulation result for an image taken from the straight road sequence.

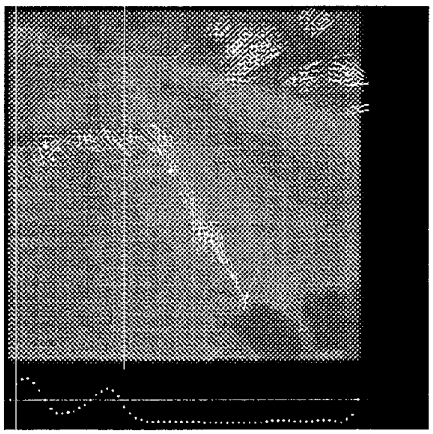

Fig. 3. Determination of heading direction using high-level abstraction simulations: the sum of the horizontal components of the optical flow field is plotted on the bottom of the figure. The presence of more than one zero-crossing is due to different types of noise in the optical flow computation (e.g. quantization errors in software simulations). The coordinate of the heading direction is computed as the abscissa of the zero-crossing with maximum steepness and closest to the abscissa of the previously selected unit.

All of the operators used in the algorithm have been carefully chosen to be suitable for implementation with analog circuits. To evaluate the horizontal component of the optical flow in real cases, a test chip using standard 2μm

technology with an array of 24 elementary motion sensors, described in section III-A, connected to transconductance amplifiers that perform the saturating non-linearity has been developed. Measurements have shown that this first processing stage is prone to errors due to device mismatch, offset variations and aliasing. Fig. 4 shows some measurement results that have been obtained by imaging on the chip expanding or contracting stimuli using black and white edges on a rotating drum and their reflection from an adjacent tilted mirror. The point of contact between drum and mirror corresponding to the simulated heading direction has been imaged approximately onto the 11th unit of the array.

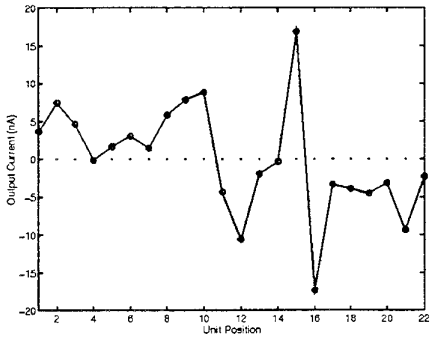

Fig. 4. Output currents of an array of transconductance amplifiers connected to elementary motion sensors for rotating drum stimulus under incandescent room illumination.

As shown, there is one isolated error point on the 15th unit. We observed that these types of errors generally tend to be isolated. Due to the nature of these errors a simple operation of linear spatial smoothing is not always sufficient to solve the problem.

D. Low-level abstraction simulations

Low-level abstraction software simulations were performed to enhance the smoothing stage and to define the structure of the architecture in order to make the system robust to isolated errors. Specifically, we designed an interaction scheme that connects neighboring units, through *recursive lateral inhibition* [3] to precede a linear smoothing stage. For this type of inhibition the connections between units are strictly local, yet each unit gains access to global information on the status of the system. The algorithm underlying the computations carried out by this scheme is governed by the following set of equations:

$$\begin{cases} y_i^+ = g\left(g(x_i) - w \cdot y_{i-1}^- - w \cdot y_{i+1}^-\right) \\ y_i^- = g\left(g(-x_i) - w \cdot y_{i-1}^+ - w \cdot y_{i+1}^+\right) \end{cases} \tag{3}$$

where y_i^- and y_i^+ represent two distinct populations of direction selective units for the two directions of motion considered; $g(\cdot)$ is the *halfwave-rectification* function; x_i represents the output of the transconductance amplifier at position i and w is a positive constant that represents the connection weight between neighboring units. To understand the principle of operation of the recursive scheme we will analyze a linear version of the system by neglecting the non-linear operator (i.e. we will consider $g(x) = x$). To evaluate the effects of the non-linearity we will then show some results obtained from numerical simulations.

If we consider $g(x) = x$ the system of coupled equations in eq. (3) can be written as:

$$\begin{cases} y_i^+ = x_i - w \cdot y_{i-1}^- - w \cdot y_{i+1}^- \\ y_i^- = -x_i - w \cdot y_{i-1}^+ - w \cdot y_{i+1}^+ \end{cases} \tag{4}$$

By expliciting the terms y_{i-1}^-, y_{i+1}^- and y_{i-1}^+, y_{i+1}^+ we obtain:

$$\begin{cases} y_i^+ = x_i + w(x_{i-i} + x_{i+1}) + 2w^2 y_i^+ + w^2(y_{i-2}^+ + y_{i+2}^+) \\ y_i^- = -x_i - w(x_{i-i} + x_{i+1}) + 2w^2 y_i^- + w^2(y_{i-2}^- + y_{i+2}^-) \end{cases} \tag{5}$$

Thus, in a first approximation, if $|w| \ll 1$, we can write:

$$\begin{cases} y_i^+ \approx x_i + w(x_{i-1} + x_{i+1}) \\ y_i^- \approx -x_i - w(x_{i-1} + x_{i+1}) \end{cases} \tag{6}$$

Hence, in the linear case the recursive lateral inhibition scheme is equivalent to a simple first order smoothing algorithm weighted by the lateral connection weight w. The recursive system is able to show much more interesting behaviors if we take into account the non-linear function $g(\cdot) = max(0, \cdot)$. Software simulations have shown that the recursive lateral inhibition scheme is able to eliminate isolated errors present in the input data in a wide variety of cases. Most of the results obtained have been inferred from statistical considerations: algorithms which incorporate Gaussian random processes to simulate different types of noise that might be present in the hardware implementation were run many times for different input data configurations. Specifically, we accounted for the fact that in an analog VLSI implementation the lateral connection value w would be affected by noise (due to device mismatch effects) with a Gaussian probability density function with standard deviations, for a 2μm fabrication process, as high as 20% (depending on the geometry of the transistors) [4] [5].

In order to compare the performance of the linear system with the non-linear one, we will consider their outputs for the measured input data shown in Fig. 4.

Fig. 5 shows the result of the recursive inhibition simulations: the isolated bad data point (on the 15th unit) has been cancelled while the rest of the data remained substantially unmodified. At this point there are no spurious zero-crossings due to device mismatches or offset variations in the single elements of the system. A further linear smoothing operation enhances the data and points out the zero-crossing corresponding to the correct heading direction position. Fig. 6 shows the results of smoothing (obtained through a simulation of a resistive network) on the data processed by the recursive inhibition scheme and, for comparison, on the original data coming from the transconductance amplifiers.

Also for this case the computational operators were chosen so that they could be implemented using analog circuits. Specifically, following a *current-mode approach* [4],

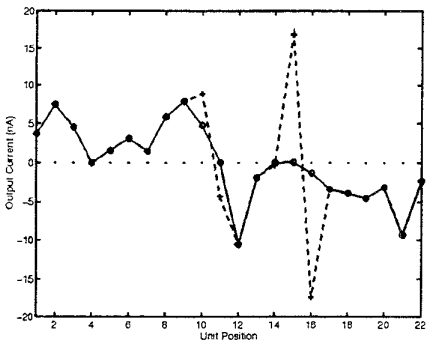

Fig. 5. Output current $(y_i^+ - y_i^-)$ of the recursive inhibition scheme simulations on chip measurement data: the crosses show the original input data while the circles show the result of the inhibition. The lateral connection weight w was set to 0.75 and the standard deviation of Gaussian random process simulating noise due to device mismatch was set to 20%.

for which the relevant signals carrying the information are currents while voltages play only a minor role, the non-linear function can be implemented by merely using current mirrors and multiplications with the connection weight w can be implemented by setting the $\frac{W}{L}$ ratio of the transistors in the current mirrors.

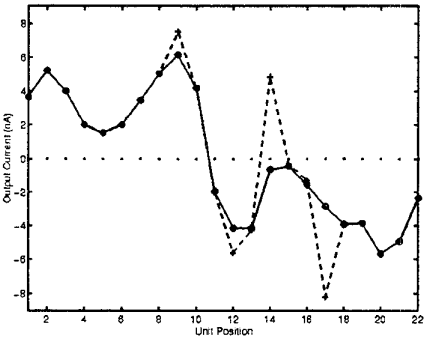

Fig. 6. Result of local smoothing with low space constant (the smoothing kernel was only 3 pixels wide) on the result of the lateral inhibition (circles) and on the original data from the transconductance amplifiers (crosses).

III. Analog VLSI Implementation

The computational operators described in the previous section have all been implemented using $2\mu m$ analog VLSI technology.

A. Elementary Velocity Sensor

The elementary velocity sensor has been previously described [6]. It implements an algorithm, that measures the time of travel of an edge between two nearby locations on the chip, unambiguously encoding 1-D velocity over considerable velocity, contrast, and illumination ranges in real time.

The input stage consists of temporal edge detectors, each using a photo-receptor circuit, whose output voltage logarithmically encodes local brightness [7]. An irradiance

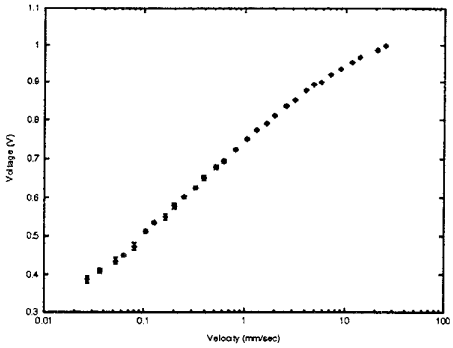

Fig. 7. Output voltage of a velocity sensing element for the preferred direction of motion of a sharp 56% contrast ON edge, versus image velocity for a.c. incandescent illumination. The object was imaged from a distance of 380 mm using a lens with a focal length of 13 mm. The pixel spacing of 300 μm corresponds to an optical angle of 1.28° and a distance of 8.5 mm on the object. Each data point represents the average of 5 successive measurements, the error bars indicate r.m.s. errors computed from the data. The logarithmic dependence of the analog output voltage on velocity allows measurements down to very low speeds.

transient thus causes a voltage transient which only depends on contrast and not on overall illumination. The output of the photo-receptor is then amplified, thresholded, differentiated, and rectified by a compact circuit, so that a positive irradiance transient with sufficiently large contrast, corresponding to a temporal dark-bright or ON edge, is converted into a current pulse.

The current pulses from each temporal edge detector are fed into a non-linear filtering circuit, consisting of a low-pass filter in the feedback path of a high-gain amplifier. The filter generates two voltage pulses, a short spike and a logarithmically-decaying signal. The outputs of the filtering circuits coupled to adjacent edge detectors are used to compute the velocity of a moving edge. The slowly-decaying pulse triggered by the edge at one location serves as a timing signal, whose analog voltage is sampled by the short spike generated upon the arrival of the edge at the adjacent location. For the direction of motion where the timing pulse is generated before the sampling pulse (preferred direction), the sampled output voltage is a measure of the time delay of the onset of the two pulses and therefore of edge velocity. For the other direction (null direction), the sampling pulse precedes the onset of the timing pulse and the residual voltage of the timing pulse triggered by the previous edge is sampled. A bidirectional motion sensor has two output stages, each responding to one direction of motion. In the absence of spatial aliasing, *i.e.* if the spacing of succeeding edges is larger than twice the photo-receptor spacing, the preferred direction signal is larger than the null direction signal, so that the residual output for the null direction can be suppressed through comparison of the two output voltages.

As implemented with a 2 μm CMOS process, the size of such a bidirectional velocity sensing element is 0.045 mm^2. We characterized an element with a photo-receptor spacing of 300 μm, using printed gray scale patterns on a rotating drum as optical stimuli. The stimuli were projected onto

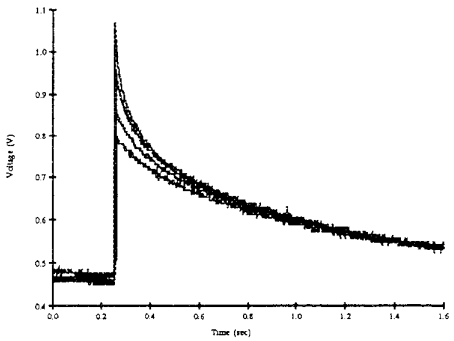

Fig. 8. Measured waveforms of the timing pulse of a velocity sensing element in response to sharp edges of different contrasts. The four waveforms represent the response to edges with 56%, 16%, 11% and 5% contrast respectively. The pulse amplitude monotonically decreases with contrast, but for sufficiently long times after their onset, all curves converge to a contrast-independent encoding of time delay and thus velocity. The noise level is smaller than the 10 mV quantization step of the measurements.

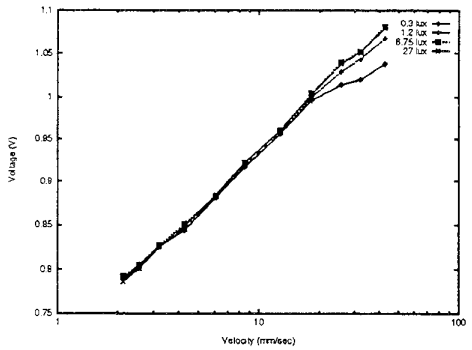

Fig. 9. Output voltage of a velocity sensing element for the preferred direction of motion of a sharp 56% contrast ON edge versus image velocity for different a.c. incandescent illumination levels. Each data point represents the average of 5 successive measurements. For low enough speeds the measured data is independent of illumination.

the chip from a distance of 380 mm by means of a lens with a focal length of 13 mm. Fig. 7 shows the response of the motion sensing element to a 56% contrast ON edge as a function of the velocity of its image on the chip in the preferred direction. The measurements were taken under a.c. incandescent lighting conditions using a lens f-number of 11, where a white paper surface provided an illuminance of about 0.03 lux on the chip. For the chosen setup, the motion sensor encodes velocity logarithmically over more than three orders of magnitude with good sensitivity.

Fig. 8 shows measured timing pulses for sharp edges of different contrasts. The initial voltage varies with the size of the current pulse from the edge detector and therefore with contrast, but eventually the voltage traces converge. This means that for sufficiently low speeds the measured velocity is contrast-independent down to contrasts of 5%. For contrasts higher than 15% contrast-invariance is obtained for all but very high speeds.

Decreasing global illumination results in similar qualitative characteristics of the timing pulse as decreasing contrast. Fig. 9 shows the response of the velocity sensing element for a 56% contrast ON edge under different ambient a.c. incandescent illumination levels with the lens f-number set to 1.8. The response is invariant to global illumination over at least two orders of magnitude, except for high speeds.

B. Circuit Architecture for Computation of Heading Direction

A diagram of the circuit blocks for the extraction of direction of heading is shown in Fig. 10.

In order to determine the local direction of motion, a saturating non-linearity is applied to the difference of the two directional outputs of each velocity sensor, implemented using a transconductance amplifier operated in the *subthreshold domain* [8]. The bidirectional current signal coming out of the transconductance amplifier is then copied into two separate paths (one for positive currents and the other for negative ones) by the "pos-neg" circuit, which consists of two current mirrors and two bias transistors. The following circuits implement the recursive lateral inhibition architecture. These circuits copy their input currents to their immediate neighbors through current mirrors with $\frac{W}{L}$ ratios set to a constant optimal value of 0.75 (computed a priori through numerical simulations). The mirroring operations implement the *halfwave rectification* function and the multiplication by a constant weight simultaneously. Linear spatial smoothing is then obtained using resistive networks, which diffuse each unit's current to its neighbors [9]; the space constant of the resistive network determines the range of current diffusion and can be set externally through a bias voltage. The following circuits, based on a modified

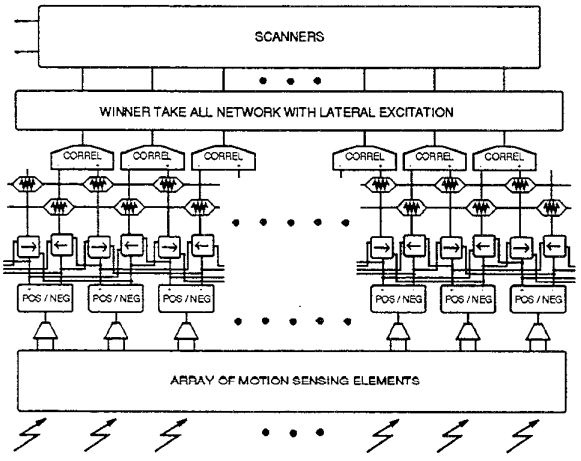

Fig. 10. Block diagram for the VLSI implementation of the architecture for heading direction computation.

version of the *simple current-correlator* [10], compute a correlation measure between the negative current of one unit and the positive current of the adjacent one, possibly detecting the presence of a zero-crossing and evaluating its steepness. The selection of the zero-crossing with maximum steepness closest to the previously selected one is implemented using a winner-take-all circuit with distributed

excitation and local hysteresis [11]. Finally, the scanners are used to read the output data reducing the number of pins that would otherwise be required.

IV. Conclusions

We have developed compact, low-power, low-cost analog VLSI elementary motion sensors and integrated them in a parallel architecture for detection of heading direction applied to the field of vehicle navigation. We bypassed the problem of low precision in analog VLSI technology by exploiting qualitative properties of the optical flow. The use of software simulations was fundamental in pointing out such properties and in determining the structure of the parallel architecture developed.

The developed chips are currently being tested and premilinary results are satisfactory. Their correct functionality, at least from a qualitative point of view, has confirmed the validity of the approach followed and induced us to continue this line of research. More accurate circuits that implement the computational operators described are being developed.

References

[1] A. Verri, M. Straforini, and V. Torre, "Computational aspects of motion perception in natural and artificial vision systems," *Phil. Trans. R. Soc. Lond. B*, vol. 337, pp. 429–443, 1992.

[2] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal on Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[3] C. Koch and I. Segev, *Methods in Neural Modeling.* MIT Press, 1989.

[4] A. Andreou, K. Boahen, P. Pouliquen, A. Pavasović, R. Jenkins, and K. Strohbehn, "Current mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Net.*, vol. 2, pp. 205–213, 1991.

[5] J. Franca and Y. Tsidivis, "Micropower techniques," in *Design of VLSI Circuits for Telecommunications and Signal Processing*, Prentice Hall, 1994.

[6] J. Kramer, R. Sarpeshkar, and K. C., "An analog VLSI velocity sensor," in *Proc. Int. Symp. Circuit and Systems ISCAS '95*, (Seattle, WA), pp. 413–416, May 1995.

[7] T. Delbrück, *Investigations of analog VLSI visual transduction and motion processing.* PhD thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1993.

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

[9] K. Boahen and A. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," in *NIPS91 Proceedings*, IEEE, 1992.

[10] T. Delbrück, ""Bump" circuits for computing similarity and dissimilarity of analog voltages," in *Proc. IJCNN*, pp. I–475–479, June 1991.

[11] T. Morris, D. Wilson, and S. DeWeerth, "Analog VLSI circuits for manufacturing inspection," in *Conference for Advanced Research in VLSI-Chapel Hill, North Carolina*, Mar. 1995.