

# Computing Motion Using Analog VLSI Vision Chips: An Experimental Comparison Among Different Approaches

TIMOTHY HORIUCHI, WYETH BAIR, BROOKS BISHOFBERGER, ANDREW MOORE AND  
CHRISTOF KOCH

*Computation and Neural Systems Program, 216-76, California Institute of Technology, Pasadena, CA 91125*

JOHN LAZZARO

*University of California, Berkeley, Computer Science EECS, Berkeley, CA 94720*

Received .

## Abstract

We have designed, built and tested a number of analog CMOS VLSI circuits for computing 1-D motion from the time-varying intensity values provided by an array of on-chip phototransistors. We present experimental data for two such circuits and discuss their relative performance. One circuit approximates the correlation model while a second chip uses resistive grids to compute zero-crossings to be tracked over time by a separate digital processor. Both circuits integrate image acquisition with image processing functions and compute velocity in real time. For comparison, we also describe the performance of a simple motion algorithm using off-the-shelf digital components. We conclude that analog circuits implementing various correlation-like motion algorithms are more robust than our previous analog circuits implementing gradient-like motion algorithms.

## 1 Introduction

There exist two broad categories of algorithms for recovering the optical flow field underlying the time-varying intensity patterns falling onto a retina or camera (for an overview, see [36]).<sup>1</sup> We will focus here on the class of motion algorithms that uses intensity or a linear function of the intensity at every location to compute the optical flow field throughout the image. During the last decade there has been increasing interest in these *intensity-based* or *short-range* methods (for a review see [10] and [31]). The two main approaches that have been proposed for determining the optical flow are the *correlation*, *second-order*, or *spatio-temporal energy* methods [8, 29, 1, 38, 41, 32, 3, 5] and the *differential methods* [28, 6, 14, 34, 9, 42, 37, 40]. It is common to all correlation methods that the intensity  $I(x, y, t)$  is passed through a linear spatio-temporal filter and multiplied with a delayed version of the filtered intensity from a neighboring receptor [29]. The output of these methods is a quadratic functional from which velocity or speed has to be extracted. Gradient methods,

on the other hand, exploit the relationship between the velocity and the ratio of the temporal to the spatial derivative: that is,  $v = -I_t/I_x$ . These methods yield a direct estimate of the optical flow field. However, they require evaluation of first- or second-order spatial and temporal derivatives of the image intensities.

Common to all intensity-based motion algorithms is a large associated computational overhead, preventing real-time machine vision applications within most industrial, military, or deep-space/planetary settings except on anything but large, costly, and power-hungry computers. Special-purpose hardware for computing optical flow in real-time becomes therefore a very attractive possibility. Here at Caltech, the laboratories of Carver Mead as well as ours have focused on a special class of such vision systems, analog, nonclocked CMOS VLSI circuits with on-chip photoreceptor arrays [26], [15]. A number of working chips, integrating image acquisition with different early vision algorithms, such as filtering, edge detection, binocular stereo, and surface interpolation, have been designed, fabricated, and successfully tested (for an up-to-date

overview, see [24]). We present data from two different analog circuits for computing the 1-D optical flow associated with an on-chip 1-D photoreceptor array. Section 2 discusses a chip approximating a Reichardt correlation algorithm, while section 3 presents data from a mixed analog-digital circuit. This system, tracking thresholded zero-crossings, bears similarity to the Marr and Ullman [23] scheme of computing velocity along zero-crossings of the  $\nabla^2 G$  operator. We compare the performance of these two analog, nonclocked chips with that of a simple system built out of a 1-D CCD imager and a programmable microprocessor in section 4. Finally, in section 5, we compare the two analog chips to a gradient model analog chip, the Tanner-Mead motion detector circuit [33], and we discuss the difficulties of implementing the gradient model in analog VLSI. More details of our comparison with the Tanner-Mead circuit, as well as most of the material in this article, have been published as a conference proceedings [18].

### *1.1 Data Acquisition and Circuit Design*

When testing the performance of our different motion chips, we tried to directly compare their output under the same test conditions, in particular using the same stimulus and speed as well as background intensity. Accordingly, we built a conveyor-belt system using an electric motor; belts with square-wave gratings of various contrasts and spatial frequencies could be moved in view of the chips, with velocities that ranged over more than an order of magnitude. Moving stripe patterns were imaged onto the silicon surface using a narrow-aperture lens directly positioned onto the chip. However, we did not achieve our initial goal of comparing all of the chips under identical operational conditions. This was mainly due to the fact that the different circuits have different optimal operating characteristics (e.g., some operate best under very low light conditions while others require higher light intensities).

All the data shown in this article is based upon measured data from working chips and not from circuit simulations. The chips were implemented with a standard  $2.0\ \mu\text{m}$  CMOS process available through the MOSIS silicon foundry.

Finally, the design of these electronic circuits is motivated by our desire to understand the function of

their biological counterparts in the motion pathway of flies, rabbits, or primates. In fact, it has been our experience that thinking about biological motion-estimation systems (e.g., [16] [40]) leads to the design of more robust electronic circuits, while thinking about machine-vision systems leads to a better understanding of the problems—such as gain-control or the limitations in the precision of components—that any biological vision systems must face.

## **2 A Pulse-Coded Correlation Circuit**

The circuit discussed in this section was directly inspired by the correlation model as well as by the computational architecture found in the auditory system of owls [19]. We designed an analog VLSI chip that contains a large array of velocity-tuned units that correlate two events in time, using a delay-line structure [11]. In building motion-detection systems using correlation methods, a clocked system measures the image shift over a fixed sampling time, while a dedicated analog hardware approach lends itself to the measurement of image shift time over a fixed distance. The latter is a local computation that gracefully scales to different velocity ranges without suffering from the problems of extended interconnection. It is this local property that we are using to compute motion.

### *2.1 System Architecture*

Figures 1a and 1b show the conceptual design of the motion detector and the organization of the chip in two stages of processing: motion detection and aggregation of data. Motion detection begins by focusing the image directly onto a one-dimensional array of 28 on-chip hysteretic photoreceptors spaced  $50\ \mu\text{m}$  apart [4]. These photoreceptors enhance temporal changes in the incident light intensity. Functionally similar to a follower, the circuit has its highest gain at higher temporal frequencies. Additionally, the circuit has a compressive gain function for the amplitude of the signal, making it responsive to both small and large signals. Each photoreceptor is connected to a half-wave rectifying neuron circuit [20] that fires a single digital pulse of constant duration when it receives a quickly rising (but not falling) light-intensity signal. The duration of the pulses can be adjusted from approximately 1 ms to 0.08 ms.

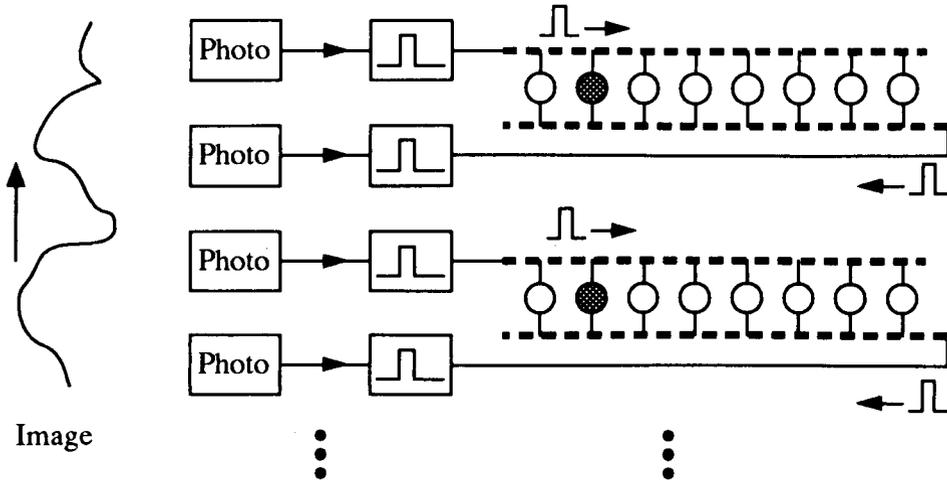


Fig. 1a. Block diagram of the pulse-coded correlation chip, showing only two motion detection units. Rising light intensity signals at the photoreceptors are converted into pulses and sent down the delay line. Velocity is determined by the location where two pulses meet. The axon delay-line is drawn as a heavy dashed line and correlators are drawn as circles. The actual chip fabricated contains 28 photoreceptors and 17 delay-line segments.

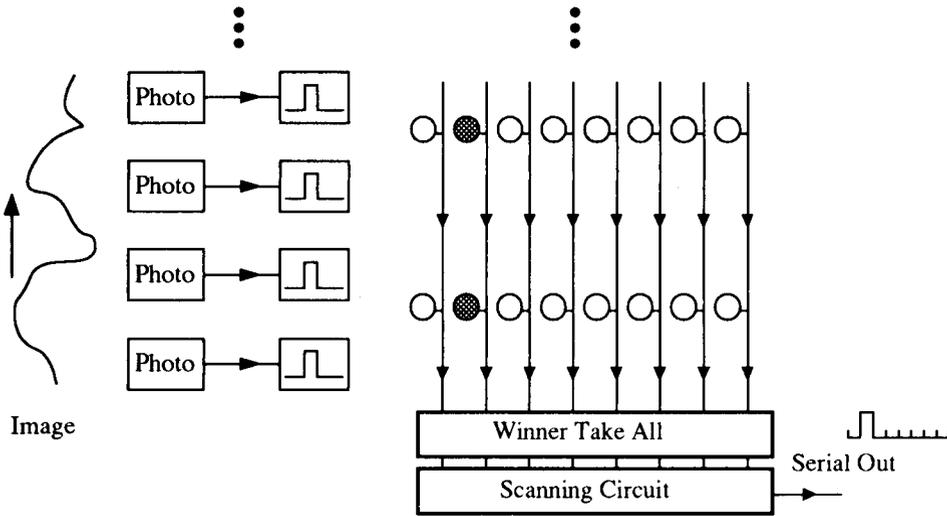


Fig. 1b. Diagram showing connections used to aggregate individual motion detector outputs. Individual correlators output current pulses which are summed according to velocity from across the field of view. The current sums for each velocity are compared and the winning velocity channel passes a voltage out of the scanning circuit that encodes that current.

This rising light-intensity signal is interpreted as a moving edge in the image passing over the photoreceptor. This signal is the image feature to be correlated. Note that from a computational point of view, we can use either the rising or the falling intensity values, corresponding to an ON or to an OFF edge, as the feature to be correlated with. Due to the faster turn-on characteristics of the photoreceptor, however, a rising signal was chosen. Each neuron circuit is connected to an axon circuit [26] that propagates the pulse down its length.

By orienting the axons in the two alternating propagation directions, as shown in figure 1a, any two adjacent receptors generate pulses that will “race” toward each other and meet at some point along the axon. Correlators between the two opposing axons detect when the two pulses pass each other, indicating the detection of a specific time difference. The width of the pulse in the axon circuits, which is adjustable, determines the pulse propagation rate down the line; the propagation rate determines the detectable velocity range.

To determine motion, the system effectively measures the time a feature takes to travel from one photoreceptor to one of its neighbors. By placing in parallel two delay lines that propagate signals in opposing directions, a temporal difference in signal start times from opposite ends will appear as a difference in the location where the two signals will meet. Between the axons, correlation units perform a logical AND with the axon signals on both sides. If pulses enter adjacent axons with zero difference in start times (i.e., infinite velocity), they meet in the center and activate a correlator in the center of the axon. If the time difference is small (i.e., the velocity is large), correlations occur near the center. As the time difference increases, correlations occur further out toward the edges. The left and right halves of the axon represent different directions of motion. At the chip level, when a single stimulus e.g., a step edge) is passed over the length of the photoreceptor array with a constant velocity, a specific subset of correlators is activated that all represent the same velocity.

The second step of processing, seen in figure 1b, is the data-aggregation stage. In order to obtain a global velocity value for the entire field of view, coincidences are computed at 27 different locations in the image. A current summing line is connected to correlators from different image locations that represent the same velocities. When a correlation occurs, a current is passed onto the line and thus the total current represents the level of confidence associated with the velocity at a given time. Because the frequency of matches affects the confidence of the data, scenes that are denser in edges provide more confident data and thus the chip responds more quickly. The current sums are then passed to a winner-take-all circuit [21] as one of the competing time-delay channels. The winner of the winner-take-all computation corresponds to the bin that is receiving the largest number of correlated inputs. The output of the winner-take-all is then scanned off the chip using an external input clock.

Since the major sources of error in the computation are related to fabrication offsets and noise. Component nonuniformities in the axon cause the pulses to be of slightly different durations, thereby changing the propagation speeds at each location in the axon. This can shift the resulting correlation position, with the accumulated errors being the largest at longer time differences. By aggregating these noisy motion outputs, we obtain a more accurate estimate of the velocity.

Note that the scheme we use to compute the velocity—estimating the coincidence event receiving the maximal amount of support—approximates the “ridge” strategy Grzywacz and Yuille [7] advocate to compute velocity from a population of spatio-temporally oriented receptive fields.

**2.1.1 Single Versus Bursting Mode.** The circuit described thus far uses a single pulse to indicate a passing edge. Due to the statistical nature of this system, a large number of samples are needed to make a confident statement of the detected time difference. By externally increasing the amplitude of the signal passed to the neuron circuit during each event, the neuron can be made to fire a burst of pulses in quick succession. With an increased number of pulses traveling down the axon, the number of correlations increases, but with a decrease in accuracy due to the multiple incorrect matches. The incorrect correlations are not at random, however, but cluster around the correct velocity. The end result is a net decrease in resolution in order to achieve confidence in the final output.

The chip output is the measured time difference of two events in multiples of  $\tau$ , the time-delay of a single axonal section. The final velocity  $v$  is given by  $\text{const}/\Delta t$ , where  $\Delta t$  corresponds to the signed time difference (measured in seconds/pixel). We set  $\text{const} = 1$ . Due to this inverse relationship, we expect to obtain the highest resolution for slow speeds. However, due to the relatively small number of correlations at slower speeds, the signal-to-noise ratio will decrease. This will be less troublesome as larger arrays of photoreceptors are implemented. The variable resolution inherent in this computation can be an acceptable feature for control of robotic motion systems since higher-velocity motions are often coarse, with fine control needed only at the lower velocities.

## 2.2 Performance

We fabricated the circuit described by figure 1 using a double polysilicon  $2\ \mu\text{m}$  process on a MOSIS Tiny Chip die containing about 8000 transistors. The chip has 17 velocity channels (8 channels in each direction as well as a center channel), and an input array of 28 photoreceptors. The voltages from the winner-take-all circuit are scanned out sequentially by on-chip scanners, the only clocked circuitry on the chip.

In testing the chip, gratings of varying spatial frequencies and natural images from newspaper photos and advertisements were mounted on a rotating drum in front of the lens. Although the most stable data were collected using the gratings, both image sources provided satisfactory data. Figure 2 shows the winning time interval channel vs. actual time delay. The response is linear as expected. In figure 3, the data from figure 2 is converted into a measured velocity vs. input velocity plot. At the lower velocities, as described above, correlations occur at a lower rate and therefore occasionally some of the lowest velocity channels fail to respond. This is interpreted as zero velocity. Increasing the number of parallel photoreceptor channels will improve this situation. The circuit has been shown to measure, with varied settings of the axonal unit time constant, velocities from about 50 pixels/s to over 1150 pixels/s. Any given setting will measure a range of velocities just over one order of magnitude.

The circuit response time depends strongly upon the type of stimuli used and the velocity range detected. Since the number of correlations per second determines the rate of current being passed to the winner-take-all circuit, channels with the highest correlation rate will win. For either faster velocities or stimuli with denser edges, the chip will respond more quickly than for slower speeds or sparser stimuli. While

strongly dependent upon many parameter settings and conditions, typical response times run from 0.5 s up to 2 s.

The performance under differing light levels depended primarily upon the ability of the photoreceptor and feature extraction circuit to deliver reliable feature detection signals. The hysteretic photoreceptor is extremely sensitive to both large and small changes in the intensity and allows the chip to operate at quite low light levels [4]. Usable data were obtained with DC illumination from  $1 \text{ mW/m}^2$  up to  $1000 \text{ mW/m}^2$  over various gain settings of the coupling circuit between the photoreceptor and axon circuits.<sup>2</sup> With any particular gain setting it is possible to operate reliably over slightly greater than one order of magnitude of light intensity. The limiting factor for illumination is at the higher end where the DC level of the photoreceptor begins to reduce the amount of signal that is coupled into the neuron circuit.

### 2.3 Summary

Our implementation of the correlation model shows promise due to its relative robustness to light levels and contrast. Some of the issues to be discussed include flicker sensitivity, noise, velocity range, and possible design expansion.

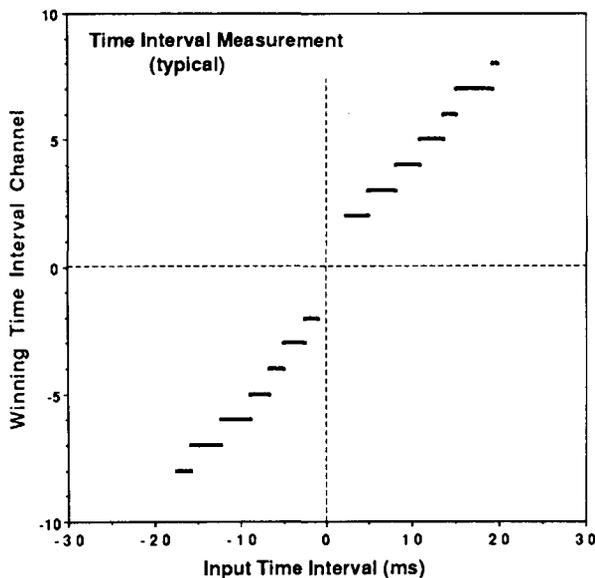


Fig. 2. Plot of winning output channel vs. input time interval. Each output channel represents a difference in the axon start times of  $2\tau$ . The highest velocities correspond to the shortest time interval. The horizontal shift in the negative velocities is believed to be due to a propagation delay in the long signal line on the chip. Its effect can be seen in the slightly different slopes in figure 3.

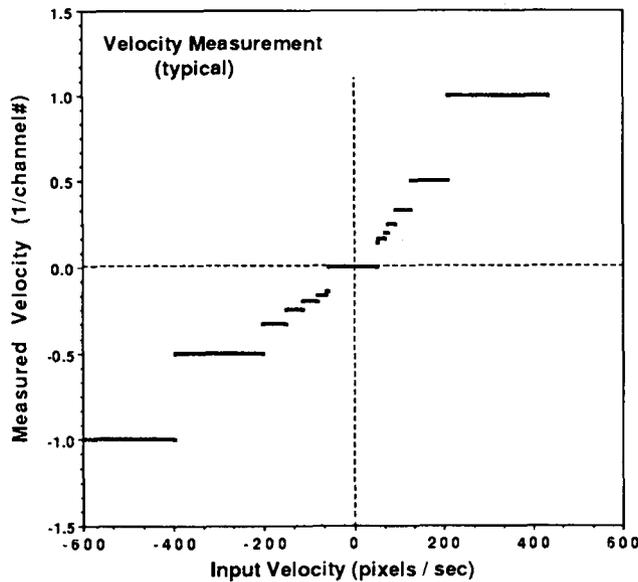


Fig. 3. Plot of measured velocity vs. input velocity using data from figure 2. Each output channel represents a specific velocity related to the inverse of the difference in axon start times. This relationship maps a large range of velocities into the short time interval channels, giving a coarse resolution at higher speeds. We only show the response of the chip up to about  $\pm 500$  pixels/s, since our conveyor-belt system fails to move at higher speeds.

The first and most limiting aspect of this particular chip is its feature-extraction component. The hysteretic photoreceptor is intended to enhance the temporal changes in the light signal and thus detect edges. This circuit is extremely sensitive and allowed the operation of the circuit to continue down into very low light levels and for low-contrast stimuli. Temporally differentiating circuits are, however, quite troublesome due to noise amplification; in our circuit this manifests itself in the form of flicker sensitivity. Under fluorescent and some AC incandescent lighting, all of the photoreceptors' circuits fire synchronously at 120 Hz, indicating infinite velocity and thus making it unusable under such lighting conditions. A modification of the circuit to include a more sophisticated feature-extraction stage would eliminate this problem.

The statistical nature of the computation allows the system to perform successfully in the presence of noise as well as to produce a usable measure of confidence level. By summing votes for specific velocities across the chip and by using the burst-mode described above, it is possible to obtain a strong signal above the noise. If a different method for extracting the detected time difference were used in the place of the winner-take-all circuit, the current levels in each of the summing lines would provide a confidence level for each particular channel. It is also interesting to note that despite

the apparent loss of resolution caused by operating in the bursting mode, the confidence level measure can provide additional information to allow interpolation between the discretized velocity outputs.

A natural next step in developing motion detection circuits is the design of a 2-D array of these motion detection units in order to integrate motion over an array. It should be remembered, however, that this particular circuit exploits the second spatial dimension on the silicon to represent time, making it necessary to use three dimensions to build a similarly designed 2-D motion detector.

### 3 Motion from Zero-Crossings

This system estimates velocity by using an analog chip to localize zero-crossings and a digital microprocessor to track the zero-crossings. It approximates the scheme proposed by Marr and Ullman [23], but without their use of  $X$  and  $Y$  cells. The analog chip is a one-dimensional 64-pixel device which exploits on-chip photoreceptors and the natural filtering properties of resistive networks to implement an edge-detection scheme similar to the Difference of Gaussians (DOG) operator proposed by Marr and Hildreth [22]. The chip localizes the zero-crossings associated with the difference of two

exponential weighting functions, and reports the locations of only those zero-crossings that have a slope greater than an adjustable threshold. A conventional digital microprocessor receives the locations of the zero-crossings from the analog chip and tracks their displacements over time to compute velocity.

### 3.1 The Analog VLSI Zero-Crossing Chip

Similar to a DOG, our chip takes the difference of two filtered versions of the input light intensity, but we avoid the difficulties associated with implementing Gaussian kernels in silicon and filter with first-order resistive networks instead. In these networks, each node is connected to an input data voltage via a conductance  $G$  and to its two direct neighbors via resistances  $R$ . The characteristic length, corresponding to the standard deviation  $\sigma$  of a Gaussian, of the resulting filter function is given by  $\lambda = 1/\sqrt{RG}$ . In the case of the meshsize going to zero, the Green's or impulse response function of this network, that is its voltage in response to a delta pulse of current, is simply  $1/2\lambda e^{-|x|/\lambda}$ . This function has a behavior qualitatively similar to the Gaussian  $e^{-x^2/2\sigma^2}$ , except around the origin where the exponential function has discontinuous first derivatives. Two resistive networks with different values of  $\lambda$ , achieved by using different resistances, then implements a discretized version of the difference-of-exponentials, or DOE, operator. This filter has some similarities to a  $\nabla^2 G$  operator; for instance, the output of this DOE operator to a constant input is zero; in general, convolving an  $n$ th-order polynomial  $f(x) = x^n$  with this operator yields a  $(n - 1)$ th-order polynomial (for more details see [2]). The rounded peak of the Gaussian around the origin makes the DOG look like a "Mexican-hat," while the pointed peak of the decaying exponential makes the operator implemented by our chip appear more like a pointed "Witch-hat." Additional circuitry then localizes zero-crossings in the input image convolved with the DOE operator, zero-crossings which ideally correspond to edges in the image and object boundaries in the scene. The entire process, from imaging to edge detection, occurs on-chip in four stages of analog circuitry: photoreceptors capture incoming light, a pair of 1D resistive networks smooth the input image with the exponential operator, transconductance amplifiers subtract the two smoothed images, and

mixed analog and digital circuitry localizes and thresholds the zero-crossings. Figure 4 shows a block diagram of the first three stages of this processing, which is described in more detail below.

The chip receives input from an array of photoreceptors spaced  $100 \mu\text{m}$  apart, encoding the logarithm of light intensity as a voltage. The set of voltages from the photoreceptors are reported to corresponding nodes of two resistive networks via transconductance amplifiers connected as followers. The followers' voltage biases can be adjusted off-chip to independently set the data conductances for each resistive network. The network resistors are implemented using saturating resistors developed by Mead [26]. Another pair of voltage biases allow independent off-chip adjustment of the resistances along the two resistive networks. The data conductance and network resistance values determine the space constant of the smoothing filter which each network implements. The sets of voltages along the networks represent the two filtered versions of the image. These filtered images are subtracted by wide-range transconductance amplifiers [26] which produce an output current proportional to the difference in voltage applied across their inputs. The array of currents produced by this circuitry corresponds to the result of applying the discretized DOE operator to the input image.

The final stage of processing detects zero-crossings in the array of currents from the wide-range amplifiers and implements a threshold on the slope of those zero-crossings. Each pair of neighboring currents charge or discharge the inputs of an exclusive OR gate. The binary output of this gate indicates the presence or absence of a zero-crossing between two nodes. A second signal is generated by subtracting a threshold current from the magnitude of the difference between the neighboring currents mentioned above. If the charging current, representing the slope of the zero-crossing, is greater than the threshold current set by an off-chip bias voltage, then this signal charges a node to logical 1, otherwise, that node is discharged to logical 0. The conjunction of a zero-crossing and a steep slope causes the chip to report the existence of an edge at that location for any of 63 possible locations. The output can be thought of as a 63-bit word where each bit codes for the presence or absence of a zero-crossing at that particular location.

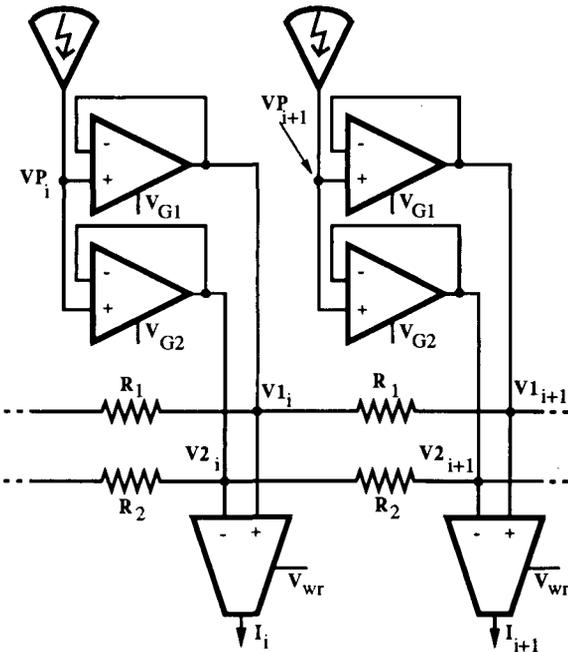


Fig. 4a. Zero-crossing chip circuit diagram. Logarithmic photoreceptors encode light intensity as voltages,  $V_P$ , which are reported to the nodes of two resistive networks via transconductance amplifiers connected as followers. The voltage biases  $V_G$  set the conductances. The network resistances,  $R_1$  and  $R_2$ , are implemented as saturating resistors and are externally adjustable from voltage biases. The filtered images,  $V_1$  and  $V_2$ , are subtracted by wide-range transconductance amplifiers which output a current,  $I$ , proportional to the voltage difference across their inputs.

### 3.2 Data from the Zero-Crossing Chip

Figure 5 shows data taken from the zero-crossing chip. The input light profile is a bright bar. Oscilloscope traces show the filtered versions of the image from the nodes along the resistive networks. By setting the space constants of the networks differently, we have achieved varying amounts of smoothing. The difference of these two smoothed voltage traces is shown in figure 5c; arrows indicate the locations of two zero-crossings which the chip reports at the output. The reported zero-crossings accurately localize the positions of the edges in the image. Other zero-crossings were not reported because their slopes were less than the adjustable threshold. Thresholding allows for noise and imperfections in the circuitry and can be used to filter out weaker edges which are not relevant to the application.

Figure 6 shows the response when two fingers are held 1 m in front of the lens and swept across the field of view. The fingers appear as bright regions against

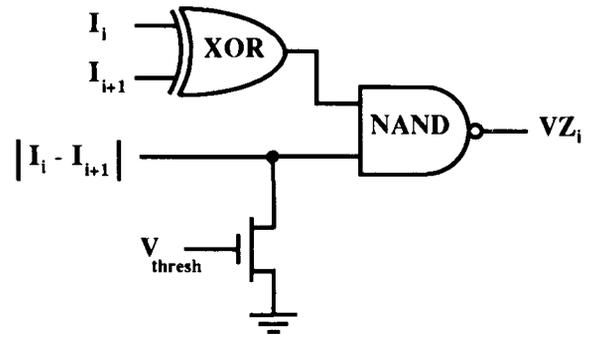


Fig. 4b. Zero-crossing detection and thresholding circuit. The final stage of processing detects zero-crossings in the sequence of currents  $I$  and implements a threshold on the slope of the zero-crossings. Neighboring currents charge or discharge the inputs of the exclusive-OR gate, which signals the presence of a zero-crossing with a logical 1 output. A threshold current is subtracted from the magnitude of the difference  $I_i - I_{i+1}$ . If the threshold current is larger than the difference, the input to the NAND is logical 0. The NAND output then signals the presence of an above threshold zero-crossing with logical 0.

a darker background. The chip accurately localizes the four edges (two per finger) as indicated by the pulses below each voltage trace. As the fingers move quickly back and forth across the field of view of the chip, the image and the zero-crossings follow the object with no perceived delay. From sequences of frames like these, we can compute optical flow. Note that these are not successive frames, but are more representative of every 100th frame that the motion detection system will receive (see below).

### 3.3 The Microprocessor and Motion Detection

The motion detection system consists of a zero-crossing chip interfaced to a 12.5 MHz 80286 microprocessor-based single-board computer. The interface allows the microprocessor to receive 63-bit frames of zero-crossing data at just over 320 frames per second. As each new frame is read, the microprocessor updates the cumulative displacement of each zero-crossing and increments the number of frames over which that displacement has occurred. The system assumes that zero-crossings will not move more than 2 pixels per frame. With our optics, this assumption is violated only at velocities in excess of approximately 700 degrees per second.

After tracking zero-crossings for  $n$  frames, individual velocities are computed by dividing the total displacement in pixels by  $n$ . Obviously, larger values of  $n$  are necessary to achieve more precision, particularly for

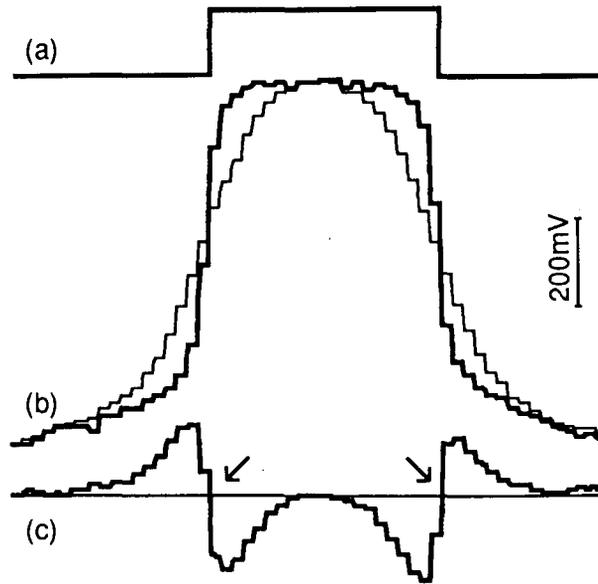


Fig. 5. Measured response of the zero-crossing chip to a light bar stimulus. (a) Input light intensity, (b) voltage traces from the two resistive networks, and (c) difference of the voltage traces, corresponding to the image intensities convolved with a difference-of-two-exponentials (DOE) operator. The circuit correctly localizes the two edges (arrows). The threshold suppresses zero-crossings with small magnitude slope.

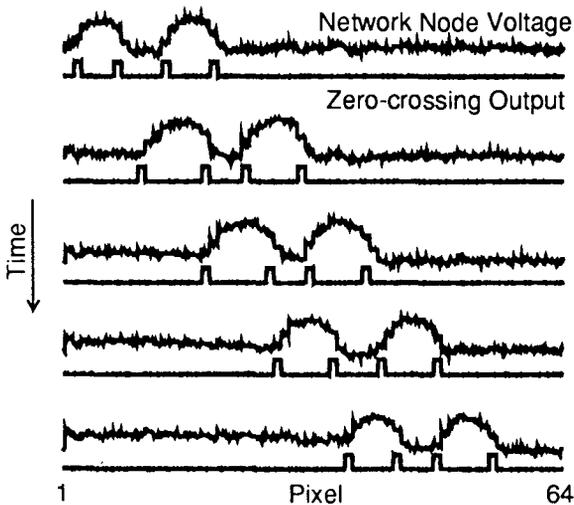


Fig. 6. Zero-crossing chip response as two fingers are waved about 1 m in front of the lens. The upper traces show voltages from one resistive network; the lower traces show positions of zero-crossings reported by the chip.

slow zero-crossings. The lowest nonzero velocity for which data are shown in figure 7 corresponds to less than one-tenth pixel per frame. The full-field average velocity is computed by averaging over the individual zero-crossing velocities. Full-field velocity may be computed after every new frame of data is received from the zero-crossing chip, but in practice it is convenient

to compute this velocity every  $n$  frames to reduce the correlation with the previous full-field velocity value. Figure 7 shows data from the system for  $n = 320$  frames (i.e., 1s). The mean and standard deviation of the system output are shown (sample size 60). Obviously, as  $n$  is decreased, the variability in the output will increase. A reasonable value for  $n$  should be chosen based on the desired precision and the expected velocity range for a particular application.

### 3.4 Performance Analysis

Figure 7 shows the output of the system for image velocities ranging from 0 to 450 pixels per second at two different light levels. The error bars show the standard deviation of the output velocity. Over most of this range, the standard deviation was less than four percent of the average value. The stimulus velocity range was limited by the speed of the stimulus and the geometry of the optics. The data shown for  $10 \text{ W/m}^2$  is representative of the system response for light levels of  $1 \text{ W/m}^2$  and higher. Below  $1 \text{ W/m}^2$ , the zero-crossing chip was unable to localize higher-velocity zero-crossings due to R-C time constants associated with the circuitry of the analog chip. Also, as seen in figure 7, the reported velocity is less than the image velocity at low intensities but remains linear. At lower light levels,

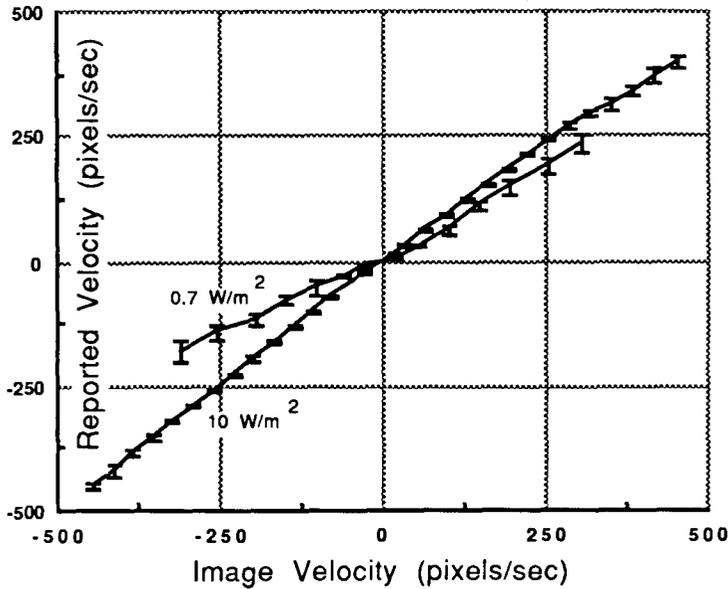


Fig. 7. Zero-crossing motion detection system output for two light intensities. At higher light intensities (about  $10 \text{ W/m}^2$  and above) the output is linear and accurate over a large range of velocities. At lower intensities, the zero-crossing chip cannot localize fast edges, and lower signal-to-offset ratios introduce spurious zero-crossings that compromise accuracy (error bars show standard deviation).

zero-crossings due to offsets are more prevalent and introduce zeros into the average velocity computation, thus lowering the reported velocity. Such spurious zero-crossings can undermine the accuracy of the average velocity in more subtle ways as well. As light intensity drops, the linear range of output for this system becomes smaller around zero. Below  $100 \text{ mW/m}^2$ , the zero-crossing chip fails to detect edges, and the system cannot even detect direction of motion. Qualitatively, the useful range of operation for this system is from bright sunlight to dim indoor fluorescent or incandescent lighting. This range is achieved without changing gain or other parameters.

The zero-crossing chip fails at low light and contrast levels due to the small signal-to-offset ratio. Imperfections in the fabrication process cause many of the signals in the analog chip to be corrupted. The magnitude of this noise, called offsets, is a substantial fraction of the magnitude of the signal reported by the logarithmic photoreceptors. Although the logarithmic receptor allows operation over a wide range of lighting conditions, it compresses the range of voltages that are used to encode any particular scene and therefore decreases the signal-to-noise ratio. A hysteretic photoreceptor similar to the one used in the correlation chip described in the previous section would improve the signal-to-noise ratio, but would also increase sensitivity to lighting changes, and possibly compromise sensitivity to small velocities.

Another limitation in the performance of the zero-crossing chip is the photoreceptor response time. The measured response time of the chip to the appearance of a detectable discontinuity in light intensity varies from about  $100 \mu\text{s}$  in bright indoor illumination to about  $10 \text{ ms}$  in a dark room, and these response times seem to be dominated by the phototransistor.

Finally, spatial and temporal aliasing limit the performance of this system. As the spatial frequency of features increases, zero-crossings appear closer together and the correspondence problem arises. This is a function of the environment, the lens, and the photoreceptor spacing on the chip. Interfacing the zero-crossing chip to a digital computer requires clocking the output from the chip. In theory, this causes temporal aliasing at higher velocities, but the slow time response of the photoreceptors causes the system to fail before temporal aliasing is noticed.

#### 4 A Fully Digital System

In order to compare our analog circuits against their digital counterparts as well as to be able to quickly test vision algorithms using the reliability of a CCD system, a completely digital circuit was built incorporating a linear 256 pixel element as well as a small microprocessor designed for real-time use. We use a Harris

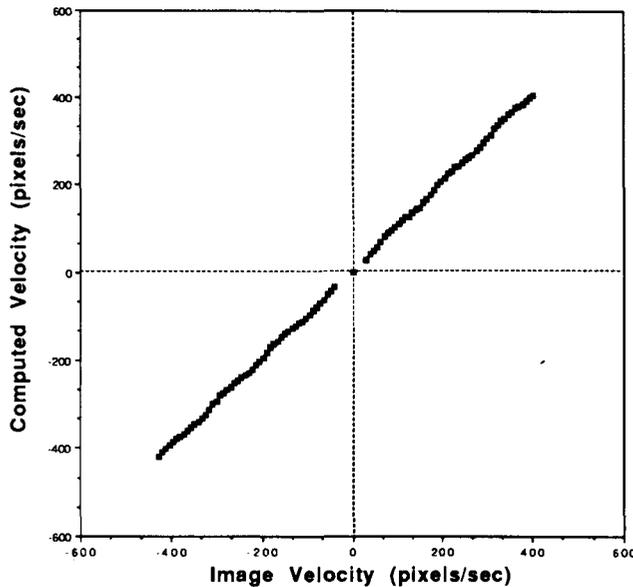


Fig. 8. The output of our real-time digital system, interpolated to subpixel accuracy, using a one-dimensional, commercial 256 pixel CCD camera. The reported velocity remains constant as long as the image intensity is above  $120 \text{ mW/m}^2$ .

RTX2001A microprocessor operating at 8 MHz. It includes 8 K of RAM memory and executes FORTH directly. The system computes a 1-D field velocity based upon a simple correlation method. The CCD camera retrieves image data at a maximum rate of 2800 images/second with greater than 12 bits of accuracy. Each pixel is sent through an 8-bit A/D converter, which updates the processor memory at a maximum rate of 2000 images/second. With this structure, the processor is able to access the image residing in memory as rapidly as possible.

The global image velocity is estimated by storing two consecutive images sampled 10 ms apart. These raw images are then subtracted from each other and the absolute value of this difference, summed over the entire 1-D image, is computed. We term this the error associated with a 0 pixel shift. The same operation is also performed when the second image is shifted by  $\pm 1$ ,  $\pm 2$ ,  $\pm 3$ , and  $\pm 4$  pixels with respect to the first image. The global motion estimate corresponds to the spatial offset with the smallest associated error divided by the temporal sampling time (10 ms). For additional accuracy, we interpolate the spatial offset to subpixel accuracy using the associated errors as weights. This very simple algorithm approximates a correlation model in a manner reminiscent of the algorithm of Bülthoff et al. [3]. The microprocessor then retains the second image, waits 10 ms, stores a new image, and performs the shift comparison again. In the interest of computa-

tional speed, we did not perform any filtering on the image. In spite of this lack of prefiltering, the algorithm performed remarkably well (figure 8). Simple modifications of this algorithm enable the system to compute the spatially varying optical flow field as well as time-to-contact (not shown here).

At any setting, the CCD provides valid output over light intensities differing by more than one order of magnitude, in DC or 120 Hz illumination. The system operates down to  $120 \text{ mW/m}^2$ , at which point features are no longer detected and the reported velocity drops sharply to zero. Preliminary tests show promising results from zero-crossing detection and from edge-tracking schemes.

## 5 Discussion

We have described three methods of motion measurement: pulse correlation on a single analog chip, digital tracking of zero-crossings provided by a single analog chip, as well as digital autocorrelation of grey levels from a CCD imager. In this section, the three methods of motion measurement are compared. Each of these motion circuits utilizes a different type of photoreceptor, leading to dramatically different performances with respect to range of light intensity, contrast, and flicker. The difference in algorithms leads to different noise-rejection levels, failure modes, and causes different concerns for possible expansion to two dimensions.

### 5.1 Photoreceptors

The pulse-correlation chip uses hysteretic photoreceptors that have a very high gain AC response for small changes in intensity [4]. These photoreceptors are also sensitive to large changes in light intensity, due to the compressive nature of their gain. High gain endows the chip with sensitivity to low contrasts, but it causes the chip to respond strongly to the 120 Hz flicker of ordinary indoor lighting. Since the motion algorithm requires features to track, any number of other feature-detecting circuits would increase performance.

The zero-crossing chip uses logarithmic photoreceptors which compress about six orders of magnitude of light intensity into a voltage range from about 2.0 volts to 3.5 volts. This extreme compression leads to a small signal which, at the low end of light intensity, becomes dominated by noise from fabrication offsets in the photoreceptors and the subsequent circuitry. The zero-crossing chip was useful only at the highest two decades of light intensity in the photoreceptor's range. Substituting a higher gain or adaptive photoreceptor would increase performance.

The photodetector of the fully digital system, a CCD camera, responds to light over a range of two orders of magnitude; the range of operation can be varied with the integration time of the CCD. The CCD has a very low-noise characteristic compared to the photoreceptors on our analog chips. This is not surprising as many hundreds of man-years of engineering have been devoted to building very accurate CCD cameras. When measuring power consumption, however, the on-chip photoreceptors with the associated feature-detection and motion circuits draw negligible power in comparison to the CCD imager alone.

### 5.2 Algorithms

The main feature of the pulse-correlation algorithm is its use of the second dimension of silicon to represent time. At each point in the one-dimensional image, there are a large number of units tuned for a specific range of velocities, instead of a single unit that responds to all possible velocities. This resulted in a simpler detection circuit at the expense of valuable silicon area. This feature makes the extension to a two-dimensional motion detector rather difficult.

Unlike traditional methods of correlation on sequential machines such as the digital system described here

which measure image shift between a fixed sampling interval, our hardware implementation favors the measurement of travel time of the image over a fixed distance. While this particular chip aggregates signals from detectors across the chip, it is also possible to obtain the velocity at each point in the image.

In contrast to this silicon-based motion algorithm, tracking zero-crossings across many pixels in software provided a good method of obtaining low noise measurements of motion. By averaging out errors due to the spatial sampling and by eliminating transient edges, more reliable velocity measurements were obtained. A possible drawback to this algorithm, which is simple in software, is its difficulty to implement in hardware. The main strength of this system is the real-time feature extraction stage performed by the analog chip, which prevents the digital microprocessor from becoming computationally overloaded.

The intensity-based correlation scheme used by the digital system had the best overall performance in our experiments. While the feature-tracking algorithm was the next best, noise in the imaging and feature-extraction stages limited its performance. The digital system performed error comparisons for shifts of the entire image to obtain robust motion estimates. The very low noise figure of the CCD camera made this technique successful. Using smaller windows for correlation, however, introduced larger errors, which can be attributed to regions in the image with constant intensity values and to flicker. The intensity-based correlation scheme was chosen for its speed and simplicity. Preprocessing requirements such as smoothing, feature extraction, or convolution are performed in parallel by resistive nets and other simple analog hardware but are computationally quite expensive in software.

We have reported elsewhere [27] our attempts at building robust circuits implementing a version of the gradient algorithm ([27]; see also [34]). These circuits are not very successful for a number of reasons. (1) The gradient algorithm is intrinsically ill conditioned, requiring heavy smoothing or equivalent operations. This makes it particularly ill suited to our hardware, given the limited accuracy we can achieve in our circuits operating in the subthreshold regime. (2) The output of these chips varies greatly with ambient light level and contrast. The ratio of the temporal and spatial gradients should be independent of the overall light intensity in theory, but is not in practice, given our hardware. (3) The division of the two gradients, implemented in our chip using a feedback scheme [34],

needs to be carried out in a different, more accurate circuit technology, such as translinear bipolar circuits. We believe that gradient methods are not robust enough to yield reliable estimates of motion, except under special circumstances (such as on an optimally lit lab bench using a high-contrast stimulus). Correlation methods are substantially more robust than methods requiring the evaluation of spatial and temporal derivatives! Of course, correlation methods do appear to be used throughout the animal kingdom, from flies to humans [35], [10], [31], [5].

The work reported here represents an effort over several years to build robust, analog motion sensors with on-chip photoreceptors. We have achieved moderate success in that we are able to compute the global velocity of a 1-D image in real-time. We are continuing to port our vision chips onto small, highly mobile and autonomous vehicles (toy cars) in order to demonstrate their use as smart sensors in a real-life environment [17]. We are also continuing our quest for more robust motion circuits. Our next major goal, however, is the design of circuits enabling us to compute the 1-D optical flow, that is, to estimate a velocity vector at different locations across the retina, in order to compute such quantities as the focus of expansion and time-to-contrast [12]. We are continuing to focus our efforts on various correlation-like motion circuits.

### Acknowledgments

We thank Carver Mead for providing laboratory resources for the design and fabrication of the analog chips and Steve DeWeerth, John Harris, Andy Moore, and John Tanner for their help in getting these motion chips to work. We thank the Office of Naval Research, the National Science Foundation, Rockwell International Science Center, and Hughes Aircraft Corporation for financial support of VLSI research in our laboratory. W.B. was supported by a NSF Graduate Fellowship and performed some of the work described here at the Hughes Aircraft AI Center.

### Notes

1. Given the topic of this article, we make no distinction here between the optical flow field induced by the time-varying image intensities and the underlying 2-D velocity field, a purely geometrical concept [12], [39].
2. Note that the solar-constant is about  $1400 \text{ W/m}^2$  while a value of  $1 \text{ mW/m}^2$  corresponds to candle-light illumination.

### References

1. E.H. Adelson and J.R. Bergen, Spatio-temporal energy models for the perception of motion, *J. Opt. Soc. Amer. A* 2:284-299, 1985.
2. W. Bair and C. Koch, An analog VLSI chip for finding edges from zero-crossings. In *Advances in Neural Information Processing Systems*, vol. 3, R. Lippman, J. Moody, and D.S. Touretzky, eds., Morgan Kaufmann, San Mateo, CA, pp. 399-405, 1991.
3. H.H. Bülthoff, J.J. Little, and T. Poggio, Parallel computation of motion: computation, psychophysics and physiology, *Nature* 337:549-553, 1989.
4. T. Delbrück and C. Mead, An electronic photoreceptor sensitive to small changes in Intensity. In *Neural Information Processing Systems 1*, D. Touretzky, ed., Morgan Kaufmann, San Mateo, CA, pp. 720-727, 1989.
5. M. Egelhaaf, A. Borst, and W. Reichardt, The computational structure of a biological motion detection system," *J. Opt. Soc. Amer.* 6:1070-1087, 1989.
6. C.L. Fennema and W.B. Thompson, Velocity determination in scenes containing several moving objects, *Comp. Graph. Image Process.* 9:301-315, 1979.
7. N.M. Grzywacz and A.L. Yuille, A model for the estimate of local image velocity by cells in the visual cortex, *Proc. Roy. Soc. London B* 239:129-161, 1990.
8. B.Hassenstein and W. Reichardt, Systemtheoretische Analyse der Zeit, Reihenfolgen, und Vorzeichenbewertung bei der Bewegungsperzeption des Rüsselkäfers *Chlorophanus*, *Z. Naturforsch.* 11b:513-524, 1956.
9. E.C. Hildreth, *The measurement of visual motion*, MIT Press: Cambridge, MA, 1984.
10. E. Hildreth and C. Koch, The analysis of visual motion: from computational theory to neuronal mechanisms," *Annu. Rev. Neurosci.* 10:477-533, 1987.
11. T. Horiuchi, J. Lazzaro, A. Moore, and C. Koch, A delay-line based motion detection chip. In *Advances in Neural Information Processing Systems*, vol. 3, R. Lippman, J. Moody, and D. Touretzky, eds., pp. 406-412, Morgan Kaufmann, San Mateo, CA, 1991.
12. B.K.P. Horn, *Robot Vision*, MIT Press: Cambridge, MA 1986.
13. B.K.P. Horn, Parallel networks for machine vision, *Artif. Intell. Lab. Memo no. 1071*, MIT, Cambridge, 1989.
14. B.K.P. Horn and G. Schunck, Determining optical flow, *Artif. Intell.* 17:185-203, 1981.
15. C. Koch, Seeing chips: analog VLSI circuits for computer vision, *Neural Comput.* 1:184-200, 1989.
16. C. Koch, H.T. Wang, B. Mathur, A. Hsu, and H. Suarez, Computing optical flow in resistive networks and the primate visual system, *Proc. IEEE Workshop on Visual Motion*, pp. 62-72, Irvine, CA, March 1989.
17. C. Koch, W. Bair, J.G. Harris, T. Horiuchi, A. Hsu, and J. Luo, Real-time computer vision and robotics using analog VLSI circuits, In *Advances in Neural Information Processing Systems*, vol. 2., D. Touretzky, ed., pp. 750-757, Morgan Kaufmann: San Mateo, CA, 1990.
18. C. Koch, A. Moore, W. Bair, T. Horiuchi, B. Bishofberger, and J. Lazzaro, Computing motion using analog VLSI vision chips: an experimental comparison among four approaches, *Proc. IEEE Workshop on Visual Motion*, pp. 312-324, Princeton, NJ, October 1991.

19. M. Konishi, Centrally synthesized maps of sensory space, *Trends Neurosci.* 4:163-168, 1986.
20. J. Lazzaro and C. Mead, Circuit models of sensory transduction in the cochlea. In *Analog VLSI Implementations of Neural Networks*, C. Mead and M. Ismail, eds., pp. 85-101, Kluwer: Norwell, MA, 1989.
21. J. Lazzaro, S. Ryckebusch, M.A. Mahowald, and C. Mead, Winner-take-all networks of  $O(n)$  complexity. In *Advances in Neural Information Processing Systems*, vol. 1, D. Tourestzky, ed., pp. 703-711, Morgan Kaufmann: San Mateo, CA, 1988.
22. D. Marr and E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London*, B 207:187-217, 1980.
23. D. Marr and S. Ullman, Directional selectivity and its use in early visual processing, *Proc. Roy. Soc. London* B 211:151-180, 1981.
24. B. Mathur and C. Koch, eds., Visual Information Processing: From Neurons to Chips, *Proc. SPIE*, San Diego, 1473, 1991.
25. C. Mead, A sensitive electronic photoreceptor. In *1985 Chapel Hill Conference on Very Large Scale Integration*, H. Fuchs, ed. Computer Science Press: Chapel Hill, NC, 1985.
26. C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley: Reading, MA 1989.
27. A. Moore and C. Koch, A multiplication-based analog motion detection chip. In *Visual Information Processing: From Neurons to Chips*. B. Mathur and C. Koch, eds., pp. 66-75, *Proc. SPIE* 1473, 1991.
28. H.H. Nagel, Analysis techniques for image sequences, *Proc. 4th Intern. Conf. Patt. Recog.* Kyoto, Japan, November 1978.
29. T. Poggio and W. Reichardt, Considerations on models of movement detection, *Kybernetik* 13: 223-227, 1973.
30. T. Poggio, V. Torre, and C. Koch, Computational vision and regularization theory, *Nature* 317:314-319, 1985.
31. T. Poggio, W. Yang, and V. Torre, Optical flow: computational properties and networks, biological and analog. In *The computing neuron*, R. Durbin, C. Miall, and G. Mitchison, eds., pp. 355-370. Addison-Wesley: Menlo Park, CA, 1989.
32. W. Reichardt, R.W. Schölgel, and M. Egelhaaf, Movement detectors of the correlation type provide sufficient information for local computation of 2-D velocity field, *Naturwiss. enschaften* 75:313-315, 1988.
33. J. Tanner and C. Mead, An integrated optical motion sensor, *VLSI Signal Processing II*, S-Y. Kung, R.E. Owen, and J.G. Nash, eds., pp. 59-76, IEEE Press: NY, 1986.
34. J. Tanner and C. Mead, Optical motion sensor. In *Analog VLSI and Neural Systems*, C. Mead, pp. 229-255, Addison-Wesley, Reading, MA, 1989.
35. S. Ullman, *The interpretation of visual motion*. MIT Press: Cambridge, MA, 1979.
36. S. Ullman, Analysis of visual motion by biological and computer systems, *IEEE Computer* 14:57-69, 1981.
37. S. Uras, F. Girosi, A. Verri, and V. Torre, A computational approach to motion perception, *Biol. Cybern.* 60:79-87, 1988.
38. J.P.H. van Santen and G. Sperling, A temporal covariance model of motion perception, *J. Opt. Soc. Amer. A* 1:451-473, 1985.
39. A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *Trans. Patt. Anal. Mach. Intell.* 11:490-498, 1989.
40. H.T. Wang, B. Mathur, and C. Koch, Computing optical flow in the primate visual system, *Neural Comput.* 1:92-103, 1989.
41. A.B. Watson and A.J. Ahumada, Model of human visual-motion sensing, *J. Opt. Soc. Amer. A* 2:322-341, 1985.
42. A.L. Yuille and N.M. Grzywacz, A computational theory for the perception of coherent visual motion, *Nature* 333:71-73, 1988.