



A Modular Multi-Chip Neuromorphic Architecture for Real-Time Visual Motion Processing

CHARLES M. HIGGINS* AND CHRISTOF KOCH

Division of Biology, 139–74 California Institute of Technology Pasadena, CA 91125, USA

E-mail: chuck@klab.caltech.edu; koch@klab.caltech.edu

Received September 3, 1999; Revised January 27, 2000

Abstract. The extent of pixel-parallel focal plane image processing is limited by pixel area and imager fill factor. In this paper, we describe a novel multi-chip neuromorphic VLSI visual motion processing system which combines analog circuitry with an asynchronous digital interchip communications protocol to allow more complex pixel-parallel motion processing than is possible in the focal plane. This multi-chip system retains the primary advantages of focal plane neuromorphic image processors: low-power consumption, continuous-time operation, and small size. The two basic VLSI building blocks are a photosensitive sender chip which incorporates a 2D imager array and transmits the position of moving spatial edges, and a receiver chip which computes a 2D optical flow vector field from the edge information. The elementary two-chip motion processing system consisting of a single sender and receiver is first characterized. Subsequently, two three-chip motion processing systems are described. The first three-chip system uses two sender chips to compute the presence of motion only at a particular stereoscopic depth from the imagers. The second three-chip system uses two receivers to simultaneously compute a linear and polar topographic mapping of the image plane, resulting in information about image translation, rotation, and expansion. These three-chip systems demonstrate the modularity and flexibility of the multi-chip neuromorphic approach.

Key Words: analog VLSI, vision chips, optical flow, stereo, neuromorphic

1. Introduction

Conventional approaches to the real-time analysis of an evolving visual scene take frames of image data from a camera and process pixels sequentially on a serial computer. A data transfer bottleneck is created between the camera and image processor, and a powerful computer is required to process high-resolution images in real-time. A more efficient way to approach such problems is to utilize a simple processor for each pixel, located in the focal plane near the imaging element. This strategy eliminates the data transfer bottleneck. The processor located at each imaging element runs in parallel with all other

processors in the focal plane, together accomplishing a tremendous amount of computation in a short period of time.

However, pixel-parallel focal plane image processing can only be taken to a certain level of complexity without incurring an unacceptably large pixel size. A passive CMOS imager pixel, incorporating no explicit image processing, can be fabricated with as few as one transistor along with the photosensitive element [1]. The addition of circuitry to allow adaptation to the mean light level over a wide range [2] enlarges the circuit to six transistors and requires some explicit capacitors. If focal plane motion processing is desired, between 30 and 50 transistors and significant capacitance are required [3–5] to perform the computation. Further processing, such as optical flow field smoothing [6] or discontinuity detection, adds significantly to the transistor

*Current address: Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA.

count. In building from a CMOS imager to a focal plane motion processor, the imager fill factor (the percentage of pixel area dedicated to light collection) drops from above 80% to less than 5%, while the pixel area grows by a factor of more than 10.

It is a long-term goal of our laboratory to emulate in VLSI the image processing strategies of biological visual systems, which incorporate many stages of simple parallel processors. As we proceed towards smart sensor designs incorporating more and more stages of pixel-parallel processing, we must either decrease the feature size of our fabrication process, resulting in higher costs and lower imager fill factors, or limit the processing that occurs in the focal plane. If an intermediate computation can be communicated off the photosensitive chip without losing the advantages of focal plane computation, the effective processing in the focal plane can be extended while retaining practical pixel resolutions. However, to retain the advantages of single-chip continuous-time focal plane image processors, this communication must be done without incurring significant delays, dramatically increasing power consumption, or introducing temporal aliasing.

The interchip communications protocol described in Section 3 provides a way of accomplishing this feat. Data is communicated asynchronously at low latency, allowing a representation of events in continuous time. Communication between chips only occurs when the input changes, thus making power consumption activity dependent and maximizing the effective bus bandwidth.

In this paper, we describe a novel CMOS VLSI chip pair designed on neuromorphic principles [7] which computes real-time optical flow using this communications protocol. The sender chip contains an array of photoreceptors and nonlinear differentiators which produce a voltage pulse upon a sudden change in local image intensity. These voltage pulses are communicated across a digital bus to a motion processing receiver chip, which computes the local velocity of motion by noting the order and timing in which edges arrive. The motion vectors are then serially scanned out of the receiver chip for display.

After characterizing the basic VLSI building blocks, we provide two examples of three-chip motion processors which can compute more complex visual motion data products.

An earlier version of this work has appeared in a conference proceedings [8].

2. Related Work

The interchip communications strategy used in this paper was originally envisioned by Mahowald [9] as a circuit analogy to the optic nerve. As such, it was first used to transmit visual signals out of a silicon retina. The protocol has since been strengthened and formalized by Boahen [10] for the same purpose. Several variants and specializations of the scheme have emerged in the last few years [11–14].

While applications of asynchronous interchip communications to neuromorphic sensory processing are still in the early stages, the results so far are quite promising. Boahen [15] has interfaced two silicon retinas to three receiver chips to implement binocular disparity-selective elements. Venier et al. [16] have used an asynchronous interface to a silicon retina to implement orientation-selective receptive fields. Deiss et al. [13] are implementing a silicon model of primate visual cortex using interchip communication, and DeWeerth et al. [14] are implementing a model of leech intersegmental coordination. Andreou et al. [17] have demonstrated the use of EPROMs for linear or nonlinear address remapping in interchip communication. Kumar et al. [18] have provided an auditory front-end chip with an asynchronous interface for further off-chip processing.

Recently, Boahen [19] has published a multi-chip vision processor that computes motion by emulating a model of primate motion computation. The photosensitive sender chip has four output channels per pixel, modeling on and off, sustained and transient responses to light stimulation. By using a serial processor to combine the outputs of channels from neighboring pixels in a receiver chip, motion-sensitive outputs were synthesized.

Kalayjian et al. [20] have created a photosensitive sender chip with similar function to the one presented in this paper: an array of photoreceptors and temporal derivative circuits are used to communicate local temporal illumination changes across a digital bus. This chip differs from the present work in two ways. Firstly, it differs in the use of a temporal derivative circuit rather than the highly nonlinear temporal edge detector used here. This chip is not intended to be an edge detector, but rather a silicon retina sensitive to temporal changes, and thus would not work well as a front end for token-based motion detection. Secondly, the communications scheme used in [20] is based on a partly analog winner-take-all arbitra-

tion, rather than the digital binary tree arbitration used in this paper (see Section 3 below). This arbitration scheme sacrifices the speed of the fully digital arbiter used in this paper to provide smaller pixel and peripheral circuitry requirements.

Indiveri et al. [21] have recently published a multi-chip motion processor which follows the present work. This system employs three stages of processing: a photosensitive sender with nonlinear temporal differentiation, a programmable interconnect processor (the Silicon Cortex [13]) that allows for arbitrary address remapping, and a motion processing receiver. The sender utilizes the same photoreceptor and nonlinear differentiator used here, but splits the edge information from rising and falling edges into two output channels, allowing for independent sensitivity adjustment. The motion receiver chip is based on the same velocity algorithm used here, but uses more sophisticated circuitry to obtain a bidirectional current output for the velocity of motion, rather than the multiple voltage outputs utilized in this paper.

3. Interchip Communications Protocol

The interchip communications protocol used in this work is known as the Address-Event Representation (AER). The original and most basic form of AER utilizes two digital control lines and several digital address lines to interface a sender chip to a receiver chip, as shown in Fig. 1. The protocol is used to communicate the occurrence of a binary event from sender to receiver in continuous time. A four-phase asynchronous handshake between sender and receiver guarantees reliable communication between chips; the address lines communicate the spatial position of a requesting sender pixel to the receiver chip, which forwards the event to the receiver pixel with the same spatial position.

This protocol effectively allows any sender pixel to communicate digital spikes to the corresponding receiver pixel. Because requests can come at any time from any pixel in the array, it is necessary to use an arbitration scheme on the sender to serialize simultaneous events onto the single communications bus. Because the asynchronous protocol operates so quickly (on nanosecond scales) relative to the time-scale of visual stimuli (on millisecond scales), the serialization caused by sharing of a single digital bus

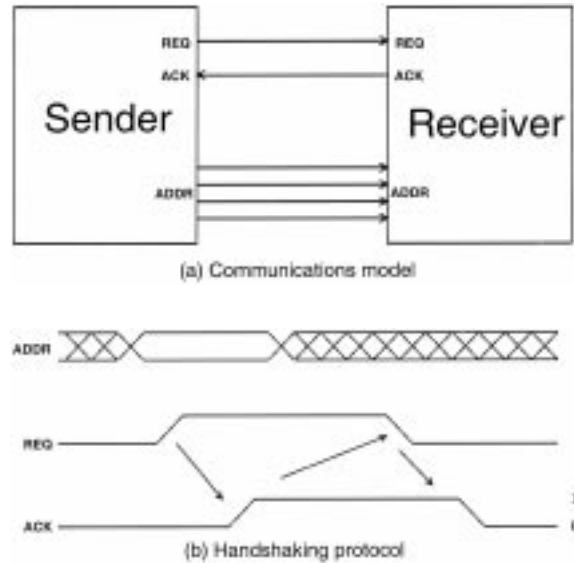


Fig. 1. AER protocol summary. In (a), the model for AER transmission is shown: a sender chip communicates with a receiver chip via request, acknowledge and address lines. In (b), the handshaking protocol for transmission using the above control and address lines is shown: a request with a valid address leads to an acknowledgment, which in turn leads to falling request and falling acknowledge.

is usually benign for sensor applications. Various schemes exist for deciding which of several simultaneously requesting sender pixels is allowed to use the bus first [9,12,22]. The scheme used in this paper is a binary tree arbiter [23], which yields a quick decision and scales well to large array sizes.

The circuitry necessary to implement the protocol varies from scheme to scheme. The particular hardware implementation of AER used in this chipset has been newly devised by Boahen; refer to [23] for further details.

4. Edge-Detecting Sender Chip

This section describes the photosensitive sender chip: the visual front end for all further processing. This chip detects moving contrast edges in an image focused directly upon it. Edge locations are communicated off-chip via the AER bus. Qualitatively, the output of this chip looks like an image filtered in real-time with an edge-enhancing operator; however, the edges disappear when no motion is present.

4.1. Sender Architecture

The core of the sender chip is a 12×12 array of sender pixels. See Fig. 2 for a layout diagram. Each sender pixel contains an adaptive photoreceptor [2] and a nonlinear differentiator circuit [24] interfaced to the interchip communication circuitry. This combination of adaptive photoreceptor and nonlinear differentiator is sensitive only to sudden changes in light intensity, and is referred to as a *temporal edge detector*, with the assumption that a sudden change in local intensity is due to a passing spatial edge. When an illumination edge passes over the pixel, the event is communicated to the receiver. In this implementation, events are communicated on the bus only when the illumination changes, resulting in an efficient use of bus bandwidth. Arbitration, address encoding, and other interface circuitry to support the protocol are located in the periphery and described in [23]. The chip also incorporates a serial scanner for readout of the raw photoreceptor image.

The photoreceptor circuit (shown in Fig. 3(a) and analyzed in detail by Delbrück [2]) adapts to the local light intensity on slow time scales (a few seconds), allowing high sensitivity to transient changes over a

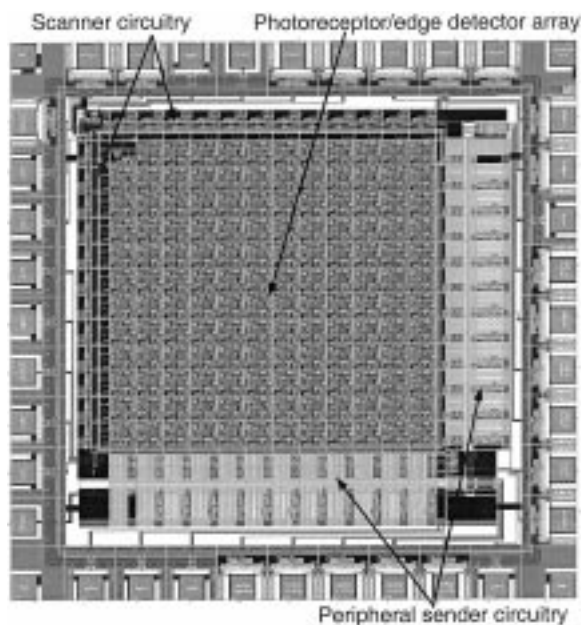
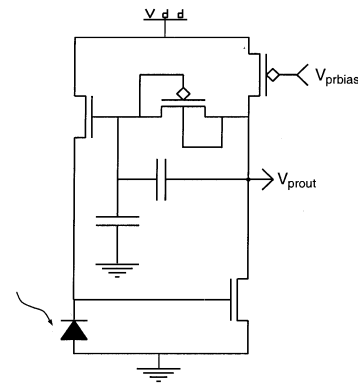
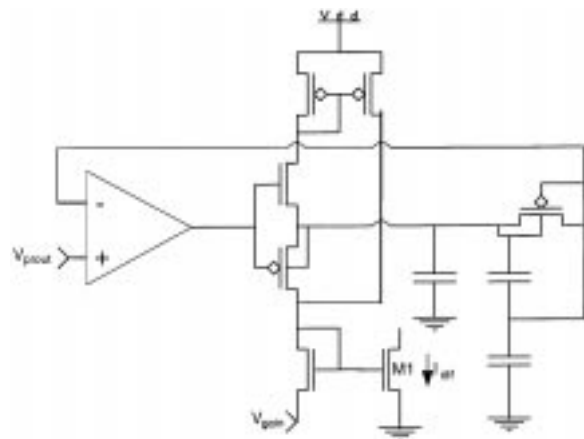


Fig. 2. Layout of the sender chip, as fabricated on a MOSIS tiny chip in a $1.2 \mu\text{m}$ standard CMOS process. Total area of the chip is $2.1 \times 2.1 \text{ mm}^2$.



(a) Adaptive Photoreceptor



(b) Nonlinear differentiator

Fig. 3. Components of the temporal edge detector: (a) adaptive photoreceptor which allows sensitivity to transient contrast changes over a wide range of illuminations, and (b) nonlinear differentiator circuit which responds only to sharp changes in illumination.

wide range of illumination without a change in bias settings. The nonlinear differentiator circuit (shown in Fig. 3(b) and analyzed in detail by Kramer et al. [24]) produces a current pulse whenever the photoreceptor output changes suddenly. This circuit is nonlinear in the sense that it produces a fairly narrow (1–10 ms) current pulse at the change of the derivative sign [25] both for sharp and smooth inputs, due to the nonlinear feedback. Note that the presence of such a current pulse does not convey motion, but rather temporal change information. The amplitude of the current pulse from this circuit can be shown to be proportional to *temporal contrast*, the product of stimulus speed and spatial contrast [5].

The sender pixel communications interface circuit, shown in Fig. 4, is slightly modified from [23]. Its input current I_{in} is taken from the nonlinear differentiator circuit output. This circuit generates a spike rate linearly proportional to the current input I_{in} . Before a request is made, R_{pix} is (inactive) low, A_{pix} is (inactive) high, and D_{pix} is (inactive) low. When sufficient current is integrated on node V_{mem} that it overcomes the threshold set by V_{thr} , V_{rp} is pulled low and the wired-OR R_{pix} shared by all pixels in the row is pulled high. When A_{pix} returns low from the row arbiter, it simultaneously resets V_{mem} to V_{dd} (and thus releases R_{pix}) and pulls up the wired-OR D_{pix} shared by all pixels in the column. D_{pix} will be held high until A_{pix} returns to inactive high. This circuit implements the required sender pixel protocol. The pass transistor M1 connected to the input node (a modification from the Boahen circuit) cuts off the input current during the reset phase, allowing stable reset even in the presence of large input currents. The transistor M2 interposed in the reset pathway is a second modification from the Boahen circuit and allows control of the speed of reset, effectively setting the maximum spike rate. Finally, a leak transistor M3 allows a minimum input current to be set.

The input current from the differentiator circuit will only exceed the threshold set by V_{leak} for a few milliseconds after a sudden illumination change. If the pixel request is not serviced within this time, the request will be withdrawn. For this reason, a slow-down in bus activity will not cause a large buildup of unserved events.

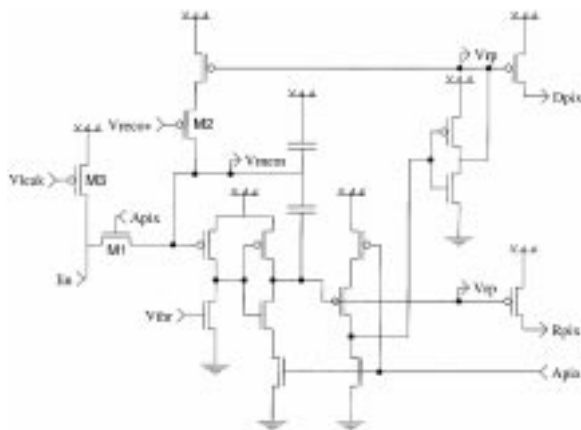


Fig. 4. Sender pixel communications interface circuitry.

4.2. Sender Performance

In this section, we experimentally characterize the sender array's AER bus response to changes in light intensity. During the period of time when the nonlinear differentiator's current output is large enough to overcome the leakage current, multiple requests from a single pixel (hereafter referred to as a *burst of spikes*) are created on the AER bus. A typical burst from a single pixel is shown in Fig. 5. Bus availability for each spike in the burst is arbitrated *independently*, so the burst from a particular pixel will, in general, appear on the interchip bus interleaved with requests from other pixels. Three parameters of these bursts—burst width, spike frequency, and latency from stimulation—have been measured as the chip is visually stimulated with the sender chip's interchip request line tied back to its own acknowledge line. This self-acknowledge yields the fastest possible event cycle, taking approximately 100 ns per request-acknowledge cycle.

Because the spike generating circuitry imposes a threshold on the output of the analog temporal edge detector circuit, each sender pixel will only fire spikes in response to a stimulus above a fixed temporal contrast threshold. Due to inevitable random noise in the analog part of this system, a

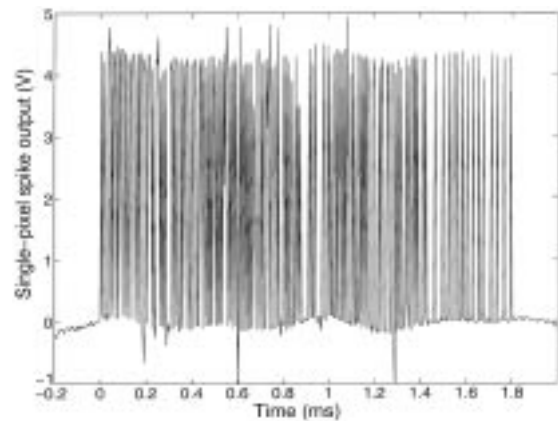


Fig. 5. Sender pixel burst response: this spike train is the response of an individual pixel to a passing edge. Because true spike width is approximately 50 ns and spike separation is on the order of 6 μ s, the spike duration has been lengthened to make individual spikes visible. This 1.8 ms burst peaks at a spike rate of approximately 160 kHz and encompasses around 200 individual spikes. The stimulus edge occurred at approximately $t = -15$ ms.

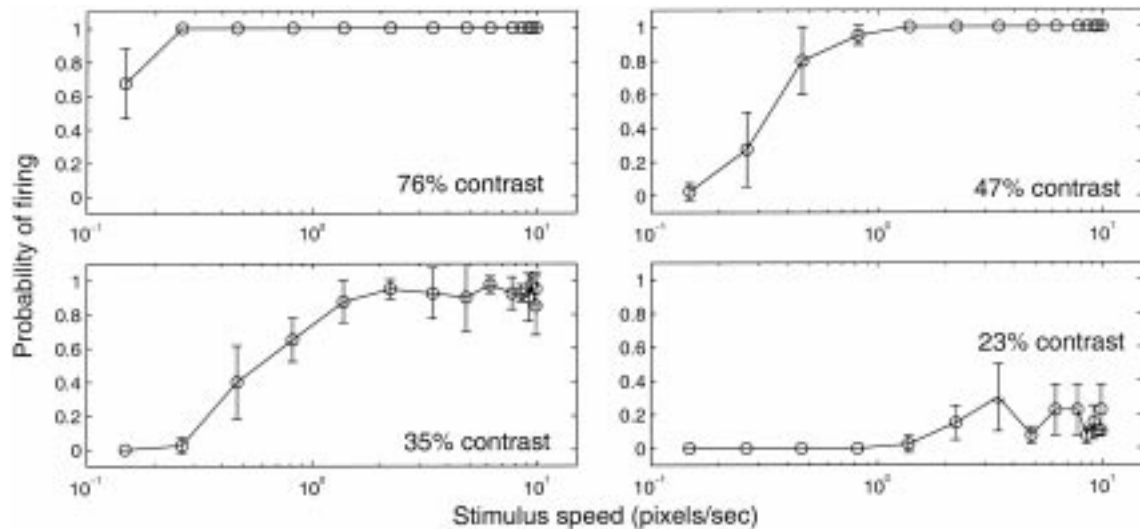


Fig. 6. Sender pixel speed/contrast response: probability of an individual pixel creating a burst (of at least one spike) is shown as the contrast and speed of the stimulus is varied. Probabilities have been calculated over five different pixels with ten stimulus presentations each. Error bars represent standard deviation of pixel probability between different pixels, and thus reflect the spatial variation. For this experiment, the nonlinear differentiator was adjusted to respond only to falling intensity changes. Pixel was stimulated with a computer-controlled intensity patch centered on the desired photoreceptor which slowly rose to the desired contrast and then fell with a controlled speed to zero contrast. Effective stimulus speed can be calculated from the geometry of the implementation. Contrast of a stimulus is calculated from luminance as $(L_{\max} - L_{\min}) / (L_{\max} + L_{\min})$.

sender pixel will fire *probabilistically* when visually stimulated by edges with temporal contrast near the threshold [5].

Fig. 6 shows the range of operation of the sender chip. As stimulus contrast and speed are varied, the probability of a burst response is shown. These plots reflect the expected dependence on temporal contrast: higher speed response thresholds are seen for lower contrasts. For high contrast, bursts are reliably produced for nearly two orders of magnitude in speed. No response is seen at very low contrast due to the threshold set by the spike-generating circuitry.

Fig. 7 shows how the burst width and spike frequency of the bursts vary with speed and between different pixels. Because the amplitude of the differentiator current pulse increases with speed and the spike generating circuit sets a fixed threshold, we expect that temporal width of the burst will increase with stimulus speed. Experimentally, burst width increases with stimulus speed until approximately 2 pixels/s, and then falls off as the speed becomes too fast for the tuning of the nonlinear differentiator. (For this paper, the differentiator was tuned to be sensitive to very low speeds to facilitate computer-controlled

stimulation. See Higgins et al. [5] for a discussion of the possible range of temporal edge detector speed tuning.) Spike frequency is linearly proportional to the differentiator current, so we also expect spike frequency to increase with stimulus speed. Experimentally, spike frequency increases with stimulus speed over the entire range, saturating at high velocities. This saturation is not due to lack of AER bus bandwidth, but rather to limitations in the nonlinear differentiator response. Mean latency (not plotted) was measured with the computer stimulus to be approximately 15 ms over a wide stimulus range, and varies with a standard deviation of less than 2 ms between sender pixels. This latency (between visual stimulation and spike response) is almost entirely due to delay in the photoreceptor and nonlinear differentiator circuit, not in the AER circuitry.

With no stimulus present, the sender chip consumes a static power of 3.2 mW at 5 V, largely due to leakage currents in minimum-size CMOS digital structures. A transient increase in power consumption is seen upon the creation of an event on the AER bus: power consumption peaks at 9.2 mW during a request. However, while creating

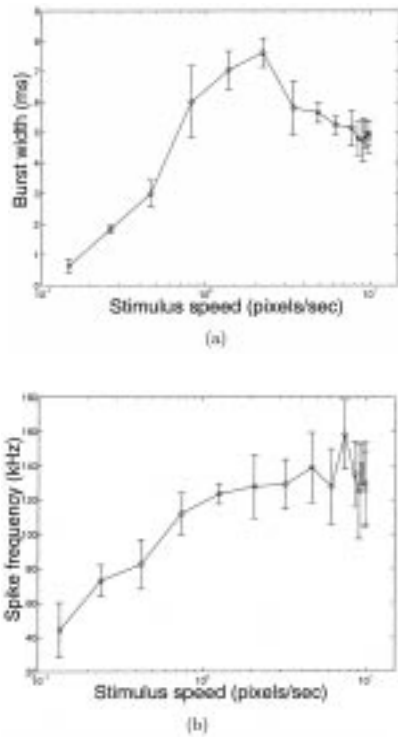


Fig. 7. Sender pixel burst parameters: for high contrast, (a) burst width and (b) spike frequency are shown as stimulus speed is varied. Stimulus is the same as in Fig. 6. Means have been calculated over five different pixels with ten stimulus presentations each. Error bars represent standard deviation of the pixel mean between different pixels, and thus reflect the spatial variation.

events at its maximum rate, the average power consumption of the sender chip is still only 3.8 mW due to the short duration of each request.

5. Motion Receiver Chip

This section describes the motion receiver chip, which calculates two-dimensional (2D) optical flow vectors from the edge information on the AER bus. A sample flow field from the sender-receiver pair is shown in Fig. 8.

5.1. Receiver Architecture

The core of the receiver chip is a 13 × 15 array of receiver pixels. See Fig. 9 for a layout diagram. Each

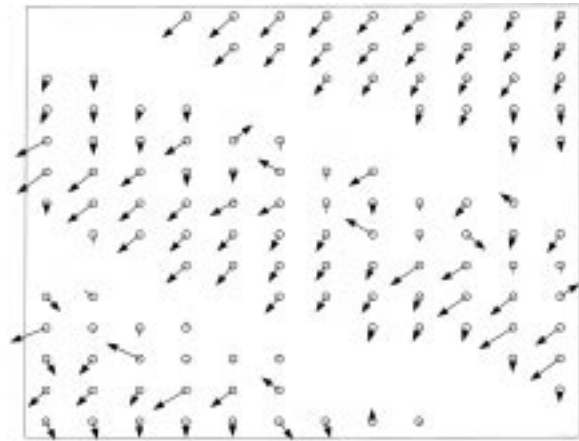


Fig. 8. Sample raw flow field from the sender–receiver pair: visual stimulus was a hand moving towards the lower left. Three fingers may be seen. The length of the band of vectors following the leading edge of each finger is dependent upon the vector persistence time setting of the receiver chip. Vectors in each band which are farther up and right have begun to decay. Some vectors in the lower two bands have also been lightly stimulated by the finger trailing edge.

receiver pixel contains the communications interface and a motion circuit implementing a 2D version of the FS (Facilitate-and-Sample) velocity algorithm

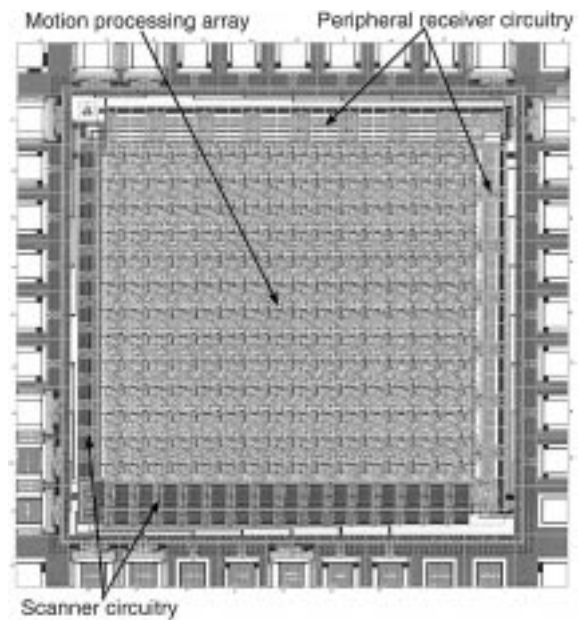


Fig. 9. Layout of the receiver chip, as fabricated on a MOSIS tiny chip in a 1.2 μm standard CMOS process. Total area of the chip is 2.1 × 2.1 mm².

[24]. The motion circuitry in each receiver pixel takes as input a current pulse from the interface circuit. Address decoding and interface circuitry to support the protocol are located in the periphery and described in [23]. This chip also incorporates a serial scanner for readout of the 2D optical flow vectors.

The receiver pixel communications interface circuit, shown in Fig. 10, is far simpler than its sender counterpart, and is changed from Boahen [23] only by the addition of a current-limiting transistor (M1). When X_{set} and Y_{set} are both active high, the source of the limiting transistor is pulled low and a current whose magnitude is set by V_{thr} flows into the motion circuit. The indirectness of this circuit is to avoid charge-pumping, which leads to a small “leakage” current even if X_{set} and Y_{set} are only asserted at non-overlapping times.

The FS (Facilitate-and-Sample) velocity algorithm (shown in Fig. 11 and analyzed in detail in [24]) computes the log velocity of a moving edge by measurement of the time between an edge’s arrival at neighboring pixels. The pulse-generation circuit (Fig. 11(a)) takes the current pulse from the receiver pixel interface and creates two voltage pulses. The fast pulse V_{fast} is a large-amplitude temporally narrow pulse which occurs when the receiver pixel is stimulated. The slow pulse V_{slow} quickly rises to a few volts and falls logarithmically over time as the capacitor is discharged. While the slow pulse is high, no further fast pulses can be created. The sample-and-hold circuit (Fig. 11(b)) is used to sample neighbouring slow pulses when a local fast pulse is generated. The stored value is effectively the

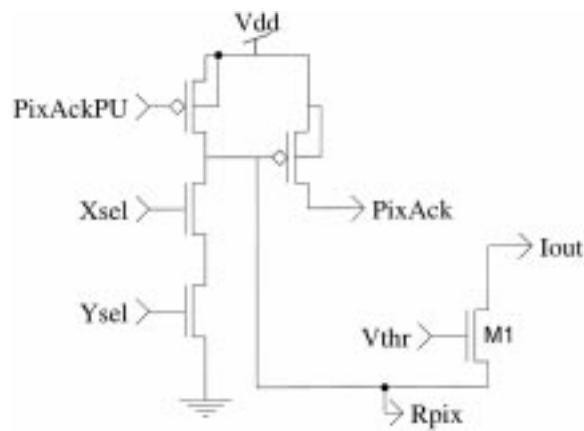


Fig. 10. Receiver pixel communications interface.

logarithm of the reciprocal of the time since that pixel was last stimulated. By using a single pulse-generation circuit and four sample-and-hold circuits in each pixel, it is possible to calculate the local 2D velocity of motion. To accomplish this, all four sampled slow pulses must be scanned from the receiver chip for each pixel. Ideally, the output of a one-dimensional (1D) FS sensor can be used to implement the following equation:

$$O_{FS,ideal} = g \cdot \text{sign}(T) \cdot \log\left(1 + \frac{1}{|T| + \sigma}\right) \quad (1)$$

where T is the pixel transit time and σ is a small number to represent the saturation of the sensor at high speeds. T may be positive or negative, representing two directions of motion. The sensor’s output is proportional to the log of the inverse pixel transit time, with the appropriate sign.

5.2. Receiver Performance

In this section, we experimentally evaluate the output of the dual-chip motion processor as a whole by measuring receiver chip responses to visual stimuli. The 12×12 sender array maps trivially into the larger 13×15 receiver array; the other receiver pixels are unused in this configuration. The request–acknowledge cycle in this system takes approximately 400 ns; the corresponding maximum spike rate is 2.5 million spikes/second. In Fig. 12 the percentage of 2D optical vectors within 15° of the correct stimulus orientation is plotted against stimulus speed and contrast. (Spatial differences between nominally identical motion sensors can account for 5–10° of variation in this process.) The low-speed threshold agrees with that seen in the sender chip. Above approximately 3 pixels/s, the correct response percentage falls off due to increasing variability in the temporal edge detectors. Correct orientation of the stimulus is calculated over more than an order of magnitude in speed and down to less than 30% contrast.

With no stimulus present, the receiver chip consumes a static power of 2.1 mW at 5 V. During stimulation, the average power consumption is 5 mW.

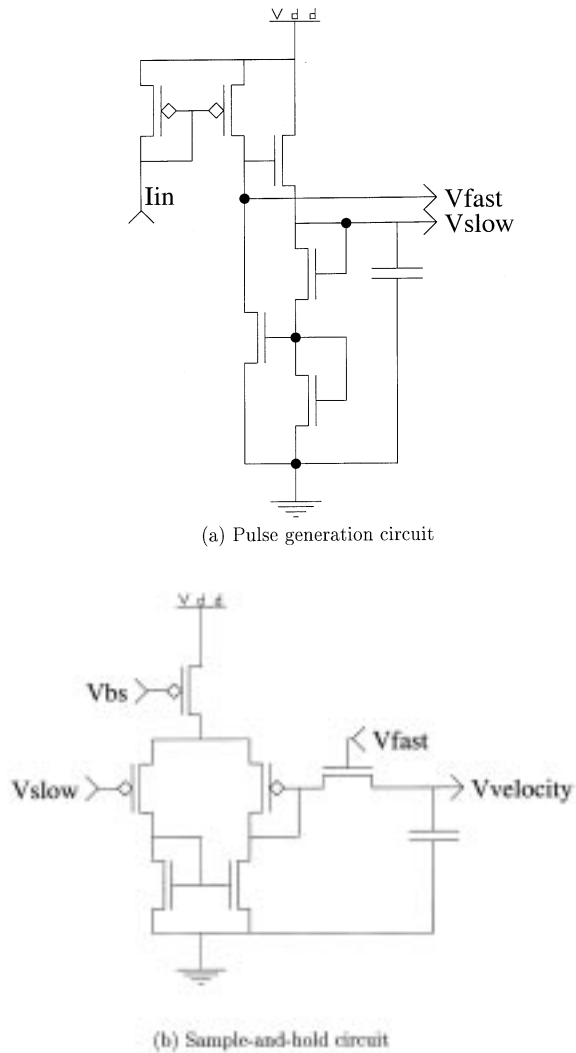


Fig. 11. Circuits implementing the FS (Facilitate-and-Sample, [24]) velocity algorithm. (a) The pulse generation circuit takes input from the receiver circuitry and generates a sharp fast pulse V_{fast} and a slowly decaying slow pulse V_{slow} . (b) The sample-and-hold circuit uses the fast pulse to sample a neighboring slow pulse.

6. Dual-Sender Motion Processor

In this section, we describe a motion processing system which uses two sender chips and a single motion receiver to compute motion tuned to a particular stereoscopic depth from the imager pair.

In application, the two sender chips would be arranged so that their fields of view converged at a certain stereoscopic depth (see Fig. 13). All disparities would be measured relative to this depth of

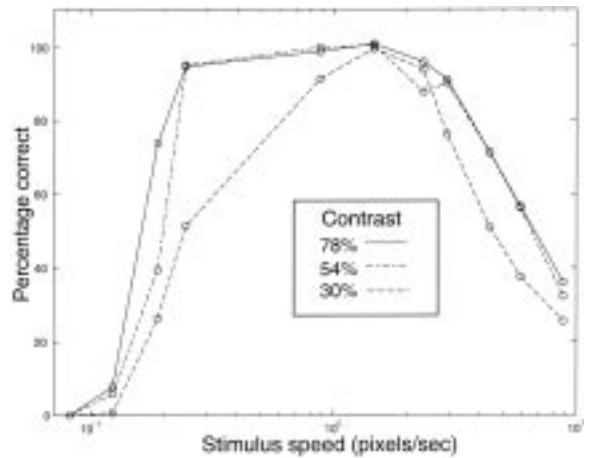


Fig. 12. Receiver chip temporal contrast response: the dual-chip motion processor was stimulated with a variable-speed rotating drum stimulus. The percentage is calculated across the entire array as the number of vectors within 15° of the correct stimulus angle.

convergence (known as the horopter), at which the images seen by the two chips are identical.

The representation of motion tuned to a particular optical disparity is similar to that seen in area MT of primate visual cortex [26], and is a biological alternative to explicit maps of motion and depth.

For the experiments in this section, the gain of the FS sensor velocity output has been increased to the point where only the local direction of motion is represented. This increased gain is necessary for the disparity tuning algorithm presented below to function properly.

6.1. Dual-Sender Architecture

See Fig. 14 for a block diagram of the hardware system. In order to converge the asynchronous requests from two sender chips, a fundamental requirement for this system is a two-input arbiter [9] to decide which request will be passed through to the single receiver. Given the choice bit from this arbiter, the appropriate address is multiplexed on-to the receiver address. An EPROM is included for address remapping; the choice bit is also input to the EPROM to allow different mappings for the two sender chips.

Static remapping of addresses with the EPROM is used to implement the disparity processor as shown

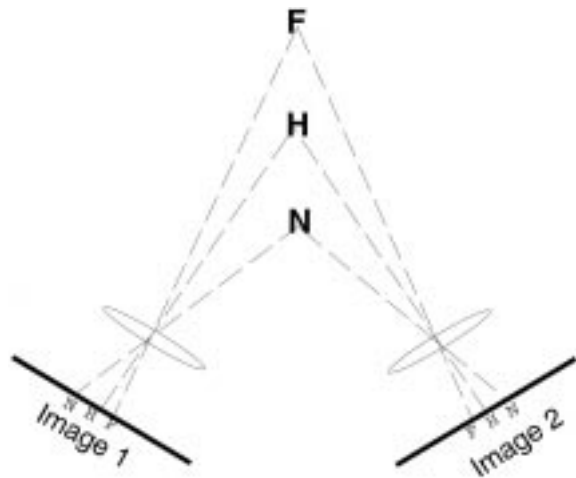


Fig. 13. Optical disparity related to depth: a point at depth H (the horopter) is shown at identical positions in images 1 and 2, because the cameras' lines of sight intersect at that depth. The difference between the position of the point in the two images is defined as the optical disparity d , so point H has zero disparity. At a larger depth F (far), the images of the point seen by the two cameras are offset by a positive disparity. At a smaller depth N (near), the images of the point seen by the two cameras are offset by a negative disparity.

in Fig. 15. The columns of the two senders are interlaced onto the receiver chip. Because the motion receiver chip expects to see columns fire in sequence as an edge passes from left to right, interlacing the columns from the two sender chips introduces a preference for motion at a particular disparity.

The algorithm for synthesizing a disparity-selective

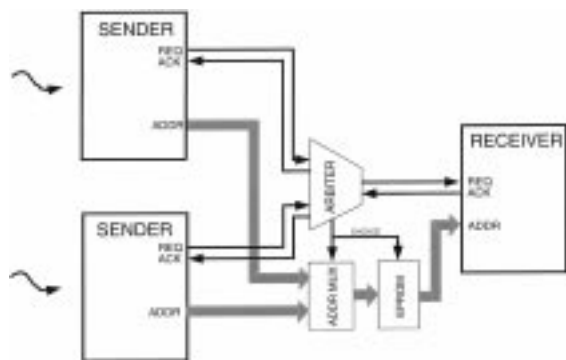


Fig. 14. Dual-sender hardware architecture: the arbiter is a standard two-input asynchronous arbiter [9] built out of discrete logic; not shown is an analog delay on the request line at the output of the arbiter to allow address setup time. Note that, aside from the custom VLSI components described, only discrete logic is used.

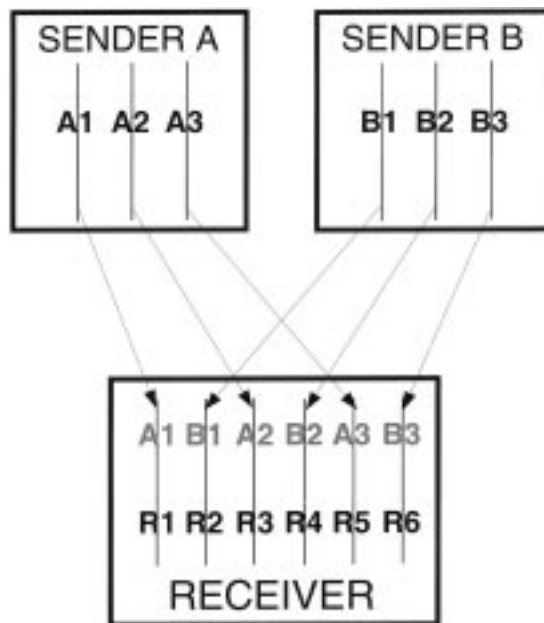


Fig. 15. Dual-sender address mapping: columns from the two senders are interlaced on the receiver chip. Because the motion chip expects the columns to be stimulated in sequence by a passing edge, this leads to a preference for a particular optical disparity.

motion system requires summing the outputs of sensors from two consecutive columns of the receiver chip. Consider the output of the receiver chip motion sensors in the left two columns $R1$ and $R2$ shown in Fig. 15. Let us model a visual stimulus with a variable depth from the imagers. For simplicity, assume that the visual stimulus is aligned with the sender chip columns, such that all rows in a column are stimulated at the same time. Let columns $A1, A2, A3, \dots$ be stimulated at times $0, \tau, 2\tau, \dots$ where τ indicates the pixel transit time of a moving edge (τ may be negative, allowing the stimulus to move in either direction). To model a stimulus with variable disparity, let columns $B1, B2, B3, \dots$ be stimulated at times $\tau_d, \tau_d + \tau, \tau_d + 2\tau, \dots$ where τ_d is proportional to the optical disparity between the two chips.

The output of the FS sensor used in this design may be approximated as

$$O_{FS}(T) = \tanh\left(g \cdot \text{sign}(T) \cdot \log\left(1 + \frac{1}{|T| + \sigma}\right)\right) \tag{2}$$

(compare to equation (1)) where again, T is the pixel transit time and σ is a small number to represent the saturation of the sensor at high speeds. This equation is a reasonable fit to the measured sensor output. The hyperbolic tangent with gain g is added to represent the increased gain of the FS sensor output, making it represent direction of motion (sign of velocity) rather than velocity.

The horizontal (X) component of the output of any motion sensor in column R1 is proportional to the time between the stimulation of column A1 and column B1. From the stimulus definition above, that time is τ_d and thus the output is $O_{FS}(\tau_d)$, the sign of which is positive whenever τ_d is positive. The horizontal component of the output of any sensor in column R2 is proportional to the time between the occurrence of column B1 and column A2, which is $\tau - \tau_d$. Its output is thus $O_{FS}(\tau - \tau_d)$, the sign of which is positive whenever τ_d is less than τ . The sum of these two quantities will be small except in the region $0 < \tau_d < \tau$, thus implementing a simple disparity tuning. Referring to equation (2), this output is

$$O_{disp} = O_{FS}(\tau_d) + O_{FS}(\tau - \tau_d) \quad (3)$$

A plot of this function is shown in Fig. 16(a). The disparity tuning is always between 0 and τ , even if τ is negative (stimulus moving in the opposite direction). Adding in the outputs of more velocity sensors does not change the tuning, but makes it more robust to noise. By shifting the interlace pattern, it is possible to create disparity tunings at any multiple of τ . For example, if B2 was mapped between A1 and A2, the disparity tuning would be in the region $\tau < \tau_d < 2\tau$.

This analysis can be extended to show that the sum of the horizontal (X) components of sensors from any two neighboring columns will be selective to the same disparity. In a more general stimulus framework, the elementary disparity-tuned element consists of the sum of two motion sensor outputs from the same row and neighboring columns.

6.2. Dual-Sender Performance

To verify the disparity tuning of the dual-sender system, a computer stimulus (diagrammed in Fig. 17) was used to simultaneously present two moving vertical bars, only one of which was visible to each

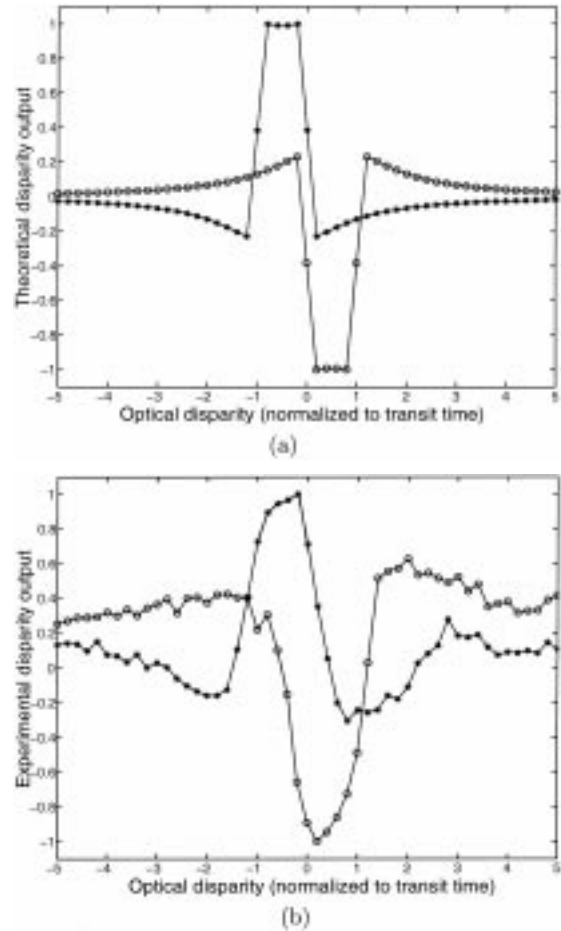


Fig. 16. Dual-sender performance: (a) Theoretical performance of the interlaced rows algorithm: a plot is shown of equation 3 with $g = 1.1$ and $\sigma = 0.1$ while τ_d is varied. The circles are for $\tau = 1.0$; the asterisks are for $\tau = -1.0$. The horizontal axis is τ_d/τ , the optical disparity normalized to the pixel transit time. The vertical axis has been normalized to unity. (b) Experimental disparity tuning of the interlaced rows algorithm. As the binocular disparity of the stimulus shown in Fig. 17 is varied, the averaged X output of the receiver chip is plotted. This output is the spatial average of the X (horizontal) component of every optical flow vector in the receiver array. It is also temporally averaged over one period of the stimulus to remove the effects of periodic variation. Circles indicate the response to a leftward-moving bar; asterisks indicate the response to a rightward-moving bar. The horizontal axis is the optical disparity normalized to the pixel transit time. The vertical axis has been normalized to unity.

sender chip. The disparity between the two stimuli was varied precisely under computer control. Fig. 16(b) shows the result of this experiment. The average X output of the entire array is plotted against stimulus disparity. The chip shows a clear preference

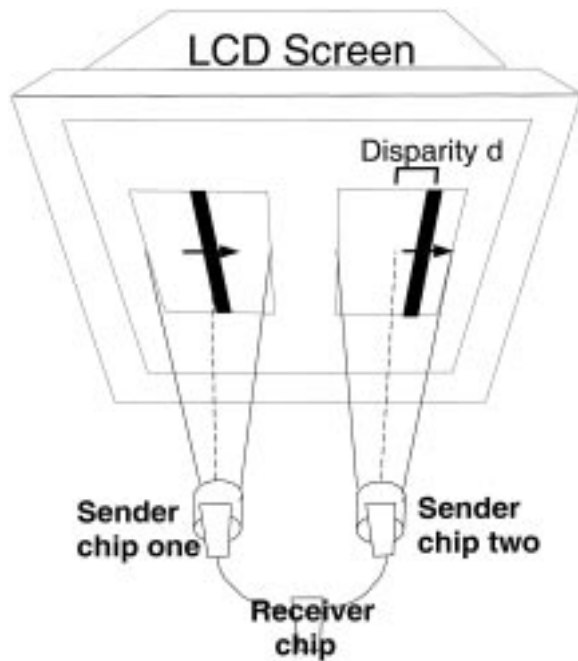


Fig. 17. Dual-sender stimulus diagram: separate moving bar stimuli were presented simultaneously to each sender chip on the same LCD Screen. The disparity between the two bars (that is, the difference in the horizontal position of the bar in the two images) was varied under computer control, and remained constant as both bars moved. This stimulus simulates a horizontally moving vertical bar with a variable depth.

for a particular disparity in each stimulus direction. Because the width of the disparity tuning is proportional to the stimulus interpixel transit time, the region of preferred disparity widens as the stimulus slows down. The tuning is smoother than the theoretical prediction because of the whole-chip average: due to transistor mismatch, each “disparity sensor” (made of the horizontal sum of two motion sensors) is tuned to a slightly different range. The request-acknowledge cycle in this system takes approximately 400 ns.

It is possible to add more receiver chips with shifted interlace patterns such that each is tuned to a unique optical disparity, resulting in a parallel map of motion and depth. Alternatively, a single receiver chip could be used, and additional address lines of the EPROM could be used to dynamically change the depth mapping.

The maximum total power consumed by the VLSI components in this design is 12.6 mW at 5 V. This figure neglects the power consumption of the CMOS

discrete logic circuitry and EPROM¹ used, which is relatively small.

7. Dual-Receiver Motion Processor

In this section, we describe a motion processing system which uses a single sender chip and two identical motion receivers with different topological mappings of the image plane.

The use of a polar (or log-polar) mapping is motivated by the biological observation [27] of the mapping from the retina to primary visual cortex, and has inspired past vision sensor designs [28].

As in Section 6, the gain of the FS sensor velocity output has been increased for these experiments to the point where only the local direction of motion is represented.

7.1. Dual-Receiver Architecture

See Fig. 18 for a block diagram of the hardware system. Because two receiver chips are present, circuitry is necessary to ensure that *both* receiver chips have acknowledged the single sender event before the system continues. This circuit is known as a C-element [29]. Two EPROMs are included for parallel static remapping of both receiver destination addresses.

The first receiver uses a pass-through sender

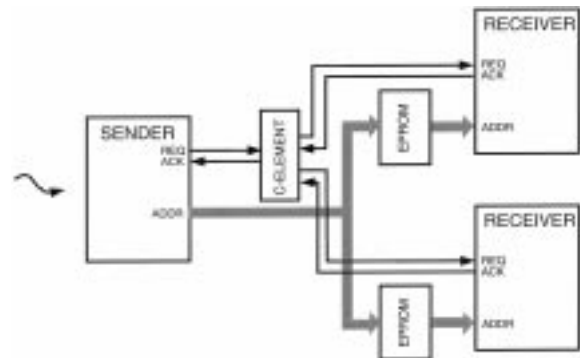


Fig. 18. Dual-receiver architecture: the C-element is a standard asynchronous communications building block [29] built out of discrete logic; not shown are timeout circuits to handle nonexistent receiver addresses, an analog delay on both receiver request lines to allow for address setup time, and EPROM enabling circuitry. Note that, aside from the custom VLSI components described, only discrete logic is used.

address mapping, which generates the same sort of optical flow field characterized in Section 5. The second receiver uses a polar coordinate mapping: let the polar coordinates of a sender pixel be described by

$$R = \sqrt{(X_{sdr} - X_{mid})^2 + (Y_{sdr} - Y_{mid})^2}$$

$$\theta = \tan^{-1}((Y_{sdr} - Y_{mid}) / (X_{sdr} - X_{mid}))$$

where (X_{mid}, Y_{mid}) is the center pixel address of the sender array. Then the receiver mapping can be described as the nearest integer to

$$X_{rcvr} = S_x \cdot R$$

$$Y_{rcvr} = S_y \cdot \theta$$

where S_x and S_y are chosen to maximally cover the receiver array. This remapping makes the second receiver sensitive to expanding and rotating motions. A pure expansion corresponds to movement only along the radial coordinate (remapped X). A pure rotation corresponds to movement only along the angular coordinate (remapped Y). Such motion must be centered on the sender chip for a maximal response.

7.2. Dual-Receiver Performance

To demonstrate the particular sensitivity of each receiver, we first present a moving bar stimulus and observe the array average X coordinate from each receiver chip. For any given stimulus presentation, the angle and speed of the moving bar are fixed. Fig. 19(b) shows the responses as the angle of the moving bar is varied across multiple stimulus presentations. The linearly mapped array shows a strong directionally-selective response, whereas the polar-mapped array shows little selectivity. In Fig. 19(d) the same outputs are shown in response to a stimulus composed of expanding circles as the position of the focus of expansion is swept horizontally across the sender chip. The output of the linearly-mapped array reflects the position of the focus of expansion, as explained in [30]. The output of the polar-mapped array is strongly negative, indicating the presence of expansion, and peaks in strength when the focus of expansion is at the center of the sender chip. The request-acknowledge cycle in this system takes approximately 500 ns.

The maximum total power consumed by the VLSI components in this design is 13.8 mW at 5 V. This figure neglects the power consumption of the CMOS discrete logic circuitry and EPROMs used, which is relatively small.

8. Discussion

We have described a flexible, modular, multi-chip neuromorphic motion processing system which retains many of the advantages of single-chip motion processors while allowing for significant further expansion. While many component subcircuits have been extended for the purpose of building this system, this paper's main contribution is intended to be at the system architecture level. In addition to characterizing the elementary motion processor, we have shown two three-chip systems which compute more complex real-time motion data products completely without the use of serial computers.

This system has been demonstrated using MOSIS tiny chips with 2.1×2.1 mm die sizes at $1.2 \mu\text{m}$ process resolution, which has resulted in very small array sizes. Due to the fully parallel architecture, there are no architectural limitations to the expansion of this system to reasonable sizes. Straightforward replication of pixels in the same process would yield a sender array of approximately 50×50 pixels on an 8×8 mm die; in a $0.35 \mu\text{m}$ process, an 8×8 mm die would yield at least 128×128 pixels. Higher resolutions of motion detection than 128×128 yield rapidly diminishing gains in machine vision applications.

The dual-receiver architecture we have demonstrated can be programmed with arbitrary topological mappings of the image plane, which can be used to perform a number of image processing tasks. In addition, the visual motion caused by changes in angle of the imaging platform can be compensated for by providing information about camera angle to the EPROMs. This can be used to compensate for unintentional camera jitter, as well as programmed movements of the camera.

A different approach to computing disparity-tuned motion with the dual-sender architecture than we have shown here would be to map corresponding pixels from each sender to the same receiver pixel and require a coincidence of bursts to create a motion output. This coincidence based stereo approach

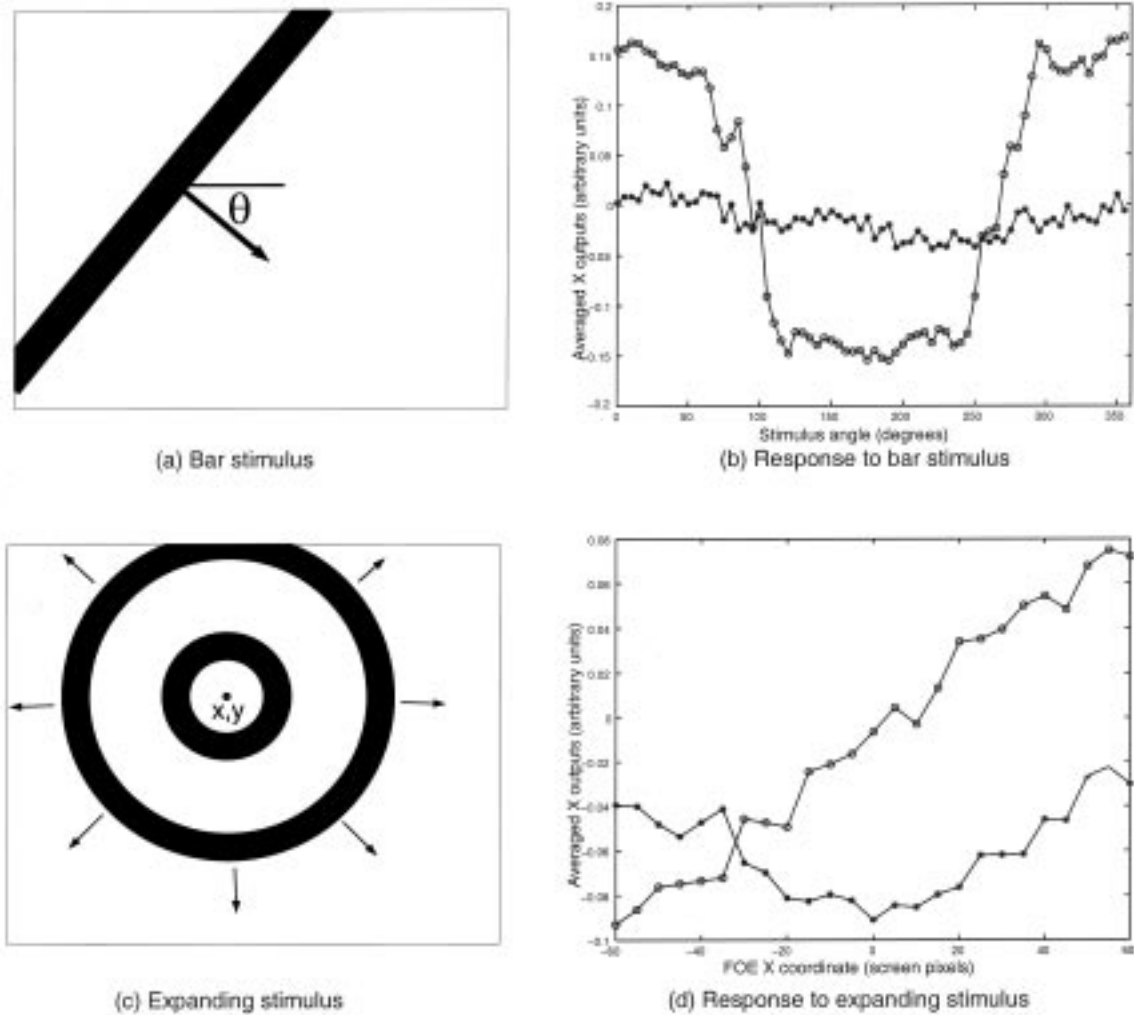


Fig. 19. Dual-receiver performance: In (a), a moving bar stimulus is diagrammed; (b) shows the output of both receiver chips as the angle θ of the bar stimulus is varied. The output shown is the pixel X velocity output averaged spatially over the entire chip, and temporally over one period of the stimulus. Circles indicate the response of the linearly-mapped receiver, asterisks indicate the polar-mapped receiver. In (c), an expanding stimulus is diagrammed. The coordinates x, y indicate the focus of expansion. The circles grow larger in the direction of the arrows; (d) shows the response of both receiver chips as the focus of expansion is swept horizontally across the chip.

would require a nonlinear threshold on the motion receiver chip to make a strong distinction between one burst and a coincident pair.

Aside from the serialization inherent in the AER bus, the motion computation demonstrated in this paper is fully pixel-parallel. Due to the small size of the current implementation, this serialization places no limitations on performance. However, it begs the question of scaling to larger array sizes. If the sender/receiver pair were scaled to a 128×128 array, would

the AER bus present a significant bottleneck? Consider the case of stimulation by a vertical moving edge spanning the entire array: 128 pixels would make simultaneous requests. Using the full bandwidth of the bus (2.5 MHz), a 4 ms burst from 128 pixels would result in 80 spikes from each pixel: more than enough to stimulate the receiver. An individual pixel might experience serialization delays as large as a few microseconds; with burst widths in the millisecond range this is insignificant. If the entire

sender chip were stimulated at once by a flash, spikes from some sender pixels would clearly be lost due to lack of bandwidth. However, in actual operation only a small fraction of sender array pixels are likely to be stimulated at one time.

Power consumption is also an important consideration in scaling designs of this type to large array sizes. The power consumption figures given in this paper for small multi-chip systems cannot be trivially scaled to larger array sizes. Rather, they contain factors which are constant, and which vary linearly, logarithmically, and as the square of the number of pixels in the array. Unfortunately, with the current VLSI components it is not possible to measure the power consumption of each factor individually. We can, however, parameterize them as follows. The power consumption for an $N \times N$ multi-chip system may be expressed as

$$P_{total} = P_0 + P_1 \cdot N + P_L \cdot \log(N) + P_2 \cdot N^2$$

The factor P_0 , comes from circuitry that is not replicated as the system gets larger, such as the off-chip interface logic. This component is relatively small. The factor P_1 comes from circuitry along the periphery of the arrays, such as serial scanners, address decoders, and sender support circuitry. The factor P_L scales with the logarithm of the array size and results from circuitry including the arbiter binary tree, padframe address drivers, and the EPROM. Finally, the factor P_2 scales with the square of the array size, and is associated with the power consumption in the pixel itself. Based on measurements of the pixel circuitry in this system from other chips, it is most likely that the factor $P_2 \cdot N^2$ in the total power for the small-chip system we have shown is overwhelmed by the other factors. However, it is obviously most important to control P_2 , since this factor will dominate for very large array sizes. The use of subthreshold MOSFET analog circuitry and asynchronous digital CMOS circuitry minimizes this power consumption, and large low-power designs such as [19] (at 104×96 pixels) help motivate that this strategy can be successful.

In addition to the obvious benefits of allowing more stages of processing and the combination of multiple imagers, the use of an efficient asynchronous communications link between chips can allow for connectivity which is not practical even in advanced fabrication processes. By simple manipulations of the

digital communications link between two VLSI chips, it is possible to modify the destinations of individual events to achieve virtual wiring [31]. As demonstrated in this paper, a memory chip can be used as a look-up table to perform a one-to-one or many-to-one rerouting of events. In addition, a very simple digital processor (or even an FPGA) can be used to transmit each event to many destinations, achieving arbitrary interconnectivity at the price of bus bandwidth.

Multi-chip systems such as these will make hardware implementations of complex multi-stage image processors like those suggested by biological vision systems a feasible prospect.

Acknowledgment

The authors gratefully acknowledge Kwabena Boahen for his copious assistance in explaining his implementation of the AER protocol, and would also like to thank Tim Horiuchi for helpful suggestions, and Rainer Deutschmann for comments on the manuscript. This research was supported by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation's Engineering Research Center program as well as by the Office of Naval Research. The authors wish to thank the anonymous reviewers for their help in clarifying this paper.

Note

1. The EPROM is held in standby (low-power) mode except when a request is made, and thus has a very low average power consumption.

References

1. E. R. Fossum, "CMOS image sensors: Electronic camera-on-a-chip." *IEEE Transaction Electron Devices* 44(10), pp. 1689–1698, 1997.
2. T. Delbrück and C. Mead, "Analog VLSI phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits." Tech. Rep. 30, Department of Computation and Neural Systems, California Institute of Technology, 1993.
3. R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "A focal plane visual motion measurement sensor." *IEEE Transactions on Circuits and Systems I* 44(1), pp. 55–56, 1997.

4. R. A. Deutschmann and C. Koch, "Compact real-time 2-D gradient based analog VLSI motion sensor," in *Proceedings of the International Conference on Advanced Focal Plane Arrays and Electronic Cameras*. Zurich/Switzerland, 1998.
5. C. M. Higgins, R. A. Deutschmann, and C. Koch, "Pulse-based 2D motion sensors." *IEEE Transaction on Circuit System II* 46(6), pp. 677–687, June 1999.
6. A. Stocker and R. Douglas, "Computation of smooth optical flow in a feedback connected analog network," in *Advances in Neural Information Processing Systems*. M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds., Cambridge, MA, 1999, Vol. 11, MIT Press.
7. C. A. Mead, "Neuromorphic electronic systems." *Proceedings of the IEEE* 78, pp. 1629–1636, 1990.
8. C. M. Higgins and C. Koch, "Multi-chip neuromorphic motion processing," in *Proceedings of the 20th Conference on Advanced Research in VLSI*, Atlanta, GA, 1999.
9. M. A. Mahowald, VLSI analogs of neuronal visual processing: a synthesis of form and function. Ph.D. thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA, 1992.
10. K. Boahen, "Retinomorph vision systems," in *Proceedings of the International Conference on Microelectronics for Neural Networks and Fuzzy Systems*. IEEE, 1996.
11. J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals." *IEEE Transaction Neural Network* 4(3), May 1993.
12. A. Mortara, E. Vittoz, and P. Venier, "A communications scheme for analog VLSI perceptive systems." *IEEE Journal Solid State Circuit* 30(6), June 1995.
13. S. R. Deiss, R. J. Douglas, and A. M. Whatley, "A pulse-coded communications infrastructure for neuromorphic systems," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds., Chapter 6, pp. 157–178. MIT Press, 1998.
14. S. DeWeerth, G. Patel, M. Simoni, D. Schimmel, and R. Calabrese, "A VLSI architecture for modeling intersegmental coordination," in *Proceedings of the 17th conference on Advanced Research in VLSI*, Ann Arbor, MI, 1997.
15. K. Boahen, *NSF Neuromorphic Engineering Work-Shop Report*, Telluride, CO, 1996.
16. P. Venier, A. Mortara, X. Arreguit, and E. Vittoz, "An integrated cortical layer for orientation enhancement," *IEEE Journal Solid State Circuits* 32(2), pp. 177–186, 1997.
17. S. Grossberg, G. Carpenter, E. Schwartz, E. Mingolla, D. Bullock, P. Gaudiano, A. Andreou, G. Cauwenberghs, and A. Hubbard, "Automated vision and sensing systems at Boston University," in *Proceedings of the DARPA Image Understanding Workshop*. New Orleans, LA, 1997.
18. N. Kumar, W. Himmelbauer, G. Cauwenberghs, and A. G. Andreou, "An analog VLSI chip with asynchronous interface for auditory feature extraction." *IEEE Transaction Circuit System II*, 45(5), pp. 600–606, 1998.
19. K. Boahen, "Retinomorph chips that see quadruple images," in *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems*, April 1999.
20. Z. Kalayjian and A. G. Andreou, "Asynchronous communication of 2D motion information using winner-takes-all arbitration." *Analog Integ. Circuit. Signal Processing* 13, pp. 103–109, 1997.
21. G. Indiveri, A. M. Whatley, and J. Kramer, "A reconfigurable neuromorphic vlsi multi-chip system applied to visual motion computation," in *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems*, April 1999.
22. Z. Kalayjian, S. Waskiewicz, D. Yochelson, and A. Andreou, "Asynchronous sampling of 2D arrays using winner-takes-all arbitration," in *IEEE International Symposium on Circuits and Systems*, Atlanta, GA, 1996.
23. K. Boahen, "A throughput-on-demand 2-D address-event transmitter for neuromorphic chips," in *Proceedings of the 20th Conference on Advanced Research in VLSI*, Atlanta, GA, 1999.
24. J. Kramer, H. Sarpeshkar, and C. Koch, "Pulse-based analog VLSI velocity sensors." *IEEE Transaction Circuit System II*, 44, pp. 86–101, 1997.
25. C. A. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, 1989.
26. J. H. R. Maunsell and D. C. Van Essen, "Functional properties of neurons in middle temporal visual area of the Macaque monkey. II. Binocular interactions and sensitivity to binocular disparity." *Journal Neurophysiology* 49, pp. 1148, 1983.
27. E. L. Schwartz, "Spatial mapping in the primate sensory projection. Analytic structure and relevance to perception." *Biological Cybernetics* 25, pp. 181–194, 1977.
28. J. Van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Bellutti, and G. Soncini, "A foveated retina-like sensor using CCD technology," in *Analog VLSI and Neural Network Implementations*. C. Mead and M. Ismail, Eds. Kluwer, 1989.
29. I. E. Sutherland, "Micropipelines." *Comm. ACM* 32(6), pp. 720–738, 1989.
30. C. M. Higgins and C. Koch, "An integrated vision sensor for the computation of optical flow singular points." in *Advances in Neural Information Processing Systems*. M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds., Cambridge, MA, 1999, Vol. 11, MIT Press.
31. J. G. Elias, "Artificial dendritic trees." *Neural Computation* 5, pp. 648–663, 1993.



Charles M. Higgins received the Ph.D. in Electrical Engineering from the California Institute of Technology in 1993. He worked in the Radar Systems Group at MIT Lincoln Laboratory until 1996, when he returned to Caltech as a postdoctoral

research fellow in the Division of Biology. In 1999, he joined the Department of Electrical and Computer Engineering at the University of Arizona as Assistant Professor. His research is in the area of analog/digital VLSI vision and robotic systems. His research interests include analog computation, asynchronous digital inter-chip communication, hardware emulations of biological neural systems, and autonomous systems.



Christof Koch, born in 1956 in Kansas City, USA, studied physics and philosophy at the University of Tuebingen in Germany and was awarded his Ph.D. in Biophysics in 1982. After four years at MIT, he moved to Caltech where he has been ever since as Professor of Computation and Neural Systems. He has written and edited Five books, including a textbook on biophysics, and over 200 technical articles on analog VLSI, vision and visual algorithms, computational neuroscience and the neuronal correlates of consciousness (together with Francis Crick).