A RETINOMORPHIC VISION SYSTEM

Kwabena A. Boahen

California Institute of Technology

This system uses neurobiological principles to accomplish the major operations of biological retinae: continuous sensing, local automatic gain control, spatiotemporal band-pass filtering, and adaptive sampling.

The retina is an exquisitely evolved piece of neuronal wetware. It contains about a hundred million black-and-white photoreceptors, complemented by three to four million color receptors. Its output—about one million axonal fibers that make up the optic nerve—conveys visual information to the rest of brain using an all-or-none pulse code. Compared to a state-of-the-art charge-coupled-device (CCD) camera, the retina accomplishes many amazing feats.

Parallel processing of visual information begins in the retina with the presence of several channels specialized for such tasks as nocturnal vision, color vision, spatial vision, and motion. Under ideal conditions, these channels allow us to detect reliably the absorption of 10 photons in a pool of 5,000 rods; to perceive color in light wavelengths ranging from 400 to 670 nm; to detect 0.5% contrast; to resolve two lines subtending an angle of 1/60 of a degree; and to tell the onset order of two lines flashed 3 to 5 milliseconds apart. In addition, we can see well both in dim starlight and in bright sunlight—a dynamic range of over 10 decades!

In contrast, though an 8-bit CCD camera's 0.4% full-scale amplitude resolution comes close to matching the retina's contrast sensitivity, the electronic camera's 1/5-degree angular resolution and its 30-ms temporal resolution are an order of magnitude worse. Its 50-dB dynamic range is six orders of magnitude short. We can therefore advance the state of the art in focal-plane image processing by studying the expanding body of knowledge gathered by neurobiologists about how the retina operates.[1]

In this article, I compare and contrast retinal design principles with standard imager design practice. I argue that neurobiological principles are best suited to perceptive systems that go beyond reproducing the dynamic scene as conventional video cameras do—to extract salient information in real time.

Retinomorphic vision systems use neurobiological principles to accomplish at the pixel level all four major operations of biological retinae: 1) continuous sensing for detection, 2) local automatic gain control for amplification, 3) spatiotemporal band-pass filtering for preprocessing, and 4) adaptive sampling for quantization. The retinomorphic system that I describe uses a random-access communication channel to read out asynchronous pulse trains from a 64×64-pixel array on the retinomorphic chip, and then transmits them to corresponding locations on a second chip that has a 64×64 array of integrators.

What are retinal design principles?

Table 1 outlines the retina's design principles, which retinomorphic systems borrow. For comparison, the table also lists design principles employed by standard imager technology. The following sections elaborate these principles.

Detection. Integrating detectors—for example, CCDs and photogates[2]—suffer from blooming at high intensity levels and so require a destructive readout, or reset, operation. The retinal approach uses continuous-sensing detectors, such as photodiodes and phototransistors. These do not bloom, and can therefore operate over a much larger dynamic range.[3] In addition, because we do not need to reset such detectors, we can eliminate redundant readout operations with considerable power savings. Continuous-sensing detectors have been shunned, though, because they suffer from gain and offset mismatches that produce salt-and-pepper noise in the image. However, preliminary results indicate that the learning

0272-1732/96/$5.00 © 1996 IEEE

Table 1. Retinal design principles.

Operation	Standard imager technology	Retinal method
Detection	Integrate/reset	Continuous
Amplification	Global AGC*	Local AGC
Preprocessing	Absent	Band-pass filter
Quantization	Fixed	Adaptive

* Automatic gain control

capability of image recognition systems can easily compensate for this fixed-pattern noise.[4]

Amplification. Imagers that use global automatic gain control (AGC) for amplification can operate only under uniform lighting, because the variation of intensity across a scene exceeds their 8-bit dynamic range when shadows are present. A CCD or photogate can achieve 12 bits (a four-decade range),[2] and a photodiode or phototransistor can achieve 20 bits (six decades)[3]—but the phototransistor's performance in the lowest two decades is plagued by slow temporal response. In addition to these capability limitations, a system's dynamic range is limited by the cost of precision analog readout electronics and A/D converters, and by video standards. These system-level constraints account for the much lower dynamic range achieved by conventional imagers.

Introducing AGC locally—at the pixel level—increases dynamic range and resolution in the darker parts of the image without increasing the number of bits per sample. Following retinal principles, we can set the gain to be inversely proportional to the local intensity, discounting gradual changes in intensity and producing an output that is proportional to contrast.[5] This adaptation greatly extends the dynamic range, because lighting intensity varies by six decades from high noon to twilight, whereas contrast varies by at most a factor of 20.

Preprocessing. The intensity pattern that falls on the imager is highly redundant in space and time; that is, differences between adjacent samples in space or time are rare. Band-pass spatiotemporal filtering is an optimal preprocessing strategy for removing redundancy in the presence of white noise.[6-7] This filtering, absent in standard imager technology, reduces the correlation among pixels by eliminating low spatial and temporal frequencies. In addition, it attenuates fluctuations in the photon flux and transistor currents by by rejecting high temporal and spatial frequencies. Coupled with adaptive quantization, this efficient image representation requires much less bandwidth for transmission than the raw image intensity values. It also enhances features at finer spatial and temporal scales, making recognition easier[4] and providing a spatial or temporal reference for motion computation.

Quantization. Converters that automatically adapt their quantization in time and amplitude to the rate of change and the amplitude probability distribution of the input signal maximize the information transmitted through the output channel. In contrast, traditional A/D converters use quantization set to match the maximum rate of change and the smallest

Glossary

Band-pass filter: A filter that passes an intermediate range of frequencies, and attenuates frequencies that are too high or too low. Such filters are usually tuned to a particular frequency, and their sensitivity drops off smoothly as you move away from that frequency.

Blooming: Bleeding of extremely bright areas of an image into neighboring dark areas. This occurs when the buckets holding photocharge overflow. This phenomenon is analogous to overexposure in photography.

CCD: Charge-coupled device. An electronic device, similar in structure to a MOSFET, which serves as a charge bucket that may be filled and emptied under electronic control. When used in imagers, this device collects the photocharge; the photocharge is then read out by shifting charge along a chain of devices, in bucket-brigade fashion. CCD technology is currently used in all video and digital cameras.

Dynamic range: The ratio between the highest and lowest signal amplitudes that a sensor can accomodate. It is usually quoted in decibels (dB), a logarithmic scale: each 20 dB corresponds to a tenfold increase in signal energy.

Excitatory and inhibitory synapses: Synapses mediate communication between neurons. They come in two varieties: excitatory and inhibitory. Excitatory synapses increase the voltage in the cell on the output (postsynaptic) side of the synapse when the voltage in the cell on the input (presynaptic) side increases, making the target cell more likely to fire. Inhibitory synapses have the opposite effect; they reduce the post synaptic potential and make firing less likely.

Fovea: Small central region of the retina where the density of cones (the photoreceptors that operate in daylight) is highest, providing the highest visual accuity. The receptor density drops off rapidly as you move away from the fovea, unlike in a video camera, which has the same resolution everywhere.

Lenslet array: Microlenses that may be placed directly over each pixel. These lenses may be designed to accept light coming from a particular direction, as in the insect eye, or to funnel light onto the light-sensitive part of the pixel. It is now possible to etch such lenses directly onto the surface of a silicon chip.

Photocurrent/photocharge: When light strikes silicon, it dislodges electrons from the lattice, and these electrons diffuse freely or are swept away by an electric field. This stream of light-generated charge constitutes a current, called a photocurrent, which is directly proportional to the photon flux or the light intensity. Alternatively, we can collect the charge carried by the photocurrent on a capacitor, in which case we get a photocharge.

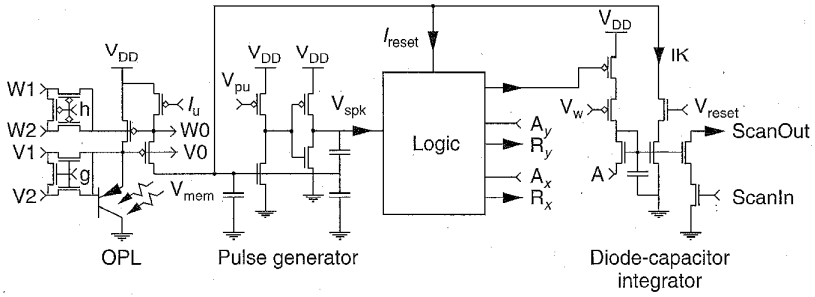


Figure 1. Pixel circuit for retinomorphic imager. The outer-plexiform-layer (OPL) circuit performs spatiotemporal band-pass filtering and local automatic gain control. The pulse generator converts the OPL circuit's output current to pulse frequency, quantizing the signal. The diode-capacitor integrator adapts the quantization process, making the step size and the sampling rate proportional to the signal's amplitude and rate of change. The logic circuit communicates the occurrence of a spike to the chip periphery, turns on I_{reset} to terminate the spike, and takes V_{adapt} low to increment the integrator. The remaining circuitry scans out the integrator's output for display on a video monitor.

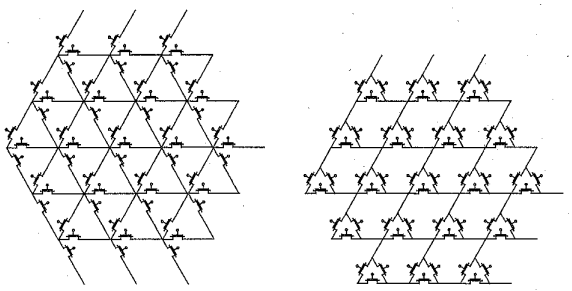


Figure 2. Tiling hexagonally connected networks using star elements (a) and delta elements (b). The star-based network requires wires running along three axes, whereas the delta-based network used in the retinomorphic pixel requires wires running along only two axes. Thus, the delta-based hexagonal grid is no more complicated than a more traditional square grid, yet it achieves a 33% improvement in peak sampling frequency for pixels of equal area.

amplitude. This encoding produces many redundant samples, because changes in the signal are rare. Also, the large amplitude codes are seldom used, since these signal amplitudes rarely occur in natural scenes. Reassigning the codes to more probable amplitudes increases the overall number of signals that can be discriminated. Thus, information is maximized when all codes are equally probable. If conversion occurs in parallel at the pixel level, each converter can adapt its quantization independently. In addition, this arrangement avoids corruption of analog signals by the switching noise that high-speed multiplexing produces.

How do we build the pixels?

The primary difference between retinomorphic and conventional vision systems is that retinomorphic imagers perform all four operations listed in Table 1 at the pixel level. Shrinking feature sizes in CMOS technology are driving the migration of more sophisticated signal processing down to the pixel level. The retinomorphic pixel circuit shown in Figure 1 has a total of 32 transistors and, fabricated in an outdated process with a 2-μm minimum feature size, occupies an area of 106×98 μm². As feature sizes are scaled down, and the size of the active devices becomes small compared to the sensor area—typically about 5 μm per side—it will become cost-effective to shrink the detector area and use lenslet arrays to focus the light. This will free up area for additional image-processing functions. With this approach, it will be possible to build a 380×380-pixel retinomorphic imager on a 1-cm² die in today's state-of-the-art, 0.4-micron CMOS process. In comparison, the human fovea has only about 500×500 cones, but the density is much higher: These cones occupy an area of just 1.5 mm×1.5 mm!

In general terms, the circuit's principles of operation are as follows. The transducer is a vertical bipolar transistor; its emitter current is proportional to the incident light intensity.[3] V0 and W0, the two nodes in the outer-plexiform-layer (OPL) circuit, which models the retina's first layer, connect to their six nearest neighbors by delta-connected transistors on a hexagonal grid, as shown in Figure 2, to form two current-spreading networks.[5,8] These networks diffuse the currents over space, and their node capacities diffuse signals over time. Thus, these two networks form low-pass filters for spatial *and* temporal frequency.

When there is a transition from black to white—an edge crosses the pixel or a light turns on—the photocurrent increases. The excess current discharges node V0, turning on the device whose gate connects to node V0. The excess current in this device discharges node W0. Thus, node W0 is also excited, turning on the device whose gate connects to it. The excess current in this second device balances the excess photocurrent, and tends to restore node V0. Thus, the second device inhibits the effect of the photocurrent. These two devices make the first network (V0) excite the second and make the second network (W0) reciprocate by inhibiting the first, in close analogy to excitatory and inhibitory synapses between cones and horizontal cells in the outer retina.[5] When the inhibitory network is biased such that its time and space constants are longer than those of the excitatory network, signals that change rapidly over time or space will escape inhibition, resulting in a high-pass frequency response. However, the frequency response will start to roll off when the period approaches the time and

space constants of the excitatory network, resulting in an overall band-pass response in spatial and temporal frequency.[5]

The low-pass-filtered version of the image from the inhibitory layer serves as a measure of the local intensity level and controls the circuit's gain. As it turns out, the gain is inversely proportional to the coupling strength in the excitatory layer.[5] We can increase this coupling strength globally by increasing the common gate voltage on the internode transistors (g), or we can change it locally by decreasing the node voltages (V0). The latter adjustment happens naturally as the light level increases, because the voltage on W0 drops as the inhibition increases to balance the increased photocurrent. As W0 drops, V0 drops as well to maintain a relatively constant current in the device between V0 and W0. The result is local automatic gain control.[5]

Figure 3. CCD camera (a) versus OPL imager chip (b) under variable lighting. The CCD camera performs global AGC, whereas the OPL chip performs local AGC and band-pass filtering.

Figure 3 shows the OPL circuit outputs (data from a chip I described previously[5]) and images of the same scenes acquired with a CCD camera.[4] The retinomorphic front end pulls out information in the shadows, whereas the CCD camera output has hit its lower limit. Local AGC indeed increases the dynamic range.

The spatiotemporal band-pass filtering removes gradual changes in intensity and enhances edges and curved surfaces. Unfortunately, the retinomorphic chip's output is noisier in the image's darker parts. This undesirable side effect of the gain control mechanisms arises because the space constant becomes shorter as the coupling strength in the excitatory network is reduced to increase the gain. When the space constant decreases, the chip can no longer attenuate salt-and-pepper noise, because the cutoff frequency shifts upward. At these intensity levels, the dominant noise source is the poor matching among the small ($4L \times 3.5L$) transistors operating in the subthreshold region—it is not fluctuations in the photon flux. Nevertheless, when the retinomorphic imager replaced a CCD as the front end of a face recognition system, this 90×90-pixel OPL chip improved the recognition rate from 72.5 to 96.3% (5% false positives) under variable illumination.[4]

A pulse generator converts analog currents from the OPL circuit into pulse frequency. The diode-capacitor integrator computes a current proportional to the short-term average of the pulse frequency, and this current is subtracted from the pulse generator's input. Hence, the more rapidly the input changes, the more rapidly the pulse generator fires. Adding a fixed-charge quantum to the integrating capacitor produces a multiplicative change in current—this is due to the exponential current-voltage dependence in the subthreshold region. Hence, the higher the current level (for a brighter region), the larger the step size. The result is adaptive quantization. In the postprocessor, the diode-capacitor integrator also integrates the pulses and reconstructs the encoded current level.

Figure 4 (next page) shows the response of the adaptive neuron circuit to a 14% change in its input current; these data also demonstrate the integration of pulse trains by the diode-capacitor integrator and the adaptive step size. It is preferable to keep the integrator's output current below the input current at all times: This ensures that the membrane voltage stays close to the threshold, making the latency shorter and less variable and keeping the integrator's output device in saturation. The circuit operates in this regime if small step sizes are used. In that case, however, adaptation occurs slowly, and the process generates many spikes.

How do we transmit the pulses?

Having quantized the signal in the pixel, we need to read out these asynchronous pulse trains from the array and transmit them off chip. For this purpose, my design uses a random-access communication channel. Random access is an alternative to the more common sequential-access protocol, which polls all the users sequentially and allocates a fixed fraction of the channel capacity to each user. That protocol's efficiency degrades as the fraction of active users decreases, because the polling of inactive users ties up bandwidth. In contrast, random access makes it possible to service only the active users, a more efficient use of channel capacity. However, this enhancement comes at the cost of sending $\log_2 N$-bit addresses to identify one out of N users, instead of using just 1 bit to indicate whether a user is active.

Nevertheless, a random-access channel better serves retinomorphic pixels for two reasons. First, activity is sparse, because band-pass spatiotemporal filtering attenuates signals that change slowly over time or space. Second, sampling is sporadic, because neurons fire rapidly when their inputs are

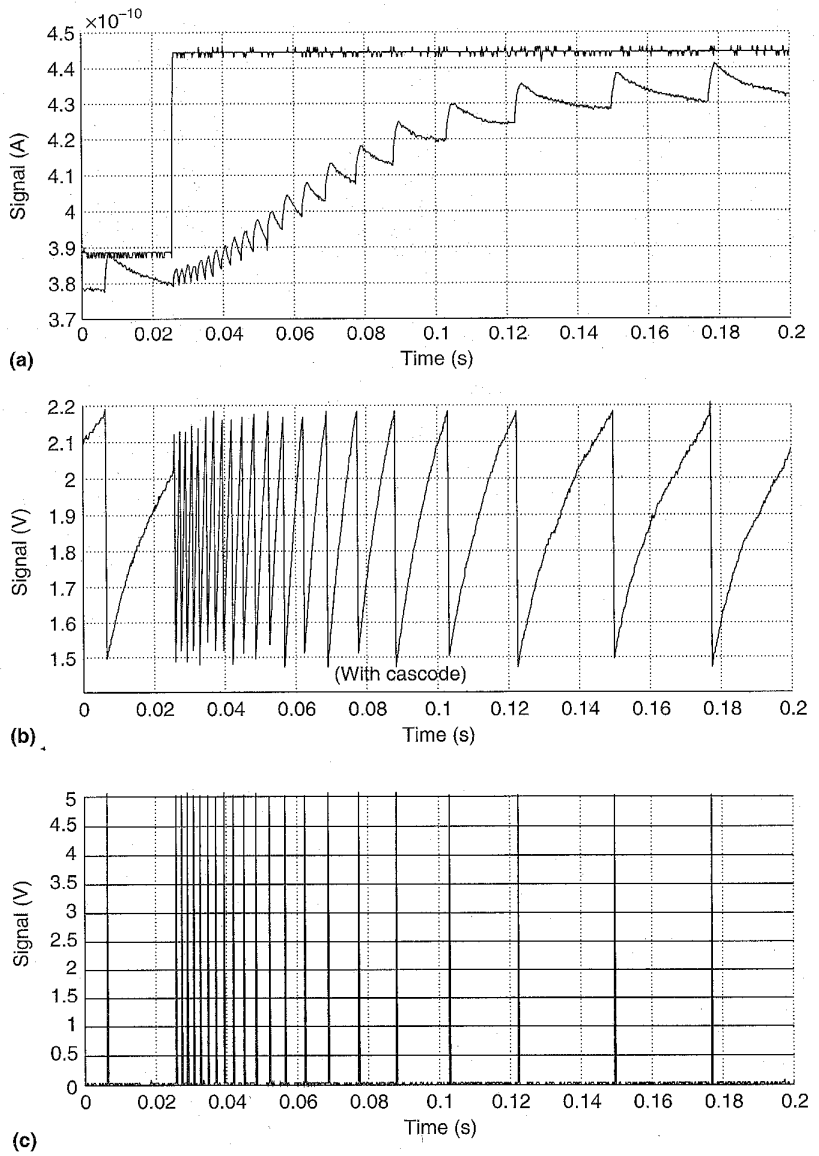


Figure 4. Adaptive neuron's step response: the neuron's input current and the integrator's output current (a); input voltage ramping up between the reset (1.5 V) and threshold levels (2.2 V) (b); and the spike train (c). The difference between the input current and the integrator's output ramps up the input voltage as it charges the input capacitance.

changing and reduce their sampling rates drastically when the signal is steady. Typically, only about 5% of the neurons are firing rapidly; the rest are quiescent, and fire at extremely low rates. If the active neurons fire 40 times faster than the quiescent ones, random access will allocate the channel capacity in a ratio of $0.05 \times 40{:}0.95 \times 1$—or 2 to 1—between the active and quiescent subpopulations. In contrast, sequential polling allocates the channel capacity according to numerical strength—that is, in the ratio 0.05:0.95, or 1 to 19—between the active and quiescent subpopulations. Hence, assuming equal channel capacity, the firing rate of active neurons must be 13.3 times lower in the sequential channel, because they can use only 5% of the channel capacity. The random-access channel, on the other hand, dynamically allocates 66.7% of its capacity to the active neurons.

The pixels must not transmit pulses at will, because that approach results in collisions when two or more pixels attempt to transmit at the same time. Instead, the communication channel includes an arbiter to deal with contention and a queue where unsuccessful contenders wait. This architecture was proposed and developed by Sivilotti[9] and Mahowald:[10] It allows graceful information degradation in the face of the heavy, but sporadic, demands on bandwidth caused by synchronous firing triggered by events occurring in the scene.

The arbiter scheme incurs some temporal dispersion of the burst of spikes when the channel capacity is exceeded briefly, but it scales linearly with the load. That is, a burst of duration T_{burst} that offers a peak channel load of F_{burst} will be dispersed over an interval no longer than $T_{burst}F_{burst}/F_{chan}$, where F_{chan} is channel capacity. In contrast, the number of collisions increases exponentially as the probability of spike occurrence rises.[11] Hence, introducing arbitration allows us to trade an exponential spike loss for a linear temporal dispersion, resulting in more graceful degradation.

The downside of arbitration is that it increases the length of the communication cycle, reducing the channel capacity, which is defined as the reciprocal of the cycle time. I have adopted three strategies to improve the throughput:

- *Pipelining.* There can be several addresses in various stages of transmission at the same time. This well-known approach to increasing throughput, called pipelining, involves breaking the communication cycle into a series of steps and overlapping the execution of these steps as much as possible.
- *Tree locality.* My system exploits locality in the arbiter tree. That is, it does not arbitrate among all the inputs every time; doing so would require spanning all $\log_2(N)$

levels of the tree. Instead, it finds the smallest subtree that has a pair of active inputs, and arbitrates between those inputs. This approach minimizes the number of levels spanned.

- *Row-column locality.* My system exploits locality in the row-column architecture as well. That is, it does not rearbitrate both rows and columns for each address transmitted. Instead, it services all requesting pixels in a selected row, rearbitrating only the columns; it rearbitrates rows only when no more requests remain in the selected row. Lazzaro et al. have also improved on the original design, and have used their improved channel to interface a silicon auditory model.[12]

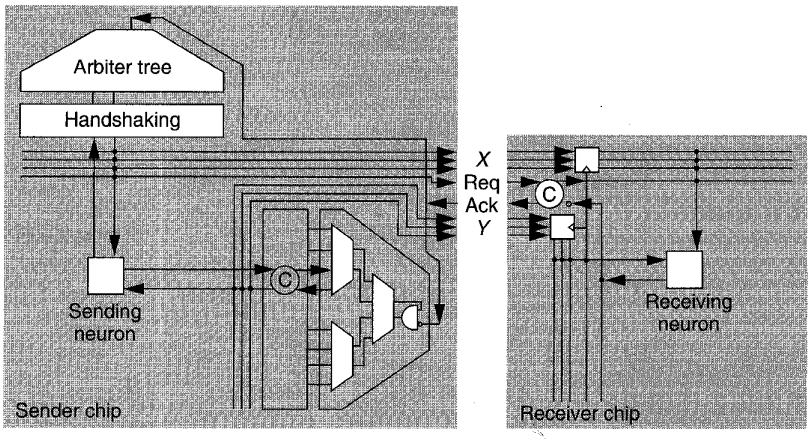


Figure 5. Pipelined address-event interchip communication channel. The arbiter consists of a tree of two-input arbiter cells that send a request signal to, and receive a select signal from, the next tree level. The row and column arbiter circuits are identical; the row and column handshake circuits (labeled C) also are identical.

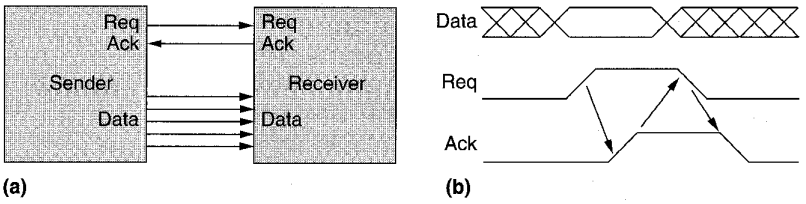


Figure 6. Self-timed data transmission protocol using a four-phase handshake: data bus (Data) and data-transfer control signals (Req and Ack) (a) and handshake cycle on control lines (b).

Figure 5 shows the pipelined design. Like the first generation, this interface is completely self-timed; thus, every communication must be acknowledged by a feedback signal, as shown in Figure 6. These acknowledge signals also implement the queue: They make a pixel wait just by refusing to acknowledge it. At the beginning of a communication cycle, the request and acknowledge signals are both low. Figure 7 (next page) shows the complete set of steps involved in the communication cycle.

On the sender side, spiking neurons first make requests to the *Y* arbiter, which selects only one row at a time. All spiking neurons in the selected row then make requests to the *X* arbiter. Concurrently, the *Y* address encoder drives the address of that particular row onto the bus. When the *X* arbiter selects a column, the pixel in that particular column, and in the row selected earier, withdraws its column and row requests. At the same time, the *X* address encoder drives the addresses of that particular column onto the bus, and takes Req high. When Ack goes high, the select signals that propagate down the arbiter tree are disabled by the AND gates at the top of the *X* and *Y* arbiter trees. As a result, the arbiter inactivates the select signals sent to the pixels and to the address encoders. Consequently, the addresses and the request signal, Req, are widthdrawn. When it is necessary, the handshake circuit (also known as a C element[13]) between the arbiters and the rows or columns delays inactivation of the select signals that drive the pixel, and the encoders, and thus gives the sending pixel sufficient time to reset. The sender's handshake circuit is designed to stall the communication cycle by keeping Req high until the pixel withdraws its row and column requests, confirming that it has reset. The exact sequencing of these events is shown in Figure 7. On the receiver side, as soon as Req goes high, the address bits are latched and Ack goes high. Concurrently, the address decoders are enabled and, while the sender chip is deactivating its internal request and select signals, the receiver decodes the addresses and selects the corresponding pixel. When the sender takes Req low, the receiver responds by taking Ack low, disabling the decoders and making the latches transparent again, at the same time. When necessary, the handshake circuit, which monitors the sender's Req and the receiving pixel's Ack, delays disabling the address decoders to give the receiving pixel sufficient time to read the spike and to generate a postsynaptic potential. The receiver's handshake circuit is designed to stall the communication cycle by keeping Ack high until the pixel acknowledges receiving the spike.

The arbiter works in a hierarchical fashion, using a decision tree built out of two-input cells.[9,10] Thus, arbitration between N inputs requires only $\log_2(N)$ two-way decisions. The two-input arbiter cell is essentially a flip-flop with negated inputs and outputs. That is, both the set and reset controls of the flip-flop are normally active, forcing both the flip-flop's outputs to be active. When one of the two requests becomes active, the corresponding control (either set or reset) is inac-

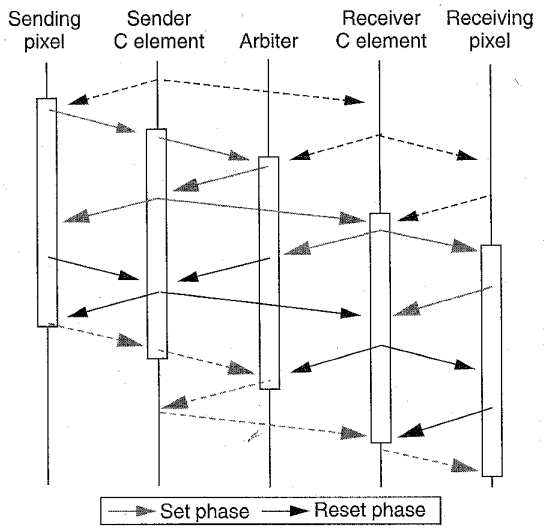


Figure 7. Pipelined communication cycle sequence for arbitration in one dimension, showing four-phase minicycles among five elements. The boxes indicate the duration of the current cycle, which may overlap with the reset phase of the preceding cycle and the set phase of the succeeding cycle. (Dashed lines in each phase indicate steps associated with preceding and succeeding cycles.) The cycle consists of three smaller interwoven minicycles: pixel to C element, C element to C element, and C element to pixel. Each pair of minicycles is synchronized by handshake circuits, also known as C elements. These circuits ensure that transitions occur in pairs.

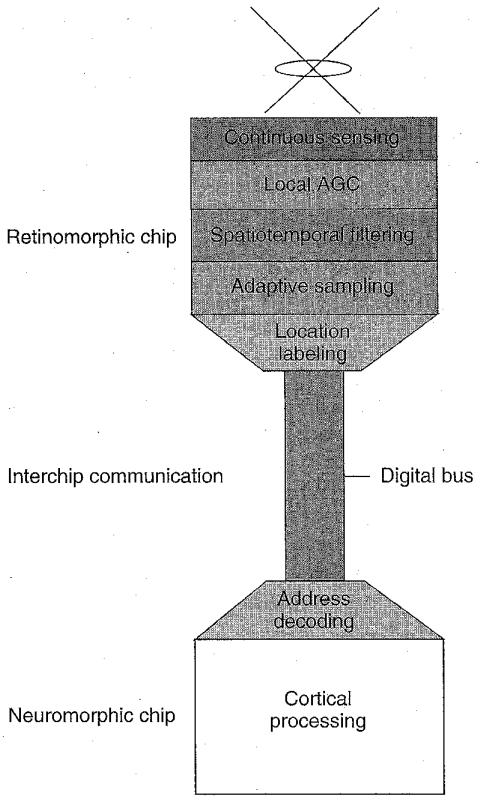


Figure 8. System concept. The retinomorphic chip acquires, conditions, prefilters, and quantizes the image—performing all these operations at the pixel level. The interchip communication channel reads out digital pulses from the pixels by transmitting the location of pulses as those pulses occur. A second neuromorphic chip decodes these addresses and recreates the pulses.

tivated, and that request is selected when the corresponding output becomes inactive. First, however, a request signal is sent up the tree, and the select signal propagates down the tree only when a select signal is received from the next level. When both requests become active simultaneously, both the set and reset controls are inactivated, and the flip-flop randomly settles into one of its stable states, with one output active and the other inactive. Hence, only one request is selected.

Since the arbiter cell continues to select an input as long as its request stays active, we can keep a row selected until all that row's active pixels are serviced. We do this simply by ORing together all its pixels' requests to generate the request to the Y arbiter. Similarly, since the request passed to the next level of the tree is simply the OR of the two incoming requests, a subtree remains selected as long as there are active requests in that part of the arbiter tree. Thus, once selected, each subtree will service all of its daughters. Using the arbiter in this way minimizes the number of levels of arbitration performed—the next input selected is always the one that requires the smallest number of levels to be crossed. To exploit the locality in the array and the arbiter, however, we must preserve the state of the flip-flops from cycle to cycle.

To reset the select signals fed into the array and to the encoder, previous designs removed the requests fed in at the bottom of the arbiter tree, and therefore did not preserve the arbiter's state.[9,10,12] Hence, these designs performed full $\log_2(N)$-level row and column arbitration for every cycle. In my design, I reset the select signals by removing the select signal from the top of the arbiter tree; thus, the request signals fed in at the bottom are undisturbed, and the arbiter state is preserved. This allows the design to fully exploit locality in the array and the arbiter tree.

This channel design achieves a peak throughput of 2.5 Mspikes/s, for a 64×64 array in 2-micron CMOS technology. The cycle time is 730 ns if arbitration takes place in both dimensions and 420 ns when arbitration takes place in only the X dimension (that is, when the pixel sent is from the same row as was the previous pixel). These cycle times represent a three to fivefold improvement over the 2-μs cycle times reported in the original work,[10] and are comparable to the shortest cycle time of 500 ns reported for a much smaller, 10×10-pixel nonarbitered array fabricated in 1.6-micron technology.[11]

Pipelining the receiver shaves off 113 ns of the cycle time—the time saved by latching the address event instead of waiting for the receiving pixel to acknowledge. Pipelining the sending pixel's reset phase is not effective in this case, because most of the delay arises from resetting the row and column wired-OR lines. However, these lines are not allowed to reset until the receiver's acknowledge disables the select signals that propagate down the arbiter tree.

Propagating these high- and low-going select signals down the six-level arbiter tree (110 ns + 40 ns) and resetting the column and row request lines (120 ns) adds 270 ns to the cycle time, when arbitration is performed in only the X dimension, and adds 400 ns when arbitration is performed in both the X and Y dimensions. Thus, we could improve the performance considerably by doubling the width ($3L$) of the devices in the pixel that pull down the wired-OR row or column lines, by allowing these lines to reset as soon as the pixel resets, and by disabling the select signals at the bottom of the arbiter tree, instead of disabling those at the top. (Tobi Delbruck suggested this change to me.) These changes, implemented together, will reduce the cycle time to 150 ns, for a full six-level arbitration in one dimension, or to 400 ns for a full six-level arbitration in both dimensions.

How does the chip fit into the system?

Figure 8 shows how my retinomorphic chip fits into a larger neuromorphic system. I've replaced the neurons in the neuromorphic chip with a two-dimensional array of diode-capacitor integrators. These integrate the pulse trains, producing slowly changing analog signals that model postsynaptic activity. Displaying these signals on a video monitor helps us visualize the retinomorphic chip's activity. Table 2 lists these chips' specifications, and Figure 9 shows their die photos.

Figure 10 shows the postprocessor output after image acquisition, analog preprocessing, quantization, address encoding, interchip communication, address decoding, and integration of charge packets in the receiver's diode-capacitor integrators. I culled these frames from a video sequence that provides a more vivid display of the system; the entire sequence is available at http://www.pcmp.caltech.edu/~buster/.

The sparseness of the output representation is evident. When the windmill moves, neurons at locations where the intensity is increasing (white region invades black) become active; hence, the leading edges of the white vanes are more prominent. These neurons fire more rapidly as the speed increases, because the temporal derivative increases.

Table 2. Specifications of the two-chip retinomorphic system.

Element	Specification	
	Imager	Postprocessor
Technology	2-μm, two-poly, two-metal p-well	
Number of pixels	64×64	
Pixel size (L^2)	53×49	31.5×23
Transistors/pixel	32	8
Die size (mm^2)	8.1×7.4	5.1×4.0
Supply	5 V	
Dissipation (0.2 MHz)	230 mW (total)	
Throughput	2 MHz	

Specifications listed between imager and postprocessor apply to both. (L = 2 μm)

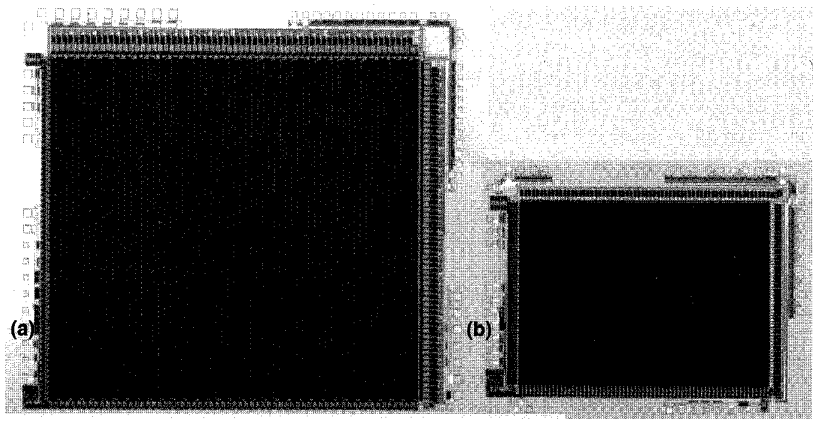


Figure 9. Die photos of retinomorphic focal-plane processor (a) and postprocessor (b).

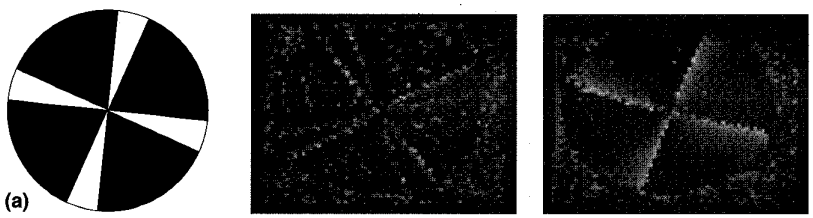


Figure 10. Video frames from diode-capacitor integrator chip showing real-time temporal integration of pulses. The stimulus is a windmill pattern (a) that rotates counterclockwise slowly (b) and quickly (c).

The time constant of the receiver's diode-capacitor integrator is shorter than that of the sender, so temporal integration occurs only at high spike rates. This mismatch wipes out DC information, and results in an overall high-pass frequency response that enhances the response to motion. The mean spike rate is 30 Hz per pixel; at this spike rate, the two-chip system dissipates 190 mW.

We can add a second set of neurons that encode an inverted version of the OPL circuit's output current to transmit decreases in intensity. This dual-rail encoding is analogous to the retina's on and off pathways, different kinds of ganglion cells that respond preferentially to light increase or light decrease.

THIS SYSTEM REPRESENTS an early attempt to build machine-vision systems that exploit regularities in natural scenes to optimize their information-gathering capacity. The retinomorphic approach reported here mimics both the structure and function of the biological retina. The system performs sophisticated signal processing at the pixel level to make the sensor maximally adaptive to inputs. The result is a sparse-output representation in time and space that uses the capacity of the output channel efficiently, and that achieves substantial power savings by eliminating redundant sampling operations and performing computations locally.

The retinomorphic imager I designed includes three retinal mechanisms to improve its information coding efficiency: local automatic gain control, band-pass spatiotemporal filtering, and phasic transient-sustained response. In addition to these mechanisms, the retina employs several other strategies to improve its coding efficiency. The following are three important ones, from which future retinomorphic imagers would benefit enormously.

- *High-pass temporal and spatial filtering* in the second stage of the retina (inner plexiform layer or IPL). This attenuates signals that do not occur at a fine spatial scale *and* temporal scale, eliminating the redundant signals passed by the OPL. The OPL responds strongly to low temporal frequencies that occur at high spatial frequencies (sustained response to static edge) or to low spatial frequencies that occur at high temporal frequencies (blurring of a rapidly moving edge).
- *Half-wave rectification*, together with dual-channel encoding (on and off output cell types), in the relay cells between the OPL and IPL (bipolar cells) and the retina's output cells (ganglion cells). This eliminates the elevated quiescent neurotransmitter release and firing rates required to signal both positive and negative excursions using a single channel.
- *Foveated architecture*, together with actively directing the gaze. This strategy eliminates the need to sample all points in the scene at the highest spatial and temporal resolution, while providing the illusion of doing so everywhere. The cell properties are optimized: smaller and more sustained at the fovea (parvocellular or *X* cell type), where the image is stabilized by tracking; and larger and more transient in the periphery (magnocellular or *Y* cell type), where motion occurs.

The compact adaptive neuron circuit described here and the enhanced connectivity provided by the address event interchip communication channel will make it possible to build large-scale neuromorphic systems. This effort will require extending the communication interface to support programmable convergent (many-to-one) and divergent (one-to-many) virtual connections between neurons. It will also require the addition of routing capability so that several neuromorphic chips may communicate with each other over a network of point-to-point connections. These enhancements will allow us to build neuromorphic systems that model the awesome parallel processing capabilities of the visual cortex, giving us the capability to compute three-dimensional shape, depth, and motion, at full motion video rates. Such computational power will make it possible, for the first time, to build compact autonomous biomorphs that interact purposefully with the environment in real time.

Acknowledgments

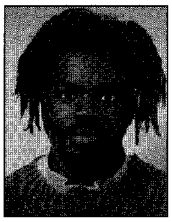
This work was partially supported by the Office of Naval Research; DARPA; the Beckman Foundation; the Center for Neuromorphic Systems Engineering, as part of the National Science Foundation Engineering Research Center Program; and the California Trade and Commerce Agency, Office of Strategic Technology.

I thank my advisor, Carver Mead, for sharing his insights into the operation of the nervous system. I also thank Misha Mahowald for making available layouts for the arbiter, the address encoders, and the address decoder; John Lazzaro, Alain Martin, and Jose Tierno for helpful discussions on address events and asynchronous VLSI; Tobi Delbruck for help with the Macintosh address-event interface; and Jeff Dickson for help with PCB design.

References

1. S.B. Laughlin, *Matching Coding, Circuits, Cells, and Molecules to Signals: General Principles of Retinal Design in the Fly's Eye,* Vol. 31, No. 1, of *Progress in Retinal and Eye Research,* Ch. 7, Pergamon Press, Oxford, England, 1994, pp. 165-196.
2. A. Dickinson et al., "Standard CMOS Active Pixel Image Sensors for Multimedia Applications," *Proc. 16th Conf. Advanced Research in VLSI,* IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 214-224.
3. C.A. Mead, "A Sensitive Electronic Photoreceptor," *Proc. Chapel Hill Conf. VLSI,* Computer Science Press, 1985, pp. 463-471.
4. J. Buhman, M. Lades, and F. Eeckman, "Illumination-Invariant Face Recognition with a Contrast Sensitive Silicon Retina," *Advances in Neural Information Processing 6,* J.D. Cowan, G.

Tesauro, and J. Alspector., eds., Morgan Kaufman, San Mateo, Calif., 1994.
5. K. Boahen and A. Andreou, "A Contrast-Sensitive Retina with Reciprocal Synapses," *Advances in Neural Information Processing 4,* J.E. Moody, ed., Morgan Kaufman, 1991.
6. J. Atick and N. Redlich, "What Does the Retina Know about Natural Scene?" *Neural Computation,* Vol. 4, No. 2, 1992, pp. 196-210.
7. J.H. van Hateren, "A Theory of Maximizing Sensory Information," *Biol. Cybern.,* Vol. 68, 1992, pp. 23-29.
8. E. Vittoz and X. Arreguit, "Linear Networks Based on Transistors," *Electronic Letters,* Vol. 29, 1993, pp. 297-299.
9. M. Sivilotti, *Wiring Considerations in Analog VLSI Systems, with Application to Field-Programmable Networks,* doctoral thesis, Dept. Computer Science, Calif. Inst. Tech., Pasadena, Calif., 1991.
10. M. Mahowald, *An Analog VLSI Stereoscopic Vision System,* Kluwer Academic Publishers, Boston, 1994.
11. A. Mortara, E. Vittoz, and P. Venier, "A Communication Scheme for Analog VLSI Perceptive Systems," *IEEE Trans. Solid-State Circuits,* Vol. 30, No. 6, 1995, pp. 660-669.
12. J. Lazzaro, J. Wawrzynek, and A. Kramer, "System Technologies for Silicon Auditory Models," *IEEE Micro,* Vol. 14, No. 3, June 1994, pp. 7-15.
13. I.E. Sutherland, "Micropipelines," *Comm. ACM,* Vol. 32, No. 6, 1989, pp. 720-738.



Kwabena A. Boahen is a doctoral student at the California Institute of Technology, Pasadena, Calif., in the Computation and Neural Systems Program. His current research interests include mixed-mode multichip VLSI models of biological sensory systems and asynchronous digital interfaces for interchip and intrachip communication. Boahen earned a BS/MSE degree in electrical engineering from the Johns Hopkins University, Baltimore, Maryland.

Direct questions about this article to Kwabena A. Boahen, Physics of Computation Laboratory, California Institute of Technology, MS 136-93, Pasadena, CA 91125; buster@pcmp.caltech.edu.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 159 Medium 160 High 161