System Implementations of Analog VLSI Velocity Sensors

Giacomo Indiveri

Jörg Kramer

Christof Koch

California Institute of Technology



These three single-chip, analog VLSI systems extract information from optical-flow fields for use in machine-vision and navigation systems. At the input stage, arrays of compact velocity sensors provide the optical-flow data.

Real-time machine-vision application tasks are computationally intensive and often require complex, costly resources. Furthermore, certain specific tasks that rely heavily on machine vision (such as space exploration, driver-assist systems for automobiles, and autonomous navigation) place additional constraints on the overall system in terms of its size, power consumption, shock resistance, and manufacturing cost. An attractive solution to the problems posed by these constraints is to use parallel image-processing architectures implemented in analog VLSI technology.

Within this framework, velocity sensors could prove extremely useful in a wide variety of applications. Analog velocity sensor circuits have been thoroughly investigated in the past. (For a review, see Sarpeshkar et al.[1]) Nonetheless, researchers were unable to obtain a device that would simultaneously be compact, robust to background brightness level and insensitive to stimulus contrast, and have a wide, unambiguous range of speed selectivity. Recently, however, we have proposed and built just such velocity sensors: They are sensitive to low-contrast stimuli, independent of contrast (for intermediate- and high-contrast values), and selective to over three orders of magnitude of velocity and over two orders of magnitude in light irradiance.[2,3] Because these sensors are also extremely compact, we can now integrate them at the system level. They are ripe for use in real-time machine-vision applications that require special-purpose parallel hardware for computing optical flow across an entire image.

However, optical flow is only an approximation of the 2D projection of the true 3D velocity field, based on spatiotemporal variations in image brightness. Moreover, analog circuits are limited by low precision in their state variable values. For these two reasons, we targeted applications that rely on integrative features of the optical-flow field rather than on the precise values of its vectors. (That is, our target applications rely on features that derive from averaging and thresholding operations.) Specifically, in this article we present three different architectures that use the facilitate-and-sample velocity sensor[2] (see the box for more information). Our architectures use this sensor to compute focus of expansion, time to contact, and motion discontinuities.

Focus of expansion

During an observer's motion through an environment, the velocity vectors generated in an instant of pure translational motion are radial in nature. These vectors expand out from a point that corresponds to the direction of heading, also called the focus of expansion (FOE).

By choosing a particular application domain—such as, for example, vehicle navigation—we can use a priori information and make assumptions that simplify the FOE detection problem for general cases. Specifically, for vehicle navigation, we can restrict our analysis to pure translational motion. This takes advantage of the fact that it is possible to compensate for a motion's rotational component using lateral accelerometer measurements from other sensors often already present on the vehicle. (We can do this, for example, by inputting the signal that encodes the rotational component of velocity and subtracting it from the measured optical flow field.)

Furthermore, being interested in determining and possibly controlling the heading direction mainly along the horizontal axis, we greatly reduce the problem's complexity by considering 1D arrays of velocity sensors. In such a case, we measure only the horizontal component of the optical-flow vec-

0272-1732/96/$5.00 © 1996 IEEE

The facilitate-and-sample velocity sensor

In earlier work, members of our group successfully built and characterized a velocity-sensing element that can be integrated into dense arrays to estimate optical-flow fields in parallel.[3] Unlike most previous implementations of analog VLSI motion sensors, this element unambiguously encodes 1D velocity over considerable velocity, contrast, and illuminance ranges, and is reasonably compact. It implements an algorithm that measures the time of travel of edges in the image between two fixed locations on the chip. It can then calculate the image velocity as the ratio of the known spatial separation of the two locations and the measured time of the edge's travel between them.

Figure A1 shows the schematic architecture of one of the sensor's functional elements. In an edge detection stage (labeled E in Figure A1), rapid dark-to-bright irradiance changes, called temporal ON edges, are converted into short current pulses at the two locations (I_1 and I_2). At each location, the current pulses feed into a pulse-shaping stage (labeled P), which generates a logarithmically decaying voltage signal (V_{s1} and V_{s2}) and a sharp voltage spike (V_{f1} and V_{f2}) in response to each edge pulse. In a motion-sensing stage (M), the slowly decaying signal from one location (V_{s1}) is sampled by the sharp spike from the adjacent location (V_{f2}), and vice versa with V_{s2} and V_{f1}. Because the slowly decaying signal can be regarded as a facilitation pulse for the velocity measurement, and because it is sampled by the spike signal, we call the algorithm facilitate and sample.

The motion-sensing stage consists of two M elements, each encoding velocity for one direction of the edge's motion. The element for which the onset of the facilitation pulse precedes the sampling spike (Figure A2) encodes the relative time delay of these signals generated by the same edge at the two locations, and therefore the edge velocity. The edge is then said to move in the element's preferred motion direction. For the other element, the sampling spike precedes the onset of the facilitation pulse, so it samples the residual voltage of the facilitation pulse triggered by the previous edge. The edge is said to move in the null direction of this element, and the sampled voltage does not contain velocity information.

In the absence of spatial aliasing—that is, if the separation of succeeding edges is larger than twice the pixel spacing—the output voltage for the preferred direction is always greater than the spurious output voltage for the null direction. The sensor suppresses the spurious output by comparing the voltages at the two outputs and setting the lower one to zero (D in Figure A1).

In a 1D array, an elementary, bidirectional velocity-sensing element consists of one edge detector, one pulse-shaping circuit, and two motion cells. When implemented with a 2-micron CMOS process, the size of such an element (including 41 transistors and 8 capacitances) is 0.05 mm^2.

Figure B shows that the experimental data confirms the predicted logarithmic encoding of time delay—and thus velocity—by the analog output voltage. We collected this data by imaging a moving high-contrast ON edge onto the chip under AC incandescent room illumination. The 300-μm spacing of the chip's adjacent photoreceptors determines the image velocity calibration in the focal plane.

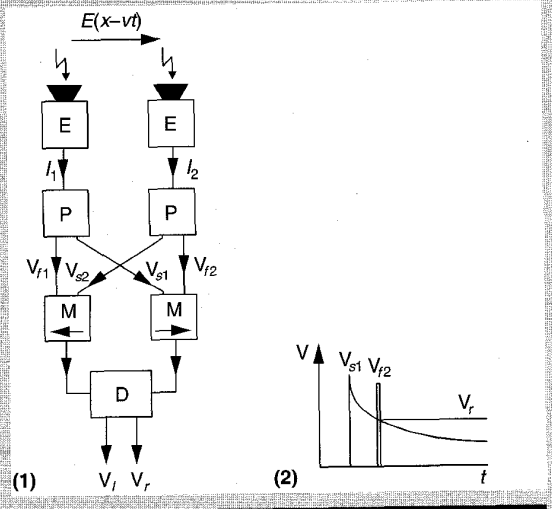


Figure A. Facilitate-and-sample velocity sensor block diagram (1) and voltage signals (2): The analog output voltage of the motion circuit for rightward motion (V_r) equals the voltage of the slowly decaying facilitation pulse (V_{s1}) at the time the narrow sampling pulse (V_{f2}) arrives. For leftward motion, the sampling pulse precedes the facilitation pulse and the output voltage is low. The analog output voltage thus encodes velocity for rightward motion only.

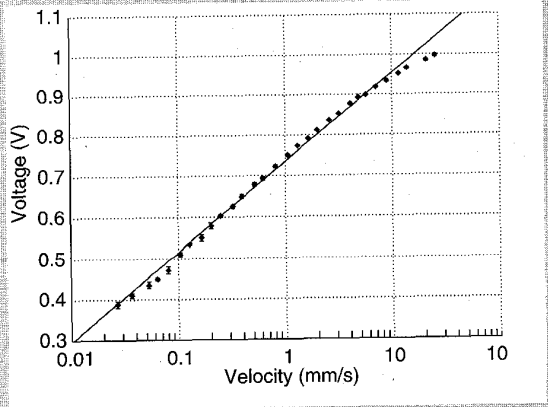


Figure B. The velocity-sensing element's output voltage for the preferred direction of motion of a sharp, high-contrast ON edge versus image velocity under AC incandescent room illumination. Each data point represents the average of five successive measurements.

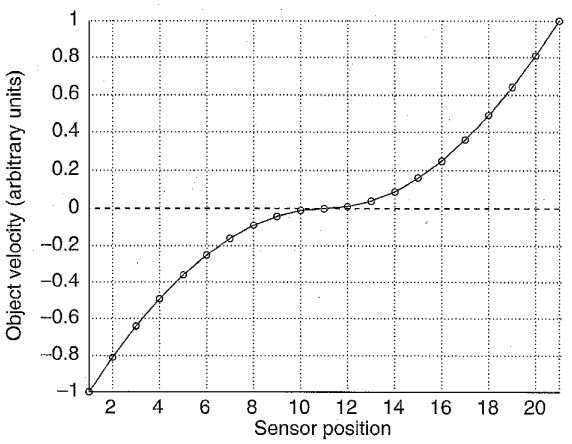


Figure 1. Signed encoding of simulated optical-flow vectors of an edge approaching an ideal observer, at different positions on the array of velocity sensors. The FOE is centered on the 11th sensor.

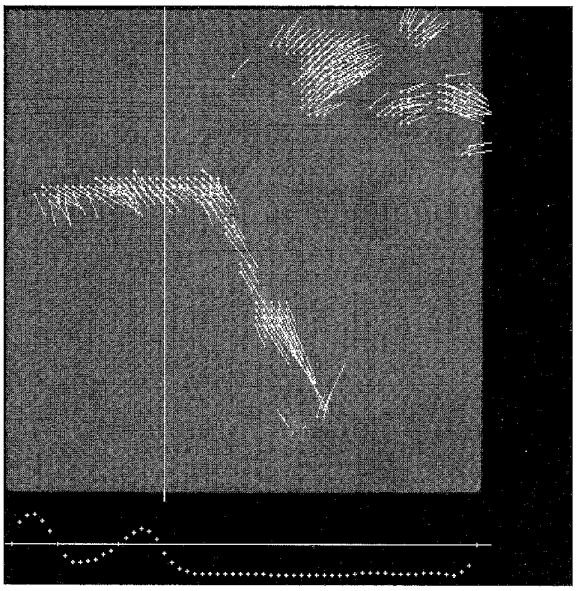


Figure 2. Image obtained from a camera mounted on a truck, moving on a straight road, with optical-flow vectors superimposed. The bottom of the figure represents the sum of the horizontal components of the optical-flow field for each column. We compute the coordinate of the heading direction as the abscissa of the zero crossing that has maximum steepness and is closest to the abscissa of the previously selected unit. An animation of the optical-flow sequence is available at http://www.klab.caltech.edu/~giacomo/opflow.html.

tors, obtained from pure translational motion in a fixed environment. Consequently, the problem of detecting the FOE reduces to detecting the point at which the optical-flow vectors change direction. If we code such vectors with positive values for one direction and negative values for the opposite direction, the problem then is to detect the data array's zero crossing. Ideally, if we use this convention, the velocity vectors of an edge translating toward an array of velocity sensors should yield a result similar to that shown in Figure 1.

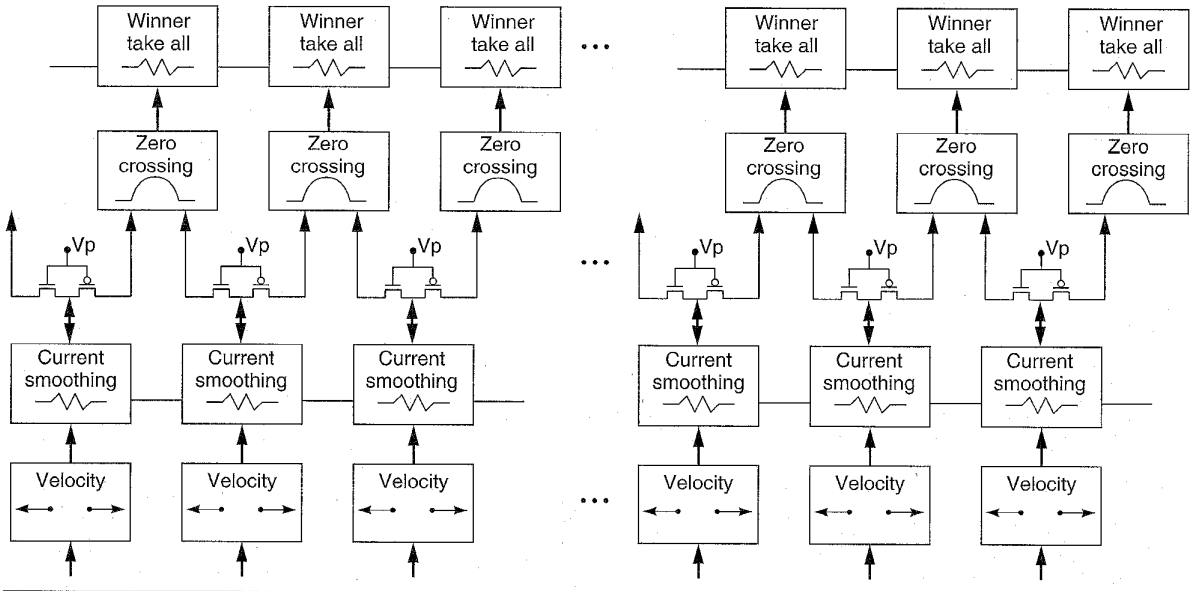


Figure 3. Block diagram of the analog VLSI architecture for determining the FOE position for an observer translating in a fixed environment.

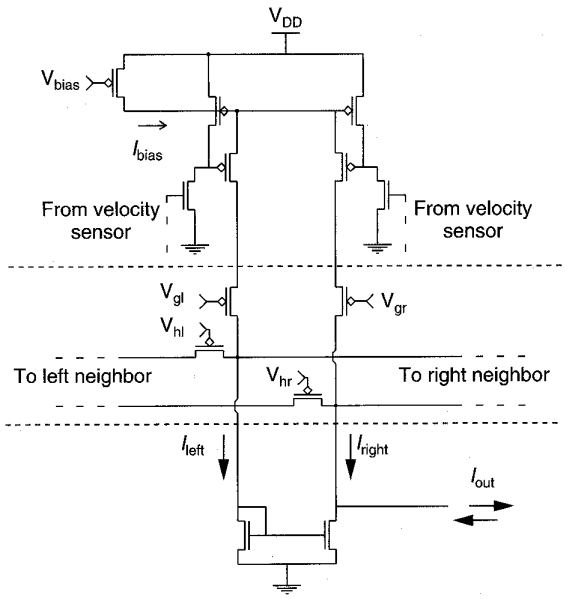


Figure 4. Circuit diagram of the current-smoothing block, the second processing stage of the FOE architecture.

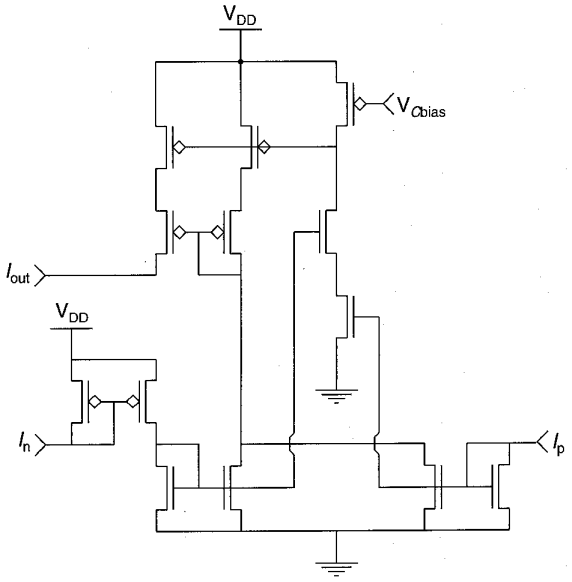


Figure 5. Circuit diagram of the zero-crossing detection block. The circuit generates a positive output current I_{out} if both input currents I_n and I_p are greater than a threshold value set by bias voltage V_{Cbias}.

To analyze the computational properties of the optical flow for typical vehicle navigation scenes in real cases, we performed software simulations on sequences of images obtained from a camera with a 64×64-pixel silicon retina placed on a moving truck (provided by Rockwell Corporation).[4] We obtained the optical-flow vectors using a standard procedure based on the image brightness constancy equation.[5] We encoded the vectors' horizontal components with positive or negative values depending on their direction and averaged them along the image columns. We then computed the heading direction position by detecting the zero crossing with maximum steepness and closest to the previously selected unit. This computation exploited the a priori assumption that the heading direction position shifts smoothly in space.

Errors inherent to the optical-flow computation and noise present in real image sequences often create spurious zero crossings in the 1D array of values encoding the flow vectors' averaged horizontal components (see Figure 2). To account for these errors and to select the zero crossing that corresponds to the correct FOE position, we designed and implemented a circuit architecture with four main processing stages (see Figure 3).

The input stage is a 1D array of elementary velocity sensors that use on-chip photoreceptors to measure the speed and direction of moving temporal edges (see the earlier box). Each velocity sensor has a differential output with one terminal for the preferred direction of motion and another for the null direction. Stimuli moving in the preferred direction cause an output logarithmically dependent on the speed of the stimulus at the first terminal (ranging from 400 mV to 1.1 V) and a null output at the second terminal (typically around 200 mV). Stimuli moving in the opposite direction generate a specular response. In the second processing stage, a circuit based on a two-node winner-take-all (WTA) network (Figure 4, top) converts the differential voltage signal of each velocity sensor into current.

Depending on the stimulus' direction of motion, each local WTA network will suppress the signal from the null-direction terminal and source a set bias current (I_{bias} in Figure 4) in the branch connected to the preferred-direction terminal. The sourced current then spreads laterally (to implement spatial smoothing) using the p-type transistors with gate voltages V_{gr}, V_{gl}, V_{hr}, and V_{hl} (middle part of Figure 4). Finally, a current mirror subtracts the smoothed currents from the two WTA branches ($I_{right} - I_{left}$), yielding a bidirectional current, the sign of which encodes the direction of motion.

The third processing stage detects zero crossings by looking for the copresence of negative currents from one unit and positive currents from the neighboring unit. The circuit that implements this operation (Figure 5) is based on a current-correlator circuit and a digitally controlled current mirror. If both input currents are greater than a set threshold value (controlled by V_{Cbias} in Figure 5), the output is activated. This generates a current I_{out}, which corresponds to the sum of the absolute values of the two inputs, $|I_n| + |I_p|$. To select the zero crossing that corresponds to the correct FOE location, we choose the one with the steepest slope (maximum current sum). We do this by feeding the output of the zero-crossing circuits to a global WTA network with lateral excitation.[6] Lateral excitation accounts for the fact that the FOE position shifts smoothly in space: It facilitates the selection of units close to the previously chosen winner and inhibits units farther away. In this way, once the system has

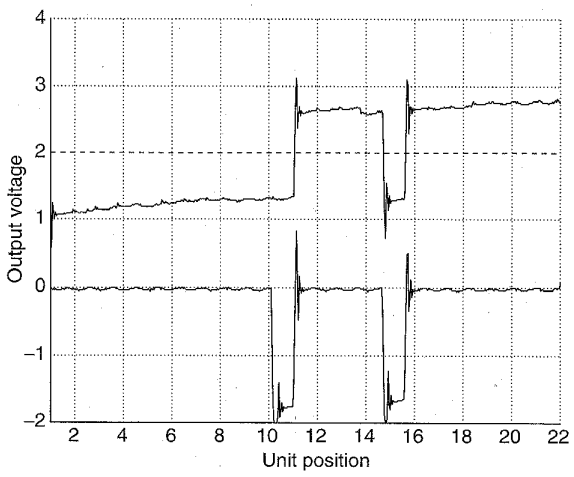


Figure 6. Data measurements from the analog VLSI device implementing the FOE detection algorithm. The top plot shows an output trace—with lateral spreading switched off—of the 22 units that convert the velocity sensors' differential voltage into a positive or negative bias current. An off-chip sense amplifier with a reference voltage set to 2 V converted the currents to voltages. For the left part of the trace, input stimuli were moving left, and for the right part, right. One of the velocity sensor's outputs (the U-shaped dip at unit 15) is incorrect due to noise intentionally added to the system. The bottom plot shows the output of the zero-crossing detection circuits. For clarity, we have shifted the data down by a constant value of 2 V.

selected a strong zero crossing, it tends to track that crossing as it moves along the array.

Figure 6 shows the outputs of the current-smoothing and zero-crossing detection stages. The input stimulus came from a rotating drum with edges moving to simulate ego-motion in a stationary environment. We used an 8-mm lens to image the input stimulus onto the chip (see Testing section). In the example shown, we deliberately switched off spatial smoothing to show how zero crossings corresponding to the correct FOE position are indistinguishable from spurious zero crossings, erroneously generated. Because the input came from moving stimuli, it was not possible to measure the smoothed output of the circuits simultaneously.

Spatial smoothing reduces the effect of spurious zero crossings by exploiting the data array's symmetry: The correct, single zero crossing maintains a high degree of steepness even after smoothing. On the other hand, spatial smoothing either eliminates spurious, double zero crossings completely—smoothing isolated bad data points so that they don't cross the zero axis—or brings them closer to the zero axis so that their slope is significantly lower than that of the correct zero crossing. If we use spatial smoothing together with high-contrast, well-controlled stimuli, the WTA network in the architecture's last processing stage can select and track the correct FOE location more reliably.

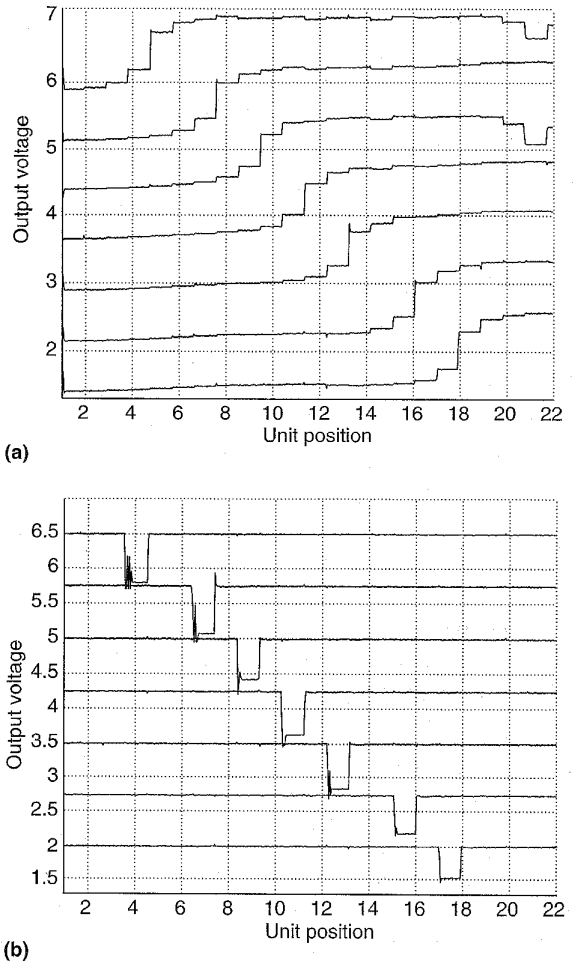


Figure 7. Output of the system at the second processing stage as the FOE position shifts from left to right: direction of velocity selection and differential voltage-to-current conversion (a). We set the bias current of the two-node local WTA networks with V_{bias} = 3.83 V. We obtained spatial smoothing by setting $V_{gr} = V_{gl}$ = 2.5 V and $V_{hr} = V_{hl}$ = 2.4 V. An off-chip sense amplifier with a reference voltage set to 2 V converted the output currents into voltages. We separated each plot from the next by adding a constant value of 0.75 V. Output of the system at the third processing stage, zero-crossing detection, for the same sequence of FOE positions (b). We set the threshold for detecting zero crossings with V_{Cbias} = 4.24 V.

Figure 7 shows the system's output, with smoothing turned on, as the FOE position shifts from left to right.

Time to contact

Another quantity that we can extract from an optical-flow field obtained with analog VLSI velocity sensors is the time to contact with an object partially or completely covering the

visual field. We define time to contact as the time it would take an observer to collide with an object, if the relative velocity between observer and object remains constant. The time to contact is thus a very useful quantity for navigation systems, especially in the case of motion in a rigid environment. For translational motion with relative speed v of the observer with respect to the object, the following equation gives time to contact τ:

$$\tau = d/v.$$

Here, d is the distance between object and observer measured along the direction of relative motion. The time to contact is therefore the inverse of the rate of looming, v/d.

Behavioral and electrophysiological evidence supports the hypothesis that the time to contact triggers landing responses in flies and birds and escape responses in a variety of animals, including humans.

For translational motion toward a planar surface perpendicular to the optical axis of the imaging system, velocity field $\mathbf{V}$ in the image is linear, and its divergence $\nabla \mathbf{V}$ is thus constant across the surface. Using the 2D version of Gauss's divergence theorem, we can then estimate the time to contact robustly from the line integral of the normal velocity component along a closed contour.[7] If the contour is circle C of radius r, we obtain[1]

$$\tau = \frac{2\pi r^2}{\int_C \mathbf{V} \cdot \mathbf{n} ds}, \quad (1)$$

where $\mathbf{n}$ denotes the unit normal vector along the contour. The focus of expansion does not have to lie within the contour. Its position, relative speed v, and distance d between object and observer do not have to be known. Furthermore, because Equation 1 depends on integrative rather than differential properties of the velocity field, the estimation of the time to contact is numerically stable. This is the case even in the presence of random noise and offsets, which are characteristic of subthreshold analog circuits.

Using the result from Equation 1, we can estimate time to contact with a circuit consisting of an array of 1D velocity-sensing elements arranged on a circle, such that each element measures radial velocity. The properly normalized sum of all sensor outputs then approximates the line integral, so that the time to contact amounts to

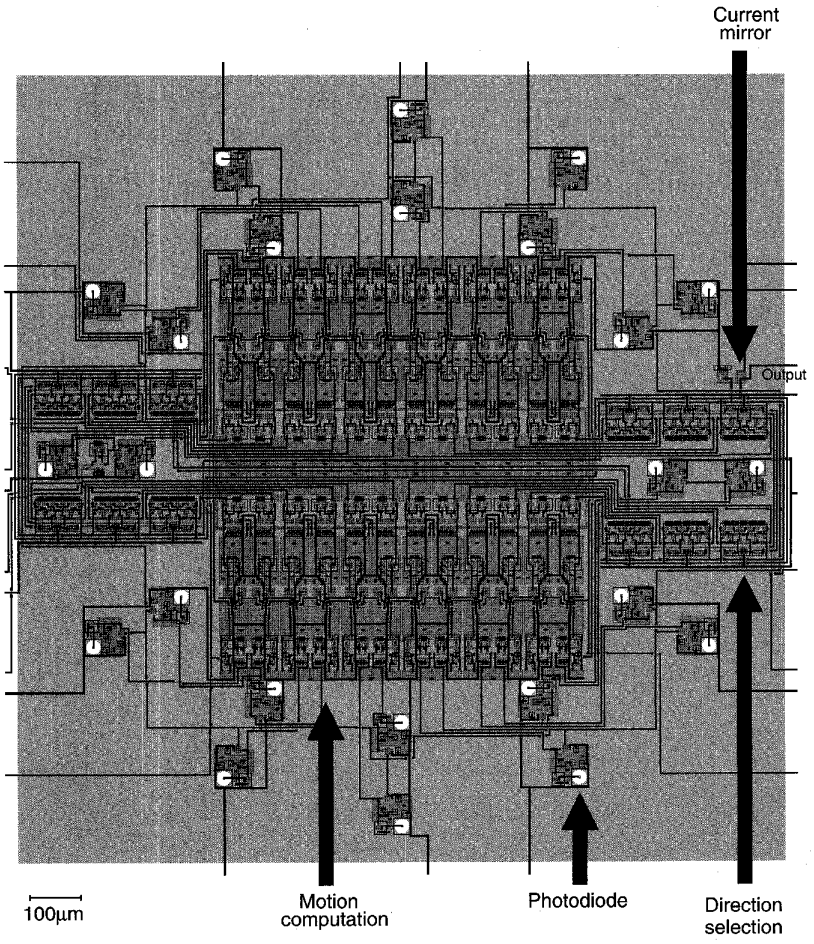


Figure 8. Layout of an analog VLSI chip that determines time to contact using facilitate-and-sample velocity sensors. The pulse-shaping circuits and motion circuits are in the center, and the direction selection circuits are on the left and right sides. The layout size is 1.6 mm×1.6 mm.

$$\tau = \frac{N \cdot r}{\sum_{k=1}^{N} V_k}. \quad (2)$$

Here, N denotes the number of elements on the circle, and V_k the radial velocity components at the locations of the elements.

We implemented such a circuit on a VLSI chip with 12 radially oriented velocity-sensing elements.[8] Figure 8 shows this chip's layout. The photodiodes of the velocity-sensing elements are arranged on two concentric circles with radii of 400 and 600 µm.

Figure 9 (next page) shows the circuit's schematic diagram. The output voltage of each velocity-sensing element controls a subthreshold transistor current. Because this voltage is logarithmically dependent on velocity, the current is proportional to velocity. Thus, we can calculate the sum of the velocity components by aggregating the currents from all elements on two lines—one for outward and one for inward motion—and taking the difference of the total currents. The

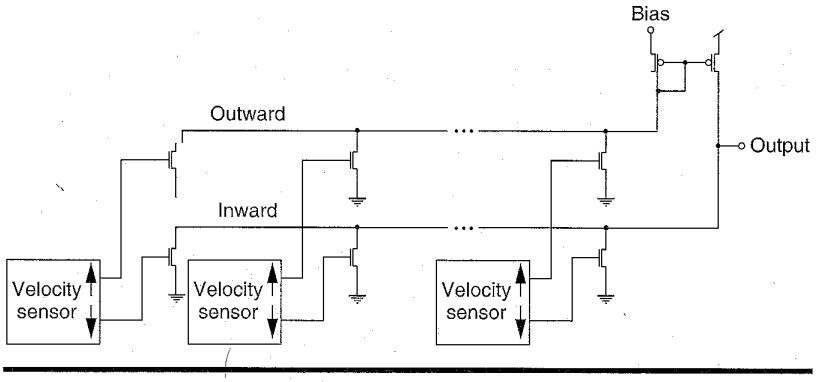


Figure 9. Schematic diagram of the time-to-contact sensor. The chip converts the output voltages of the velocity-sensing elements into currents proportional to the measured velocities. It then aggregates the currents representing outward and inward motion on two separate lines and subtracts them from each other. The resulting output current is proportional to the inverse of the signed time to contact.

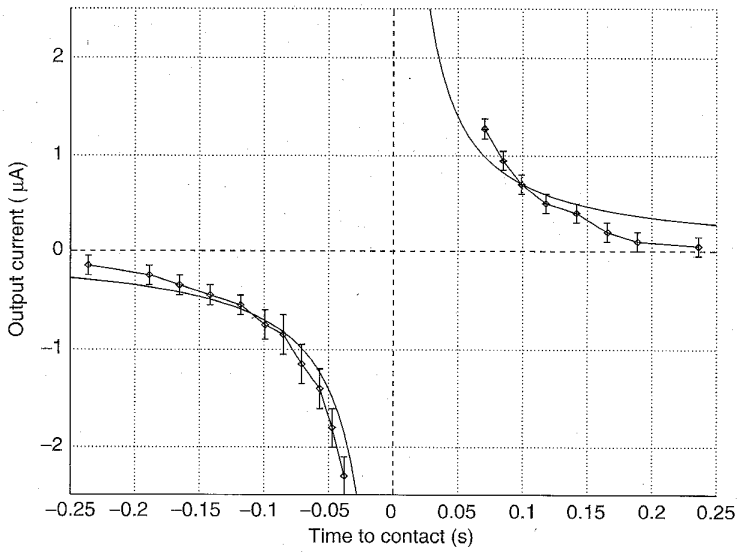


Figure 10. Output current of the time-to-contact sensor as a function of simulated time to contact under AC incandescent room illumination. The theoretical fit predicts an inverse relationship.

resulting bidirectional output current is then an inverse function of the signed time to contact according to Equation 2.

The circuit yields reasonably accurate estimates of time to contact for 1) an approaching or receding pattern of high-contrast concentric rings centered on the focus of expansion, and 2) a spiral stimulus on a rotating disk that simulates approaching or receding motion (see Testing, later). Figure 10 shows the averaged output current in response to the spiral stimulus as a function of simulated time to contact with a theoretical fit using Equation 2.

The data shown in Figure 10 allows us to observe qualitatively the expected inverse relationship of output current and time to contact, and robustly encodes the sign (expansion or contraction). However, the deviation of the output current from its average can be substantial. The output voltage of each velocity-sensing element gradually decays due to leak currents, and the spiral stimulus causes a serial update of the velocity values along the array. Thus, upon each update we observe a step change in the output current followed by a slow decay. The effect is aggravated if the individual elements measure significantly differing velocities. This is generally the case, because the FOE is often not centered on the sensor. There are also inaccuracies in the velocity measurements. These are due to circuit offsets, noise, and the aperture problem[9]—the fact that the direction of motion of a linear image feature cannot be determined from strictly local measurements.

Nonetheless, because the algorithm is integrative, we can expect more robust results from stimuli with higher edge densities and arrays with a larger number of sensing elements. Such arrangements will reduce statistical errors, and the high edge detection rate will prevent the signal from decaying significantly or changing abruptly. In this way, we should obtain reasonable estimates for the time to contact in more general scenes.

Motion discontinuities

Segmentation of images into different objects and a background is an important step in most image-processing systems. In dynamic scenes, we can segregate objects by their different apparent velocities with respect to the observer. These are either induced by motion parallax due to the observer's motion or by independent motion of the objects. For reasonably fast motions, segmentation based on motion discontinuities is less error-prone in complex environments than segmentation based on extracted edges. It therefore lends itself well to implementation in navigation systems mounted on rapidly moving platforms. We can extract motion discontinuities from an array of velocity-sensing elements by comparing the velocities measured by neighboring elements.

We implemented a circuit that finds motion discontinuities in a 1D image and outputs current signals at the discontinuity locations.[10] It uses a linear array of velocity-sensing elements as a front end. The circuit (see Figure 11) compares

voltage outputs of pairs of adjacent elements for both directions of motion separately. If the absolute value of the voltage difference for either direction of a pair exceeds a set threshold, the circuit activates a current at the pair location, signaling a discontinuity. This current's value is a monotonic function of the absolute value of the speed difference, which saturates at large differences. If the voltage difference remains below the threshold, the current remains shut off. We set the threshold with a bias voltage to exceed the fixed-pattern and temporal noise of the velocity-sensing array for uniform image motion.

For testing, an on-chip scanner periodically reads the output currents at the array's different locations. We passed the resulting output signal through a linear current-to-voltage converter and displayed it on an oscilloscope. The array contains 24 velocity-sensing elements with a pitch of 60 μm, giving 23 discontinuity measurements. The total size of the circuitry is 1.5 mm×1.1 mm, implemented with 2-micron technology.

Figure 12 shows a scope trace of the circuit response to a black bar translating in front of a background of uniformly moving black and white stripes. The two current peaks mark the locations of the edges of the black bar. The velocities differed enough to saturate the output currents at the edge locations. We took the data for Figure 12 with the bar moving in the same direction as the background, but we measured the same type of signal for the bar and background moving in opposite directions. For uniform image motion, we observed no output.

Suppose we feed a copy of the velocity-sensing array's output signals into a resistive network to smooth out random noise and circuit offsets. This would allow us to use motion discontinuity signals to diminish or eliminate locally the amount of smoothing across object boundaries by increasing the local resistance or by opening switches at their respective locations.[11] With such an architecture, we could implement smoothing and segmen-

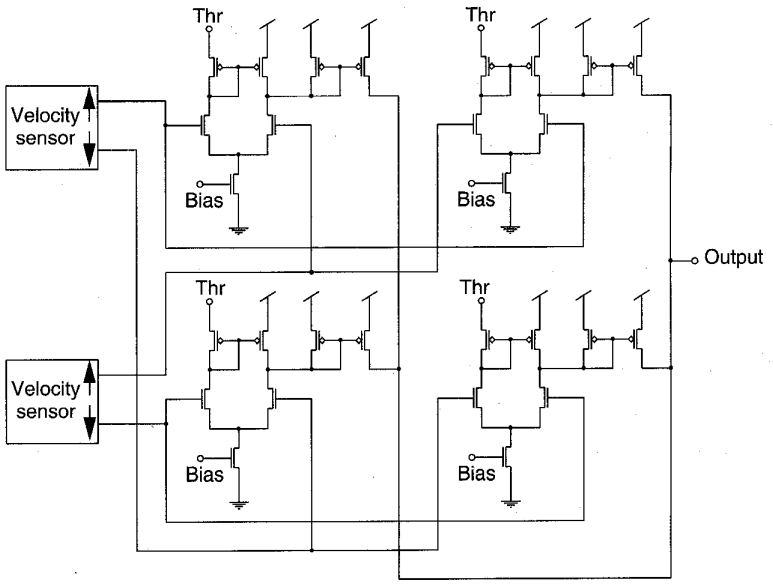


Figure 11. Schematic diagram of the motion discontinuity detection circuitry for a pair of adjacent velocity-sensing elements. The circuit compares the absolute value of the difference of the output voltages against a threshold (set by the Thr bias voltage) for both directions of motion. If the difference exceeds the set threshold for one or both directions of motion, the circuit outputs a current signal.

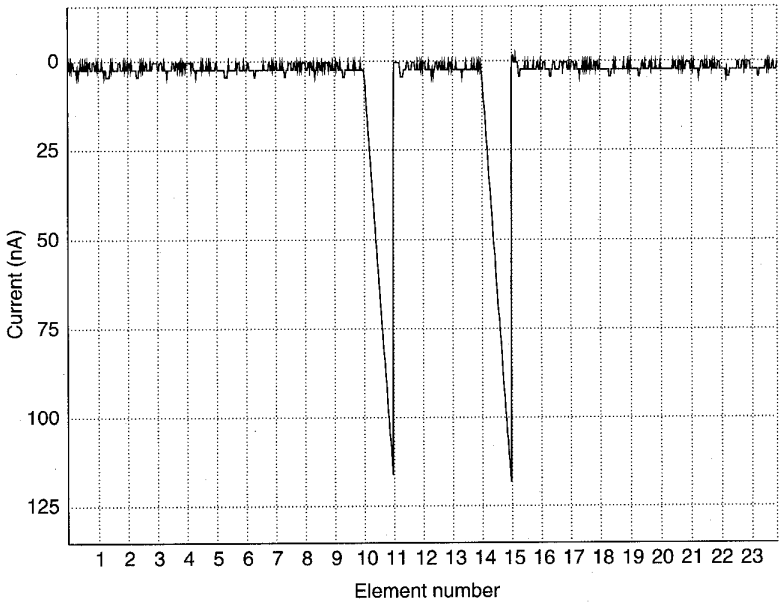


Figure 12. Response of the motion discontinuity chip to a black-bar stimulus traveling across a striped background, which is moving in the same direction at a different velocity. The velocities on chip were 5 mm/s for the bar and 30 mm/s for the background. The voltage peaks on the scope trace show the locations of the bar's edges on the imaging array. We obtained these peaks by converting the current signals, scanned off the chip at a rate of 250 Hz, into voltages. The offset of the *I-V* conversion was 3.7 V, and the conversion factor was –1 V per 50 nA.

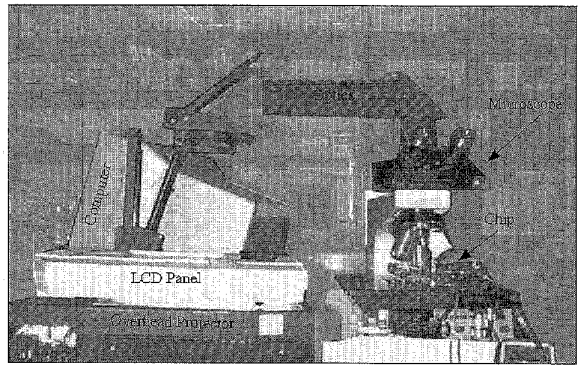


Figure 13. Test setup for analyzing analog VLSI devices with on-chip photoreceptors. A computer generates optical stimuli and displays them on an LCD panel. An overhead projector and a periscope-like structure containing lenses and mirrors image the display into the tube of a microscope. The microscope optics project the image in the tube onto the chip surface.

tation simultaneously. In a resistive fuse network,[10,12] discontinuities tend to persist so that for a given input stimulus, multiple stable states may exist depending on the stimulus' history. However, a feed-forward network like the one we describe would always yield a well-defined, unique solution.

Because the discontinuity signals are analog in nature, we could use them to vary the local resistance of the network so that the smoothing would gradually decrease with increasing velocity difference. This would correspond to a weighting of smoothing and segmentation depending on the confidence level for the presence of an object boundary. If the velocity differences to be segmented are significantly above the offset and noise levels, binary switches could be used in conjunction with a low-resistivity network. This would allow the chip to measure velocities of the different objects in the image robustly through averaging, and determine their boundaries accurately at the same time. Work is in progress to implement such schemes on chip.

Testing

An important step in the process of designing a working analog VLSI device is testing. Even if it is not often mentioned in research articles, this step can easily take up to 90% of the design process effort. This is especially true in the case of analog circuits that employ on-chip photoreceptors. When prototyping or debugging an analog circuit, it would be nice to keep the (low-power, compact, portable) device on a bench and connect it to measuring equipment such as oscilloscopes and current and voltage meters. Consequently, for the devices we describe in this article, one of the most challenging problems was generating motion stimuli that would give rise to the desired optical-flow fields while we kept the chips still.

Specifically, for the FOE device we wanted to simulate ego-motion in a fixed environment; for the time-to-contact device, to simulate translation towards a fixed plane at a controlled velocity; and for the motion-discontinuity device, to simulate backgrounds moving at different velocities from objects in the foreground.

We solved all of these problems using a DC motor connected to a rotating drum, to which we attached different types of figure drawings. To simulate ego-motion we imaged expanding edges onto the 1D photoreceptor array by placing a drawing of a V-shaped curve on the rotating drum. In this way, the 1D array would see sections of the V-shaped curve, starting from the bottom (edges at the center) all the way to the top (edges at the periphery).

To simulate translation towards an obstacle, we attached to the base of the rotating drum a drawing of a spiral and imaged the center of the rotating spiral inside the circle of concentric photoreceptors described earlier.

We generated motion discontinuity stimuli by using high-contrast edges translating at a controlled velocity as background, and moving dark bars at a different velocity in the foreground (between the rotating drum and the lenses mounted on the chip).

This type of testing methodology does not always allow us to characterize the properties of the designed circuits precisely. We intended the tests performed in this way to demonstrate qualitatively the correct behavior of each device. To generate well-controlled stimuli and provide input signals with single-photoreceptor resolution, we designed the apparatus shown in Figure 13. A computer generates optical stimuli. An overhead projector and some additional optics image these stimuli from an LCD panel into the camera tube of a microscope. The microscope optics project this intermediate image onto the chip surface. Through the microscope we can monitor the focus and position of the image with respect to the surface. This system enables us to selectively illuminate circuit parts to be analyzed with a well-defined stimulus.

The resolution of the stimulus pattern depends on the magnification of the microscope objective. With a 50× objective, we obtain resolution on the order of 1 μm in the focal plane, which is below the minimum feature size on the chip. Since the system has no moving parts, its mechanical stability is excellent and does not limit performance.

Despite the fact that this methodology gives us much more control over the stimulus and its position on the chip, it also has its disadvantages. There are two main problems: the artificial flicker introduced by the computer signal and the overhead projector, and the fact that we are unable to generate high-contrast stimuli with the LCD panel. Nonetheless, this methodology complemented by the rotating drum let us identify the circuits that were either causing errors in the systems or limiting their performances, and make the necessary corrections to obtain the devices we have described.

OUR THREE ANALOG VLSI ARCHITECTURES use robust, elementary velocity sensors to selectively integrate features of the optical-flow field for detecting focus of expansion, time to contact, and motion discontinuities. By choosing applications that rely on integrative properties of the optical flow, we demonstrated how to use these compact, low-power, smart-vision chips for stand-alone applications. We have had the circuits fabricated in old, low-cost, 2-micron

VLSI technology. With more aggressive technologies, we could enhance the performance of the proposed architectures—by increasing the pixel resolution, for example—and apply them to industrial applications.

Acknowledgments

We developed the elementary velocity sensor used in these architectures in collaboration with Rahul Sarpeshkar. Pedro Melo assisted in the design of the optical test setup. Support for this work came from Daimler-Benz as well as from grants by the US Office of Naval Research, the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program, and the Office of Strategic Technology of the California Trade and Commerce Agency. MOSIS provided fabrication of the integrated circuits.

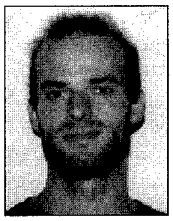
References

1. R. Sarpeshkar et al., "Analog VLSI Architectures for Motion Processing: From Fundamental Limits to System Applications," *Proc. IEEE*, Vol. 84, No. 7, July 1996, pp. 969-987.
2. J. Kramer, "Compact Integrated Motion Sensor with Three-Pixel Interaction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, April 1996, pp. 455-460.
3. J. Kramer, R. Sarpeshkar, and C. Koch, "An Analog VLSI Velocity Sensor," *Proc. Int'l Symp. Circuits and Systems*, IEEE, Piscataway, N.J., 1995, pp. 413-416.
4. G. Indiveri, J. Kramer, and C. Koch, "Analog VLSI Architecture for Computing Heading Direction," *IEEE Proc. Intelligent Vehicles*, IEEE, 1995, pp. 24-29.
5. J. Barron, D. Fleet, and S. Beauchemin, "Performance of Optical Flow Techniques," *Int'l J. Computer Vision*, Vol. 12, No. 1, 1994, pp. 43-77.
6. G. Indiveri, "Winner-Take-All Networks with Lateral Excitation," to appear in *J. Analog Integrated Circuits and Signal Processing*, 1996.
7. T. Poggio, A. Verri, and V. Torre, "Green Theorems and Qualitative Properties of the Optical Flow," Internal Lab. Memo 1289, MIT, Cambridge, Mass., 1991.
8. G. Indiveri, J. Kramer, and C. Koch, "Parallel Analog VLSI Architectures for Computation of Heading Direction and Time-to-Contact," *Advances in Neural Information Processing Systems*, Vol. 8, D. Touretzky, C. Mozer, and E. Hasselmo, eds., MIT Press, Cambridge, Mass., 1995, pp. 720-726.
9. A. Verri, M. Straforini, and V. Torre, "Computational Aspects of Motion Perception in Natural and Artificial Vision Systems," *Philosophical Trans. Royal Soc. London*, Vol. 337, No. 1282, Sept. 1992, pp. 429-443.
10. J. Kramer, R. Sarpeshkar, and C. Koch, "Analog VLSI Motion Discontinuity Detectors for Image Segmentation," *Proc. IEEE Int'l Symp. Circuits and Systems*, Vol. II, IEEE, 1996, pp. 620-623.
11. C. Koch, J. Marroquin, and A. Yuille, "Analog 'Neuronal' Networks in Early Vision," *Proc. Nat'l Acad. Science USA*, Vol. 83, 1986, pp. 4263-4267.
12. J. Harris, C. Koch, and J. Luo, "A Two-Dimensional Analog VLSI Circuit for Detecting Discontinuities in Early Vision," *Science*, Vol. 248, No. 4960, June 1990, pp. 1209-1211.



Giacomo Indiveri is a research fellow in the Department of Biology at the California Institute of Technology, Pasadena, California, where he works on the design of analog VLSI subthreshold neuromorphic architectures for low-level visual tasks and motion detection. He formerly held a fellowship in the National Research Program on Bioelectronic Technologies in the Department of Bioelectronic and Electronic Engineering at the University of Genova and at SGS-Thomson Microelectronics. His research interests are in the areas of neural computation, analog VLSI, and biological signal processing.

Indiveri received the Laurea degree in electrical engineering from the University of Genova, Italy.



Jörg Kramer is a postdoctoral fellow at the California Institute of Technology, working on analog VLSI vision systems. He graduated in physics from the Swiss Federal Institute of Technology Zurich (ETHZ), and obtained an MS degree in applied optics from Imperial College, London. He received a PhD degree in physics from ETHZ for a project in optoelectronics carried out at the Paul Scherrer Institute Zurich.



Christof Koch is a professor of computation and neural systems at the California Institute of Technology. His research focuses on understanding the biophysical mechanisms underlying information storage and processing in single neurons. In particular, he is interested in the computations underlying motion and visual attention in cortical networks in the mammalian visual system. His laboratory builds neuromorphic, analog, smart-vision chips to solve a host of applied vision problems. Together with Francis Crick, he works on the neuronal basis of visual awareness and consciousness.

Koch received his PhD in biophysics in Tübingen, Germany, and spent four years at MIT's Artificial Intelligence Laboratory before moving to Caltech. He has published three books and well over a hundred technical articles, and has numerous patents in the area of vision chips.

Direct questions about this article to Giacomo Indiveri, Institute of Neuroinformatics, Univ. of Zurich, Gloriastr. 32, 8006 Zurich, Switzerland; giacomo@neuroinf.ethz.ch.

Reader Interest Survey

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

Low 162 Medium 163 High 164