# System Implementations of Analog VLSI Velocity Sensors

Giacomo Indiveri, Jörg Kramer and Christof Koch
Computation and Neural Systems Program
California Institute of Technology
Pasadena, CA 91125, U.S.A.
E-mail: giacomo@klab.caltech.edu

## Abstract

*We present three different architectures that make use of analog VLSI velocity sensors for detecting the focus of expansion, time to contact and motion discontinuities respectively. For each of the architectures proposed we describe the functionality of their component modules and their principles of operation. Data measurements obtained from the VLSI chips developed demonstrate their correct performance and their limits of operation.*

## 1: Introduction

Analog velocity sensor circuits have been thoroughly investigated in the past years [19, 2, 1, 7, 4, 17, 5]. Nonetheless, researchers were unable to obtain a device that would simultaneously be compact, robust to background brightness level, insensitive to stimulus contrast and have a wide, unambiguous, range of speed selectivity. Recently novel velocity sensors that are sensitive to low contrast stimuli, independent of contrast (for intermediate and high contrast values), selective to over 3 orders of magnitude of velocity and over 2 orders of magnitude in light irradiance have been proposed [13, 11]. These sensors being extremely compact, we are at a point now in which we can actually integrate such sensors at a system level and apply them to real-time machine-vision applications that require special-purpose parallel hardware for computing motion across the entire image. Since analog circuits are limited by low precision in the values of their state variables, we target applications that rely on integrative features of the optical flow field rather than on the precise value of its vectors. Specifically, in this paper we present three different architectures that make use of the velocity sensor defined as *facilitate and sample* in [13] for focus of expansion, time to contact and motion discontinuity detection respectively.

## 2: Focus of Expansion

During observer motion through the environment, the velocity vectors of the optical flow field generated in an instant of pure translational motion are radial in nature and expand out from a point that corresponds to the direction of heading, also referred to as the focus of expansion (FOE) [6]. By choosing a particular application domain, such as for example the one of vehicle navigation, we can use *a-priori* information and make assumptions that simplify the problem of FOE detection in general cases. Specifically, for vehicle navigation, we can restrict our analysis to pure translational motion taking advantage of the fact that it is possible to compensate for the *rotational* component of motion using lateral accelerometer measurements from other sensors often already present on the vehicle. Furthermore, being interested in determining, and possibly controlling,

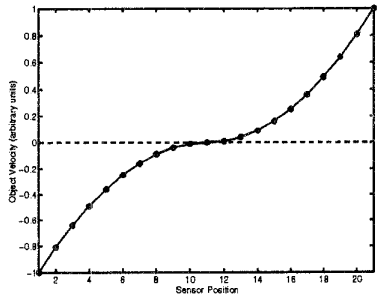



**Figure 1. Signed coding of simulated optical flow vectors of an object approaching an ideal observer at different positions in the array of velocity sensors. The FOE is centered on the $11^{th}$ sensor.**

1086-1947/96 $5.00 © 1996 IEEE
*Proceedings of MicroNeuro '96*

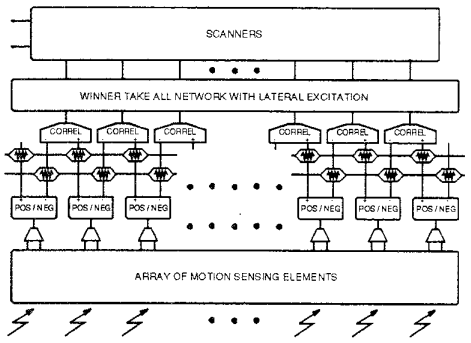



**Figure 2. Block diagram of the analog VLSI architecture for determining the FOE position for an observer translating in a fixed environment**

the heading direction mainly along the horizontal axis, we can greatly reduce the complexity of the problem by considering one-dimensional arrays of velocity sensors. In such a case, where only the horizontal component of the optical flow vectors, obtained from pure translational motion in a fixed environment, is measured, the problem of detecting the FOE reduces to detecting the point in which the optical flow vectors change direction. Coding such vectors with positive values for one direction and negative values for the opposite direction, the problem then is to detect the *zero-crossing* in the data array. Ideally, using this convention, the velocity vectors of an object translating toward an array of velocity sensors should yield a result as the one shown in Fig. 1. 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 $64 \times 64$ pixel silicon retina placed on a moving truck. Such simulations revealed that real image sequences may generate erroneous optical flow vectors that give rise to spurious zero-crossings, arising from noise in the input images, extreme lighting conditions, sparseness of the data or noise in the state variables (*e.g.* due to device mismatch) of the hardware implementation [9]. To account for such errors and select the zero-crossing corresponding to the correct FOE position we designed, simulated and implemented the architecture shown in Fig. 2. The input stage of this architecture is a 1-D array of 25 elementary velocity sensors. Each velocity sensor has a differential output with one terminal for the preferred direction of motion and the other for the non-preferred direction of motion, such that stimuli moving in the preferred direction will cause an output proportional to the speed of the stimulus on the first terminal and a null output on the second terminal and *vice versa* for stimuli moving in the opposite direction. This differential signal is then fed into a wide-range transconductance amplifier operated in the *subthreshold* domain. The output current of each amplifier will hence be proportional to the hyperbolic tangent of the differential voltage input [14]. At this stage the output current, which can be both positive or negative, is half-wave rectified so that the positive part of the current is copied into one branch of the architecture and the negative part is copied into the other branch. Having separated positive and negative parts of the currents, we can perform spatial smoothing on both parts using two separate resistive networks [3]. Following the smoothing stage there is the zero-crossing detection stage, which is implemented by using a "summing correlator" circuit: negative currents from one velocity sensor are inverted and fed into one terminal of the circuit and positive currents from the neighboring velocity sensor are fed in the other terminal of the circuit so that the co-presence of low negative currents from one unit and high positive currents from the neighboring unit is signaled. Since more than one zero-crossing could be detected at a given time the summing-correlator circuits are connected to a winner-take-all network with lateral excitation [15]. Lateral

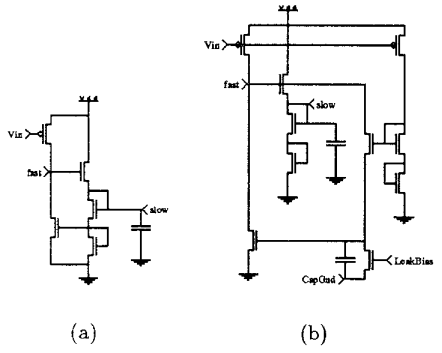



**Figure 3. Pulse-shaping circuits. (a) Original version. (b) Modified version. The modified version responds reliably also to extremely low speeds and can be used to measure optical flow vectors around the FOE.**

excitation facilitates units close to the chosen winner and inhibits units farther away. Thus the network selects the zero-crossing with maximum steepness closest to the previously selected unit. This operation is extremely useful since it accounts for the fact that gener-

ally the FOE doesn't jump abruptly from one part of the image to another but shifts smoothly in time.

The velocity sensors used as input stage of the architecture are based on a minor modification of the ones described in [12]. Specifically, the part of the circuit that has been modified is the one defined in [12] as the *pulse-shaping* circuit. The pulse-shaping circuit is

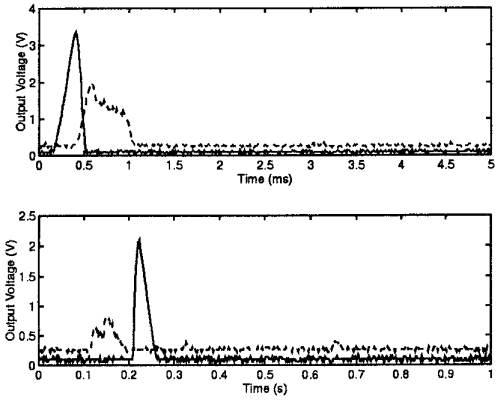
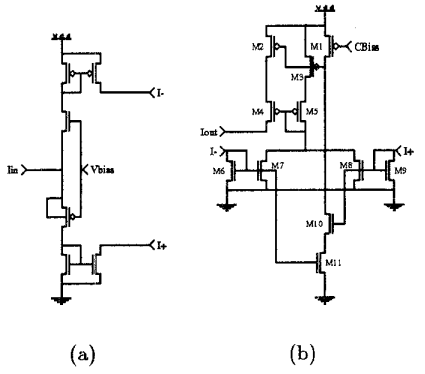



**Figure 4. Fast sampling pulses of the original pulse-shaping circuit (dashed line) and of its modified version (solid line). The top plot shows the response to a high-contrast edge imaged onto the chip through a lens with 13 mm focal length, moving at a speed (on chip) of 1.26 mm/s while the bottom plot shows the response to an edge moving at a speed of 16.8 $\mu$m/s.**

responsible for generating two types of pulses at the onset of a voltage pulse at its input node: the first one should be a slowly decaying pulse whereas the second one should be a sharp fast pulse used to sample the slowly decaying pulse generated from a neighboring velocity sensor. Since the original circuit is unable to reliably generate sharp and high enough fast pulses for extremely low velocities (lower than approximately $17\mu m/s$), we designed a new pulse-shaping circuit, using 5 additional transistors and a small capacitor (of $60fF$), able to respond also to such ranges of velocities. This operation was necessary because stimulus velocities close to the FOE position can be extremely low (see Fig. 1). To evidence the modifications made to the circuit, both its original and modified versions are shown in Fig. 3. Fig. 4 shows the shape of the fast pulses for both circuits for a typical stimulus velocity (in the top trace) and for an extremely low stimulus velocity (in the bottom trace). As shown, in the latter case the fast pulse of the original pulse-shaping circuit is low to a point that could be insufficient to activate the sample-and-hold circuit used to sample the slowly decaying pulse.




(a) (b)

**Figure 5. (a) Positive and negative half-wave rectifying circuit. (b) Summing correlator circuit.**

The remaining circuits used in the proposed architecture are very compact and operate in *current mode* (*i.e.* signals are represented as currents whereas voltages play only a minor role). The circuit used to rectify

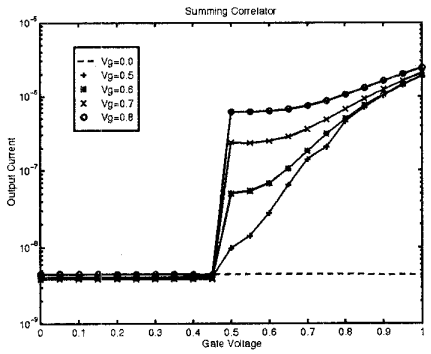



**Figure 6. Output current of "summing correlator" for a family of input voltages. The threshold voltage CBias was set to 0.75 V.**

the current generated from the wide-range transconductance amplifier that codes for the direction of motion is shown in Fig. 5(a). The resistive networks that implement the smoothing operation are implemented using one transistor per node operated in the subthreshold domain. The "summing correlator" circuit

that is responsible for the zero-crossing detection is shown in Fig. 5(b). This circuit behaves as an analog AND gate: if either one of two inputs is *zero* (*i.e.* below a set threshold) no current can flow through transistor M1, hence the output is switched off, whereas if both input currents are above the threshold, transistors M10 and M11 are turned on and the output is activated. The output current then corresponds to the sum of the two input currents. The threshold value that determines the minimum input current from both terminals necessary to activate the output can be controlled by changing the voltage on the node CBIAS. Fig. 6 shows the transfer characteristic of this circuit for a family of input pairs. The inputs were provided by changing the voltage values at nodes I+ and I- of Fig. 5(b). Since for transistors operated in the subthreshold domain there is an exponential dependence between gate voltage and drain current the output current of the correlator circuit is $I_{out} \approx e^{\alpha V_{G+}} + e^{\alpha V_{G-}}$, provided that both inputs are above a given threshold.

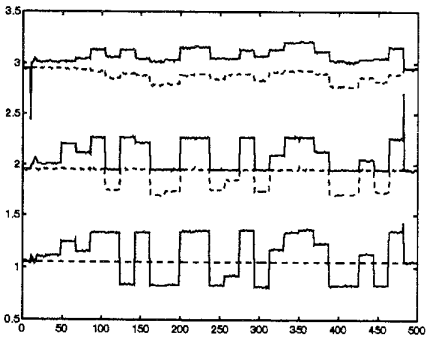


**Figure 7. Data obtained from measurements of the output current of the transconductance amplifiers, having provided a random moving stimulus as input (bottom trace), from the output of the half-wave rectifying circuits (middle trace) and from the resistive networks (top trace). The currents are scanned out and converted off chip with a linear current to voltage converter and displayed using different bias voltages as reference.**

To test the functionality of the proposed architecture we designed two different chips. The first chip contains the circuitry from the velocity sensor array up to the resistive networks (see Fig. 7), while the second chip has as input stage an array of transconductance amplifiers, with one terminal connected to a common line and the other to external potentiometers, and contains the circuitry up to the winner-take-all network. The top trace of Fig. 8 shows the output of the array of correlator circuits for the input configuration shown on the bottom trace. As shown, the output of the correlator circuits is proportional to the steepness of the zero-crossings for those cases in which the currents from neighboring units code for expanding stimuli (while contracting cases are neglected). This will allow the winner-take-all network to select the correct FOE position and, by means of the lateral excitation, lock on to it in time.

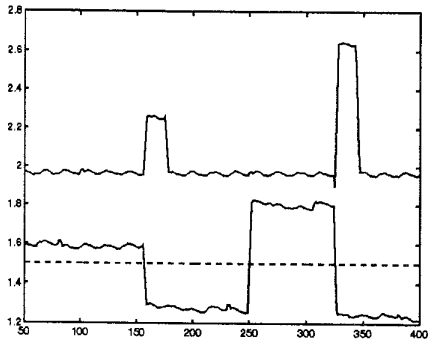


**Figure 8. Data obtained from measurements of the output current of the transconductance amplifiers, having set a desired input configuration, on the bottom trace, and from the output of the array of correlator circuits (with smoothing biases set to zero) on the top trace.**

## 3: Time to contact

Another quantity that can be extracted from an optical flow field obtained with analog VLSI velocity sensors is the *time to contact* with an object that partially or completely covers the visual field. The time to contact is defined as the time it would take the observer to collide with the object, if the relative velocity between observer and object would remain 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 translatory motion with a relative speed $v$ of the observer with respect the object, the time to contact $\tau$ is given by

$$\tau = \frac{d}{v}, \tag{1}$$

where $d$ 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 electro-physiological evidence supports the hypothesis that the time to contact is used to trigger landing responses in flies and birds and escape responses in a variety of animals (including humans) [21]. For translatory motion towards a planar

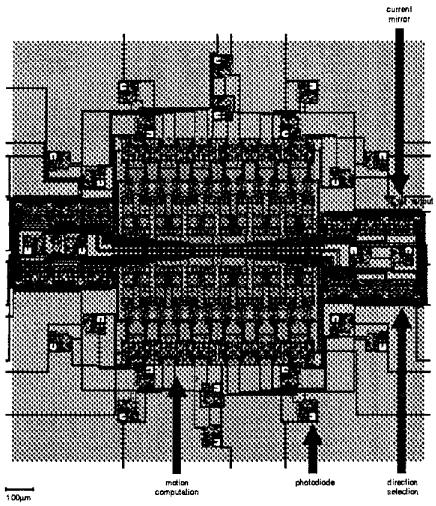



**Figure 9. Layout of an analog VLSI chip for the determination of time to contact. The 12 pairs of photoreceptors arranged on two concentric circles with radii of 400 $\mu$m resp. 600 $\mu$m are coupled to 12 velocity-sensing elements that estimate the radial components of the optical flow field. The pulse-shaping circuits and motion circuits are located in the central part of the chip and the direction-selection circuits are located on the left and right sides. Their output currents are summed for each direction (outward and inward) and subtracted (using a current mirror) to yield the final output that is inversely proportional to the time to contact. The size of the layout is 1.6 mm $\times$ 1.6 mm.**

surface perpendicular to the optical axis of the imaging system, the velocity field $\mathbf{V}$ in the image is linear and its divergence $\nabla \cdot \mathbf{V}$ is thus constant across the surface. Using the 2-D version of Gauss's divergence theorem the time to contact can then be estimated robustly from the line integral of the normal velocity component along a closed contour [16]. If the contour is a circle C of radius $r$, we obtain [18]:

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

where $\mathbf{n}$ denotes the unit normal vector along the contour. Note that the focus of expansion does not have to lie within the contour and that its position, the relative speed $v$ and the distance $d$ between object and observer do not have to be known. Furthermore, since eq. (2) depends on integrative rather than differential properties of the velocity field, the estimation of the time to contact is numerically stable, even in the presence of random noise and offsets, that are characteristic of subthreshold analog circuits.

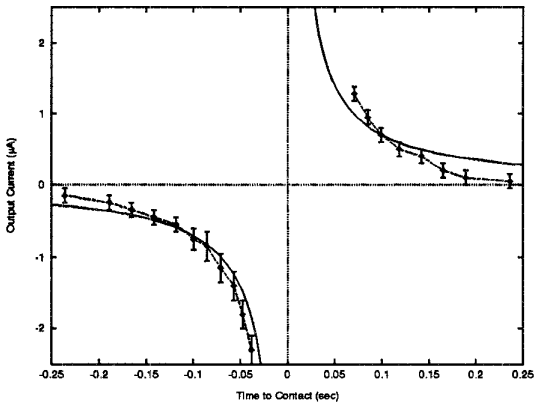



**Figure 10. Output current of the time-to-contact sensor as a function of simulated time to contact under incandescent room illumination. The theoretical fit predicts an inverse relationship.**

Using the above result, the time to contact can be estimated with a circuit consisting of an array of 1-D velocity-sensing elements arranged on a circle, such that each element measures radial velocity. The line integral is then approximated by the properly normalized sum of all sensor outputs, so that the time to contact amounts to

$$\tau = \frac{N \cdot r}{\sum_{k=1}^{N} V_k}, \qquad (3)$$

where $N$ 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. The layout of this chip is shown in Fig. 9. The photodiodes of the velocity-sensing elements are arranged on two concentric circles with radii of 400 $\mu$m and 600 $\mu$m respectively. The size of the layout is $(1.6\ \text{mm})^2$. The output voltage of each velocity-sensing element is used to control a subthreshold transistor current. Since this voltage is logarithmically-dependent on velocity, the current is proportional to velocity and the sum of the velocity components can be calculated by aggregating

the currents from all elements on two lines, one for outward motion and one for inward motion, and taking the difference of the total currents. The resulting bidirectional output current is then an inverse function of the signed time to contact.

The circuit yields reasonably accurate estimates of the time to contact for an approaching or receding pattern of high-contrast concentric rings centered on the focus of expansion and for a spiral stimulus on a rotating disk that simulates approaching or receding motion. Fig. 10 shows the averaged output current in response to the spiral stimulus as a function of simulated time to contact with a theoretical fit. The stimulus was projected onto the chip via a microscope lens under incandescent room illumination, such that the simulated focus of expansion was approximately centered with respect to the photodiode circles. The expected inverse relationship of output current and time to contact is qualitatively observed and the sign (expansion or contraction) is robustly encoded. However, the deviation of the output current from its average can be substantial. Since the output voltage of each velocity-sensing element gradually decays due to leak currents and since the spiral stimulus causes a serial update of the velocity values along the array, a step change in the output current is observed upon each update, followed by a slow decay. The effect is aggravated, if the individual elements measure significantly differing velocities. This is generally the case, because the focus of expansion is often not centered on the sensor and because of inaccuracies in the velocity measurements due to circuit offsets, noise, and the *aperture problem* [20]. Nonetheless, due to the integrative character of the algorithm, more robust results may be expected from stimuli with higher edge densities and arrays with a larger number of sensing elements, so that reasonable estimates for the time to contact in more general scenes should be obtained.

## 4: Motion discontinuities

Segmentation of images into different objects and a background is an important step in most image processing systems. In dynamic environments, objects can be segregated by their different apparent velocities with respect to the observer. These are either induced by motion parallax due to motion of the observer 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. Motion discontinuities can be extracted from an array of velocity-sensing elements by comparing the velocities measured by neighboring elements.

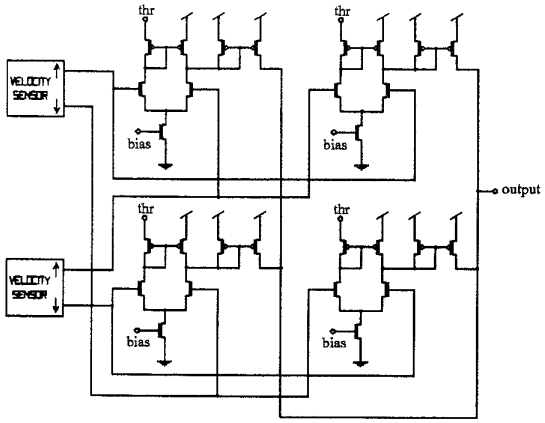



**Figure 11. Schematic diagram of the motion discontinuity detection circuitry for a pair of adjacent velocity-sensing elements. The absolute value of the difference of the output voltages is compared against a threshold (set by the *thr* bias voltage) for both directions of motion. If the difference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

We implemented a circuit that finds motion discontinuities in a one-dimensional image and outputs current signals at their locations. It uses a linear array of velocity-sensing elements as a front end. The voltage outputs of pairs of adjacent elements are compared for both directions of motion separately. If the absolute value of the voltage difference for either direction of a pair exceeds a set threshold, a current is activated at the location of the pair, signaling a discontinuity. The value of this current is a monotonic function of the speed difference, that saturates at large differences. If the voltage difference remains below the threshold, the current remains shut off. The threshold is set with a bias voltage to exceed the fixed-pattern and temporal noise of the velocity-sensing array for uniform image motion. For testing purposes, the output currents at the different locations along the array are scanned off the chip using an on-chip scanner, passed through a linear current-to-voltage converter, and displayed on an oscilloscope. The array contains 24 velocity-sensing elements with a pitch of 60 $\mu$m, giving 23 discontinuity measurements. The total size of the circuitry is 1.5 mm $\times$ 1.1 mm, as implemented with 2 $\mu$m technol-

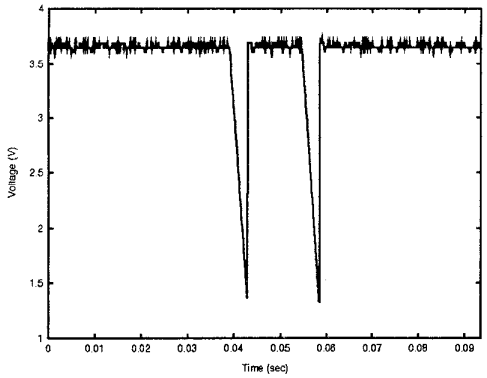



**Figure 12. Response of the motion discontinuity chip to a black bar stimulus traveling across a striped background, that moves in the same direction at a different velocity. The velocities on chip were 5 mm/sec for the bar and 30 mm/sec for the background. The voltage peaks on the scope trace show the locations of the bar's edges on the imaging array. They were obtained by converting the current signals, scanned off the chip at a rate of 500 Hz, into voltages. The offset of the I-V conversion is 3.7 V and the conversion factor is −1 V per 50 nA. No current is output at locations without motion discontinuities.**

ogy. Fig. 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 velocity difference was large enough to saturate the output currents at the edge locations. The data of Fig. 12 was taken with the bar moving in the same direction as the background, but the same type of signal was measured for opposite directions of motion of bar and background. For uniform image motion no output was observed.

If we consider a resistive network connecting the outputs of neighboring velocity-sensing elements to smooth out random noise and circuit offsets, we can use the motion discontinuity signals to diminish or eliminate the amount of smoothing across object boundaries by increasing the local resistance or by opening switches at their locations respectively [10, 8]. With such a feedback architecture it would be possible to implement smoothing and segmentation simultaneously. Due to the analog nature of the discontinuity signals, they could be used to vary the local resistance of the network in such a way that the smoothing gradually decreases 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 can be used in conjunction with a low-resistivity network, so that the velocities of the different objects in the image can be robustly measured through averaging, and their boundaries can be determined accurately at the the same time. Work is in progress to implement such schemes on chip.

## 5: Conclusions

We described three analog VLSI architectures that make use of robust elementary velocity sensors to selectively integrate features of the optical flow field for detecting the focus of expansion, time to contact and motion discontinuities. By choosing applications that rely on integrative properties of the optical flow, we demonstrated how it is possible to use these compact, low-power *smart-vision* chips for stand-alone applications. The chips developed have been fabricated using old and low-cost $2\mu m$ VLSI technology. By using more aggressive technologies it would be possible to enhance the performance of the architectures proposed (*e.g.* by increasing the pixel resolution) and apply them to industrial applications.

## Acknowledgments:

The elementary velocity sensor used in the architectures proposed was developed in collaboration with R. Sarpeshkar. This work was supported by Daimler-Benz as well as by grants from the Office of Naval Research, the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program, and by the Office of Strategic Technology of the California Trade and Commerce Agency. Fabrication of the integrated circuits was provided by MOSIS.

## References

[1] A. Andreou, K. Strohbehn, and R. Jenkins. Silicon retina for motion computation. In *Proc. Intnl. Symp. on Circ. and Systems ISCAS'91*, Singapore, 1991.

[2] R. Benson and T. Delbrück. Direction selective silicon retina that uses null inhibition. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 4*, pages 756–763, San Mateo, CA, 1991. Morgan Kaufmann.

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

[4] T. Delbrück. Silicon retina with correlation-based, velocity-tuned pixels. *IEEE Trans. Neural Net.*, 4:529–541, May 1993.

[5] R. Etienne-Cummings, S. Fernando, N. Takahashi, V. Shtonov, J. Van der Spiegel, and P. Mueller. A new temporal domain optical flow measurement technique for focal plane VLSI implementation. In M. Bayoumi, L. Davis, and K. ValavanisTouretzky, editors, *Proc. Comp. Arch. Machine Perception*, pages 241–250, 1993.

[6] J. Gibon, J. *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin, 1979.

[7] T. Horiuchi, W. Bair, B. Bishofberger, J. Lazzaro, and C. Koch. Computing motion using analog VLSI chips: an experimental comparison among different approaches. *International Journal of Computer Vision*, 8:203–216, 1992.

[8] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computers*, 21:52–63, 1988.

[9] G. Indiveri, J. Kramer, and C. Koch. Analog VLSI architecture for computing heading direction. In *IEEE Proc. Intelligent Vehicles, Detroit*, 1995.

[10] C. Koch, J. Marroquin, and A. Yuille. Analog 'neuronal' networks in early vision. In *Proc. National Academy of Science USA*, volume 83, pages 4263–4267, 1986.

[11] J. Kramer. Compact integrated motion sensor with three-pixel interaction. submitted for publication to IEEE Trans. Pattern Anal. Machine Intell., 1995.

[12] J. Kramer, R. Sarpeshkar, and C. Koch. An analog VLSI velocity sensor. In *Proc. Int. Symp. Circuit and Systems ISCAS '95*, pages 413–416, Seattle, WA, May 1995.

[13] J. Kramer, R. Sarpeshkar, and C. Koch. Pulse-based analog VLSI velocity sensors. submitted for publication to IEEE Trans. Circuits and Systems II, 1995.

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

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

[16] T. Poggio, A. Verri, and V. Torre. Green theorems and qualitative properties of the optical flow. Technical report, MIT, 1991. Internal Lab. Memo 1289.

[17] R. Sarpeshkar, W. Bair, and C. Koch. An analog VLSI chip for local velocity estimation based on reichardt's motion algorithm. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 781–788, San Mateo, CA, 1993. Morgan Kaufmann.

[18] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch. Analog VLSI architectures for motion processing: from fundamental limits to system applications. submitted for publication in *Proc. IEEE*, 1995.

[19] J. Tanner and C. Mead. An integrated analog optical motion sensor. In N. York, editor, *VLSI Signal Processing, II*, pages 59–76. IEEE Press, 1986.

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

[21] Y. Wang and B. J. Frost. Time to collision is signalled by neurons in the nucleus rotundus of pigeons. *Nature*, 356:236–238, 1992.