A Reconfigurable Neuromorphic VLSI Multi-Chip System Applied to Visual Motion Computation

Giacomo Indiveri, Adrian M. Whatley and Jörg Kramer
Institute of Neuroinformatics, University/ETH Zürich
[giacomo|amw|kramer]@ini.phys.ethz.ch

Abstract

We present a multi-chip neuromorphic system in which an Address Event Representation *is used for inter-chip communication. The system comprises an analog VLSI transient imager with adaptive photoreceptors, an analog VLSI motion receiver chip and a prototyping communication infrastructure which allows for programmability of connections between the elements on the two chips. We describe the properties of the two VLSI chips and of the communication infrastructure. To characterize the whole system, we present examples of connectivity tables which allow it to compute translational motion and expanding motion and show data from the transient detector array and motion receiver chips.*

1 Introduction

Recently several types of analog VLSI velocity sensors integrated on single substrates have been presented [1, 4, 5, 6, 8, 10, 15]. Unfortunately, due to the complexity of the circuits comprised in the elementary velocity sensors, all designs proposed suffer from the same problem: a trade-off between resolution of the imaging area and complexity of the computation carried out by the chip. To overcome the general problem of resolution versus complexity of the circuits, one can resort to multi-chip systems [7]. For such systems the limited number of the chips input/output ports requires time multiplexing of the communication channels. If the vision chips transmitting data have sparse pixel activity, as is often the case with adaptive silicon retinas [9], sequential multiplexing techniques make inefficient use of the communication channel bandwidth [13]. Asynchronous event-based communication, on the other hand, represents a very efficient communication protocol for multi-chip neuromorphic systems [2, 11]. The most popular representation used for such systems is the *Address Event Representation* (AER). Events in this representation are stereotyped digital pulses, whereas the interval between events is continuous. Each event is represented by a digital word encoding the address of the sending node that has been activated. The address of an event is broadcast during the pulse to all the receiving elements linked to it. For example, for an AER silicon retina, sender events are digital pulses that encode the position of the pixel and that carry the analog nature of the signal in their temporal pattern.

This type of representation lends itself naturally to multi-chip systems for token based motion algorithms: rather than measuring the time for a feature (token) to travel across pixels on the same chip, the interval between events generated by neighboring pixels on a sender array can be measured by a receiver chip. In this paper we present a system which exploits such a representation to compute the motion of edges between pixels and spatially integrate the measured values. The system we constructed comprises an AER transient imager, sending events from each pixel location in response to fast temporal brightness changes, a motion receiver chip, measuring delays between events arriving at neighboring units, and the so called *silicon cortex* (SCX), a fully arbitered address-event communication infrastructure that can be used to test inter-chip communication in simple neuromorphic systems [3].

2 The Transient Imager

A 16×16 array of irradiance transient detectors has been constructed to generate the events driving the system. The circuit responds with digital pulses in real time to local changes of a brightness distribution projected onto its surface. These pulses latch binary addresses encoding the positions of the activated pixels onto a bus. Fig. 1 shows a block diagram of the irradiance transient detector located at each pixel position. It consists of an adaptive photoreceptor [4] with a rectifying temporal differentiator [10] in the feedback loop and the interface to the communication circuitry. A photosensor converts the optical radiation into an electrical current signal i. This current signal is logarithmically mapped onto a voltage signal v. A feedback loop with a high gain G_o virtually clamps the voltage v_p on the

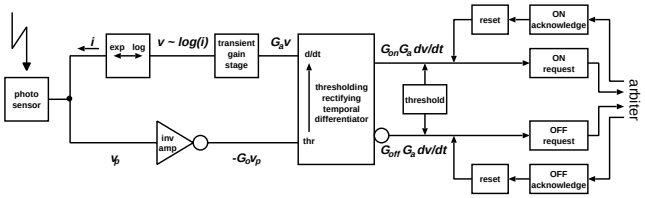


Figure 1. Block diagram of irradiance transient detector with event-based communication interface.

photosensor node and makes the signal voltage v appear at the other terminal of the logarithmic element. A transient in the signal v on that terminal is linearly amplified by a factor $G_a << G_o$ onto an adaptive node across a transient gain stage. The resulting voltage is then slowly adapted back to the voltage of the feedback node. The adaptive node and the output of the high-gain amplifier are connected to a circuit element that differentiates the voltage on the adaptive node with respect to time. The differentiation is thresholded, such that noise from the photoreceptor and the slow adaptation of the voltage signal do not give rise to an output signal. In addition, the differentiator circuit rectifies the output signal, such that positive transients, corresponding to dark-to-bright or ON transitions, and negative transients, corresponding to bright-to-dark or OFF transitions, appear at different output terminals. ON and OFF responses are individually amplified with gain factors G_{on} and G_{off} respectively. If the resulting signals exceed a chosen threshold they generate digital pixel request pulses. By appropriately setting the respective gain factors and the threshold, the circuit can be made to respond only to ON or only to OFF transients or to both types of transients. The request pulses are passed through arbitration circuitry that determines the order of the requests from the different pixel locations and latches the corresponding addresses onto the bus in that order. Each request is reset by a binary signal triggered by an acknowledge pulse from the arbitration circuitry after the address encoding its terminal has been latched onto the bus. A new request can only be activated after the reset pulse has terminated. The duration of the reset pulse, and thus a refractory period, for each pixel can be set with a control voltage. Depending on the chosen refractory period and the duration of the irradiance transient the pixel responds with a single spike or a burst of spikes. In the present application, the irradiance transients are caused by edges of moving objects imaged onto the chip and the relative timing of the detection of these edges at neighboring locations on the chip is used by a receiver chip to compute the velocity of these edges. Since the receiver works more accurately if it only receives a single pulse per edge and location, the refractory period on the sender is set accordingly.

The pixels are arranged on a rectangular grid. The position of a pixel along a row is encoded with a 4-bit column address and its position along a column with a 4-bit row address. An additional address bit is used to distinguish between ON and OFF transients. The arbitration scheme is implemented with separate binary row and column arbiter trees, such that first a row is selected and then the position along the row [2]. The positions of the ON and OFF terminals for each pixel are interlaced along the rows. The row and column addresses are latched onto the bus as soon as they have been selected. A control bit, which is set high whenever a column address is selected, is used as a request signal informing the subsequent off-chip processing stage that an event has taken place and been selected. This request signal initiates a handshaking cycle with the off-chip processing stage (the SCX, in this case) in the course of which the address of the sending pixel is transmitted. Each pixel contains 34 transistors, including the communication interface, and has a size of $(89\mu\text{m})^2$, as implemented with a 1.2μm CMOS process.

We imaged onto the transient detector array dark bars on a light background moving parallel to one edge of the pixel array in the direction of increasing pixel coordinate X_{td}. The gain factors and the threshold were set so that the chip responded only to OFF edges. There was a general trend for longer inter-event times (lower velocities) to be recorded between pixels at the same Y_{td} with lower X_{td} coordinates, and shorter inter-event times (higher velocities) to be recorded between pixels at the same Y_{td} with higher X_{td} coordinates (Fig. 2 (a) and (c)). We attribute this to a slight misalignment of the chip with the stimulus, which was provided by a pattern of stripes on the surface of a drum revolving at constant angular velocity. Away from the center-line of the drum, the stripes appear to be moving with lower linear velocity; this is consistent with the longer inter-event times recorded where this part of the drum is imaged onto the chip at lower X_{td} values and shorter inter-event times where a more central region of the drum is imaged onto the chip at higher X_{td} values.

In the direction parallel to the imaged bar, all pixels at the same X_{td} responded at similar times Δt after the pixels in the preceding row (Fig. 2 (a) and (d)), as expected.

Fig. 2 (b) shows at the same scale as (a) the standard deviations recorded for the inter-event intervals. These were typically of the order of 2ms (about 10% of the mean inter-event interval), except for a few cases towards the low X_{td} edge of the chip such as for the inter-event interval recorded between pixels $(1, 11)$ and $(2, 11)$. In these cases, the second pixel of the pair often responded very quickly ($\sim 1\mu$s) after the first. In other cases, typically between $X_{td} = 0$ and $X_{td} = 1$, pixels responded out of sequence with respect to the direction of motion of the imaged bar. In these

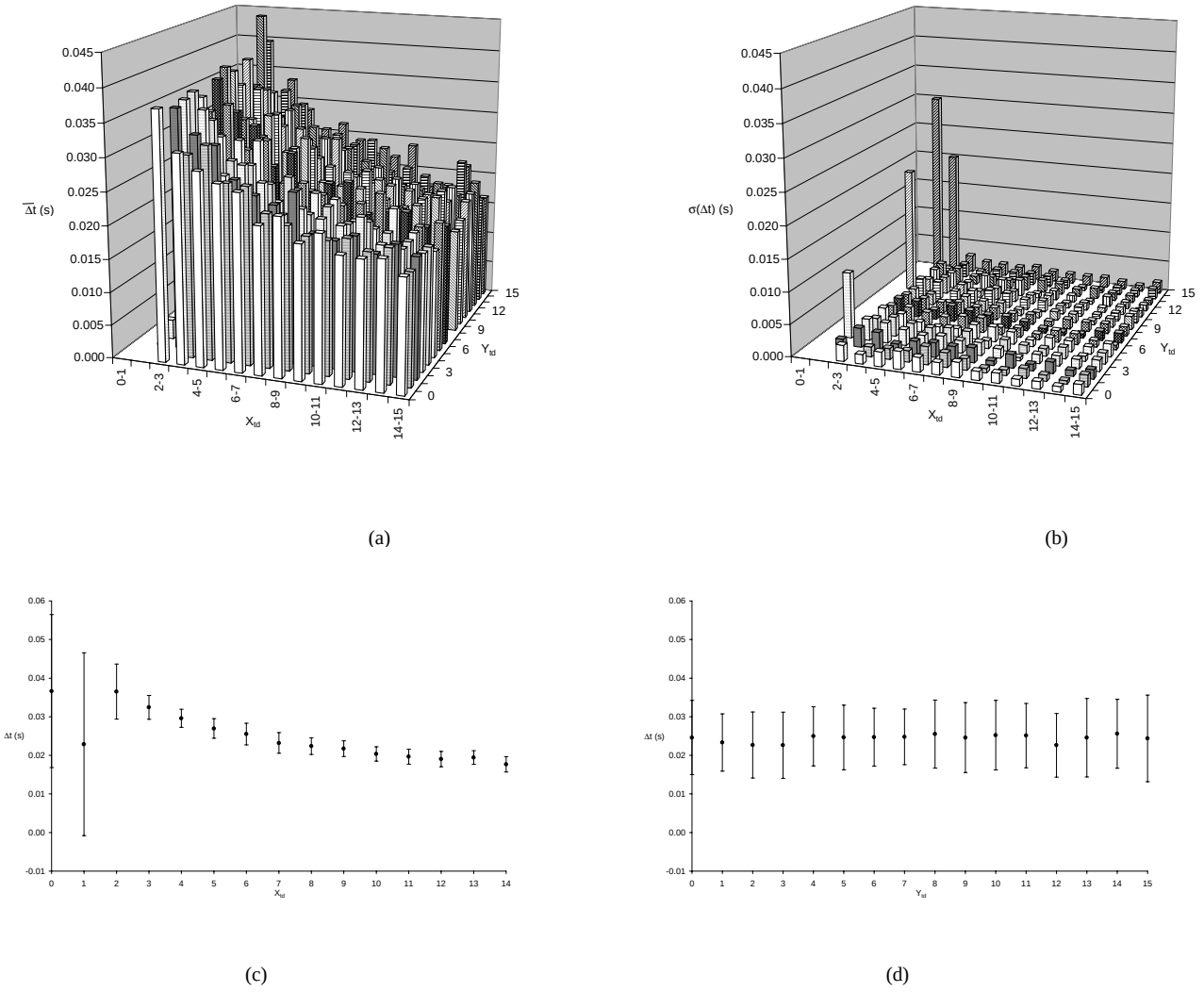


Figure 2. OFF response of the transient imager to a dark bar on a light background moving in the direction of increasing X coordinate. Data were taken for 16 sweeps of the image of the bar across the chip. (a) Mean, and (b) standard deviation of time Δt between the response from pixel (X_{td}, Y_{td}) and pixel $(X_{td}+1, Y_{td})$. No bar is drawn for pixels which failed to generate an response or which generated an out of sequence response during at least one sweep (e.g. pixel $(0,0)$). (c) Mean and standard deviation of time Δt between the response from pixel (X_{td}, Y_{td}) and pixel $(X_{td}+1, Y_{td})$ averaged over $Y_{td} = 0, \ldots, 15$ for each X_{td}. (d) Mean and standard deviation of time Δt between the response from pixel (X_{td}, Y_{td}) and pixel $(X_{td}+1, Y_{td})$ averaged over $X_{td} = 0, \ldots, 14$ for each Y_{td}.

latter cases, no bar is plotted. We attribute the unreliability of pixels at low X_{td} positions to fixed pattern noise effects, due to inhomogeneities in the silicon fabrication process.

3 The Motion Receiver

The motion receiver chip contains an array of 8×8 cells which measure delays between events arriving at neighboring units along the horizontal axis (see Fig. 3).

The circuits which carry out the motion computation are based on the *facilitate and sample* elementary velocity sensor [10]. A block diagram describing the architecture of such a sensor and its principle of operation is shown in Fig. 4. The transient detectors, E (within the dashed rectangle of Fig. 4(a)) have been substituted by the pixels of the AER sender, described in Section 2. The pulse shaping

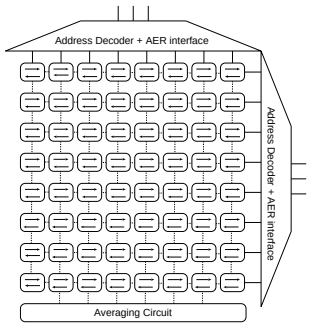


Figure 3. Architecture of the motion receiver chip. An array of 8×8 cells measures the delay between events arriving at neighboring units in the two directions along the horizontal axis. The velocity measurements are averaged along each column.

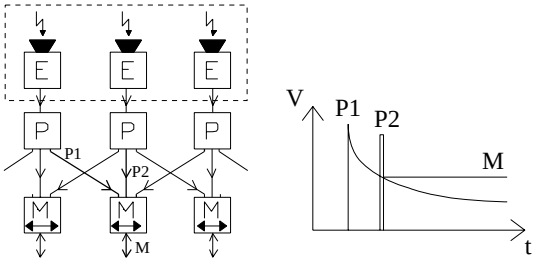
circuits P of Fig. 4(a) are connected to an interface which allows them to receive events from the AER bus and complete the handshaking cycle initiated by the off-chip processing stage (see Fig. 5). Any time an event reaches a pixel on the receiver chip, two pulses V_{fast} and V_{slow} are generated. The triggering of V_{fast} (P2 in Fig. 4)causes the motion circuits to compute both direction and amplitude of the stimulus velocity along one dimension. The direction of motion is computed by comparing the V_{slow} signals (P1 in Fig. 4) from left and right nearest neighbors and stored in the binary output voltage V_{dir} of a digital latch circuit (see top part of Fig. 6). The amplitude of the stimulus velocity is computed by sampling the voltage of the most recent V_{slow} pulse coming from the neighboring pixels and by converting it into a current I_{vel} (see bottom part of Fig. 6). The transistor generating the output current I_{vel} connected to the sample-and-hold circuit has a gate-to-source voltage difference $V_{gs} = (V_{slow} - V_{cgnd})$. It can be readily shown that, if this difference is below the transistor's threshold voltage, the following equation holds:

$$I_{vel} = C_0 {I_{slow}}^{(1+\frac{1}{\kappa})} \quad (1)$$

where C_0 is a proportionality constant, κ is the subthreshold slope coefficient [12], and I_{slow} is the drain current of the diode-connected transistor with gate voltage V_{slow} of Fig. 5.

The binary value of V_{dir} is used to gate the output current I_{vel}. In case of rightward motion, the output current is negative. In case of leftward motion, the output current is of opposite polarity.

This signed current representation allows us to easily average the motion signal over all the rows of the chip architecture (see Fig. 3) by simply connecting together the outputs of all the motion cells belonging to the same col-



(a) (b)

Figure 4. Facilitate and sample velocity sensor. (a) Transient detectors (E) generate current pulses in response to fast image brightness transients. Pulse-shaping circuits (P) convert each current pulse into two voltage signals: a slowly decaying pulse (P1) and a fast sampling pulse (P2). The fast pulse P2 of the central pixel is used by the motion circuit (M) to sample the decaying voltage signals P1 of its adjacent pixels. (b) The analog output voltage M of the motion circuit for rightward motion, equal to the voltage of the slowly-decaying pulse (P1) at the time of arrival of the fast sampling pulse (P2), encodes the stimulus velocity.

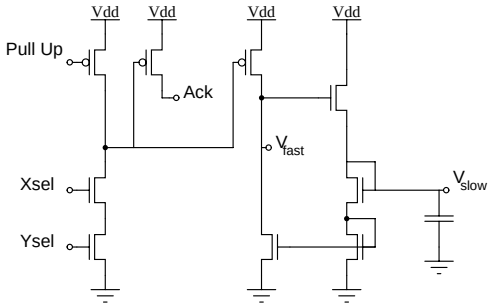


Figure 5. Pulse generating circuit with AER interfacing circuitry. The four transistors on the left implement the interface between the pulse shaping circuit and the AER signals. The five transistors on the right implement the pulse shaping circuit itself. The voltage signal labeled $\mathbf{V_{fast}}$ corresponds to the pulse P2 of Fig. 4 and the signal labeled $\mathbf{V_{slow}}$ corresponds to P1 of Fig. 4.

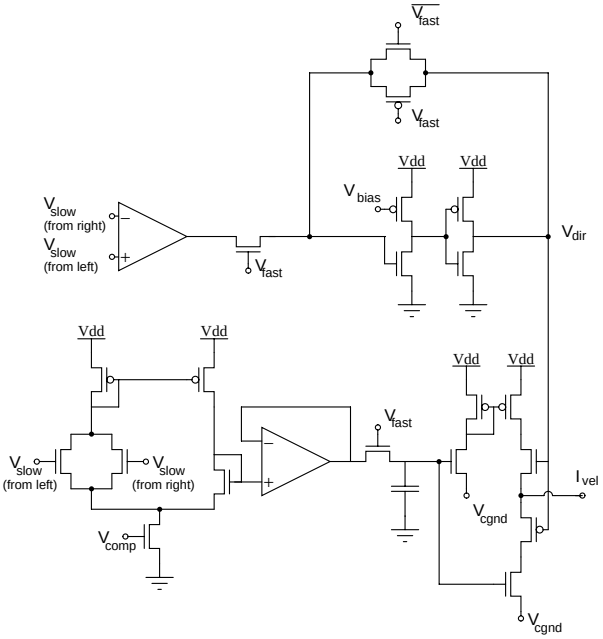


Figure 6. Direction selective circuit (upper part of the figure) and speed computation circuit (lower part of the figure). The value $\mathbf{V_{dir}}$ encodes direction of motion and $\mathbf{I_{vel}}$ its amplitude. $\mathbf{I_{vel}}$ is a positive (outward) or negative (inward) current depending on the direction of motion.

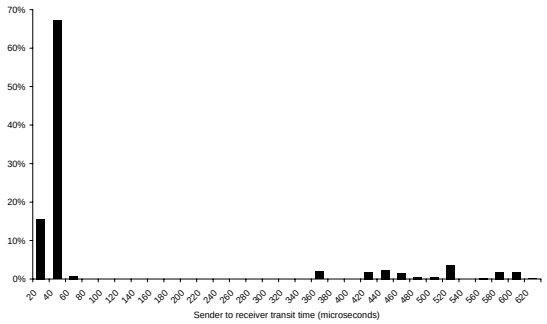


Figure 7. Histogram of transmission delays of AE addresses, from sender to receiver. The small proportion of transit times of 360 μs or more is present due to concurrent tasks using the SCX resources.

umn. The possibilities of integrating motion signals over many elements (given by the receiver chip), and of selectively choosing the spatial region over which to integrate (given by the SCX environment) lead to interesting applications, described in the following section.

4 The SCX Address-Event Communication Infrastructure

The SCX framework [3] is designed to be a flexible prototyping system, providing easily reprogrammable connectivity among on the order of 10^4 computational nodes such as silicon retina pixels and/or silicon neurons etc. spread across multiple chips on a single board, or more across multiple boards. This framework allows researchers to connect multiple sensors (sending devices) to multiple processing elements (receiving devices) and to create many different types of projective/receptive field mappings between them.

In the present work, we use the SCX to create a mapping between pixels on a transient detector array of the type described in section 2 and motion receiver elements on a chip of the type described in section 3 (see Fig. 8). Each detector pixel address may be mapped by the SCX to one or more addresses of receiving elements by specifying the appropriate connection table.

4.1 One-To-One Mapping

We wrote and downloaded to the SCX a connection table expressing a simple one-to-one mapping, from an 8×8 subregion in the center of the retinal plane to the 8×8 cells on the receiving chip. Table 1 illustrates the mapping being performed by the SCX.

In a larger sample of Address-Events (AEs) containing 4095 source AEs spanning just over nine seconds, 807 destination events were recorded, of which 84% transit from sender to receiver via the SCX connection table lookup process in under 60μs (see Fig. 7). The remaining 16% take between 360 and 620μs due to the microprocessor on the SCX running another task at the time the source AE arrives, such as the task responsible for communicating with the host PC.

If AEs generated at pixel locations on the sender array by visual stimuli have inter-event intervals significantly longer than the delays artifactually introduced by the SCX processing, the computations being carried out at the receiver side can be assumed to be correct. In our experiment, visual stimuli produce AEs with delays of the order of milliseconds (see Table 1). As there is at least one order of magnitude difference between the inter-event intervals generated by the input stimulus and the delays introduced by the SCX, the response of the motion receiver chip is substantially unaffected by the SCX processing artifacts. Fig. 9 shows data measured from the motion receiver chip in response to a

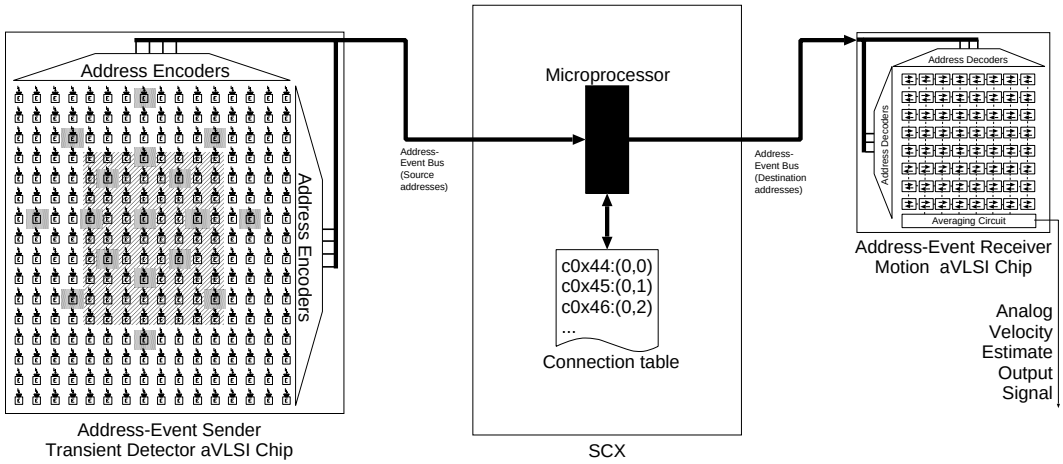


Figure 8. Events generated by the pixels of the transient imager chip are encoded as Address-Events (AEs) and transmitted to the SCX. A microprocessor onboard the SCX performs a table-lookup operation using a connection table downloaded from a PC to map the incoming source addresses to the desired destination AEs. These are then read and decoded by the receiver chip to form the pulse inputs to its individual velocity computation elements. The hatched area shows the central 8×8 sub-region of the sender chip from which we have chosen to map events to the receiver elements in a simple one-to-one mapping example, using the connection table, the first three elements of which are shown. The other shaded pixels are those we used for radial velocity computation, simply by changing the connection table.

moving stimulus being projected onto the transient detector sender array, for a simple one-to-one mapping (see also Fig. 8). For this particular example, the system computes the average velocity of edges along the horizontal axis. The measured current is monotonic with velocity and, for most of its output range, it can be fitted with equation 1, as expected by theory (see Fig. 9(b)).

By using a larger AE address space we can map the same AEs from the sender array to multiple instances of the same type of motion receiver chip. Depending on the nature of the connection table, we would thus be able to compute, in parallel, motion along different directions.

4.2 Radial Mapping

An other useful type of motion computation can be obtained by designing a connection table such that the receiver chip computes radial motion. The computation of radial motion along a closed contour can be used to determine the *time-to-contact* (τ) [14]. It has been shown that, for translational motion toward a planar surface at a constant velocity:

$$\tau = \frac{2\pi r^2}{\int_C \mathbf{V} \cdot \mathbf{n} ds} \tag{2}$$

where r is the radius of the circle C over which the line integral is computed, $\mathbf{V}$ the velocity field, and $\mathbf{n}$ the unit normal vector along the circle.

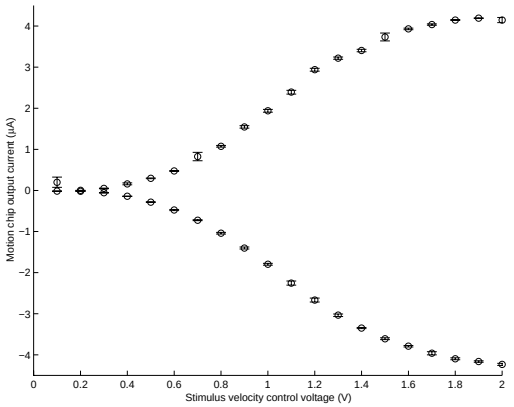
To generate the instantaneous optical flow field of an approaching surface moving at constant velocity, we imaged onto the transient detector array expanding circles with velocities increasing linearly with radius. The connection table we used has the following characteristics: pixels organized along radii of 8 different orientations are mapped to neighbouring motion cells of 8 different rows. We minimized the geometric distortion introduced by attempting to fit circles onto a square grid by using a spacing of three pixels along the horizontal and vertical radii of the sender array and a spacing of two pixels along the diagonals (see dark shaded individual pixels on the sender array in Fig. 8). The AE generated at the center of the sender array is mapped to the first motion cell of all 8 rows of the receiver.

When this connection table is downloaded to the SCX by issuing a command at the host PC, and without any modification to the hardware, the motion cells on the receiver chip respond to expansion or contraction of the optical flow field. Averaging the motion signal over the rows of the receiver chip results in approximating the integral $\int_C \mathbf{V} \cdot \mathbf{n}\, ds$.

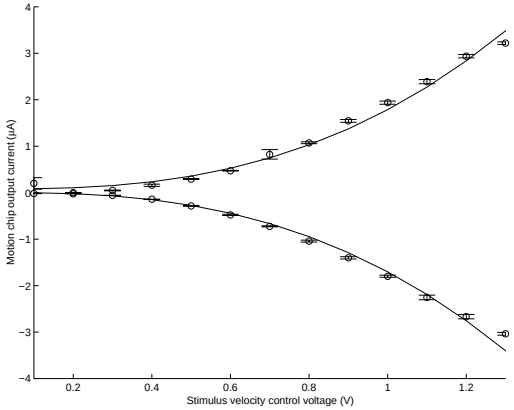
If we furthermore consider N concentric circles on the sender chip, each mapping onto a column on the receiver chip, and sum over columns, we approximate the sum of

t/ms	X_{td}	Y_{td}	$t_{trans}/\mu s$	X_{mot}	Y_{mot}
0.00	10	3		Not connected	
0.31	10	1		Not connected	
1.03	10	6	414.85	6	2
1.12	10	0		Not connected	
1.14	10	2		Not connected	
1.25	10	15		Not connected	
1.42	10	9	28.67	6	5
1.44	10	12		Not connected	
1.47	10	4	26.64	6	0
1.48	10	14		Not connected	
1.76	10	11	27.49	6	7
1.91	10	5		Not connected	
2.43	10	8	18.43	6	4
2.67	10	13		Not connected	
2.93	10	10	20.19	6	6
3.19	10	7	17.61	6	3
20.78	11	1		Not connected	
21.08	11	3		Not connected	
21.37	11	2		Not connected	
21.60	11	4	18.26	7	0
21.75	11	11	18.06	7	7
21.84	11	0		Not connected	
21.84	11	5		Not connected	
22.15	11	14		Not connected	
22.16	11	12		Not connected	
22.25	11	8	19.52	7	4
22.29	11	13		Not connected	
22.45	11	6	347.80	7	2
22.81	11	7	22.89	7	3
22.83	11	9	19.92	7	5
22.84	11	10	23.65	7	6
23.80	11	15		Not connected	
40.81	12	13		Not connected	
40.91	12	1		Not connected	
41.24	12	11		Not connected	
41.27	12	2		Not connected	
41.44	12	5		Not connected	

Table 1. 41.5 ms sample of Address-Event (AE) data taken from the one-to-one mapping case. The AE emitted by the transient detector sender chip at time t is shown decomposed into $\mathbf{X_{td}}$ and $\mathbf{Y_{td}}$ components. The source AEs from the central 8×8 sub-region of the sender are translated by the SCX into destination AEs ($\mathbf{X_{mot}}, \mathbf{Y_{mot}}$) and these are transmitted to the motion receiver chip, arriving at time $\mathbf{t_{trans}}$ after the corresponding source address was emitted from the sender. AEs from the periphery of the sender are discarded by the SCX, and are marked 'Not connected'. Destination AEs with $\mathbf{Y_{mot}}$ = 1 are also deliberately not being transmitted in this example and hence the corresponding source AEs (with $\mathbf{Y_{td}}$ = 5) are also marked as 'Not connected'.



(a)



(b)

Figure 9. (a) Output of the receiver chip for the one-to-one mapping connection table in response to a bar drifting at different velocities along the horizontal axis. Currents of opposite polarity correspond to motion in opposite directions. Each data point corresponds to the average over ten successive measurements; (b) Fit of part of the data to equation 1 with $\mathbf{C_0 = \pm\ 1.706e^{-6}}$ ($\mathbf{A^{-1/\kappa}}$) and $\kappa = 0.6131$.

integrals:

$$\sum_{k=1}^{N} \int_{C_k} \mathbf{V} \cdot \mathbf{n}\, ds\,,$$

where k represents the kth column on the receiver chip and C_k the kth concentric circle on the sender chip. This measurement will remain inversely proportional to the time-to-contact, as long as the response of the motion cells is linear with stimulus velocity. Fig. 10 shows such a measurement for simulated approaching objects at different times-

to-contact, with $N = 2$. As measured velocities become bigger, the measurements saturate (see also Fig. 9(a)) and the data departs from the theoretical prediction. Nonetheless, the integrative properties of the computation carried out make it robust to noise both in the input signal and in the system components. The possibility of considering more than 8 orientations and/or more concentric circles (*e.g.* by using more elaborate connection tables) would allow for improvements in the computation of time-to-contact.

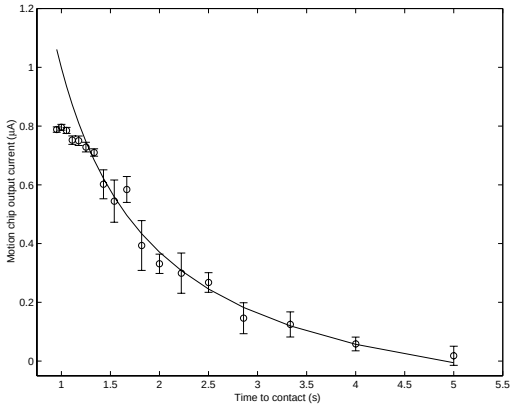


Figure 10. Output current of the motion receiver chip as a function of simulated time-to-contact. The theoretical fit predicting an inverse relationship (solid line) matches the data in the region where the response of the motion cells is given by equation (1).

5 Conclusions

In this paper we have shown how it is possible to use the SCX to interconnect an address-event transient imager to motion cells on a receiver chip in order to compute different properties of the optical flow field. We programmed the system to measure simple translational motion or expanding motion. By programming appropriate connection tables it would furthermore be possible to measure other features of the optical flow field, such as its *axis of rotation*, or its *focus of expansion*. These measurements could be done in parallel by multiple instances of the same type of receiver chip, each connected to the sender chip via the proper connection table.

Acknowledgments

This work was supported by the Swiss National Science Foundation SPP Grant and the U.S. Office of Naval Research. Fabrication of the integrated circuits was provided by MOSIS. The SCX PCB was designed and produced by S. R. Deiss of Applied Neurodynamics, and the SCX project has additionally been supported by the Gatsby Charitable Foundation and RM plc.

References

[1] A. Andreou, K. Strohbehn, and R. Jenkins. Silicon retina for motion computation. In *Proc. 1991 IEEE Int. Symp. on Circuits and Systems*, Singapore, 1991.

[2] K. Boahen. A retinomorphic vision system. *IEEE Micro*, 16(5):30–39, Oct. 1996.

[3] S. R. Deiss, R. J. Douglas, and A. M. Whatley. A pulse-coded communications infrastructure for neuromorphic systems. In W. Maass and C. M. Bishop, editors, *Pulsed Neural Networks*, chapter 6, pages 157–178. MIT Press, 1998.

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

[5] R. A. Deutschmann, C. Higgins, and C. Koch. Real-time analog VLSI sensors for 2-D direction of motion. In *Artificial Neural Networks - ICANN'97*, volume 1327 of *Lecture Notes in Computer Science*, pages 1163–1168. Springer Verlag, 1997.

[6] 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. Valavanis, editors, *Proc. Comp. Arch. Machine Perception*, pages 241–250, 1993.

[7] C. Higgins and C. Koch. Multi-chip motion processing. In *1999 Conference on Advanced Research in VLSI*, Atlanta, GA, 1998. In Press.

[8] J. Kramer. Compact integrated motion sensor with three-pixel interaction. *IEEE Trans. Pattern Anal. Machine Intell.*, 18:455–460, 1996.

[9] J. Kramer and G. Indiveri. Neuromorphic vision sensors and preprocessors in system applications. In *Advanced Focal Plane Arrays and Electronic Cameras (AFPAEC'98)*, Zürich, Switzerland, May 1998.

[10] J. Kramer, R. Sarpeshkar, and C. Koch. Pulse-based analog VLSI velocity sensors. *IEEE Trans. on Circuit and Systems*, 44(2):86–101, Feb. 1997.

[11] M. Mahowald. *An Analog VLSI System for Stereoscopic Vision*. Kluwer, Boston, 1994.

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

[13] A. Mortara. A pulsed communication/computation framework for analog VLSI perceptive systems. In T. S. Lande, editor, *Neuromorphic Systems Engineering*, pages 217–228. Kluwer Academic, Norwell, MA, 1998.

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

[15] A. Stocker and R. Douglas. Computation of smooth optical flow in a feedback connected analog network. In *Advances in Neural Information Processing Systems*, volume 12, 1998. In Press.