An Integrated Analog Optical Motion Sensor

John Tanner and Carver Mead

California Institute of Technology
Computer Science Department, 256-80
Pasadena, California

1 Introduction

Future machines that interact flexibly with their environment must process raw sensory data and extract meaningful information from them. Vision is a valuable means of gathering a variety of information about the external environment. The extraction of motion in the visual field, although only a small part of vision processing, provides signals useful in tracking moving objects and gives clues about an object's extent and distance away.

This paper describes the theory and implementation of an integrated system that reports the uniform motion of a visual scene. We have built a VLSI circuit that reports the motion of an image focused directly on it. The chip contains an integrated photosensor array to sense the image and has closely coupled custom circuits to perform computation and data extraction.

The integrated optical motion detector was designed to use local analog image intensity information as much as possible to extract image motion. We combined a new high performance photosensor with analog computation elements and used a novel approach to extracting velocity information from a uniformly moving image. The new motion detector has a number of features that overcome the shortcomings of previous designs [1,2]:

- The continuous, non-clocked analog photosensor has been demonstrated to operate over more than four orders of magnitude of light intensity [3]; this is a much greater range than video camera based vision systems.
- The system analyzes information in analog light intensity variations in the image. Sharp edges can be utilized but are not required. The image contrast requirements are small.
- Local image gradients rather than global patterns are used; global notions such as object boundaries are not needed.
- The analog circuitry prevents the information loss inherent in thresholding or digitization.
- Temporal aliasing is avoided by continuous computation; there is no clocking or temporal sampling (there is spatial sampling).

2 A Two-Dimensional Analog Motion Detector

An algorithm for two-dimensional velocity detection must address the problem that arises from an inherent ambiguity between motions along the two axes. This ambiguity occurs when the field of view is limited, as is the view through an aperture. The *aperture problem* is well known for binary-valued images.

Figure 1 shows the view through a square aperture. A black-and-white image containing a single straight edge is moving at some velocity; the position of the edge at a later time is shown by the dashed line. The velocity cannot be uniquely determined from these two "snapshots". There is an infinite family of possible velocities, as illustrated by the arrows. We can view the image velocity components v_x and v_y as the x and y coordinates in a *velocity plane*. In this plane, the actual velocity of the image defines a point. The family of possible image velocities defines a *constraint line* in velocity space that has the same orientation as does the edge in physical space. To be consistent with the visual information from the local aperture, the actual velocity point is constrained to lie on the line in velocity space.

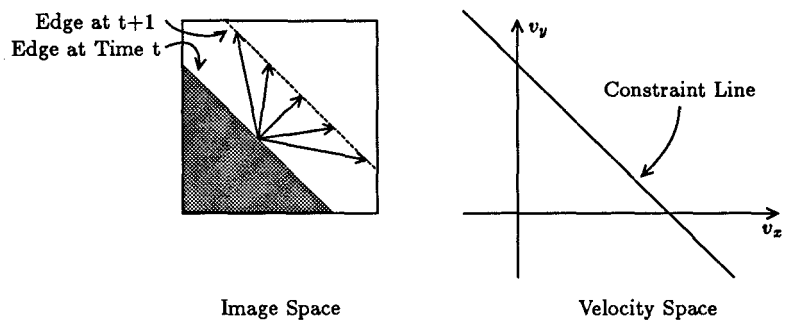

Figure 1: The aperture problem: local information is not sufficient to determine two-dimensional velocity uniquely. The family of possible velocities, denoted by the arrows in image space, define the constraint line in velocity space. The constraint line has the same orientation as does the edge in physical space.

Note that Figure 1 illustrates the ambiguity problem but does *not* depict the operation of our system. The velocity detector described in this paper (1) represents intensity values continuously—the images we consider are *not* just black and white; and represents time continuously—there is *no* notion of snapshots of the image or of clocking in this system.

Using analog values for intensities and gradients does not eliminate the ambiguity problem. Figure 2 shows the intensity plot of an image that contains gray-scale information and varies smoothly in intensity throughout. Brightness is plotted as height above the image plane for each point in the x-y image plane and is a function of the objects in view. As the image moves, the shape of the intensity surface will stay the same but will translate in the image plane. A stationary sensor will detect changes in intensity due to the movement of the image. The change in intensity with time, $\frac{\partial I}{\partial t}$, is a change in height of the intensity surface at a fixed point. The local sensor cannot tell whether

its upward or downward movement is due to motion of the intensity surface along only the x axis, only the y axis, or to motion along both axes. Two of these possibilities are shown as arrows in Figure 2. The inherent ambiguity cannot be resolved by strictly local information.

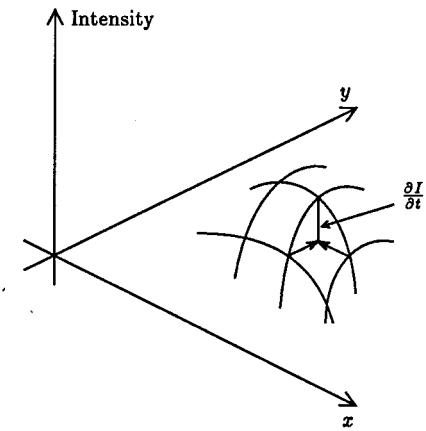

Figure 2: The intensity surface of a two-dimensional image. The change in intensity from a present high value to a future lower one, $\frac{\partial I}{\partial t}$, could be the result of many possible motions of the intensity surface. The arrows represent two possible motions.

Following Horn and Schunk [4], the expression that relates the intensity derivatives to the velocity is:

$$\frac{\partial I}{\partial t} = -\frac{\partial I}{\partial x}v_x - \frac{\partial I}{\partial y}v_y. \quad (1)$$

The intensity of Figure 2, a function of the two spatial variables x and y, has spatial derivatives $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$. The x and y components of velocity are v_x and v_y, and $\frac{\partial I}{\partial t}$ is the change in intensity with respect to time at a stationary observation point.

This equation shows that the values of three local derivatives of the intensity do not allow velocity to be uniquely determined; there is an inherent ambiguity.

The local intensity derivatives do provide some useful information—they constrain the possible values of the x and y components of velocity, just as the black-and-white images in the aperture problem constrained the velocity. Writing Equation 1 in the form of the line equation $Ax + By + C = 0$, we get:

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0.$$

Each local set of three derivatives defines a line in the velocity plane along which the actual velocity must lie. The slope of this constraint line is $-\frac{\partial I}{\partial x}/\frac{\partial I}{\partial y}$. If we view the gray-scale image as having a fuzzy "edge" with orientation perpendicular to the intensity gradient, the constraint line has the same orientation in velocity space as the "edge" has in physical space. This is the same orientation as the constraint line of the black-and-white image.

The aperture problem for binary-valued images is just a special case of the general two-dimensional velocity ambiguity. Local images, gray-scale or black-and-white, can provide only a family of possible velocities. This set of velocities can be represented by the coefficients of the equation for the constraint line.

It is much easier to determine the constraint line if the analog information is retained. For gray scale images, the coefficients of the constraint line equation are just the three partial derivatives that can be measured locally—$\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$. An image with continuous intensity values can be made into a black-and-white image by thresholding. Determining the orientation of the edge of the binary-valued image (and so its constraint line) is a more global problem of determining the boundary between black regions and white regions and fitting a line to the boundary. To determine the velocity constraint line, it is much easier to measure the coefficients locally than to throw away the information and then try to reconstruct it with a global process.

The ambiguity of a single local set of measurements can be resolved by using another set of local values from a nearby location. These values define another line in the velocity plane. The intersection of these two lines uniquely determines the actual velocity (Figure 3).

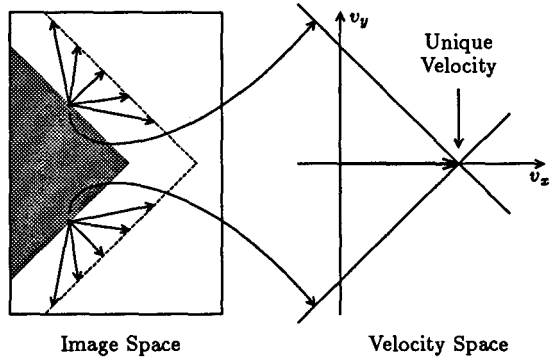

Figure 3: Uniquely determining velocity by the intersection of constraint lines.

2.1 Solution of Simultaneous Constraints

In practice, to find the actual velocity, we use the constraint contributions from all sites on the sensor array. Using only a small number of sites that are close together relative to an object size results in a few constraining lines in the velocity plane that are nearly parallel. A small error in any of the derivatives or in the constraint solver can then result in a large error in the computed velocity. Errors will be kept to a minimum when two lines in the velocity plane cross at right angles. This intersection occurs when there are contributions from two sites on edges that are perpendicular. An "edge" in this case is used loosely to mean a line perpendicular to the direction of greatest intensity change. Taking contributions from a large number of sites will then insure that we have pairs of orthogonal constraints for any reasonable image.

The barber-pole is a well-known example of an illusion that occurs because the orthogonality of constraint lines cannot be assured. In this illusion, the rotating cylinder produces a purely horizontal velocity. Our vision system erroneously reports "seeing" a vertical velocity. Images such as gratings and stripe patterns with intensity variations along only one axis cause this mistake. All constraint lines are coincident, so their intersection is not unique. It is not possible for man, beast, computer, or chip to disambiguate the motion of such a pattern.

In practice, there is no such thing as a perfect stripe pattern. The ability to correctly disambiguate velocity is thus a matter of degree. Our chip should reliably report the actual velocity unless the signals resulting from intensity variation along one axis lie below the noise level.

2.2 Constraint-Solving Circuits

Our constraint-solving circuit contains a set of global wires that distribute a best guess of velocity to all the individual constraint-generating sites (Figure 4). Each locale performs some computation to check whether the global velocity satisfies its constraint. If there is an error, circuitry within the local site then supplies a "force" that tends to move the global velocity toward satisfying the local constraint. The global velocity components are represented as analog voltages on the set of global wires. The correcting forces are currents that charge or discharge the global wires.

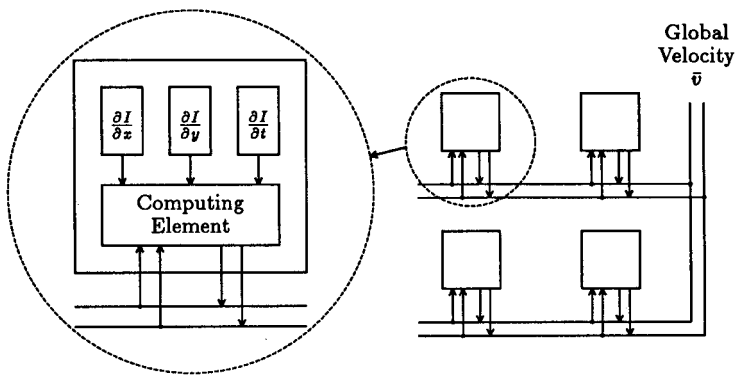

Figure 4: Block diagram of the constraint solver cell and array. The three locally derived derivatives, $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$, along with the present value of the globally distributed velocity components are inputs to a circuit that continuously computes a correction to the global velocities.

Finding the intersection of many lines is an over-constrained problem. Any errors will result in a region of intersection in which the real desired point is most likely to lie. To compute a most-probable intersection point (velocity), we must know what types of errors to expect, to define "most probable" and select on the basis of that definition a forcing function that varies with detected error. In the absence of rigor,

we can make reasonable guesses for the forcing function. It should be monotonic—the greater the error, the harder we should try to move it in the right direction. Because a forcing function linear in error distance is easiest to implement, we have selected it. The constraint solver minimizes the error by finding the least-squares fit of the velocity point to all the constraint lines.

The direction of the correction force should be perpendicular to the constraint line (Figure 5). Based on local information alone, the cell knows that the global velocity should lie on its locally defined constraint line, but it knows nothing about where along the line the global velocity lies. If a non-orthogonal component in the correction force were generated by the cell, an undue preference for the position of the global velocity on the constraint line would be expressed.

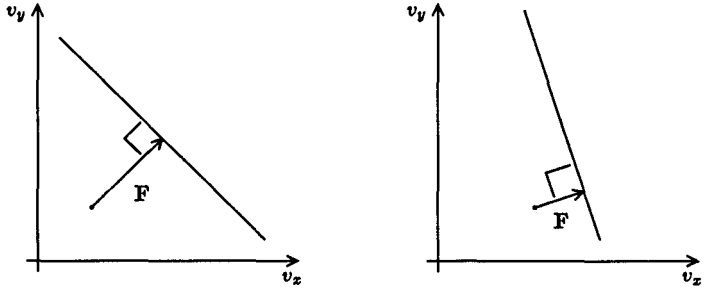

Figure 5: The correction force should be perpendicular to the constraint line in order to express only information obtained from the image.

We ensure orthogonality by using a direct perpendicular construction for the correction force. If we rearrange the constraint line equation $\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0$ from the implicit form $Ax + By + C = 0$ to the slope-intercept form $y = \mathrm{m}x + \mathrm{b}$, we get:

$$v_y = -\frac{\frac{\partial I}{\partial x}}{\frac{\partial I}{\partial y}} v_x - \frac{\frac{\partial I}{\partial t}}{\frac{\partial I}{\partial y}}.$$

The slope of the constraint line, where defined, is $\mathrm{m} = -\frac{\partial I}{\partial x}/\frac{\partial I}{\partial y}$, so the slope of the desired perpendicular correction force is then the negative reciprocal, $-\frac{1}{\mathrm{m}} = \frac{\partial I}{\partial y}/\frac{\partial I}{\partial x}$. A unit vector in this direction is:

$$\widehat{\Delta \mathbf{v}} = \frac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|} = \left\langle \frac{\frac{\partial I}{\partial x}}{\sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}}, \frac{\frac{\partial I}{\partial y}}{\sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}} \right\rangle.$$

The magnitude of the correction force should be greater if the present point in velocity space is farther away from the constraint line. The force should be zero for points lying on the constraint line. The direction of the force should always be perpendicular to the constraint line, and should have a sign such that the global velocity point will move toward the constraint line. A forcing function that is linear with error distance fulfills all these requirements and can be easily computed as:

$$D = \frac{\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t}}{\sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}}.$$

If we just substitute the values for the velocity components into the line equation and normalize by the quantity under the radical, we get D, a signed distance. The magnitude of D is the distance from the present velocity point $\langle v_x, v_y \rangle$ to the constraint line. The sign of D indicates on which side of the line the point is and therefore the direction of the correcting force. The vector, $\Delta\mathbf{v}$, from the current velocity to the point on the constraint line is then:

$$\begin{aligned}\Delta\mathbf{v} &= D \cdot \widehat{\Delta\mathbf{v}} = \left\langle D\frac{\frac{\partial I}{\partial x}}{\sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}}, D\frac{\frac{\partial I}{\partial y}}{\sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}} \right\rangle \\ &= \left\langle \frac{(\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial x}}{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}, \frac{(\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2} \right\rangle.\end{aligned}$$

Each cell should produce a force (electrical current), $\mathbf{F}$, that will tend to move the global velocity proportional to the detected error, $\Delta\mathbf{v}$. We would also like to scale this correcting force according to our confidence in the local data, C:

$$\mathbf{F} = C \cdot \Delta\mathbf{v}.$$

There is more information in a higher contrast edge, or at least there is a higher signal-to-noise ratio. A greater weight should be afforded to the correcting forces in those higher contrast areas. Our measure of contrast in the image is the intensity gradient, a vector quantity $\nabla\mathbf{I} = \langle \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \rangle$. Confidence is related to the magnitude of the gradient, $|\nabla\mathbf{I}| = (\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2)^{\frac{1}{2}}$. If we choose our confidence, C, to be the square of the magnitude of the intensity gradient, we have:

$$C = |\nabla\mathbf{I}|^2 = \frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2.$$

This choice greatly simplifies the correcting force calculation by canceling out the denominator. Our force equation becomes:

$$\mathbf{F} = \left\langle (\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial x}, (\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial y} \right\rangle.$$

Writing the two components of this vector equation separately we have:

$$\boxed{\begin{aligned}F_x &= (\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial x} \\ F_y &= (\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t})\frac{\partial I}{\partial y}.\end{aligned}} \quad (2)$$

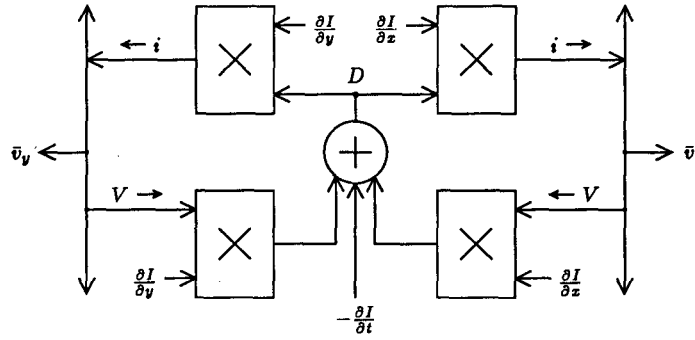

Figure 6: Block diagram for each cell's motion detection circuitry.

Analog computational elements within each cell of a two-dimensional array implement the calculations described by the above pair of equations. The block diagram of an implementation of the orthogonal two-dimensional formulation is shown in Figure 6.

The two-dimensional constraint solver can be viewed in terms of feedback. Each cell computes an error, the signed scalar quantity D, the distance in velocity space from the global average velocity to the locally known constraint line. This distance error is used as feedback to correct the system. It is multiplied by the appropriate two-dimensional vector perpendicular to the constraint line, to generate a correction force in the same direction to correct the global velocity vector.

The constraint solving system for two-dimensional motion detection is collective in nature. Local information, weighted by confidence, is aggregated to compute a global result. Each cell performs a simple calculation based on moving the global velocity state into closer agreement with its locally measured information. The collective behavior that emerges is the tracking of the intersection of constraint lines to solve the two-dimensional ambiguity, when possible, and report accurately the two-dimensional analog velocity of the image.

3 Results

The 8×8 array chip has been tested extensively by electronically simulating motion and by projecting actual moving images. The first set of experiments used an electronically controlled light source to apply an intensity field to the chip that varied spatially across the chip and varied with time. The space and time derivatives of intensity were controlled to simulate a moving intensity pattern while the velocity outputs from the chip were monitored. A second set of experiments focused actual images onto the chips and measured the chip's response. The constraint line behavior was verified and the correction forces were mapped for different images.

3.1 Characterization of the Motion Output

The motion simulation test setup is shown in Figure 7. Any changes in the light intensity were assumed to be due only to motion of the image, not to changes in the illumination level. Rapidly changing the illumination level under experimental control allowed the motion of a spatial intensity gradient to be simulated.

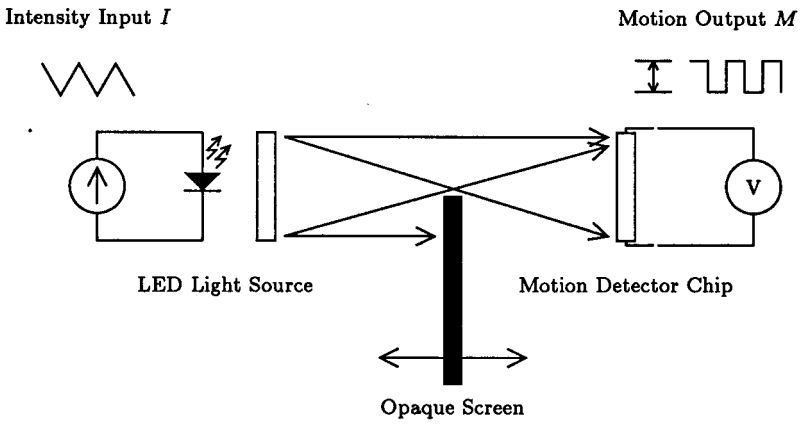

Figure 7: The test setup to simulate motion electronically. An LED cast light directly on the motion detector chip. Varying the LED current produced a controlled $\frac{\partial I}{\partial t}$. An opaque screen made a shadow edge on the chip. The distance from the chip to the screen controlled the sharpness of the edge, $\frac{\partial I}{\partial x}$.

A time derivative was generated by changing the current through the LED light source. A triangle wave intensity, used in these experiments, made a $\frac{\partial I}{\partial t}$ that is a square wave. The magnitude of the $\frac{\partial I}{\partial t}$ square wave is the slope of the triangle wave, which is dependent on the amplitude and frequency of the triangle wave. Frequency was used to vary $\frac{\partial I}{\partial t}$.

An opaque screen between the LED and the chip partially occluded the light and caused a spatial derivative of intensity (edge) to fall on the chip. Moving the screen closer to the chip made a greater $\frac{\partial I}{\partial x}$; that is, a sharper edge. The position of the screen was adjusted until the measured spatial gradient was the desired value.

When a spatial intensity gradient that varied in time as a triangle wave was applied to the motion detector chip, the differential voltage on the chip's velocity outputs was a square wave. Figure 8 shows an oscilloscope trace of the LED input current and the velocity output of the chip. For these experiments, the screen producing the spatial gradient was aligned with the y axis of the chip. As the intensities varied with time, the y component of velocity reported by the chip was very nearly zero.

With experimental control of $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial t}$, the velocity output of the chip was tested to verify that the reciprocal relationship for velocity, $v_x = -\frac{\partial I}{\partial t} / \frac{\partial I}{\partial x}$, held. This relationship

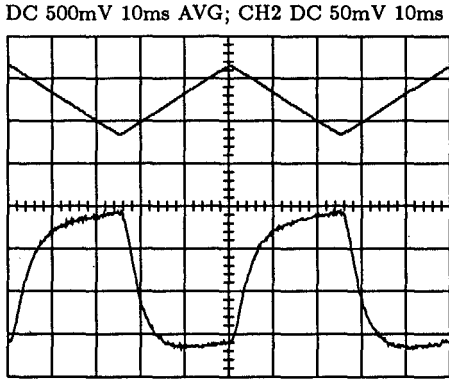

Figure 8: Oscilloscope traces of LED current (top) and reported velocity from the chip (bottom) for frequency of 20Hz.

comes from Equation 1 with v_y set to zero. Two sets of measurements were taken. First, reported velocity as a function of $\frac{\partial I}{\partial t}$ was plotted for fixed values of $\frac{\partial I}{\partial x}$; we expected a straight line graph. Second, the triangle wave frequency generating $\frac{\partial I}{\partial t}$ was held constant and the spatial gradient was varied; the expected curve was a hyperbola. For all plots, the reported velocity was the amplitude of the square wave of the x component output from the chip.

Figure 9 plots reported velocity versus $\frac{\partial I}{\partial t}$ (frequency) for three fixed values of $\frac{\partial I}{\partial x}$. The straight lines represent the theoretical proportional behavior, as shown on the log-log plot. The experimental results matched the predicted ones in the range from 1Hz to 40Hz. Beyond that frequency, the amplitude of the output rolled off.

Figure 10 is a plot of reported velocity on the vertical axis versus applied $\frac{\partial I}{\partial x}$ on the horizontal axis for three fixed values of $\frac{\partial I}{\partial t}$. The curves approximate a hyperbola over most of their range. The behavior to the left of the peaks deviates significantly from a hyperbola.

As $\frac{\partial I}{\partial x}$ decreases, the reported velocity should increase; the resistance of the circuitry will increase, however, causing it to have less effect on the reported velocity. When $\frac{\partial I}{\partial x}$ equals zero, the reported velocity could take on any value because it is not affected at all by the local cells. The current source loads on the v_x and v_y lines in our implementation have large but finite resistance so, in the absence of any information from the visual field, the reported velocity will tend toward zero. The schematic representing this effect is shown in Figure 11. For simplicity, we consider only the one-dimensional case, which can be derived from the two-dimensional case by setting $\frac{\partial I}{\partial y} = 0$. For large $\frac{\partial I}{\partial x}$ (contrast ratios), the resistance of the local circuits is much smaller than that of the loads, so the load's effect on reported velocity is negligible. As the contrast ratio is reduced, the load resistance must be taken into account.

So far we have referred to the output of the chip as the velocity v. We have now

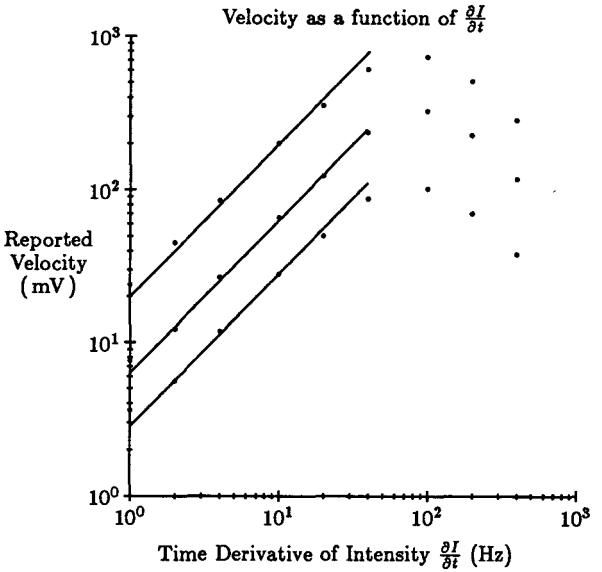

Figure 9: Measured motion response of the chip as a function of $\frac{\partial I}{\partial t}$ for fixed values of $\frac{\partial I}{\partial x}$. The straight lines represent an ideal linear response.

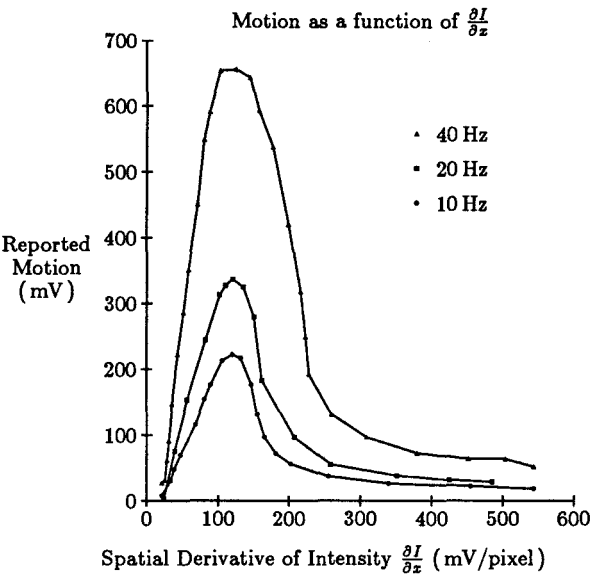

Figure 10: Measured motion response of the chip as a function of $\frac{\partial I}{\partial x}$ for fixed values of $\frac{\partial I}{\partial t}$. The response approximates a hyperbola over most of its range, as expected for velocity. Near zero, the response is linear with $\frac{\partial I}{\partial x}$.

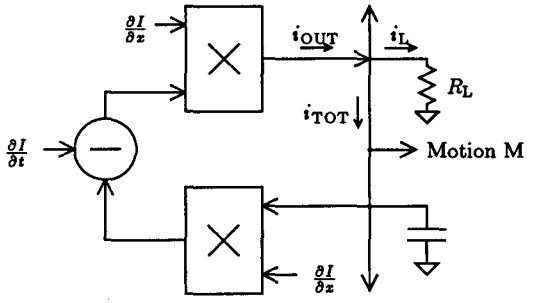

Figure 11: Schematic of the motion cell circuitry with the finite load resistance, R_L.

shown, however, that under some circumstances the values on these global output lines may not be velocity, so we will distinguish the chip's output by calling it "reported motion," M.

Because $M = \int i_{TOT}\, dt$, the condition for the system to be in steady state is $i_{TOT} = 0$. Before considering the load resistance, $i_{TOT} = i_{OUT}$. Including the load resistance, $i_{TOT} = i_{OUT} - i_L$ so for steady state $i_{OUT} = i_L = M\frac{1}{R_L}$. The output current, i_{OUT}, produced by the top multiplier is:

$$i_{OUT} = (M\frac{\partial I}{\partial x} - \frac{\partial I}{\partial t})\frac{\partial I}{\partial x}.$$

Steady state becomes:

$$M\frac{1}{R_L} = (M\frac{\partial I}{\partial x} - \frac{\partial I}{\partial t})\frac{\partial I}{\partial x}.$$

Solving for M we get:

$$M = \frac{\partial I}{\partial t}\frac{\frac{\partial I}{\partial x}}{(\frac{\partial I}{\partial x})^2 + \frac{1}{R_L}}. \tag{3}$$

A plot of this mathematical function is shown in Figure 12. For sufficiently large $\frac{\partial I}{\partial x}$, $(\frac{\partial I}{\partial x})^2 \gg \frac{1}{R_L}$, so Equation 3 reduces to:

$$M = -\frac{\frac{\partial I}{\partial t}}{\frac{\partial I}{\partial x}} = v. \tag{4}$$

As $\frac{\partial I}{\partial x}$ approaches zero, so that $(\frac{\partial I}{\partial x})^2 \ll \frac{1}{R_L}$, the equation becomes:

$$M = \frac{\partial I}{\partial t}\frac{\partial I}{\partial x}R_L. \tag{5}$$

This analysis allows us to determine the chip's behavior for images that have different contrast ratios. When there is sufficient difference between light and dark areas, the motion detector chip will report velocity accurately. As contrast in the image is reduced, the motion output M will change smoothly to become a function proportional to both $\frac{\partial I}{\partial t}$ and $\frac{\partial I}{\partial x}$. It seems a particularly graceful way for a system to fail as the contrast ratio in its field of view is reduced to the point at which velocity can no longer be extracted. For a strict velocity detector, the zero-contrast case is undefined, so a device

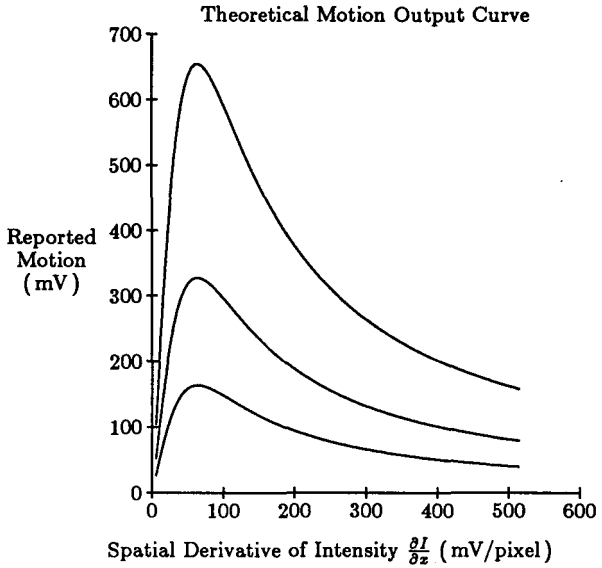

Figure 12: Plot of the theoretical motion response curve as a function of $\frac{\partial I}{\partial x}$ according to Equation 3. This curve approximates a hyperbolic response to the right and a linear response near zero.

that reported "true velocity" could take on any value in the absence of information. Our motion detector behaves better—it reports zero motion when it can detect no spatial intensity variation.

Note that, if the multiplicative definition of motion is desired over the entire operating range, the motion detector chip can be easily made to do this calculation. Setting the control current to zero on the feedback multiplier makes $i_{OUT} = \frac{\partial I}{\partial t}\frac{\partial I}{\partial x}$. This current can be turned into a voltage by the load resistor or a higher performance current sensing arrangement off chip.

Over the complete range of $\frac{\partial I}{\partial x}$'s, and in particular in both the hyperbolic and linear regimes of motion, Equations 4 and 5 show that the magnitude of the motion response should be proportional to $\frac{\partial I}{\partial t}$. The three curves of Figure 10, taken at frequencies of 10Hz, 20Hz, and 40Hz are scaled versions of the same curve and so bear out this proportionality over the range of $\frac{\partial I}{\partial x}$. Figure 9 shows a plot of reported motion as a function of $\frac{\partial I}{\partial t}$. The three curves are for fixed $\frac{\partial I}{\partial x}$'s, one chosen for each of the regimes of operation and one for midway in the transition region between them.

3.2 Verification of Constraint Line Behavior

The collection of circuits in each cell (see Figure 6), working in concert, tries to satisfy the constraint between the x and y components of velocity according to the line equation:

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0.$$

This constraint is defined by the three inputs, the locally measured intensity derivatives, $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$. If we force the value of one of the components of velocity, the circuit will drive the other component of velocity until its value satisfies the constraint. For a given image input, the entire constraint line can be determined by sweeping the forced velocity value. Figure 13 plots three constraint lines from the measured response of the motion detector chip. A single edge was projected onto the chip so that all the constraint lines of each cell in the array would coincide. The x component of velocity was driven to a sequence of values. For each value, the chip determined the y component and the resulting point in velocity space was plotted. The image was not moving relative to the chip, so we predicted that the constraint line would pass through the origin. The constraint line was plotted for three different orientations of the edge. To ensure the relative angles of the three orientations, we used a single triangle as the image for each trial. Between trials, the part of the image falling on the chip was adjusted by translations only. Although the data deviated from the ideal slightly, this experiment clearly demonstrated the constraint line behavior of the motion detector chip.

3.3 Velocity Space Maps

To demonstrate the two-dimensional collective operation of the motion detector chip, we focused an image of a single high contrast edge onto the chip. Because the image was stationary, the chip should report zero motion. The global output lines were driven externally to take on a sequence of values chosen to scan the velocity space in a regular grid. For each x,y pair of voltages driven onto the chip, the chip responded with a current intended to move the global point in velocity space into agreement with the velocity of its image input; namely, zero velocity. These resulting x,y pairs of currents were measured for each point and displayed as a small vector originating at the forced point in velocity space. The resulting maps of these vectors (Figures 14 through 16) show the direction and magnitude of the force exerted by the chip on the global velocity lines. The point of stability—the attractor point—is near zero, as it should be. The amount by which the chip pulls as the global line moves farther away from the attractor depends on the structure of the applied image. A one-dimensional image, such as the single edge used in this experiment, provides information about the velocity only perpendicular to the edge. Thus, the chip should pull harder when the velocity lines are forced away from the real velocity in a direction perpendicular to the one-dimensional image stimulus. The image contains less information about the velocity parallel to the applied edge, so forced displacements of velocity away from zero in that direction result in much smaller restoring forces.

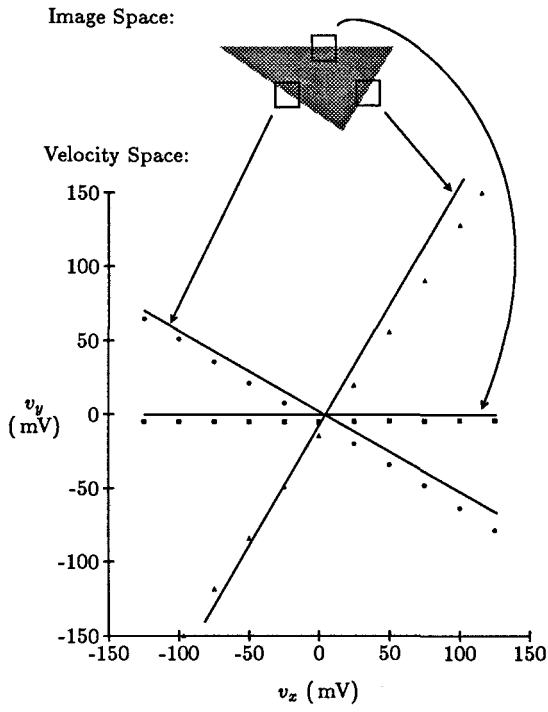

Figure 13: Demonstration of the constraint line behavior of the motion detector chip. The x component of velocity was swept while v_x versus v_y was plotted. The three trials were for the edge in the image oriented at 0°, 60°, and −30°. The lines are the ideal constraint lines.

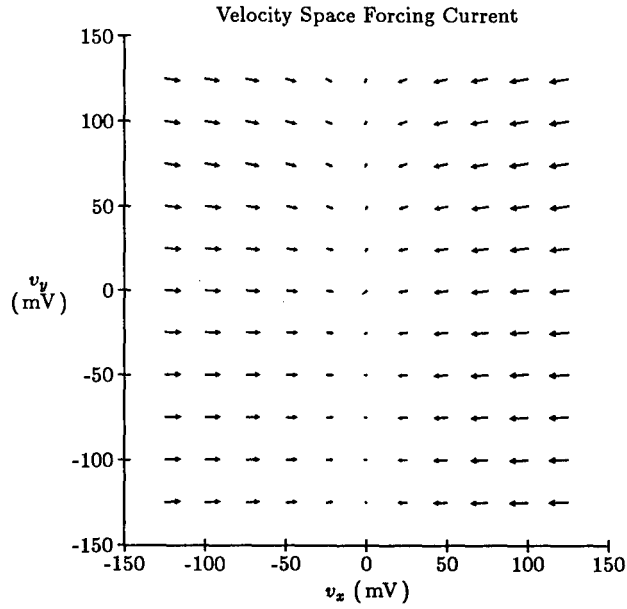

Figure 14: Velocity space map of the restoring forces generated by the motion detector chip in response to an edge at 90°.

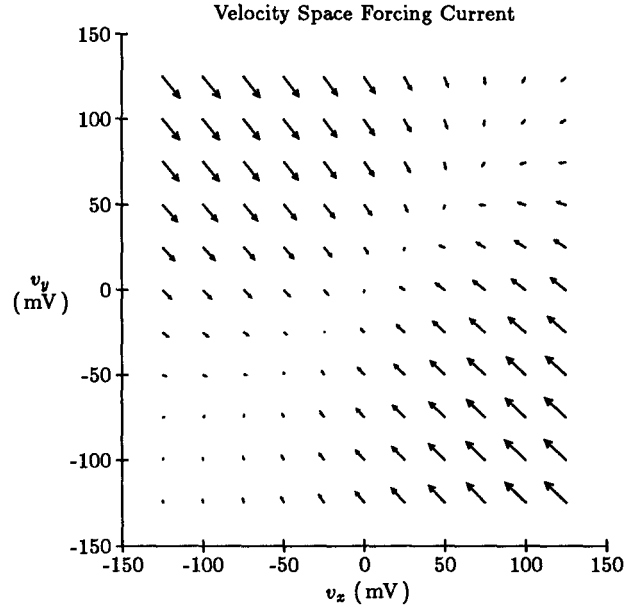

Figure 15: Velocity space map of the restoring forces generated by the motion detector chip in response to an edge at 45°.

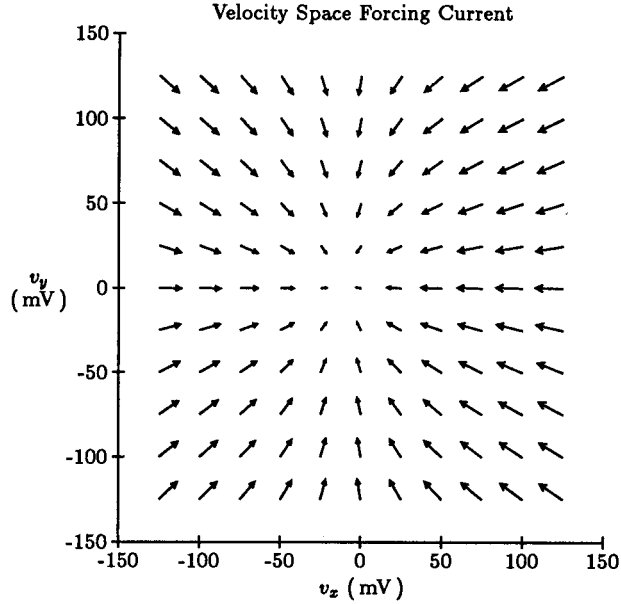

Figure 16: Velocity space map of restoring force for a bright circle on a dark background. Edges of all orientations are represented.

4 Conclusions

We have designed and built the first integrated motion detector using analog circuit elements and collective computation. We formulated the algorithm and designed the architecture to exhibit collective behavior through the aggregation of locally derived quantities. We used closely coupled analog photosensors and small analog computational elements making feasible the parallel operation of large arrays of sensors and computing elements. The system makes extensive use of local intensity information and therefore operates in the presence of global intensity gradients. This local computing also eliminates the need for any prior higher-level processing like edge detection or object recognition.

The motion detector architecture, along with local analog computational elements, provides a dense, reliable means of processing high bandwidth visual data. The motion detector chip demonstrates the suitability of analog VLSI circuits for processing of sensory data.

The motion detector is one of the first of a growing class of systems that employ collective computation. As we gain experience with this type of system, we should expand the range of application beyond that of processing sensory data. The motion detector proved to be a good first example. Although it has a crisp, solid mathematical foundation, the motion detector exhibits the collective behavior that is so necessary for processing sensory data from the fuzzy real world.

References

[1] R. F. Lyon, "The optical mouse, and an architectural methodology for smart digital sensors," *CMU Conference on VLSI Systems and Computations*, pp. 1–19, 1981.

[2] J. E. Tanner and C. Mead, "A correlating optical motion detector," *1984 MIT Conference on Very Large Scale Integration*, pp. 57–64, 1984.

[3] C. Mead, "A sensitive electronic photoreceptor," *1985 Chapel Hill Conference on VLSI*, pp. 463–471, 1985.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.