University of Pennsylvania
ScholarlyCommons

Departmental Papers (ESE)   Department of Electrical & Systems Engineering

September 1999

Hardware Implementation of a Visual-Motion Pixel Using Oriented Spatiotemporal Neural Filters

Ralph Etienne-Cummings
Johns Hopkins University

Jan Van der Spiegel
University of Pennsylvania, jan@seas.upenn.edu

Paul Mueller
Corticon, Inc.

Follow this and additional works at: http://repository.upenn.edu/ese_papers

Recommended Citation

Hardware Implementation of a Visual-Motion Pixel Using Oriented Spatiotemporal Neural Filters

Abstract

A pixel for measuring two-dimensional (2-D) visual motion with two one-dimensional (1-D) detectors has been implemented in very large scale integration. Based on the spatiotemporal feature extraction model of Adelson and Bergen, the pixel is realized using a general-purpose analog neural computer and a silicon retina. Because the neural computer only offers sum-and-threshold neurons, the Adelson and Bergen's model is modified. The quadratic nonlinearity is replaced with a full-wave rectification, while the contrast normalization is replaced with edge detection and thresholding. Motion is extracted in two dimensions by using two 1-D detectors with spatial smoothing orthogonal to the direction of motion. Analysis shows that our pixel, although it has some limitations, has much lower hardware complexity compared to the full 2-D model. It also produces more accurate results and has a reduced aperture problem compared to the two 1-D model with no smoothing. Real-time velocity is represented as a distribution of activity of the 18 X and 18 Y velocity-tuned neural filters

Keywords

Vision chips, visual motion detection, VLSI neural filters, Adelson and Bergen

Hardware Implementation of a Visual-Motion Pixel Using Oriented Spatiotemporal Neural Filters

Ralph Etienne-Cummings, *Member, IEEE*, Jan Van der Spiegel, *Senior Member, IEEE*, and Paul Mueller

Abstract— A pixel for measuring two-dimensional (2-D) visual motion with two one-dimensional (1-D) detectors has been implemented in very large scale integration. Based on the spatiotemporal feature extraction model of Adelson and Bergen, the pixel is realized using a general-purpose analog neural computer and a silicon retina. Because the neural computer only offers sum-and-threshold neurons, the Adelson and Bergen's model is modified. The quadratic nonlinearity is replaced with a full-wave rectification, while the contrast normalization is replaced with edge detection and thresholding. Motion is extracted in two dimensions by using two 1-D detectors with spatial smoothing orthogonal to the direction of motion. Analysis shows that our pixel, although it has some limitations, has much lower hardware complexity compared to the full 2-D model. It also produces more accurate results and has a reduced aperture problem compared to the two 1-D model with no smoothing. Real-time velocity is represented as a distribution of activity of the 18 X and 18 Y velocity-tuned neural filters.

Index Terms—Vision chips, visual motion detection, VLSI neural filters.

I. INTRODUCTION

THE VISUAL-motion detection mechanism employed by insects, such as flies, and primates are quite different. The effective computations performed by these organizms, however, have been shown to be identical [1], [2]. For insects, motion estimation is performed very early, immediately following the light-sensitive omatidia, while for primates, the motion center is area medial temporal (MT) in the cortex [3], [4]. As a result, the fly's motion-detection neural circuits can be modeled with mostly nearest neighbor correlation of the omatidia outputs to extract both direction and speed of moving light patches. There are obvious survival benefits for the early implementation of visual motion detection in flies. Hence, the insect visual motion neural networks are designed to optimally match the expected spatiotemporal characteristics of the environment in which they live, and are consequently narrowband and velocity-specific detectors. For primates, on the other hand, the visual tasks required for survival are much more complex. The environment in which primates live has a wide range of spatiotemporal frequencies, and motion

estimation must be performed on targets with a wide velocity dynamic range. Since primates are visual animals, a large portion of the cortex is dedicated to visual processing, which is also integrated with other sensing, reasoning and behavioral centers of the brain. Consequently, the method employed for primate visual-motion estimation exploits the availability of a large amount of wetware to realize robust, wide dynamic range, general-purpose motion-estimation circuits.

Attempts to implement visual-motion estimation in VLSI hardware have followed the fly's model [5], [6]. This has been influenced strongly by the limited space and power available to realize focal-plane motion detection circuits which can be easily integrated on behaving systems such as robots, autonomous vehicles, and gaze controllers. Consequently, the last few years have produced some elegant focal-plane motion detection circuits based on the fly's model, while expanding its spatiotemporal bandwidth and dynamic range [6]–[9]. The approach presented here is not a competitor for these compact focal-plane solutions. Our method requires too much hardware for focal-plane implementation. It is, however, intended for the scenario where a functional "silicon cortex," with many neurons, synapses, and axon/dendrites, is available and applied to a problem requiring visual motion as a prerequisite. In this paradigm, the motion-detection mechanism must be compatible with the other computations performed by the "silicon cortex." Hence, the implementation presented here mimics the physical organization of the primate visual system, and the computation model for cortical visual-motion estimation.

Visual-motion estimation is an area where spatiotemporal computation is of fundamental importance. Each distinct motion vector traces a unique locus in the space–time domain. Hence, the problem of visual-motion estimation reduces to a feature extraction task, with each feature extractor tuned to a particular motion vector. Since neural networks are particularly efficient feature extractors, they can be used to implement these visual-motion estimators. Such neural circuits have been recorded in area MT of macaque monkeys, where cells are sensitive and selective to two-dimensional (2-D) velocity [4].

Here, a hardware pixel for two 1-D visual-motion estimation with spatiotemporal feature extractors is presented. A silicon retina with parallel continuous-time edge-detection capabilities is the front-end of the system. Motion detection neural networks are implemented on a general-purpose analog neural computer, which is composed of programmable analog neurons, synapses, axon/dendrites, and synaptic time constants [12]. The additional computational freedom introduced by the synaptic time-constants is required to realize
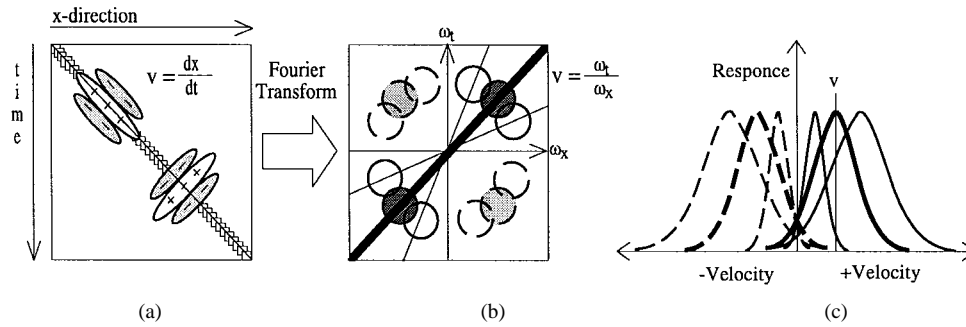
Fig. 1.   (a) 1-D motion is orientation in the space–time domain. (b) Motion detection can be realized with oriented spatiotemporal filters. (c) The distribution of the velocity-tuned filter responses encode the stimulus velocity.

the spatiotemporal motion estimators. The motion detection neural circuits are based on the early one-dimensional (1-D) model of Adelson and Bergen and recent 2-D models of David Heeger [1], [10], [11]. Since the neurons only computed delayed weighted sum-and-threshold functions, the models are modified. The original models require a division for intensity normalization and a quadratic nonlinearity to extract spatiotemporal energy. In our model, a silicon retina performs intensity normalization with high contrast sensitivity (a binary edge image is produced), and the quadratic nonlinearity is replaced by rectification. In an effort to handle 2-D motion detection with a tractable hardware complexity, we use smoothing prefilters orthogonal to the direction of motion and two 1-D detectors. We show that this approach reduces the amount of hardware required per pixel considerably, compared to the full 2-D implementation of Heeger, however, it is not as general. Compared to the two 1-D method without prefiltering, our model produces more accurate results and has less of an aperture problem. It does, however, require that the spectral components of the stimuli exist in the ω_x-ω_t and ω_y-ω_t planes. If none of the spectral power of the stimulus appears in the ω_y-ω_t and ω_y-ω_t planes, our approach would produce no output, while the strict two 1-D approach would produce incorrect results. Fortunately, the binary image produced by the silicon retina places spectral components in these planes for most images (intensity gradients with too small contrast do not produce a binary image). The measured tuning curves of 18 X and 18 Y velocity-selective neural filters are presented. The visual-motion vector is implicitly coded as a distribution of neural activity, and the explicit velocity is computed by the first moment of the distribution.

II. THEORY OF SPATIOTEMPORAL FEATURE EXTRACTION AS MOTION DETECTORS

A. Overview

The technique of estimating motion with spatiotemporal feature extraction emerged from the observation that a point moving with constant velocity traces a line in the space-time domain, as is shown in Fig. 1(a). The slope of the line is proportional to the velocity of the point. Hence, the velocity is represented as the orientation of the line. Spatiotemporal orientation detection units, similar to those proposed by Hubel and Wiesel for spatial orientation detection, can be used

for detecting motion [13]. Fig. 1(a) shows an oriented filter sensitive only to symmetric moving features; an antisymmetric filter is also required. Hence, quadrature pairs of oriented filters at various scales are used to measure visual motion in a realistic scene. In the frequency domain, the motion of a point is also a line. The slope of the line is the velocity of the point. Hence orientation detection filters, shown as circles in Fig. 1(b), are used to measure the point's velocity relative to their preferred velocity. A population of these tuned filters, Fig. 1(c), can be used to measure general image motion.

B. 1-D Oriented Spatiotemporal Filters

The construction of the spatiotemporal motion detection units is based on the frequency domain representation of visual motion. In the frequency domain, constant 1-D motion of a point is represented as a line through the origin with a slope $\mathbf{v_o} = \omega_t/\omega_x$, where $\mathbf{v_o}$ is the velocity, ω_t is the temporal frequency, and ω_x is the spatial frequency. That is, the Fourier transform of $\delta(x - v_x t)$ is $\delta(v_x\omega_x - \omega_t)$. Oriented spatiotemporal filters tuned to this velocity can be easily constructed using separable quadrature pairs of spatial and temporal filters tuned to ω_{to} and ω_{xo}, respectively, where the preferred velocity $\mathbf{v_o} = \omega_{to}/\omega_{xo}$. The $\pi/2$ phase relationship between the filters allows them to be combined such that they cancel in opposite quadrants, leaving the desired unseparable oriented filter. Hence, quadrature pairs of oriented filters are created and must be considered together to measure the complete influence of the motion of a general image patch. Fig. 2 shows the construction of oriented filters for 1-D motion. Examples of separable spatial and temporal filters and the constructed motion detectors, employed by Heeger, are given in (1)–(3) [11]. The only requirements for successful candidate functions are that they should be matched, band-limited and quadrature counterparts. Equation (3) shows how they are combined to realize oriented spatiotemporal filters

$$g_o(t) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(\frac{-t^2}{2\sigma_t^2}\right) \sin(2\pi\omega_{to}t) \qquad (1a)$$

$$g_e(t) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(\frac{-t^2}{2\sigma_t^2}\right) \cos(2\pi\omega_{to}t) \qquad (1b)$$

$$g_o(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(\frac{-x^2}{2\sigma_x^2}\right) \sin(2\pi\omega_{xo}x) \qquad (2a)$$

$$g_e(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(\frac{-x^2}{2\sigma_x^2}\right) \cos(2\pi\omega_{xo}x) \qquad (2b)$$
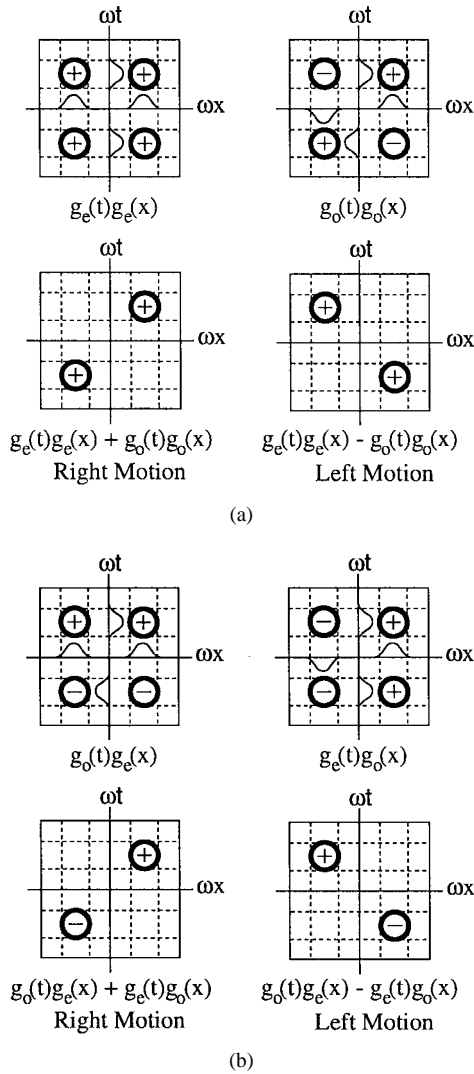
Fig. 2. In the frequency domain, motion filters are constructed with quadrature pairs of oriented spatiotemporal filters. (a) Even-oriented filters. (b) Odd-oriented filters.

$$\text{Right}_e(v_x, x, t) = g_e(t)g_e(x) + g_o(t)g_o(x)$$

or

$$\text{Right}_o(v_x, x, t) = g_e(t)g_o(x) + g_o(t)g_e(x) \quad (3a)$$

$$\text{Left}_e(v_x, x, t) = g_e(t)g_e(x) - g_o(t)g_o(x)$$

or

$$\text{Left}_o(v_x, x, t) = g_e(t)g_o(x) - g_o(t)g_e(x). \quad (3b)$$

The complete velocity selective filter is constructed by combining the even and odd oriented filters, i.e., $\text{Right}^2 = (\text{Right}_e)^2 + (\text{Right}_o)^2$. Since this technique measures the energy of the input signal about the preferred orientation of the filter, it suffers from velocity/contrast ambiguity. That is, a bright object with velocity far from the preferred value may solicit a stronger response than a dim object closer to the preferred value. To eliminate this effect, the output of the velocity sensitive filter is normalized by the contrast of the image patch at the spatial scale of the oriented filter. The final frequency domain response of an oriented motion detection
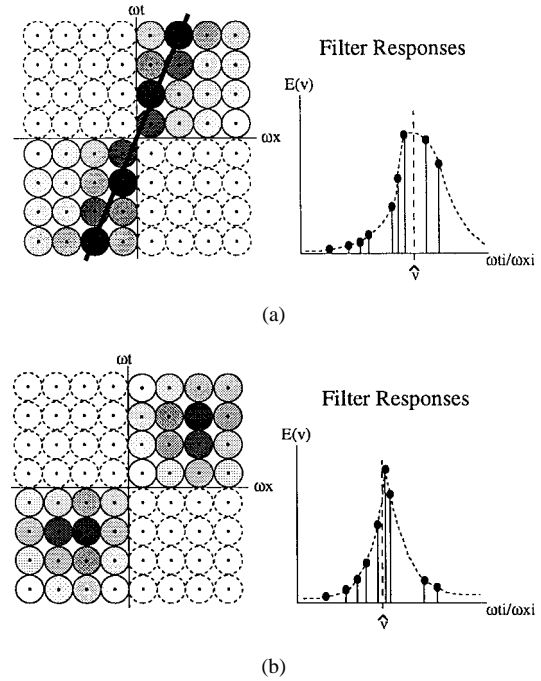

Fig. 3. (a) 1-D motion of a point produces a broad distribution of filter responses. (b) Motion of sinusoidal grating produces a narrow distribution of filter responses. The centroid relays the explicit velocity.

filter is subsequently given by (4)

$$E(v_x, \omega_{\text{xo}}, \omega_{\text{to}})$$
$$= \frac{[\text{Right}(v_x, \omega_{\text{xo}}, \omega_{\text{to}})]^2 - [\text{Left}(v_x, \omega_{\text{xo}}, \omega_{\text{to}})]^2}{[g_e(\omega_{\text{xo}})]^2 + [g_o(\omega_{\text{xo}})]^2}. \quad (4)$$

E is the output energy of the filter and is a bipolar representation of velocity. Because this filter is tuned to a specific velocity at a specific spatiotemporal frequency, the complete model is then composed of a population of similar detectors with various spatiotemporal tuning. The velocity is extracted from the distribution of filter outputs, much like the motion sensitive cells from area MT of macaque monkeys [4], [14], [15]. Fig. 3(a) shows a possible placement of the motion filters in the ω_x-ω_t plane. The responses of the filters to a moving point are shown in gray scale, where black implies maximum activity. Since the moving point has a flat power spectrum, all filters along its velocity will be stimulated. Hence, a broad distribution of responses is observed. The centroid of the distribution of responses provides an optimal estimate (in the least-squared sense) of the motion, which improves as more filters with more overlapping passbands are used [14]. Other heuristics may be used to extract the explicit velocity. Equation (5) gives the relationship for the estimated velocity, where M is the number of oriented filters and $(\omega_{ti}/\omega_{\text{xi}})$ is the tuned velocity for each filter

$$\hat{v}_x = \frac{\sum_{i=1}^{M} \left(\frac{\omega_{ti}}{\omega_{\text{xi}}}\right) E(v_x, \omega_{\text{xi}}, \omega_{ti})}{\sum_{i=1}^{M} E(v_x, \omega_{\text{xi}}, \omega_{ti})}. \quad (5)$$
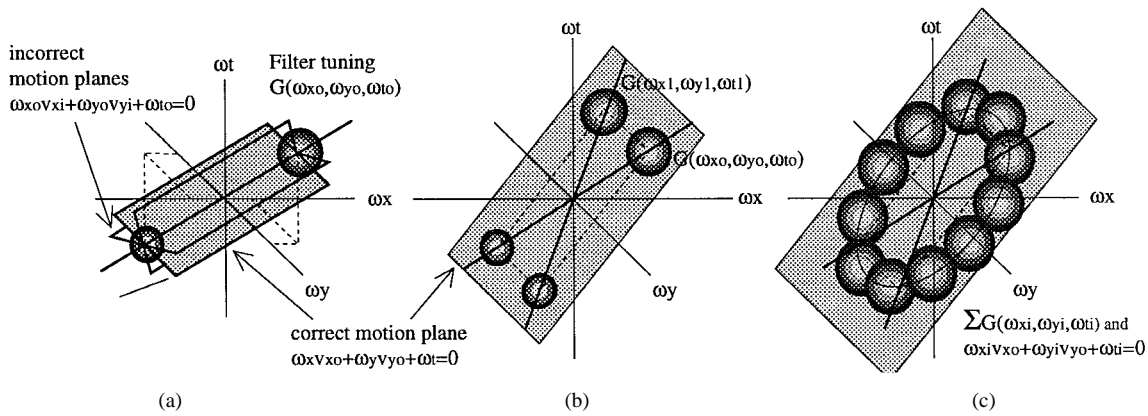
Fig. 4. Measurement of 2-D motion with oriented filters. (a) Single filter responds to a continuum of 2-D motion planes. (b) Two oriented filters isolate the correct motion plane, but do not respond to all spatiotemporal frequencies. (c) A collection of filters isolates the correct motion plane and spans all spatiotemporal frequencies.

In Fig. 3(b), a moving sinusoidal grating is presented as input. The grating's spectrum is localized at a single point in the ω_x-ω_t plane. Consequently, the filter responses are narrowly distributed about the filter closest in spatiotemporal tuning to the grating. None-the-less, a centroid of the responses provides a good estimate of the grating's velocity. Grzywacz and Yuille have argued that only two filters are required to estimate the velocity, however, the accuracy of the estimation will decrease as the spatiotemporal spectrum of the image and the oriented filters differ [14]. This is illustrated by the sinusoidal grating stimulus, which produces no filter responses if its spatiotemporal spectrum does not appear within the passband of any of the motion detectors. Clearly, a general-purpose motion detection system using oriented spatiotemporal filters must cover the entire frequency range of the intended stimulus. A mosaic of individual filters located at various tunings, as shown in Fig. 3, can be used. In addition, this collection of filters must also be replicated at every pixel. Evidently, this approach is hardware implementable only in a large system.

C. 2-D Oriented Spatiotemporal Filters

If the point exhibits 2-D $(x, y; v_x, v_y)$ motion, the problem is substantially more complicated [11], [15]. A point executing 2-D motion spans a plane in the frequency domain. The approach presented above for constructing oriented filters still applies in two dimensions. The oriented filters of the form in (6), derived from (4), respond not only to the desired motion, but also to a continuum of motion satisfying $\omega_{xo}v_{xi} + \omega_{yo}v_{yi} + \omega_{to} = 0$, where v_{xi} and v_{yi} are the velocity components defining the plane, and $(\omega_{xo}, \omega_{yo}, \omega_{to})$ is the spatiotemporal tuning of the filter. The filters G and F in (6) are orthogonal to each other. This is a manifestation of the famous aperture

problem, and results from the observation that a single 2-D oriented filter solves one equation of two unknowns. This problem is graphically shown in Fig. 4(a)

$$
\begin{aligned}
G_e(x,&\, y,\, t\!: \omega_{xo}, \omega_{yo}, \omega_{to})\\
&= g_e(x)g_e(y)g_e(t) + g_e(x)g_o(y)g_o(t)\\
&\quad + g_o(x)g_e(y)g_o(t) + g_o(x)g_o(y)g_e(t) \qquad (6a)
\end{aligned}
$$

$$
\begin{aligned}
G_o(x,&\, y,\, t\!: \omega_{xo}, \omega_{yo}, \omega_{to})\\
&= g_e(x)g_e(y)g_o(t) + g_e(x)g_o(y)g_e(t)\\
&\quad + g_o(x)g_e(y)g_e(t) + g_o(x)g_o(y)g_o(t) \qquad (6b)
\end{aligned}
$$

$$
\begin{aligned}
F_e(x,&\, y,\, t\!: \omega_{xo}, \omega_{yo}, \omega_{to})\\
&= g_e(x)g_e(y)g_e(t) + g_e(x)g_o(y)g_o(t)\\
&\quad - g_o(x)g_e(y)g_o(t) - g_o(x)g_o(y)g_e(t) \qquad (6c)
\end{aligned}
$$

$$
\begin{aligned}
F_o(x,&\, y,\, t\!: \omega_{xo}, \omega_{yo}, \omega_{to})\\
&= g_e(x)g_e(y)g_o(t) + g_e(x)g_o(y)g_e(t)\\
&\quad - g_o(x)g_e(y)g_e(t) - g_o(x)g_o(y)g_o(t). \qquad (6d)
\end{aligned}
$$

If two sets of oriented filters are used, the correct motion plane can be isolated (i.e., two equations of two unknowns). However, the two filters would only cover the spatiotemporal bands defined by their tuning and bandwidth, as shown in Fig. 4(b). Hence, this configuration is effective only for broad-band images. A collection of filters spanning all combinations of $(\omega_x, \omega_y, \omega_t)$ in the plane of the correct motion is then required. Fig. 4(c) shows schematically one such filter tuned to a band of frequencies on the preferred motion plane. To respond to a general image patch at this preferred velocity, a number of these torus-shaped filters, with increasing radii, is required to cover the bandwidth of the image. Equations (7a) and (7b), shown at the bottom of the page, give the expression for one of the torus-shaped filters. The equation assumes that individual filters in the torus are placed such that they overlap

$$
E^2(v_{xi}, v_{yi}, C_k) = \sum_{j=0}^{N} E_k^2(\omega_{xj}, \omega_{yj}, \omega_{tj}), \qquad \text{where } v_{xi}\omega_{xj} + v_{yi}\omega_{yj} + \omega_{tj} = 0 \quad \text{and} \quad \omega_{xj}^2 + \omega_{yj}^2 + \omega_{yj}^2 = C_k{}^2 \quad (7a)
$$

$$
E_k^2(\omega_{xj}, \omega_{yj}, \omega_{tj}) = \frac{[G_e^2(\omega_{xj}, \omega_{yj}, \omega_{tj}) + G_o^2(\omega_{xj}, \omega_{yj}, \omega_{tj})] - [F_e^2(\omega_{xj}, \omega_{yj}, \omega_{tj}) + F_o^2(\omega_{xj}, \omega_{yj}, \omega_{tj})]}{[g_e(\omega_{xj})g_e(\omega_{yj})]^2 + [g_o(\omega_{xj})g_o(\omega_{yj})]^2 + [g_o(\omega_{xj})g_e(\omega_{yj})]^2 + [g_e(\omega_{xj})g_o(\omega_{yj})]^2} \qquad (7b)
$$

at $(\sigma_{\mathrm{xj}}^2 + \sigma_{\mathrm{yj}}^2 + \sigma_{\mathrm{tj}}^2)^{1/2} = \sigma_j$, and the centers of the filters are $2\sigma_j$ apart, where $(\sigma_{\mathrm{xj}}, \sigma_{\mathrm{yj}}, \sigma_{\mathrm{tj}})$ are, respectively, the bandwidths of the separable filter components. Hence, $N = \pi(C_k)/\sigma_j = \pi(\omega_{\mathrm{xj}}^2 + \omega_{\mathrm{yj}}^2 + \omega_{\mathrm{tj}}^2)^{1/2}/\sigma_j$ filters are required per torus, where C_k is the radius of the kth torus, and N is the number of individual velocity-tuned filters used to create the torus. The bandwidth of the wideband velocity-tuned filter is the sum of the bandwidths of the filters ($2\sigma_j$) in each of the M tori as in (7a) and (7b), shown at the bottom of the previous page.

The construction of the wideband velocity filter tuned to this one motion plane requires a total of $8MN$ separable spatiotemporal filters per pixel. To cover a large range of speed and direction, many such filters are required. Assuming the same coverage strategy as for each torus, N^2 velocity planes are required, resulting in a total of $8MN^3$ separable filters per pixel. In comparison, for the 1-D case, assuming the same coverage algorithm, the number of separable filters required is $4MN$. (In a sparsely distributed example where $M = 3$ and $N = 6$, the number of separable filters required is 5184 per pixel for the full 2-D model, compared to 144 for the two 1-D case.) The speed and direction of a general image patch can be extracted from the distribution of filter responses as given by (8)

$$|v| = \frac{\displaystyle\sum_{k=0}^{M}\sum_{i=0}^{N^2} E^2(v_{\mathrm{xi}}, v_{\mathrm{yi}}, C_k)\sqrt{v_{\mathrm{xi}}^2 + v_{\mathrm{yi}}^2}}{\displaystyle\sum_{k=0}^{M}\sum_{i=0}^{N^2} E^2(v_{\mathrm{xi}}, v_{\mathrm{yi}}, C_k)}$$

and

$$\theta = \arctan\left[\frac{\displaystyle\sum_{k=0}^{M}\sum_{i=0}^{N^2} E^2(v_{\mathrm{xi}}, v_{\mathrm{yi}}, C_k)\left(\frac{v_{\mathrm{yi}}}{v_{\mathrm{xi}}}\right)}{\displaystyle\sum_{k=0}^{M}\sum_{i=0}^{N^2} E^2(v_{\mathrm{xi}}, v_{\mathrm{yi}}, C_k)}\right]. \quad (8)$$

Evidently, the hardware complexity for realizing the full 2-D motion detection filters is too large and completely intractable in current very large scale integration (VLSI) systems. Using two 1-D detectors reduces the complexity for $O(N^3)$ to $O(N)$, but there are some penalties. Below, we discuss the hazards of measuring 2-D motion with two 1-D detectors. We also propose an approach to reduce the disadvantages. We show that our two 1-D method only approximates 2-D motion detection in special cases. Our system promotes the special cases.

III. MEASURING 2-D MOTION WITH TWO 1-D DETECTORS

A. Overview

Measuring 2-D motion with two 1-D motion detectors is fraught with problems. In special cases, however, the two 1-D detectors do produce the correct 2-D results. First, we present the problems with the two 1-D approach, and subsequently, we identify the cases where it produces the correct results. Lastly, we show how our system promotes the special cases and how it is constructed.
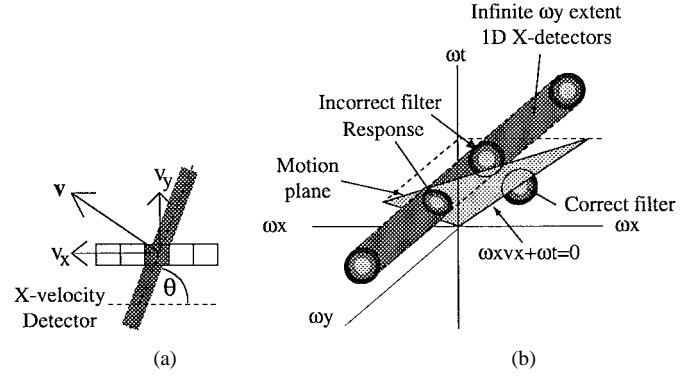

Fig. 5. (a) Measuring 2-D motion with a 1-D detector is aperture limited and (b) produces errors.

B. Two 1-D Motion Detection

Consider a 1-D motion detector for X-velocity. It is realized with a row of pixels, and the measured motion is reported at the central pixel. When a 2-D stimulus falls across its receptive field, a single sample of the stimulus is taken in the Y-direction, and in the X-direction, the number of samples corresponds to the spatial scale of the detector. Clearly, the aperture problem in the Y-direction is more severe than in the X-direction for an X-velocity detector. Fig. 5(a) shows this process schematically. The velocity reported by an X- and Y-detector pair is given by (9) for a line oriented at θ to the horizontal and moving with velocity (v_x, v_y)

$$v_x^{\text{1-D}} = v_x + v_y/\tan\theta \quad \text{and} \quad v_y^{\text{1-D}} = v_y + v_x\tan\theta. \quad (9)$$

Equation (9) illustrates both the aperture problem and the propensity for two 1-D detectors to provide extremely wrong measurements as the orientation of the 2-D stimulus approaches $0°$ or $90°$ On the other hand, if the stimulus is a vertical or a horizontal line, the correct aperture limited measurement is obtained.

To understand the impact of a general 2-D image patch on 1-D detectors, the frequency domain analysis is more illuminating. A 1-D X-velocity detector is oriented in the ω_x-ω_t plane, but extends infinitely in ω_y. The inverse is true for the Y-velocity detector. Consequently, as shown in Fig. 5(b), the motion of any image patch from a continuum of motion planes matching the passband of the filter will produce an incorrect response. The only correct response is produced by the detector located in the ω_x-ω_t plane, where the motion plane intersects the ω_x-ω_t plane. Unfortunately, the latter detector is triggered only if the image patch has spectral components in this plane. Nonetheless, it is better to report zero motion rather than the incorrect velocity. Two 1-D motion detection measures the correct 2-D motion if the passband of the 1-D detectors are restricted to the ω_x-ω_t and ω_y-ω_t planes, and if the stimulus has spectral energy in those planes. This implies that the infinite extent of the passband of the 1-D detectors in the direction orthogonal to motion must be suppressed. Furthermore, the image must either already have spectral components in the ω_x-ω_t and ω_y-ω_t planes, or must be forced to have so. Our system attempts to realize these two constraints.

C. Realizing 2-D Motion Estimation with Two 1-D Detectors

To restrict the frequency response of the 1-D motion detectors to the ω_x-ω_t and ω_y-ω_t planes, a smoothing prefilter is applied in the direction orthogonal to motion. If the prefilter has infinite spatial scale, this is identically realized. In our system, an approximation is obtained with a smoothing "Gaussian" filter with twice the spatial scale (15 pixels) of the largest scale motion filter (7 pixels). An approximation of a Gaussian prefilter is chosen to reduce the high frequency side-lobes of the smoothing filter. Furthermore, smoothing at a larger scale than the motion detection and using a tapered smoothing kernel suppress edge effects.[1] Moreover, the receptive field of the 1-D detector is widened in the smoothing direction. Consequently, the aperture problem is reduced in our model compared to the two 1-D case without prefiltering, however, motion blurring across object boundaries will be more prevalent. The two 1-D images that are produced are used as inputs for the 1-D detectors.

The power spectrum of the image must exist in the ω_x-ω_t, and ω_y-ω_t planes for the two 1-D detectors to measure 2-D motion, albeit aperture limited. For wideband images, such as points, lines or abrupt edges, the power spectrum already exists in these planes. In these cases, other than the aperture limitations, the two 1-D detectors produce the correct 2-D results if the detectors are limited to the ω_x-ω_t and ω_y-ω_t planes. For a narrow-band image, such as a sinusoidal gratings and plaids, there are no spectral components in these planes. Here, our 2-by-1-D detectors would produce no results, while the version without prefiltering would produce incorrect results. Our system, however, goes one step further and places some spectral components in the ω_x-ω_t and ω_y-ω_t planes. This is performed by the silicon retina front end, which converts all images to binary images of edges. The sinusoidal plaid is thus converted into a square plaid, which is a wideband image. The additional benefit of the silicon retina is outlined below. Equation (10) outlines, mathematically, how the two components of motion can be extracted. Fig. 6 shows schematically how correct 2-D motion can be extracted with the two orthogonal sets of 1-D oriented filters

$$W_y(i) * I(\omega_x v_x + \omega_y v_y + \omega_t)$$
$$\xrightarrow{\omega_y = 0} I'(\omega_x v_x + \omega_t)\delta(\omega_y) \Rightarrow v_x = \frac{\omega_{ti}}{\omega_{xi}} \qquad (10a)$$

$$W_x(j) * I(\omega_x v_x + \omega_y v_y + \omega_t)$$
$$\xrightarrow{\omega_x = 0} I''(\omega_y v_y + \omega_t)\delta(\omega_x) \Rightarrow v_y = \frac{\omega_{tj}}{\omega_{yj}}. \qquad (10b)$$

D. Modified Two 1-D Construction

In addition to the modifications mentioned above, there are two further changes required for implementing oriented spatiotemporal filters with a relatively small number of analog neurons, synapses, axons/dendrites, and synaptic time constants. The first modification handles the need for contrast normalization. In Section II-A, we argued that contrast

[1] An oriented line that extends beyond the smoothing window moving in the direction orthogonal to the 1-D detector will produce and error similar to (9) unless these edge pixels are suppressed.
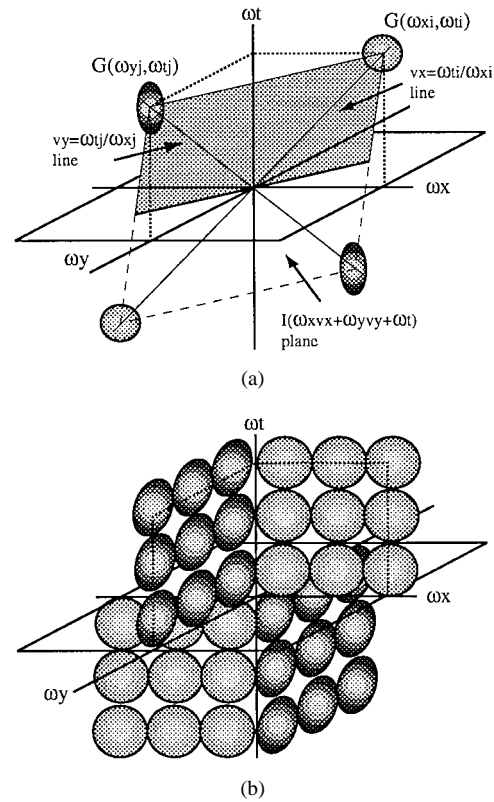

Fig. 6.   (a) 2-D motion detection with two 1-D filters isolated to the ω_x-ω_t and ω_y-ω_t planes produce correct results. (b) The oriented filters are organized in the ω_x-ω_t and ω_y-ω_t planes for wideband wide-range velocity detection.

normalization was required to remove the speed-contrast ambiguity. This is realized by dividing the filter response by the magnitude of the spatial contrast falling within its receptive-field. Unfortunately, ratios can not be easily implemented with linear sum-and-threshold neurons. We eliminate the need for contrast normalization by performing edge detection and labeling before motion detection. That is, a silicon retina is used to image the scene, to compute, in parallel, a 2-D convolution of the image with a band-limited Laplacian operator and to form a binary image of edges [6], [16]. The binary image is realized by thresholding the positive lobe of the edges $(> + V_\varepsilon)$ to the maximum voltage. This process has the effects of creating a spatiotemporally robust wideband image of strong edges with identical contrast. Furthermore, this procedure removes the potential singularity in (4) and (7) for noisy and low contrast images. This is equivalent to incorporating a confidence measure, as proposed by Uras *et al.*, when reporting visual-motion measurements [17]. Here we measure the motion of robust edges, which produce high confidence results. Fig. 7 shows a block diagram of the silicon retina.

The second modification is to replace the quadratic nonlinearity with a full-wave rectification. In Sections II-B and II-C, the filter response is squared to extract its energy content. This has the additional feature of rectifying the response. Since multiplication is also hard to realize with simple linear sum-and-threshold neural networks, the squaring operation is replaced by a full-wave rectification. In this case, the output of the filter is the square-root of the energy. Full-wave
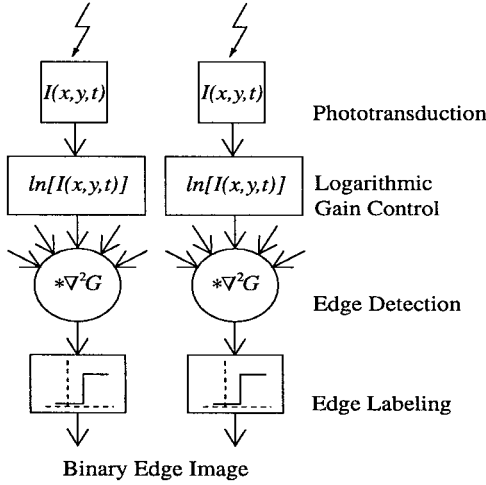
Fig. 7. The block diagram of silicon retina.

Fig. 8. Spatial presmoothing orthogonal to the direction of motion produces two 1-D images.

rectification can be easily implemented with half-wave rectified linear neurons. In our system, each 1-D oriented filter is given by

$$|\text{Right}(v_x, \omega_{xo}, \omega_{to})|$$
$$= |\text{Right}_e(v_x, \omega_{xo}, \omega_{to})| + |\text{Right}_o(v_x, \omega_{xo}, \omega_{to})| \quad (11\text{a})$$
$$|\text{Left}(v_x, \omega_{xo}, \omega_{to})|$$
$$= |\text{Left}_e(v_x, \omega_{xo}, \omega_{to})| + |\text{Left}_o(v_x, \omega_{xo}, \omega_{to})| \quad (11\text{b})$$
$$E(v_x, \omega_{xo}, \omega_{to})$$
$$= |\text{Right}(v_x, \omega_{xo}, \omega_{to})| - |\text{Left}(v_x, \omega_{xo}, \omega_{to})|. \quad (11\text{c})$$

IV. CONSTRUCTING THE SPATIOTEMPORAL MOTION FILTERS

A. Overview

Two 1-D motion detection can be used to approximate the full 2-D model with a much reduced hardware complexity per pixel. Section II-B outlines the steps for creating oriented filters tuned to 1-D motion. Coupled with the preconditioning of the image, using the silicon retina and the smoothing prefilters, the two 1-D detectors are realized with two 1-D sets of filters tuned to the X and Y directions, respectively. Quadrature pairs of spatial and temporal filters are required to construct the oriented filters. The spatial and temporal filters are used to realize the separable spatiotemporal filters that are not velocity selective. Due to their independence and separability, the velocity nonselective spatiotemporal filters are obtained simply by cascading the spatial and temporal filters. The individual spatial and temporal filters have narrow bandwidths, hence a number of these filters are required to cover the expected bandwidth of the input image. Here, three spatial and three temporal quadrature pairs are used, resulting in 36 velocity nonselective filters. Using these velocity nonselective separable filters, 18 oriented velocity selective nonseparable spatiotemporal filters are created. These 18 filters are composed of nine pairs tuned to the same spatiotemporal frequency and speed, but opposite directions.
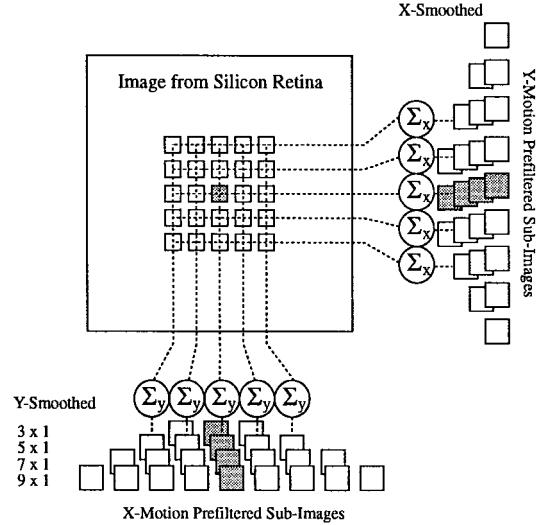
B. Spatial Prefiltering

Section III indicates that a smoothing prefilter is required to eliminate frequency components in the spatial direction orthogonal to the 1-D motion filters (see fig. 8). The scale of the prefilter is set to provide a good approximation of a delta function in the frequency domain, while not over smoothing the image. Clearly, as the scale of the smoothing filter increases, multiple independent objects can be combined. The measured motion will then be the average motion in the pixel's receptive field. An approximation of a Gaussian low-pass filter of 15 pixels, half kernel = [1, 0.97, 0.9, 0.8, 0.6, 0.35, 0.05, 0.01], is used since the motion detector is constructed of 7×7 image pixels.

C. Spatial Filters

The chosen spatial filters quadrature pairs are discrete approximations of first- and second-order derivatives of Gaussian functions. The first-order derivative provides an odd function, while the second-order derivative provides an even function. The space constant of the filter pair is set by the number of pixels used. That is, the spatial extent of the filter will be three pixels for the highest frequency filters and seven pixels for the lowest frequency filters. The sums of the positive and negative coefficients are $+0.5$ and -0.5, respectively, for all scales. Table I shows the coefficients used to implement the three scales of the spatial filter pairs. Equation (12) gives the general expression for filters in frequency space, where m is the number of coefficients in the half kernel. Equation (13) gives the expression for $g_2(\omega_x)$. This is also repeated in the y-direction

$$g_m(\omega_x)_e = 2a_m \cos[m\omega_x] + 2a_{m-1} \cos[(m-1)\omega_x]$$
$$+ \cdots + a_0 \quad (12\text{a})$$
$$g_m(\omega_x)_o = 2j(b_m \sin[m\omega_x] + b_{m-1} \sin[(m-1)\omega_x]$$
$$+ \cdots + b_1 \sin[\omega_x]) \quad (12\text{b})$$
$$g_2(\omega_x)_e = -0.18 \cos[2\omega_x] - 0.32 \cos[\omega_x] + 0.5 \quad (13\text{a})$$
$$g_2(\omega_x)_o = -2j(0.17 \sin[2\omega_x] + 0.33 \sin[\omega_x]). \quad (13\text{b})$$

TABLE I
SPATIAL FILTER COEFFICIENTS

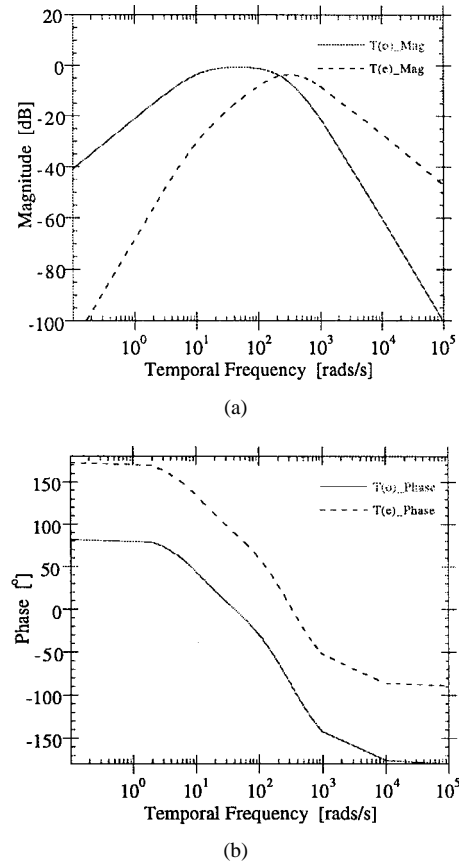| Spatial Filter	Odd	Even
	0.125	-0.080
	0.250	-0.170
	0.125	0.125
g3(x)	0	0.250
	-0.125	0.125
	-0.250	-0.170
	-0.125	-0.080
	0.170	-0.090
	0.330	-0.160
g2(x)	0	0.500
	-0.330	-0.090
	-0.170	-0.160
	0.500	-0.250
g1(x)	0	0.500
	-0.500	-0.250

(a)

(b)

Fig. 9. The magnitude and phase plots for the even and odd temporal filters for $\alpha = 11.11$ rads/s from (14). (a) Magnitude responses of the temporal filters. (b) Phase response of the temporal filters.
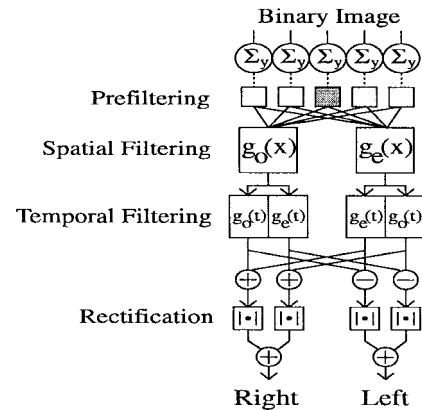

Fig. 10. Cascading the spatial and temporal quadrature filter pairs realizes the nonseparable oriented filters.

D. Temporal Filters

Bandpass filters are chosen to realize the temporal filter quadrature pairs. These filters must have nearly identical magnitude responses and delays, but the phase difference must be $\pi/2$. For identical delays, the characteristic equations for the pair must the same. Zeros at the origin are used to alter the phase of the filters. Equation (14) gives the general expressions for the odd and even pair. The pole locations govern the passband location of the filters. Here, the value of α_m is the dominant pole, while δ_{m1} and δ_{m2} control the frequency cut-off of the filters. The cut-off frequencies δ_{m1} and δ_{m2} are 20 and 40 times α_m, respectively. Fig. 9 shows plots of the magnitude and phase of (14) with $\alpha_m = 11.11$ rads/s. Three temporal filters are implemented for each spatial scale, $\alpha_m = 33.33$, 11.11, and 5.0 rads/s, in each direction

$$g_m(\omega_t)_o = \frac{j\omega_t \delta_{m1} \delta_{m2}}{(j\omega_t + \alpha_m)(j\omega_t + \delta_{m1})(j\omega_t + \delta_{m2})} \quad (14a)$$

$$g_m(\omega_t)_e = \frac{-\omega_t^2 \delta_{m2}}{(j\omega_t + \alpha_m)(j\omega_t + \delta_{m1})(j\omega_t + \delta_{m2})}. \quad (14b)$$

E. Oriented Spatiotemporal Filters

The oriented filters are realized by cascading the spatial and temporal filters as shown in Fig. 10. The figure shows the implementation of one pair of velocity-tuned filters, tuned to opposite directions. The remaining eight pairs per dimension are similarly constructed. Equation (15) gives the frequency domain representation of the filter for $\alpha_2 = 11.11$, $\delta_{21} = 222.2$, $\delta_{22} = 444.4$ rads/s and a spatial sampling frequency

$$
\begin{aligned}
&|\text{Right}(v_x, \omega_{x2}, \omega_{t2})| \\
&= \left| \frac{2\omega_t \delta_{21} \delta_{22}(0.17\,\sin[2\omega_x] + 0.33\,\sin[\omega_x])}{(j\omega_t + \alpha_2)(j\omega_t + \delta_{21})(j\omega_t + \delta_{22})} + \frac{\omega_t^2 \delta_{22}(0.5 - 0.18\,\cos[2\omega_x] - 0.32\,\cos[\omega_x])}{(j\omega_t + \alpha_2)(j\omega_t + \delta_{21})(j\omega_t + \delta_{22})} \right. \\
&\quad + \left. \frac{j\omega_t \delta_{21} \delta_{22}(0.5 - 0.18\,\cos[2\omega_x] - 0.32\,\cos[\omega_x])}{(j\omega_t + \alpha_2)(j\omega_t + \delta_{21})(j\omega_t + \delta_{22})} + \frac{2j\omega_t^2 \delta_{22}(0.17\,\sin[2\omega_x] + 0.33\,\sin[\omega_x])}{(j\omega_t + \alpha_2)(j\omega_t + \delta_{21})(j\omega_t + \delta_{22})} \right|. \quad (15)
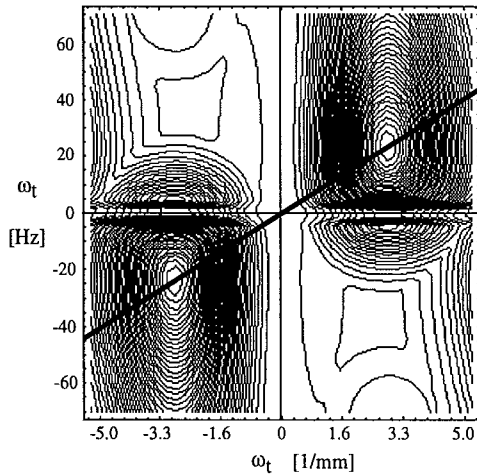\end{aligned}
$$

Fig. 11. The power spectrum of the velocity-tuned filter in (15).

of 10 mm^{-1}. This filter is realized using $g_2(\omega_x)$ and $g_2(\omega_t)$. Fig. 11 shows a plot of the power spectrum of the filter in (15). From the plot, and noting that 2π corresponds to 10 mm^{-1}, a theoretical velocity tuning of $+8.8$ mm/s is obtained. Similar plots provide the preferred velocity for all 18 filters per dimension. Note that the same tuning speed is obtained but in opposite direction for Left and Right filters (from Fig. 10). A mosaic of nine such filters is formed per direction per dimension, to cover a maximum spatiotemporal bandwidth of $\omega_t = 666.67$ rads/s and $\omega_x = 5$ mm^{-1}. The upper limit of the temporal bandwidth is controlled by δ_{m1}, while the maximum spatial frequency is limited by the spatial sampling rate of the silicon retina in (15), shown at the bottom of the previous page.

V. HARDWARE IMPLEMENTATION

A. Overview

A general-purpose analog neural computer and a silicon retina are used to realize the two 1-D visual-motion detection pixel. The silicon retina performs 40×40 edge detection and thresholding at the focal plane in parallel and continuous time. It is implemented in a 2-μm n-well CMOS process with a pixel pitch of 100-μm in both X and Y. The retina is scanned at >1 MHz and outputs >25 K binary frames/s in a row parallel manner. A demultiplexer board is used to present a 15×15 subimage to the neural computer virtually in parallel and continuous time due the high frame rate and 1-ms response time of the silicon retina. The 15×15 subimage is smoothed, in the appropriate direction, to create two 1-D images of 7×1 and 1×7 pixels. Two 1-D motion detection neural networks are realized using the 1-D images.

B. General Purpose Analog Neural Computer

The neural computer is intended for fast prototyping of neural-network-based applications. It offers the flexibility of programmability, combined with the real-time performance of a custom parallel hardware system [18]. It is modeled after the biological nervous system, i.e., the cerebral cortex, and consists of electronic analogs of neurons, synapses, synaptic

time constants and axon/dendrites. The hardware modules capture the functional and computational aspects of their biological counterparts. The main features of the system are: configurable interconnection architecture, programmable neural elements, modular and expandable architecture, and spatiotemporal processing. These features make the network ideal to implement a wide range of network architectures and applications.

The system, shown in part in Fig. 12, is constructed from three types of modules (chips): 1) neurons; 2) synapses; and 3) synaptic time constants and axon/dendrites. The neurons have a piecewise linear rate coded transfer function with programmable (8 bit) threshold and minimum output at threshold. The synapses are implemented as a programmable resistance whose values are variable (8 bit) over a logarithmic range between 5 kΩ and 10 MΩ. The time constant, realized with a load-compensated transconductance amplifier, is selectable between 0.5 ms and 1 s with a 5-bit resolution (a bypass mode is also available). The axon/dendrites are implemented with an analog cross-point switch matrix. The neural computer has a total of 1024 neurons, distributed over 64 neuron modules, with 96 synaptic inputs per neuron, a total of 98 304 synapses, 8192 time constants and 589 824 cross point switches. Up to 3072 parallel buffered analog inputs/outputs and a neuron output analog multiplexer are available. A graphical user interface software, which runs on the host computer, allows the user to symbolically and physically configure the network and display its behavior [19]. Once a particular network has been loaded, the neural network runs independently of the digital host and operates in a fully analog, parallel, and continuous-time fashion.

C. Neural Implementation of Spatiotemporal Filters

The output of the silicon retina is presented to the neural computer to implement the oriented spatiotemporal filters. The first layer of processing, realized with converging synaptic connections, compresses one 2-D image into two 1-D images through the Gaussian spatial prefiltering. The first and second derivatives of Gaussian functions are chosen to implement the odd and even spatial filters, respectively. Synaptic weights are used for the kernels' coefficients. Three parallel channels with varying spatial scales are implemented for each dimension. Within each channel, the spatial filters are subsequently fed to three sets of parallel temporal filters, which also have varying temporal tuning. Hence, six nonoriented pairs of spatiotemporal filters are realized for each scale. Six quadrature pairs of oriented filters are realized by summing and differencing the nonoriented pairs. The quadrature pairs are rectified and combined to produce the velocity-tuned spatiotemporal filters. Lastly, the responses of the filters tuned to opposite motion are sharpened with cross inhibition. This step is used to further reduce the response of the null direction filter, resulting in lower aliasing effects. Without this cross inhibition, the null direction responds slightly, but simultaneously, to opposite motion due to the discrete nature of the spatial filters. This is evident in the Fig. 11, where spectral energy in the opposite quadrants of the preferred motion is visible. Notice, however, that the
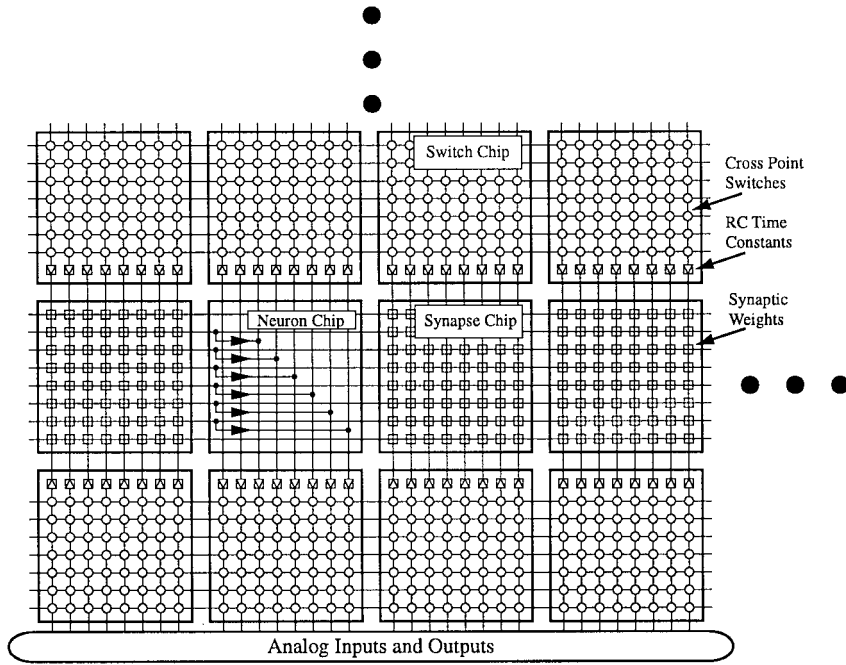
Fig. 12. The block diagram of the neural-network computer.

contour lines are sparser in the null quadrants, implying small responses at low speed. This is also observed in the measured data as shown by Fig. 17. Fig. 13 shows a schematic of the neural circuitry used to implement the velocity-tuned filters.

The pixel is composed of 1×7 and 7×1 1-D detectors with 15×7 and 7×15 receptive fields, respectively. Since the outputs of the spatial and temporal filters can be negative, and the neurons do not output negative values, they are offset to rest at half of full scale. The total number of neurons used to implement this pixel (nine oriented filters per direction per dimension) is 360, the number of synapses (including offsets) is 2900 and the number of time-constants is 144. The time-constants range from 0.75 to 200 ms. Once the networks have been programmed into the VLSI chips of the neural computer, the system operates in full parallel and continuous-time analog mode. This system realizes a silicon model for biological visual-motion measurement, starting from the retina to the visual cortex.

VI. EXPERIMENTAL RESULTS

Since the networks for the two dimensions are identical (except for expected variations among the circuits and chips), detailed results will be presented for only one of the dimension. The outputs of the neurons can be sampled at 1 MHz by an on-chip analog multiplexer and a high speed A/D card. The neuron chips have provisions that allow only selected chips to be sampled. Hence, the sampling period of each neuron ranges from 16 μs (only one chip is selected) to 1024 μs (all chips are selected).

The responses of the three quadrature pairs of spatial filters as a point (white spot on a black background) moves with a constant on-retina X-velocity of 1 cm/s (100 pixels/s) across their receptive fields are shown in Fig. 14. The silicon retina also contains photodiodes at the borders of the imaging array
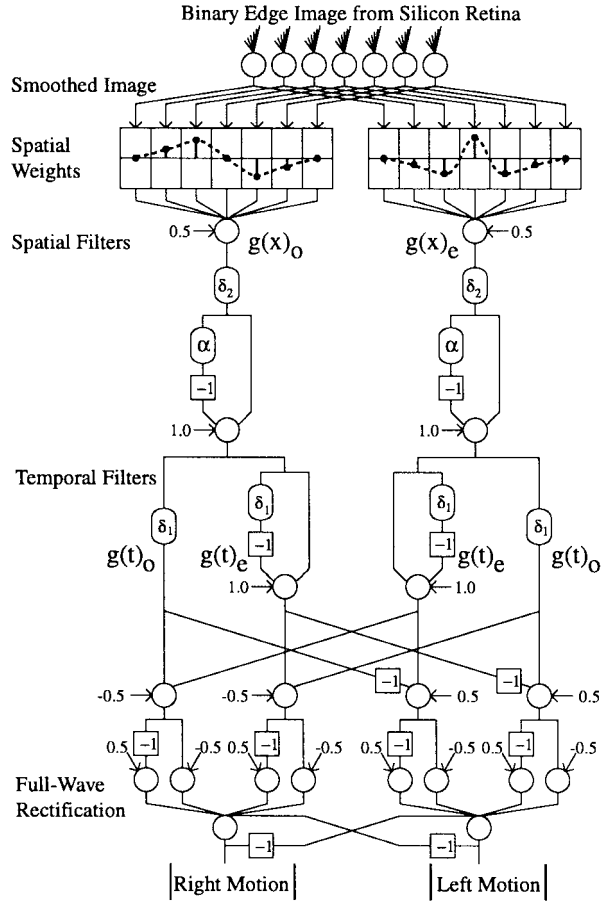

Fig. 13. Neural networks for velocity tuning are realized using synaptic weights as the coefficients of the spatial filters and synaptic time-constants for the temporal filters. The neurons have half-wave rectified transfer functions.

that are used to measure the focal-plane speed of the stimulus. The x-axis represents time; however, it can also represent space for the point moving at constant speed. As expected, the
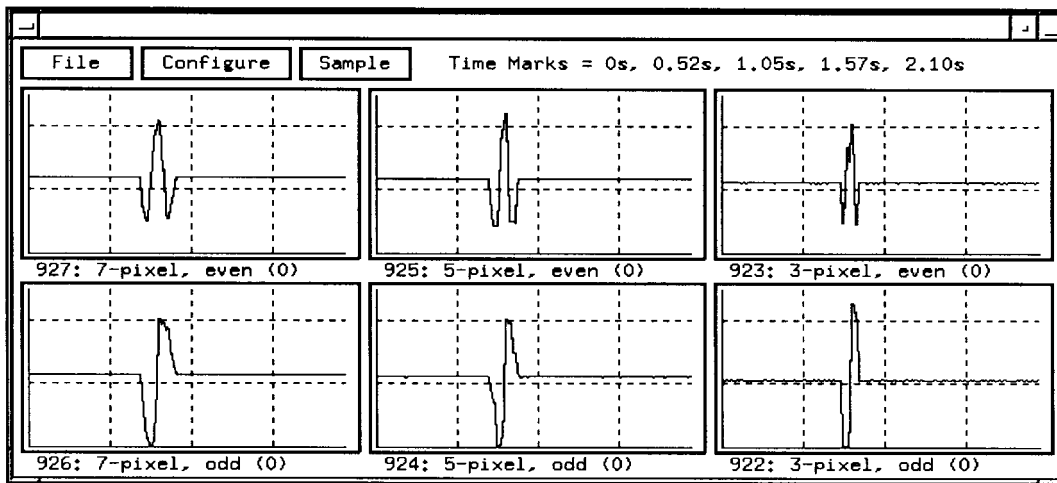
Fig. 14.   The top traces show the responses of the three even spatial filters for a point moving at 1 cm/s (100 pixels/s). The bottom traces show the responses of odd filters. The spatial scale decreases from left to right.
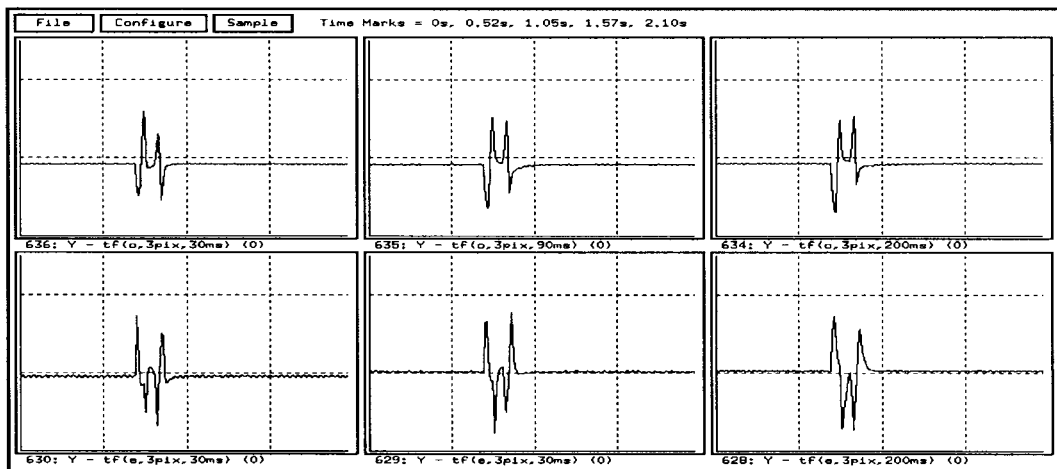

Fig. 15.   The outputs of the odd nonorientation selective filters (top) $g_e(x)g_o(t)$ and (bottom) $g_o(x)g_e(t)$ are shown. The temporal scale increases from let to right. Constructive and destructive interference produce directional selectivity.

filters with wider spatial support are broader than their higher frequency counterparts. Since all three filters are implemented on the same image patch and are centered at the same location, all filters respond simultaneously.

Fig. 15 shows the output of pairs of nonoriented spatiotemporal filters for the same point. The traces are sampled at the output of the temporal filters. The top row shows the outputs of the highest frequency even spatial filters through the three odd temporal filters, i.e., $g_e(x)g_o(t)$, resulting in odd nonoriented spatiotemporal filters. The bottom row shows the outputs of the highest frequency odd spatial filters through the three even temporal filters, i.e., $g_o(x)g_e(t)$, also resulting in odd nonoriented spatiotemporal filters. The temporal frequency tuning of these filters decrease from left to right. The sum and difference of the nonoriented filters create the velocity-tuned filters.

Fig. 15 also provides interesting insight into how the motion detection procedure is obtained with these filters. Notice that for a particular quadrature pair of temporal filters, the outputs are similar, but 180° out of phase. Hence, summing the signals

will result in destructive interference, while differencing them will result in constructive interference. Therefore, the right motion signal will vanish, while the left signal will survive. If the point was moving in the opposite direction, the even spatial filters will produce the same output, but the sign of the odd spatial filters will be inverted. Hence, the top traces of Fig. 15 will be unchanged, while the bottom traces will be inverted. Consequently, the sum of the signals, i.e., right motion, will now survive, while the difference, i.e., left motion, will vanish. In this way, the direction selectivity of the filters is obtained. The speed selectivity is governed by the spatiotemporal frequency tuning of the filters.

Fig. 16 shows the outputs of the left and right motion detectors for the moving point. The figure shows the rectified outputs of the velocity selective spatiotemporal filters. Because the point is moving to the left at 1 cm/s (100 pixels/s), all the right motion detectors are relatively silent. In contrast, the left motion detectors are active. A filter's response to the motion is given by the windowed average of its signal. The host computer calculates the average of the outputs of the
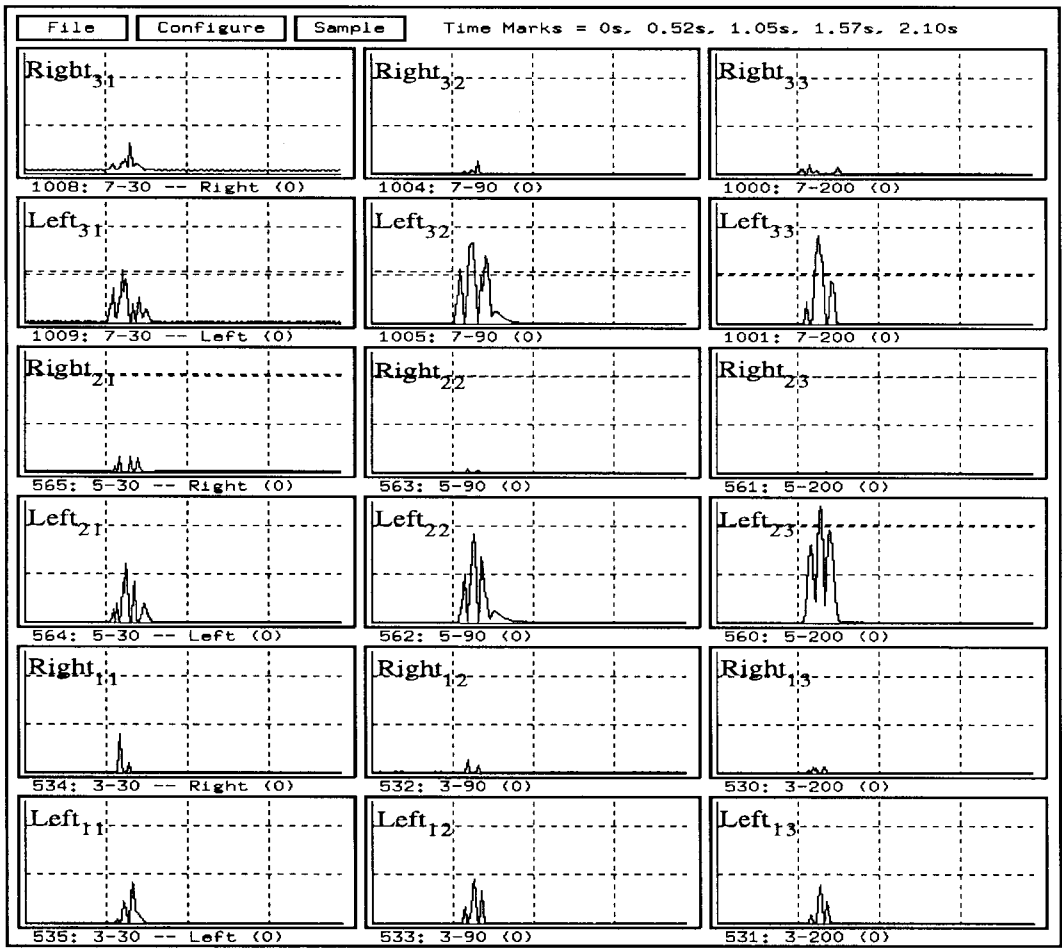
Fig. 16. The responses of 18 X-velocity selective filters are shown for a point moving at 1 cm/s (100 pixels/s) to the left.
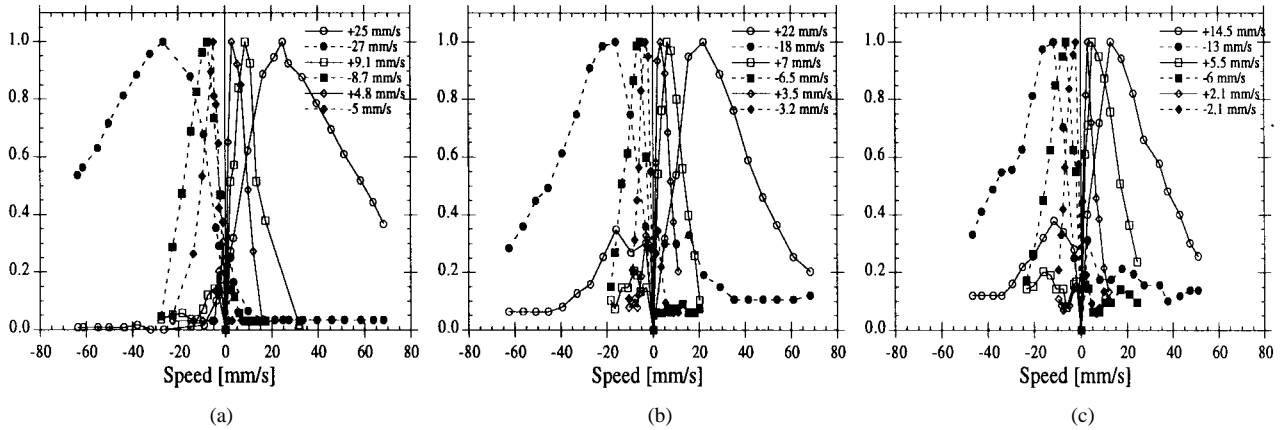

Fig. 17. Plots show the tuning curves for X-velocity selective filters as a function on on-chip speed (1 mm/s = 10 pixels/s). (a) 7-pixel detectors. (b) 5-pixel detectors. (c) 3-pixel detectors.

motion neurons after each data collection cycle. The average is computed using all the samples above a threshold (typically 10 mV). The length of the cycle is set by the user, and is bounded by the time required to read all the motion detectors once and the time to fill the memory of the host. Hence, the implicit representation of the velocity (the distribution of detector outputs) is available in continuous time, while the explicit report (time average and centroid determination) depends on the data collection cycle time. Typically, 2 s of data is collected before the explicit computation is done.

High contrast vertical and horizontal white lines on black background, moving at various speeds, are used to measure the tuning curves of the two 1-D filter sets. The silicon retina, however, can produce a binary image for edge contrasts as low as 10% [6]. The set of tuning curves for the X-velocity detectors is shown in Fig. 17. Similar curves are obtained for Y-velocity detectors. In the figure, the responses have been normalized. The variations in the responses are due to variations in the analog components of the neural computer. Some aliasing is noticeable in the tuning curves when there
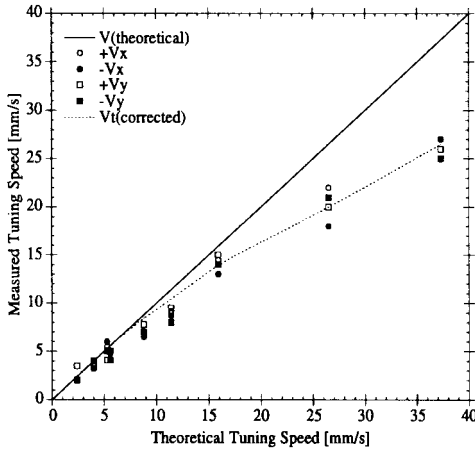
Fig. 18. A comparison of the theoretical and measured tuning speeds shows that the dynamics of the silicon retina (dotted line) must be considered in order to improve agreement for large speeds.
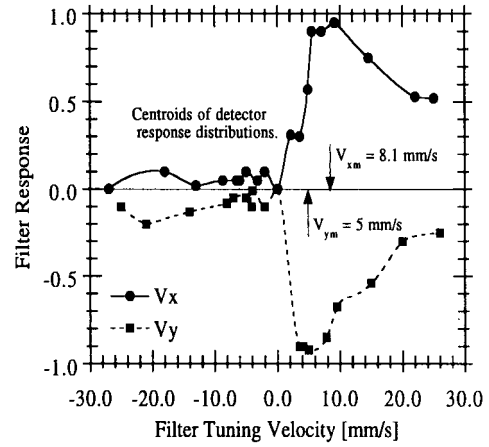

Fig. 19. The centroid of the detector outputs gives the explicit velocity of the point. Correct motion is $v_x = 8.66$ mm/s (86.6 pixels/s) and $v_y = 5$ mm/s (50 pixels/s).

is a minor peak in the opposite direction. This results from the discrete properties of the spatial filters, as can also be observed in the theoretical plot in Fig. 11. The additional fact that the minor peaks in the null direction occur at low speeds is also consistent with theory. Cross inhibition among oppositely tuned detectors keeps these aliasing effects small compared to the responses in the preferred direction. One could employ shunting inhibition to completely suppress these effects, however, this was not done here. Similar curves are obtained for the filters tuned to Y motion.

The theoretical tuning velocities, as indicated by (15) and Fig. 11, are compared to the measured values for both dimensions in Fig. 18. The filters tuned to low speed have good agreement with the theoretical value, however, as the tuning speed increases, so does the discrepancies. The measured values are consistently lower than the theoretical values. This can be explained by the observation that the theoretical tuning speed is obtained by the ratio of the temporal and spatial tunings of the motion filters (ω_t/ω_x). The temporal tuning is set by the poles of (14). The silicon retina has a response time which is comparable to the smallest time constant in the filters, i.e., $\delta_{12} = 0.75$ ms and $\tau_{sr} = 1$ ms (the time constant τ_{sr} is due to the slow tuning off response time of a photoreceptor when a dark image arrives and is a very weak function of the previous pixel brightness. The turning on response time is at least an order of magnitude smaller and decreases as intensity increases). Hence, an additional pole at 1 krad/s is required in the spatiotemporal filters. This has the effect of reducing the temporal tuning of the higher speed filters, as δ_{m2} approaches τ_{sr}, while not affecting the slower speed filters. In Fig. 18, the dotted line, labeled $v_{t(\text{corrected})}$, shows the corrected theoretical speed tuning if the dynamics of the silicon retina is taken into account. Better agreement is obtained. Other variations can be explained by mismatches in the circuit components. All subsequent measured explicit velocity computations use the measured tuning velocity of each oriented spatiotemporal filter.

The explicit velocity of the stimulus is given by the centroid of the distribution of the responses of the two 1-D detectors.
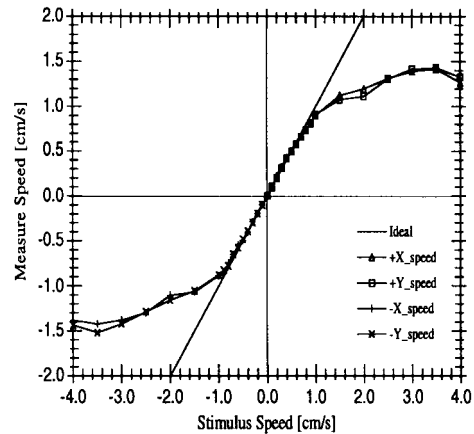

Fig. 20. The comparison of the measured and actual velocity shows that the distribution provides a good estimate at small speeds, but saturates and decreases toward zero for large speeds.

Because we do not have *a priori* knowledge of the direction of motion in real situations, the centroid of all the responses are considered. Fig. 19 shows the distribution of activities for both X and Y motion filters for a stimulus consisting of a bright spot on a dark background moving at 1 cm/s at 30° to the horizontal. Applying (5) yields $v_{xm} = 8.1$ mm/s (81 pixels/s) and $v_{ym} = 5.00$ mm/s (50 pixels/s), compared to $v_{xc} = 8.66$ mm/s (86.6 pixels/s) and $v_{yc} = 5$ mm/s (50 pixels/s). With more filters, the accuracy can be further improved. The effects of having too few filters can be seen in Fig. 20. At low speeds, all 18 filters contribute toward the computation of the explicit velocity. Hence, the measured value is fairly accurate. As the number of responsive filters decrease with increasing speed, the accuracy of the measured velocity also decreases. The measured response saturates and eventually drops to zero as the speed is increased further. At first glance, one would expect the speed at which the measurement saturates to be the tuning velocity of the fastest filter. Since our filters are realized with real circuit components, mismatches and the discrete properties of the spatial filters allow a small response to persist, especially in the null direction where aliasing is
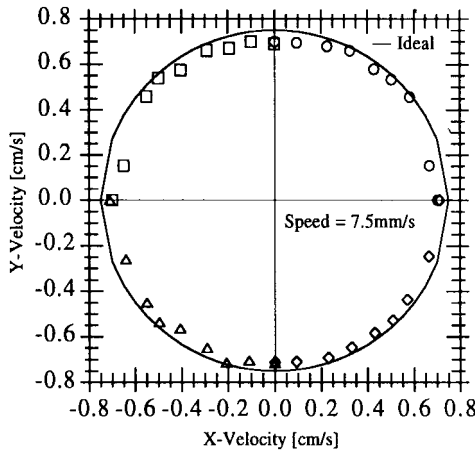
Fig. 21. The measured velocity components for a point moving at constant speed of 7.5 mm/s (75 pixels/s) at various angles highlights the increasing errors of the explicit motion detection method for larger speeds.

already present. Consequently, the computed centroid is lower than expected. Using the plot as a calibrating curve, higher speeds can still be estimated, provided that the stimulus is limited to the monotonic part of the curve.

To investigate the behavior of the pixel for 2-D motion, a bright point moving at 7.5 mm/s (75 pixels/s) at various angles is presented to the system. This point is representative of a variety of stimuli since the silicon retina creates a binary image of edges for the neural computer. Hence, all images seen by the motion detection filters are either binary points, lines or bars. The explicit velocity reported by the pixel is plotted in Fig. 21. The actual velocity of the point is plotted as the solid line. The speed of 7.5 mm/s is chosen to show the deviation of the measured motion from the ideal as the speed increases. The plot shows that the measured motion is more accurate for smaller velocity components produced by motion of ±45° to the horizontal. At the 90° and 0°, the measurement diverges more noticeable from the ideal. The measured dynamic range of the pixel for 2-D motion of a point is then given by Figs. 20 and 21.

Last, the effect of the aperture problem on the pixel is investigated. To simplify matters, a long bright line on a dark background, oriented at various angles to the horizontal, moving a constant velocity of 5 mm/s in the X-direction is presented to the pixel. The correct 2-D motion for this stimulus is $(v_x, v_y) = (5 \text{ mm/s}, 0)$. With the aperture problem taken into account, a 2-D normal vector is obtained, which varies with the orientation of the line. The components of this vector are plotted in Fig. 22. As expected, the X-component is maximum at 90° and vanishes at 0°. The Y-component peaks at 45° but is zero at both 90° and 0°. If a two 1-D detector is used, the X-velocity is correct until the orientation of the line approaches the 0°, at which point the X-velocity vanishes according to the aperture problem. The Y-component, however, displays a rapid increase as the orientation of the line approaches 90°. This error is due to the point sampling property of the 1-D motion detector orthogonal to the direction of motion, as explained in Section III-B. To demonstrate that this effect is measurable, the
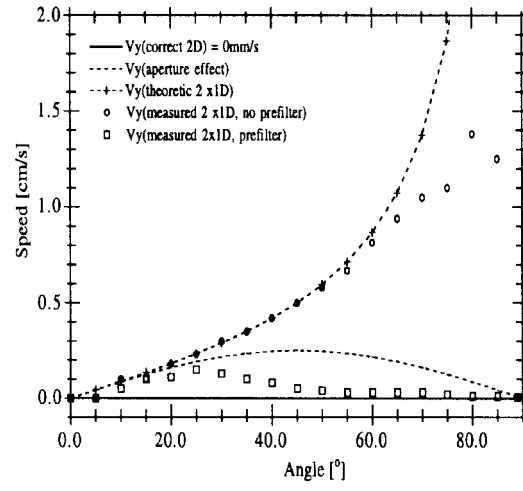
(a)
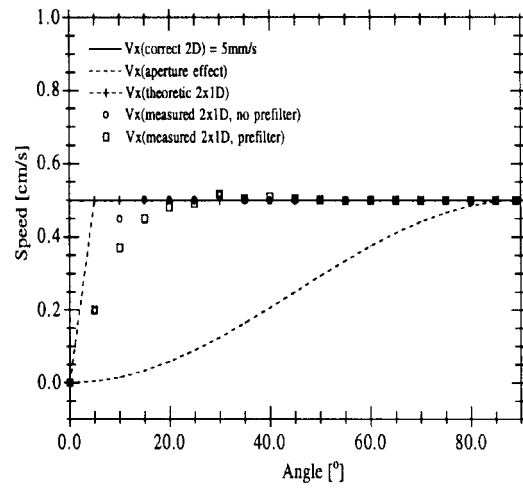
(b)

Fig. 22. The plot shows the effects of the aperture and two 1-D motion detection problems for a long line oriented at various angles, moving at $v_x = 5$ mm/s (50 pixels/s). Spatial presmoothing helps to measure the more accurate, but aperture-limited 2-D velocity. (a) Motion estimation for a long-oriented line. (b) Motion estimation for a long-oriented line.

smoothing prefilters are removed from the pixel. The recorded motion is plotted in Fig. 22. As expected, the X-component is correct and aperture limited at small angles, while the Y-component increases with orientation angle. As the line approaches vertical, the detector starts to saturate and return to zero. When the prefilters are used, the large Y-component error is virtually eliminated. At small angles ($<30°$), the X-component approaches zero because of the aperture problem and the tapered smoothing filter that reduces edge effects. Edge effects exist because the finite size smoothing window causes a moving edge to be seen in the 1-D image when the line enters or leaves the receptive field of the receptor. The small filter coefficients at the edges reduce this effect. Edge effects produce the only measurable X-motion for a line with small orientation angle. Above 20°, the 1-D X-motion detector receives a strong signal and produces the correct measurement. Y-motion detection for small angles is also influenced by aperture and edge effects to produce

a small velocity measurement. Motion due to edge effects depends on the tangent of the orientation angle; the measured Y-motion increases slightly with angle. Above $25°$, the smoothing prefilter begins to suppress all features in the Y-direction, causing the measured motion to drop to zero. As a result, the escalating error, observed with the two 1-D motion detector with no prefilter, is not present. Figs. 21 and 22 show that the two 1-D motion detectors with orthogonal spatial presmoothing produces much better results than the strict two 1-D detectors, has a reduced aperture effect and produces reasonable 2-D motion estimates in some cases. For vertical and horizontal lines, only the aperture-limited measurement is obtained.

VII. CONCLUSION

A two 1-D image motion-estimation pixel based on spatiotemporal feature extraction has been implemented in VLSI hardware using a general-purpose analog-neural computer and a silicon retina. The neural circuits capitalize on the temporal processing capabilities of the neural computer. The spatiotemporal feature-extraction approach is based on the 1-D cortical motion-detection model proposed by Adelson and Bergen, which was extended to 2-D by Heeger. To reduce the complexity of the model and to allow realization with simple sum-and-threshold neurons, we further modify the 2-D model by working with a binary edge image, by placing filters only in the ω_x-ω_t and ω_y-ω_t planes, and by replacing the required quadratic nonlinearity with full-wave rectification. These modifications do not affect the performance of the 1-D model, and approximate the 2-D model in some cases. Measured results agree with theoretical expectations. While this technique of image motion detection requires too much hardware for focal plane implementation, our results show that it is realizable when a silicon "brain," with large numbers of neurons and synaptic time constant, is available. This is very reminiscent of the biological master.

REFERENCES

[1] E. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer.,* vol. A2, pp. 284–99, 1985.
[2] J. van Santen and G. Sperling, "Temporal covariance model of human motion perception," *J. Opt. Soc. Amer.,* vol. 1, no. 5, pp. 451–73, 1984.
[3] W. Reichardt, "Autocorrelation: A principle for the evaluation sensory information by the central nervous system," in *Sensory Communication.* New York: Wiley, 1961.
[4] J. Maunsell and D. Van Essen, "Functional properties of neurons in middle temporal visual area of the Macaque monkey—I: Selectivity for stimulus direction, speed and orientation," *J. Neurophysiol.,* vol. 49, no. 5, pp. 1127–1147, 1983.
[5] C. Koch and H. Li, Eds., *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits.* New York: IEEE Press, 1995.
[6] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "A focal plane visual motion measurement sensor," *IEEE Trans. Circuits Syst. II,* vol. 44, pp. 55–66, Jan. 1997.
[7] R. Sarpeshkar, J. Kramer, and C. Koch, "Analog VLSI architectures for motion processing: From fundamentals to system applications," *Proc. IEEE,* vol. 84, July 1996.
[8] R. Harrison and C. Koch, "An analog VLSI model of the fly elementary motion detector," in *Advances in Neural Information Processing Systems 10,* M. Jordan, M. Kearns, and S. Solla, Eds. Cambridge, MA: MIT Press, 1998, pp. 880–886.
[9] A. Yakovleff and A. Moini, "Motion perception using analog VLSI," *J. Analog Integ. Circuits and Signal Processing.* Norwell, MA: Kluwer, 1998, vol. 15, no. 2, pp. 183–200.
[10] D. Heeger, E. Simoncelli, and J. Movshon, "Computational models of cortical visual processing," in *Proc. Nat. Acad. Sci.,* 1996, vol. 92, no. 2, p. 623.
[11] D. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Amer.,* vol. 4, no. 8, pp. 1455–1471, 1987.
[12] J. Van der Spiegel, D. Blackman, P. Chance, C. Donham, R. Etienne-Cummings, and P. Kinget, "An analog neural computer with modular architecture for real-time dynamic computations," *IEEE J. Solid-State Circuits,* vol. 27, pp. 82–92, Jan. 1992.
[13] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Phys.,* vol. 160, pp. 106–154, 1962.
[14] N. Grzywacz and A. Yuille, "A model for the estimate of local image velocity by cells in the cortex," *Proc. R. Soc. London B,* vol. 239, pp. 129–161, 1990.
[15] E. Simoncelli, "Distributed representation and analysis of visual motion," Ph.D. dissertation, Dept. Elect. Eng., MIT, Cambridge, MA, 1993.
[16] R. Etienne-Cummings, J. Van der Spiegel, C. Donham, S. Fernando, R. Hathaway, P. Mueller, and D. Blackman, "A general purpose analog neural computer and a silicon retina for real time target acquisition, recognition and tracking," in *Proc. CAMP'93,* M. Bayoumi, L. Davis, and K. Valavanis, Eds., 1993, pp. 48–58.
[17] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biol. Cybern.,* vol. 60, pp. 79–97, 1988.
[18] P. Mueller, J. Van der Spiegel, D. Blackman, C. Donham, and R. Etienne-Cummings, "A programmable analog neural computer with applications to speech recognition," in *Proc. Comput. Info. Sci. Symp.,* May 1995. Baltimore, MD: Johns Hopkins Press.
[19] C. Donham, "Real time speech recognition using a general purpose analog neurocomputer," Ph.D. dissertation, Dept. of Electrical Engineering, Univ. Pennsylvania, Philadelphia, PA, 1995.

Ralph Etienne-Cummings (S'94–M'98) received the B.Sc. degree in physics in 1988 from Lincoln University, PA. He also received the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Pennsylvania in 1991 and 1994, respectively.

Currently, he is an Assistant Professor of Electrical and Computer Engineering at the Johns Hopkins University, Balitmore, MD. His research interest includes mixed-signal very large scale integration systems, computational sensors, computer vision, neuromorphic engineering, smart structures, mobile robotics, and robotics-assisted surgery.

Dr. Etienne-Cummings is a recipient of the National Science Foundation Career Development Award.

Jan Van der Spiegel (M'72–SM'90) received the engineering degree in electro-mechanical engineering and the Ph.D. degree in electrical engineering from the University of Leuven, Belgium, in 1974 and 1979, respectively.

From 1980 to 1981, he was a Post-Doctoral Fellow at the University of Pennsylvania, then an Assistant Professor of Electrical Engineering from 1981 to 1987.. In 1987, he became an Associate Professor, and in 1995, a Full Professor of Electrical Engineering at the University of Pennsylvania, Philadelphia. He is currently Chairman of the Department of Director of the Center for Sensor Technology. His research interests include analog and digital integrated circuits for intelligent sensors, data acquisition, sensory dataprocessing systems, and acoustic-phonetic feature extraction for automatic speech recognition. He holds the UPS Distinguished Education Term Chair. He is the Editor for N&S America of *Sensors and Actuators*, and is on the editorial boards of the *International Journal of High Speed Electronics* and the *Journal of the Brazilian Microelectronics Society*.

Dr. Van der Spiegel has served on several IEEE Program Committees and is currently on the Program and Executive Committees of the ISSCC. He was the recipient of the Bicentennial Chair of the Class of 1940, the Presidential Young Investigator Award, and the S. R. Warren and C. & M. Lindback Award for distinguished teaching. He is a member of Phi Beta Delta and Tau Beta Pi.

Paul Mueller received the M.D. degree from Bonn University, Germany.

Since 1953, he has worked in molecular and systems neuroscience and has been involved in theoretical studies and hardware implementation of neural networks since the early 1960's. He was with Rockefeller University and the University of Pennsylvania and is currently Chairman of Corticon, Inc., Philadelphia, PA.