Design of Linear Cellular Neural Networks for Motion Sensitive Filtering

Bertram E. Shi, Tamás Roska, and Fellow, IEEE, Leon O. Chua, Fellow, IEEE

Abstract—Recently, several researchers have proposed using spatio-temporal filters for image motion analysis. For example, the optical flow field can be calculated from the output of a set of spatio-temporal filters. Some of the most popular spatio-temporal filters are the space-time Gabor filters, obtained by convolving a time varying image with a space-time Gabor function. Based on the cellular neural network paradigm, we propose a new architecture for spatio-temporal filtering called a CNN filter array and demonstrate the design of CNN filter arrays for motion sensitive filtering. One advantage of this approach to motion sensitive filtering is that a global convolution in space and time can be performed by using only spatially local interconnections and exploiting the continuous time dynamics of the CNN filter array. No storage of any past image frames is required.

I. INTRODUCTION

RECENTLY, several researchers have proposed using spatio-temporal filters for image motion analysis. This interest has been motivated by several different factors. On the physiological side, it has been discovered that motion sensitive cells in the primary visual cortex are sensitive to certain directions and spatio-temporal frequencies [1]-[3]. On the computational side, several researchers have pointed out that image motion at a specified velocity can be characterized as an orientation in the space-time domain [4]-[6]. This orientation manifests itself in the spatio-temporal frequency domain as skewing of the image spectrum so that all of the energy of a pattern translating at a velocity \( \mathbf{v} \) is concentrated in a specific region of the spatio-temporal frequency space. The spatio-temporal filtering approach to image motion analysis uses the outputs of appropriately designed filters tuned to these regions, as input to later processing stages.

For example, the problem which has attracted the most attention in this area is the derivation of the optical flow. The optical flow is defined to be the two-dimensional velocity field in the image plane resulting from relative motion between the observer and objects in the environment. Because this flow field contains information about the structure of the environment, there exist many algorithms that attempt to recover information about the environment given the optical flow. Several researchers have proposed different methods for computing the optical flow by comparing the outputs of a set of spatio-temporal filters tuned to different areas of the spatio-temporal frequency domain [7]-[10].

One of the spatio-temporal filters for image motion analysis which has attracted the most attention is the space-time Gabor filter, which we discuss in greater detail in Section II. One disadvantage of using Gabor filters for image motion analysis is that the velocity at which a sine wave grating must translate in order for the Gabor filter to have its maximal response increases as the spatial frequency of the grating decreases [9, fig. 2].

Typically, space-time Gabor filters have been implemented by convolving an image sequence by a space-time Gabor function. Digital computation of the convolution requires that many image frames be digitized and stored for later recall. Since each frame is already a two-dimensional array of pixel intensity values, the memory storage requirements are quite high. In order to operate at speeds on the order of speeds of objects moving in the environment, a system implementing the Gabor space-time filters also requires a wide bandwidth to handle the large amount of data which must be manipulated.

In this paper, we propose an alternative approach to implementing these types of filters. We assume that the image that we wish to filter is sampled in space by a rectangular \( n_0 \) by \( n_1 \) array of photodetectors, but not sampled in time. Our approach is to design an \( n_0 \) by \( n_1 \) array of continuous time temporal filters, each of which is associated with one pixel in the image. These filters are spatially locally interconnected in such a way that when the filter array is input with the set of analog waveforms from the photodetector array, only the desired spatio-temporal frequencies in the input will appear at the output of the array. Since our architecture is based upon the cellular neural network (CNN) paradigm [11], [12] as introduced by Chua and Yang, we will refer to it as a CNN filter array.

This approach attempts to overcome the previously mentioned disadvantages of Gabor filters. First, these filters can be designed so that each filter has maximal response at the same velocity for translating sine wave gratings of most spatial frequencies. Second, this method does not require any explicit storage of any frames of the input image. Instead, we use the natural dynamics of the filter array to perform the desired convolution in both time and space. Finally, the image does not need to be digitized as the filter operates directly on the analog signal coming from the sensor array.

Fleet and Jepson [13] have described work along a similar vein to the results presented here. They define a hierarchical computational structure based upon layers of linear processing.
units, which they show can be used for orientation and velocity sensitive filtering. The first layer of their hierarchical computation structure is very similar to a CNN filter array in which the temporal filters operate in discrete time. In fact, it is slightly more general. Since their biologically motivated first layer displays no orientation or velocity selectivity, they must add additional processing layers which operate on the output of this first layer. Our work demonstrates that with a structure simpler than the first layer of Fleet and Jepson’s hierarchical architecture, we can achieve an orientation and velocity selective filter.

Motion detection algorithms have also been proposed previously for the original CNN paradigm and for the CNN with delay type templates [14]. These algorithms have several drawbacks which this work attempts to address. The major drawback to all of these algorithms is that are based upon the assumption that the images are black and white. Although this assumption simplifies the task of motion detection, it is not a very realistic assumption for real world images. CNN arrays for motion detection which operate on a continuously varying input require the implementation of delay in the processing. This may be difficult because the delay must be on the order of the motion in the image, which is an order of magnitude slower than the dynamics of the processing. Finally, these methods are based on local pixel differences rather than on the global properties of translating images. Therefore, we would expect that any implementation of these types of algorithms for gray level images will be extremely sensitive to noise.

The paper is organized as follows. After reviewing some preliminaries of spatio-temporal filtering and some of the notation which we will use in this paper in Section II, we introduce the CNN filter array architecture in Section III and show that there exist a spatio-temporal frequency response and a spatio-temporal transfer function describing the mapping from the input to the output of the array. In Section IV, we begin our design of CNN motion sensitive filter arrays by starting the design of a one-dimensional motion sensitive CNN filter array. However, we cannot completely specify the interconnections based on the spatio-temporal frequency response alone. One of the most important considerations which must also be taken into account is stability. As with any analog filter which incorporates feedback, there is a possibility that the filter may be unstable. We address this issue in Section V where we give necessary and sufficient stability criteria for these types of CNN’s. These stability criteria take into account the massive redundancy resulting from the assumption of spatial invariance, resulting in simple graphical Nyquist type stability tests. Armed with these results, we complete the design of the one-dimensional CNN motion sensitive filter array in Section VI and give examples of simulation results. Finally, we illustrate how to extend these results to two dimensional images in Section VII before summarizing our results in Section VIII.

II. PRELIMINARIES AND NOTATION

In this section, we briefly discuss the theory underlying spatio-temporal filtering for image motion analysis and introduce one popular filter, the space-time Gabor filter, which has been used by many researchers. We then introduce the notation we use in the following sections to describe multidimensional arrays. The advantage of this notation is that the notation for one-dimensional and higher dimensional arrays is identical.

Consider the special case of a one dimensional sine wave grating translating at speed \( v \). In this case, the time varying image intensity at point \( x \) and time \( t \) is given by

\[
\hat{i}(x, t) = \sin(\Omega_t(x - vt)).
\]

See Fig. 1. The image intensity oscillates both in space and time with frequencies \( \Omega_x \) and \( \Omega_t \), i.e., \( \hat{i}(x, t) = \sin(\Omega_x x + \Omega_t t) \). Equating the arguments in the two expressions yields \( \Omega_t = -v\Omega_x \).

This relationship between the spatial and temporal frequencies of an image undergoing uniform translation also holds for general images. Define \( \hat{i}(x, t) = \hat{i}_0(x - vt) \), where \( \hat{i}_0(\cdot) \) is the image intensity at time 0. Fourier transforming in space and time yields

\[
\hat{i}(\omega_x, \omega_t) = \int \hat{i}(x, t) e^{-j\omega_x x} e^{-j\omega_t t} dx dt = \hat{i}_0(\omega_x) 2\pi \delta(\omega_t + v\omega_x).
\]

The function \( \hat{i}(\cdot, \cdot) \) is the Fourier transform in time and space of the time-varying waveform \( i(\cdot, \cdot) \). The function \( \hat{i}_0(\cdot) \) is the spatial Fourier transform of the image intensity at time 0. In general, we will use the hat notation, \( \hat{\cdot} \), to denote transforms in
Fig. 2. (a) The even-phase Gabor function is a cosine function in space and time windowed a Gaussian function. (b) The spatio-temporal frequency response of the even-phase Gabor Filter consists of the sum of two Gaussian functions.

time, the upside hat notation, $\hat{x}$, to denote transforms in space, and the tilde notation, $\tilde{x}$, to denote transforms in both space and time. Equation (1) indicates that for uniform translation in a one-dimensional image plane, all of the energy in the signal lies along the line $\omega_t = -w_\omega x$ in the spatio-temporal frequency plane. In the two-dimensional case, all of the energy in the signal lies along a plane in the spatio-temporal frequency domain given by $\omega_t = -(\omega_x, \omega_y)$.

The essential idea behind image motion analysis using spatio-temporal filters is to design filters which are sensitive to spatio-temporal frequencies corresponding to different image velocities. One popular such filter is the even-phase Gabor filter, obtained by convolving the image in space and time by a spatio-temporal even-phase Gabor function $g_e(x,t)$, See Fig. 2(a). This filter has a spatio-temporal frequency response which is equal to the spatio-temporal Fourier transform of $g_e(x,t)$,

$$g_e(x, t) = \frac{1}{2\pi \sigma_x \sigma_t} e^{-\left(\frac{x^2}{2 \sigma_x^2} + \frac{t^2}{2 \sigma_t^2}\right)} \cos(\omega_x x + \omega_t t).$$

See Fig. 2(a). This filter has a spatio-temporal frequency response which is equal to the spatio-temporal Fourier transform of $g_e(x,t)$,

$$\hat{g}_e(\omega_x, \omega_t) = \pi^d e^{-\left(\frac{\omega_x^2}{2 \sigma_x^2} + \frac{\omega_t^2}{2 \sigma_t^2}\right)} \cos(\omega_x x + \omega_t t).$$

See Fig. 2(b). The frequency response consists of the sum of two Gaussian functions centered at $(\Omega_x, \Omega_t)$ and $(-\Omega_x, -\Omega_t)$. Therefore, given an image consisting of a translating sine wave grating with spatial frequency $\Omega_x$, this filter has its greatest response when the grating is oscillating in time with frequency $\Omega_t$, corresponding to a velocity equal to $-\Omega_t/\Omega_x$. Thus, changing the ratio of $\Omega_t$ and $\Omega_x$ changes the velocity sensitivity of the filter.

However, as we mentioned in the introduction, the maximal output of the Gabor filter for translating sine wave gratings of different spatial frequencies occurs at different velocities, depending upon the spatial frequency. This effect is due to the fact that if $\Omega_x \omega_x > 1$, the temporal frequency at which the maximum of the frequency response is maximal is approximately equal to the constant $\Omega_t$, for all spatial frequencies $\omega_x$. Therefore, the velocity at which the response is maximal is approximately equal to $\Omega_t/\omega_x$.

These linear filters result in an oscillating output which may be difficult to interpret for later processing steps. One way to handle this is to sum the squares of the outputs of two filters which have nearly the same magnitude spatio-temporal frequency response, but which are $90^\circ$ out of phase with each other. Adelson and Bergen [4], [5] propose to sum the squares of the outputs of the even-phase Gabor filter with the odd-phase Gabor filter, obtained by convolving the input with the odd-phase Gabor function

$$g_o(x, t) = \frac{1}{2\pi \sigma_x \sigma_t} e^{-\left(\frac{x^2}{2 \sigma_x^2} + \frac{t^2}{2 \sigma_t^2}\right)} \sin(\Omega_x x + \Omega_t t).$$

This sum of the squared outputs is denoted the motion energy.

In the following, we adopt the convention that all vectors and arrays will be indexed starting from zero. For a $d$-dimensional array, define the size of the array to be the $d$-dimensional vector $n = (n_0, \cdots, n_{d-1})$, where $n_i$ is the size of the array in the $i$th dimension. Define $I_d$ to be the set of multi-indexes which index this array, i.e.,

$$I_d = \{(i_0, \cdots, i_{d-1}) \mid i_c \in \{0, \cdots, n_c - 1\} \forall c \in \{0, \cdots, d - 1\} \}.$$

For example, an $n_0 \times n_1$ two-dimensional array has size $n = (n_0, n_1)$ and is indexed by

$$I_d = \{(i_0, i_1) \mid i_0 \in \{0, \cdots, n_0\}, i_1 \in \{0, \cdots, n_1\}\}.$$

For the purposes of defining multiplication and division between two elements of $I_d$, we shall consider a multi-index as a $d$-dimensional vector in $R^d$ and define multiplication between two elements of $R^d$ using the normal scalar product in $R^d$. We define division as the following mapping from $R^d \times R^d$ to $R^d$

$$\frac{\mathbf{i}}{\mathbf{k}} = \left(\frac{i_0}{k_0}, \cdots, \frac{i_{d-1}}{k_{d-1}}\right).$$

However, we define addition and subtraction between two elements of $I_d$ or between an element of $I_d$ and a $d$-dimensional vector in $Z^d$ somewhat differently to ensure that the sum is an element of $I_d$. For all $i, n \in Z$, define $i \mod n = m$ where $m \in \{0, \cdots, n - 1\}$ such that there exists an $l \in Z$ such that $i = l n + m$. Define addition between two elements of $I_d$, $\mathbf{i}$ and $\mathbf{k}$, by

$$\mathbf{i} + \mathbf{k} = (i_0 + k_0 \mod n_0, \cdots, (i_{d-1} + k_{d-1}) \mod n_{d-1}).$$

Define subtraction similarly. The same definitions hold if $\mathbf{i} \in I_d$ and $\mathbf{k} \in Z^d$. 

III. CNN FILTER ARRAYS

In this paper we will be considering a special type of CNN, which we will refer to as a CNN filter array. This architecture is similar to the generalized CNN architecture, except that it is purely linear [19]. CNN's are a general class of analog recurrent neural networks in which the "cells" or "neurons" are arranged in a regular array and are only interconnected with their nearest neighbors [11], [12], [20]. In the case of CNN filter arrays, each of these cells is simply a linear continuous time temporal filter whose input/output behavior is described by the transfer function \( g(s) \). A CNN filter array designed to process an \( n_0 \times n_1 \) pixel time varying image will consist of an \( n_0 \times n_1 \) array of such filters where the \( 
\)

The input to each filter is equal to the sum of a feedback term and a feedforward term. The feedback term is a weighted sum of the outputs of the filters in a small neighborhood of the filter. The feedforward term is a weighted sum of the intensities of the pixels in a small neighborhood of the associated pixel in the input. We will assume that the weighting factors are spatially invariant.

More specifically, denote the input and the output of the \( k \)th filter at time \( t \) by \( v(k, t) \) and \( y(k, t) \) respectively and the intensity of the associated pixel by \( u(k, t) \). For a given \( r, d \in \mathbb{Z}_+ \), define the \( r \)-ball in \( d \) space, \( B^d_r \), as

\[
B^d_r = \{(l_0, \ldots, l_{d-1}) \in \mathbb{Z}^d \mid |l_i| \leq r, i \in \{0, \ldots, d - 1\}\}
\]

The input to the \( k \)th filter at time \( t \) is given by

\[
v(k, t) = \sum_{l \in B^d_r} a(l) y(k + l, t) + \sum_{l \in B^d_r} b(l) u(k + l, t).
\]

Fig. 3 depicts a one-dimensional CNN filter array with \( r = 1 \).

Our definition of addition between a multi-index and a \( d \)-dimensional vector implies that the boundary conditions of this array are periodic. Thus, the cells on one edge of the array are connected to the cells at the opposite end of the array. Along with the transfer function of the filter, \( g(s) \), the feedback coefficients, \( A = \{a(l)\}_{l \in B^d_r} \), and the feedforward coefficients, \( B = \{b(l)\}_{l \in B^d_r} \), completely specify the array. We will refer to the \( A \) and \( B \) coefficients as the CNN cloning templates. In the one- and two-dimensional cases, we will specify these \((2r + 1)\) vectors or \((2r + 1) \times (2r + 1)\) matrices where the center elements correspond to the \( 0 \)th element of the cloning template.

This type of array is related to the original definition of CNN's defined in [10] and [11] in that if we set \( g(s) = R/(1 + sRC) \), where \( R \) is the parallel resistance in each cell and \( C \) is the value of the capacitor in each cell, the differential equation governing the outputs of the CNN filter array is the same as the equation governing the evolution of the state of the original CNN, if the output non-linearity in the original system is equal to the identity. This is equivalent to assuming that the input to the CNN is sufficiently small and the dynamics evolve in such a way so that the outputs of the CNN never enter the saturation region of the piecewise linear output non-linearity.

Due to the assumption of spatial invariance, there is a simple intuitive way to think about these types of arrays, which is mathematically justifiable. Consider a one-dimensional CNN filter array, as shown in Fig. 3, and assume that the array is infinite. Because of spatial invariance, the computation of the weighted feedforward and feedback sums is simply a discrete cross-correlation of the \( B \) and \( A \) cloning templates with the input and output, respectively. These operations have spatial frequency responses given by the complex conjugates of the discrete space Fourier transforms of the waveforms corresponding to the \( B \) and \( A \) coefficients. Thus, we can think of the system as the block diagram system shown in Fig. 4. The effect of the cross-correlation of the output with the \( A \) template is characterized by the spatial frequency transfer function

\[
a(\omega_k) = \sum_{l \in B^d_r} a(l) e^{j\omega l - \omega_k^2}.
\]

where \( \omega_k \) is the spatial frequency in radians per pixel. Similarly, the effect of the cross-correlation of the input with the \( B \) template is characterized by the spatial frequency transfer function

\[
b(\omega_k) = \sum_{l \in B^d_r} b(l) e^{j\omega l - \omega_k^2}.
\]

By naively writing the loop equations, we can derive what appears to be a spatio-temporal transfer function,

\[
\hat{h}(\omega_k, s) = \frac{g(s)b(\omega_k)}{1 - a(\omega_k)g(s)}.
\]

Similarly, substituting \( s = j\omega \) into the above we obtain a what appears to be a spatio-temporal frequency response,

\[
\hat{h}(\omega_k, j\omega) = \frac{g(j\omega)b(\omega_k)}{1 - a(\omega_k)g(j\omega)}.
\]
Assuming our intuition is correct, if the input of the CNN filter array is a moving sine wave grating, \( u(\vec{k}, t) = \sin(\omega_{t}t + \vec{\omega}_{x} \cdot \vec{k}) \), the output will be another sine wave grating of the same spatial and temporal frequencies, but scaled in amplitude and shifted in phase,

\[
y(\vec{k}, t) = [h(j\omega_{t}, j\omega_{x})] \sin(\omega_{t}t + \vec{\omega}_{x} \cdot \vec{k} + \vec{\beta}h(\vec{k}, j\omega_{t})).
\]

This intuition can be made more rigorous in a way that also takes into account the fact that in practice we will never have an infinite array. In the finite array case, the CNN linear filter array can be analyzed as an MIMO system by listing the inputs and outputs of the array as a vector. In the two-dimensional case, using the correct ordering, the matrices representing the cross correlations by the \( A \) and \( B \) templates coefficients are block circulant with circulant blocks [21]. Using the special properties of these types of matrices, it can be shown that the spatio-temporal transfer function and frequency responses of the finite array of size \( n \) with periodic boundary conditions can be obtained by sampling (3) and (4) at \( \vec{\omega}_{x} = 2\pi(\vec{k}/n) \) where \( \vec{k} \in \mathcal{K} \). Therefore, when discussing the spatio-temporal transfer function or frequency response, we will generally mean (3) and (4).

The existence of the spatio-temporal transfer function suggests a convenient way to think about the operation of the CNN filter array. The CNN filter array is essentially a set of decoupled temporal filters, each of which processes a single spatial frequency component of the input. If we take the spatial Fourier transform of the image waveform at each time \( t \), we obtain a time-varying set of Fourier coefficients. The temporal evolution of the Fourier component of the image waveform at frequency \( \vec{\omega}_{x} \) is specified by a complex scalar valued function of time. The CNN filter array filters this scalar function by a temporal filter that has a transfer function given by (3). Therefore, (3) can be thought of as a spatial frequency-dependent SISO temporal transfer function.

**Example 1: The Silicon Retina:** Mahowald and Mead have pointed out that the resistive grid of their silicon retina [22] computes a temporally and spatially weighted average of the photoreceptor outputs. A schematic diagram of a one-dimensional silicon retina is shown in Fig. 5. Roughly speaking, the temporal averaging is due to the capacitance associated with each node of the resistive grid and the spatial averaging is due to the effect of the resistors linking adjacent nodes in the grid. The space constant of the array, \( \alpha = \sqrt{RG} \), determines the amount of spatial smoothing. The interaction between the spatial and temporal averaging can be easily explained using the concept of the spatio-temporal frequency response.

We can define a CNN filter array which is the input/output equivalent to the resistive grid by setting the filter transfer function \( \tilde{y}(s) \) to be equal to the impedance of the nodal capacitance, \( 1/sC \), and the \( A \) and \( B \) templates equal to

\[
A = \begin{bmatrix} 1 & 0 \\ \frac{-2}{R + G} & 1 \\ \frac{1}{R} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & G \\ R & 0 \end{bmatrix}.
\]

Using this equivalent CNN filter array, the spatio-temporal frequency responses of the resistive grid and of the silicon retina can easily be computed. Substituting the actual values of \( \tilde{y}(s) \) and the \( A \) and \( B \) templates into (4), we obtain the spatio-temporal frequency response of the resistive grid,

\[
\tilde{h}(j\omega_{t}, \omega_{x}) = \frac{\tilde{y}(j\omega_{t}, \omega_{x})}{\tilde{u}(j\omega_{t}, \omega_{x})} = \frac{\omega_{t}RC + \alpha^{2} + 2(1 - \cos \omega_{x})}{\omega_{t}RC + \alpha^{2} + 2(1 - \cos \omega_{x})}.
\]

Since the output of the silicon retina is equal to the difference between the output of the photodetectors and the nodal voltages of the resistive grid, the spatio-temporal frequency response of the one-dimensional silicon retina is

\[
\tilde{h}(j\omega_{t}, \omega_{x}) = \frac{\tilde{y}(j\omega_{t}, \omega_{x})}{\tilde{u}(j\omega_{t}, \omega_{x})} = \frac{\omega_{t}RC + \alpha^{2} + 2(1 - \cos \omega_{x})}{\omega_{t}RC + \alpha^{2} + 2(1 - \cos \omega_{x})},
\]

which has magnitude

\[
|\tilde{h}(j\omega_{t}, \omega_{x})| = \sqrt{\left(\frac{(\omega_{t}RC)^{2} + 4(1 - \cos \omega_{x})^{2}}{((\omega_{t}RC)^{2} + (\alpha^{2} + 2(1 - \cos \omega_{x}))^{2}}\right)}.
\]

The spatio-temporal frequency response of the silicon retina offers a rigorous explanation of the effects observed by Mahowald and Mead. For example, they have noted that the time response of the silicon retina varies with the space constant, \( \alpha \), of the resistive grid. For \( \alpha \) low, a test flash of any limited size will produce a sustained output, but as \( \alpha \to \infty \), a test flash will have no sustained output. The sustained output is essentially the steady-state response, i.e., \( \omega_{t} = 0 \). Fig. 6(a), which plots the cross section of the spatio-temporal frequency response at \( \omega_{t} = 0 \) for varying values of \( \alpha \), shows that the amplitude of the spatio-temporal transfer function decreases with increasing \( \alpha \).

In addition, Mahowald and Mead have noted that for a fixed space constant, the peak value of the output a small test flash is larger than that resulting from a larger test flash. Fig. 6(b) plots cross-sections of the magnitude of the spatio-temporal frequency response at different spatial frequencies. The magnitude of the spatio-temporal frequency response increases as the spatial frequency increases. Since a small test flash has more high spatial frequency content than a larger flash, the peak of the transient response of the smaller flash should be greater than the peak of the transient response of the larger flash. Interestingly, for higher temporal frequencies, the
by sampling the following function in $\omega_x$:

$$h(\omega_x, j\omega_t) = \frac{\sum_{k=-R}^{R} h(k)e^{j\omega_t k}}{\sum_{k=-R}^{R} \alpha(k)e^{j\omega_t k}}.$$  

Separating the real and imaginary parts results in the equation at the bottom of the page.

Our design strategy will be to design the real and imaginary parts of the numerator and denominator of the transform separately. The real part of the numerator (denominator) is determined by the sums of pairs of coefficients located symmetrically with respect to the center element in the $B(A)$ cloning template. The imaginary part of the numerator (denominator) is determined by the differences between those pairs. Since specifying the sum and the difference of two numbers uniquely determines the values of those numbers, this design strategy completely specifies the desired $A$ and $B$ cloning templates.

In order to determine the desired form of the real and imaginary parts of the spatio-temporal frequency response, we will consider only sine wave grating input. There are three main reasons for this. First, as a necessary condition for proper execution, a velocity sensitive filter should work properly on a subset of the possible inputs. Second, any more complicated input can be expressed as a weighted sum of sine wave gratings. Third, this technique is well suited for the design of CNN filter arrays since, as pointed out in Section III, a CNN filter array is essentially a set of decoupled filters, each operating on one spatial frequency component of the input.

The only interaction between the spatial and temporal frequencies in the frequency response occurs in the imaginary part of the denominator. Since the spatio-temporal spectrum of an image translating at velocity $v$ is concentrated along the line $\omega_t = -v\omega_x$ for each spatial frequency, $\omega_x$, the spatio-temporal frequency response of a CNN filter array tuned to velocity $v$ should achieve its maximum with respect to $\omega_t$ at $\omega_t = -v\omega_x$. This ensures that, at least for sine-wave gratings, the filter has its greatest response for stimuli translating at velocity $v$.

To satisfy this requirement, we choose the differences $a(i) - a(-i)$ such that

$$-\omega_x \approx \sum_{l=1}^{r} (a(l) - a(-l)) \sin(\omega_x l)$$

for $\omega_x \in [-\pi, \pi]$. The region where we fit the two curves is limited to $\omega_x \in [-\pi, \pi]$ because of aliasing due to sampling the image in space. If $r = \infty$, this can be done exactly by setting the differences equal to the Fourier sine coefficients of the sawtooth waveform. However, since, we are interested in keeping the connection radius small in the interests of ease of implementation, we are limited to only a few Fourier coefficients. Simply truncating the Fourier coefficients of the sawtooth waveform leads to an excessive

$$h(\omega_x, j\omega_t) = \frac{\{b(0) + \sum_{l=1}^{r} b(l) + b(-l)] \cos(\omega_x l) + j \{\sum_{l=1}^{r} [b(l) - b(-l)] \sin(\omega_x l)\}}{-\{a(0) + \sum_{l=1}^{r} [a(l) + a(-l)] \cos(\omega_x l) + j \{\omega_t - \sum_{l=1}^{r} [a(l) - a(-l)] \sin(\omega_x l)\}}.$$
amount of "ringing." Therefore, the Fourier coefficients must be appropriately windowed to reduce these effects. Assume \( r = 3 \); the following coefficients were chosen in an attempt to maximize the spatial frequency range within which the two curves match, while minimizing the amount of ringing in that range.

\[
\begin{align*}
    a(1) - a(-1) &= -1.90 \\
    a(2) - a(-2) &= 0.637 \\
    a(3) - a(-3) &= -0.2.
\end{align*}
\]

See Fig. 7.

To completely specify the \( A \) template coefficients, we must also choose their sums. However, considerations based upon the form of the spatio-temporal frequency response are not sufficient. As with any system which incorporates feedback, we must be concerned about stability. Therefore, we digress in the next section to discuss stability criteria for CNN filter arrays, before returning to the design of CNN motion sensitive filter arrays in Section VI.

V. STABILITY

In this section, we state stability criteria for CNN filter arrays. Due to the assumption of spatial invariance, these criteria can be checked easily using simple Nyquist-type graphical tests. In addition, these tests can be interpreted as checking the "poles" of the spatio-temporal transfer function, a concept familiar to engineers from SISO filter design.

The results stated here apply only to linear space-invariant CNN filter arrays. These two powerful assumptions allow us to state both necessary and sufficient conditions for stability, which can be easily checked for any template. These features differentiate these results from previously reported results on the stability of CNN’s by Chua and Wu [23] and Guzelis and Chua [19]. Chua and Wu consider the original definition of CNN’s and give only sufficient conditions for stability in the case of positive cell-linking sign symmetric and acyclic templates, extending the positive cell-linking condition introduced in [24]. Our results are more closely related to those of Guzelis and Chua, except that the assumption of spatial invariance made here allows for a simple interpretation of the stability tests in terms of suitably defined poles of the spatio-temporal transfer function and for simple graphical Nyquist type tests.

We define stability for CNN filter arrays based upon the stability for CNN filter arrays based upon the stability of MIMO feedback systems [25].

**Definition 2:** A linear space-invariant CNN is exponentially stable if the associated MIMO feedback system is exponentially stable, i.e., it is

1) proper; 2) all of its poles have negative real parts.

Exponential stability implies bounded input-bounded output stability. Thus, if a CNN filter array is exponentially stable, then a bounded input will result in a bounded output. It should be clear that stability is independent of the form of \( B \) template.

From Section III, the CNN filter array can be expressed as a set of decoupled single input-single output systems, each of which determines the evolution of one spatial frequency component of the output. The SISO system governing the evolution of the spatial frequency component at frequency \( \omega_s \) is characterized by the transfer function,

\[
\hat{h}(\omega_s) = \frac{\hat{h}(\omega_s)g(s)}{1 - \hat{a}(\omega_s)g(s)}.
\]

This SISO system is exponentially stable if and only if the roots of the equation \( 1 - \hat{a}(\omega_s)g(s) \) for \( s \in C \) all lie in the open left-half side of the complex plane. The CNN filter array is exponentially stable if and only if all of the decoupled SISO systems governing the evolution of the spatial frequency components of the output are stable. Therefore, defining the poles of the spatio-temporal transfer function as those values of \( s \in C \) for which there exists a \( \omega_s \in [0, 2\pi]^d \) such that the denominator of the spatio-temporal transfer function is equal to zero, the CNN filter array is exponentially stable if and only if all of the poles of the spatio-temporal transfer function line in the open left-hand side of the complex plane.

In the following, we define \( \partial\omega = [0, 2\pi]^d \) where \( k \) is a scalar if \( i_k > k \) and \( \partial\omega \in [0, 2\pi]^d \). In other words, each component of the vector or multi-index \( \bar{i} \) is greater than \( k \).

**Proposition 1:** Consider the set of all \( d \)-dimensional CNN filter arrays of size \( \bar{n} > N = 2r + 1 \), connection radius \( r \), and feedback cloning template \( \{ a_i \}_{i \in B} \). Let \( \hat{g}(s) = \hat{n}(s)/\hat{d}(s) \) be proper. Let \( 1 - \hat{a}(\bar{\omega})\hat{g}(\infty) = 0 \) for all \( \bar{\omega} \in [0, 2\pi]^d \).

Under these assumptions, if

\[
\hat{d}(s) - \hat{a}(\bar{\omega})\hat{n}(s) \neq 0 \forall \bar{s} \in \partial\omega, \forall \bar{\omega} \in [0, 2\pi]^d
\]

then all CNN’s in this set are exponentially stable.

Furthermore, this condition is strong in the sense that if there exist \( \omega_s \in [0, 2\pi]^d \) and \( s_s \in C_+ \) (the open right-half plane) such that \( \hat{d}(s_s) - \hat{a}(\omega_s)\hat{n}(s_s) = 0 \), then there exists an \( M \geq N \) such that for all CNN’s in this set of size \( \bar{n} \geq M \) are unstable.

**Example 2:** Consider the set of one-dimensional CNN filter arrays with connection radius \( r = 3 \), feedback cloning template

\[
A = [0 \ -0.637 \ 0.606 \ -0.96 \ -1.194 \ 0 \ -0.2]
\]

and filter transfer function \( \hat{g}(s) = 1/s \). By the above proposition, to check the stability of this array, we must verify...
that
\[ s - \hat{a}(\omega) \neq 0 \forall s \in \mathbb{C}, \forall \omega \in [0, 2\pi). \]

This is equivalent to checking that the real part of \( \hat{a}(\omega) \) is negative for all \( \omega \in [0, 2\pi) \). Fig. 8 shows the locus of points \( \hat{a}(\omega) \) in the complex plane as \( \omega \) ranges from 0 to 2\( \pi \). Since this locus is contained entirely within the left half plane, all CNN filter arrays in this set with size \( \tilde{n} > 7 \) are stable.

Checking the above condition is easy in the case of Example 2, where we need only verify that the real part of \( \hat{a}(\omega) \) is less than zero for all \( \omega \in [0, 2\pi) \). However, in the case of higher order \( g(s) \), direct verification of (5) requires finding the roots of an infinite number of polynomials. Fortunately, we can avoid this problem by using the following graphical test, based on the Nyquist criterion.

**Corollary 2**: Consider the set of all \( d \)-dimensional CNN filter arrays of size \( \tilde{n} > N = 2\pi + 1 \), connection radius \( r \), and feedback cloning template \( \{a_i\}_{i \in \mathbb{N}} \). Let \( g(s) \) be proper. Let
\[ 1 - \hat{a}(\omega)g(\infty) \neq 0 \text{ for all } \omega \in [0, 2\pi)^d. \]
Define \( P \) to be the number of poles of \( \hat{g}(s) \) in \( \mathbb{C}_- \).

Under these assumptions, if the Nyquist plot of \( 1/(\hat{g}(s)) \)
1) does not intersect the set of points, \( \{\hat{a}(\omega) \mid \omega \in [0, 2\pi)^d\} \);
2) encircles set of points, \( \{\hat{a}(\omega) \mid \omega \in [0, 2\pi)^d\}, \)
\( P \) times in the counterclockwise sense, then all CNN filter arrays in this set are exponentially stable.

Furthermore, this condition is strong in the sense that if there exists an \( \omega \) such that the Nyquist plot of \( 1/(\hat{g}(s)) \) does not encircle the point \( \hat{a}(\omega) \) \( P \) times in the counter-clockwise sense, then there exists an \( M \geq N \) such that all of the CNN’s of size \( \tilde{n} \geq M \) in the above set are unstable.

**Example 3**: In an actual implementation of the CNN filter arrays discussed in Example 2, the filters \( \hat{g}(s) = 1/s \) might be implemented as capacitors, the voltage across which would be the filter outputs. The input would be applied as a current injected into the capacitor. The feedback coefficients would be implemented as voltage controlled current sources. The model presented in Example 2 assumes that these transconductance amplifiers have infinite bandwidth, a condition unobtainable in practice. Therefore, we should be concerned about the effect of the limited bandwidth of the transconductance amplifiers on the stability of the array. A more realistic model of the transconductance amplifier is as a single pole system, with a cutoff frequency of \( p \) radians per second, i.e., \( a(\omega)/((s/p)+1) \).

For the sake of simplicity, we have assumed that the cutoff frequency of all of the transconductance amplifiers are the same. Due to the linearity of the system, we can take this modification into account by modifying the filter \( \hat{g}(s) \) to be \( \hat{g}(s) = 1/(s((s/p)+1)) \). Using the above corollary, instead of checking the roots of \( s^2/p + s + \hat{a}(\omega) \) for all \( \omega \), we need only plot the locus of \( \hat{a}(\omega) \) for \( \omega \in [0, 2\pi) \) and check that the Nyquist plot of \( 1/\hat{g}(s) = (s^2/p)+s \) does not intersect or encircle any of the points of this locus. See Fig. 9.

VI. DESIGN OF MOTION SENSITIVE FILTERS II

In this section, we complete the design of our one-dimensional motion sensitive filter array. We also present simulation results of the filter, demonstrating that it works as expected.

Recall from Section IV that our design strategy consisted of independently designing the real and imaginary parts of the denominator and numerator of the spatio-temporal frequency response. In that section, we discussed how the requirement of velocity selectivity constrained the choice of the imaginary part of the denominator of the spatio-temporal frequency response.

On the other hand, considerably more freedom exists in choosing the real part. Several different factors must be taken into account and weighed against each other. We will discuss four factors in the choice of the even part of the feedback coefficients:

1) stability;
2) velocity tuning;
3) localization in space/time;
4) robustness to parameter variation.
Fig. 10. The dotted line shows the ideal real part of the spatio-temporal frequency response specified by (7), which results in a filter with constant velocity bandwidth. However, several other considerations must be weighed against the desire for constant velocity bandwidth, resulting in the real part plotted by the solid line.

Of course, the most important of these considerations is stability. Since \( g(s) = 1/s \), the results of Section V show that the stability of the CNN filter array depends upon the real part of \( \bar{a}(\omega_x) \) being less than zero. Therefore, the sums of the feedback coefficients must be chosen so that the resulting real part of the denominator of the spatio-temporal transfer function is negative.

Velocity tuning refers to how sharply tuned the motion sensitive filter is to a particular velocity. Consider a sine wave grating of spatial frequency \( \Omega_x \). For a velocity filter tuned to velocities in the range \( v - \Delta v \) to \( v + \Delta v \), the passband of the \( \Omega_x \) spatial frequency filter should run from \( \omega_x = (v - \Delta v)\Omega_x \) to \( \omega_x = (v + \Delta v)\Omega_x \). The lower the spatial frequency, the sharper the required temporal tuning must be for the same velocity tuning. Intuitively, the lines in the spatio-temporal frequency domain corresponding the nonzero parts of the spectra of images undergoing uniform translation all pass through the origin. Therefore, for constant velocity bandwidth, the passband of a spatio-temporal filter must be narrower near the origin.

Define the bandwidth of each spatial frequency temporal filter to be the difference between the frequencies at which the magnitude of the temporal frequency response is 3 dB below the peak value. This occurs when the magnitude of the real part of the denominator equals the magnitude of the imaginary part. This implies that the bandwidth is equal to two times the magnitude of real part. For a constant velocity bandwidth across all spatial frequencies of \( 2\Delta v \), we must set

\[
-\Delta v|\omega_x| = a(0) + \sum_{l=-r}^{l=a} (a(l) + a(-l)) \cos(\omega_x l) \quad (7)
\]

for all \( \omega_x \in [-\pi, \pi] \). Recall that for stability reasons, the real part must negative.

Velocity tuning must be weighed against localization of the filter output in time and space. For a given spatial frequency temporal filter, the sharper the temporal tuning, the longer the transient response of that filter. If we choose the real part according to (7), the temporal transients of the low spatial frequency filters will be very long. This has two important ramifications. First, any initial conditions in the array with low spatial frequency components will take a long time to die out. Second, the response of the filter will not be very well localized in time. In other words, the filter will not respond quickly to changes in the input velocity. It turns out that localization in time implies localization in space for more general inputs than sine wave gratings.

To avoid these problems, we make the magnitude of the real part very large for low spatial frequencies, while using (7) to

![Fig. 11](image1)

(a) The real part of the spatial frequency response of \( B_1 \). The imaginary part is equal to zero. (b) The imaginary part of the spatial frequency response of \( B_2 \). The real part is equal to zero.

![Fig. 12](image2)

(a) The magnitude of the spatio-temporal frequency response of the CNN motion sensitive filter with feedback cloning template \( A \) and feedforward cloning template \( l \). (a) as a surface in three dimensions and (b) as a contour diagram. The filter with feedforward cloning template \( B_2 \) has approximately the same magnitude, but is shifted in phase by 90°.
chose the real part for high spatial frequencies. Not only does this ensure that the transients die away quickly, it also has the added benefit of decreasing the gain of the low spatial frequency filters, making the filters relatively insensitive to changes in the overall illumination level.

How negative we can make the real part is limited by the last consideration: robustness in the presence of parameter variation. In our design procedure, we specify the sum and the difference of pairs of feedback coefficients, rather than the parameters themselves. If the sum of two parameters is very large, but the difference is small, a relatively small change in one of the parameters may result in a large change in the difference. A similar effect can occur if the difference is large, but the sum is small. If the magnitudes of the sum and the difference are approximately equal, this effect is minimized.

Taking into account all of these factors and setting the desired velocity bandwidth to be 40%, we have designed the following cosine coefficients:

\[ a(0) = -0.96 \]
\[ a(1) + a(-1) = -0.588 \]
\[ a(2) + a(-2) = -0.637 \]
\[ a(3) + a(-3) = -0.2 \]

resulting in the real part plotted in Fig. 10.

Combining the sums and differences specified above and in Section IV results in the following \( A \) template:

\[ A = \begin{bmatrix} 0 & -0.637 & 0.606 & -0.96 & -1.194 & 0 & -0.2 \end{bmatrix} \]

Fig. 13. (a) The response of the sine phase filter to a bar translating velocities \( r \) pixels is 90° out of phase with (b) the response of the cosine phase filter to a bar translating at the same velocity.
This template is the same as that discussed in Examples 2 and 3. Based on our consideration of robustness in the presence of parameter variation, we chose the sum and difference of two pairs of the coefficient pairs to be exactly equal. This results in two of the template elements being equal to zero, which is advantageous if one intends to implement these filters using capacitors and transconductance amplifiers, since the corresponding transconductance amplifiers can just be omitted.

Finally, we must choose the $B$ cloning template coefficients. We have two goals in choosing these coefficients. One is to increase the spatial frequency tuning. In particular, we wish to eliminate any dc response as the output of the filter should be insensitive to changes in the overall illumination. We also wish to decrease the gain of the filters for very high spatial frequencies where the imaginary part departs from the line $-\omega_x$, as the filter is tuned to the incorrect velocity at these frequencies. The second goal is to create two filters that are $90^\circ$ out of phase with each other, so that the outputs of the two filters can be squared and summed to obtain the motion energy. To create these two filters, we use the same $A$ template and design two $B$ templates with nearly the same magnitude spatio-temporal frequency response, but which are $90^\circ$ out of phase. The resulting $B$ templates are given by

\[
B_1 = \begin{bmatrix} 0 & -0.25 & 0.50 & -0.25 \end{bmatrix} \\
B_2 = \begin{bmatrix} 0.132 & 0 & -0.368 & 0.368 & 0 & -0.132 \end{bmatrix}
\]

and their spatial frequency responses are plotted in Fig. 11. Fig. 12 displays the spatio-temporal frequency response of the final CNN motion sensitive filter array. As designed, the passband is oriented along the line $\omega_1 = -\omega_2$, resulting in velocity sensitivity for patterns translating at 1 pixel/s. These same feedback and feedforward coefficients can be used in a filter tuned to a velocity $v$. We simply choose $\hat{g}(s) = 1/|v|s$. For negative velocities, the order of the $A$ template coefficients must also be reversed. This results in a filter tuned to the velocity $v$, with the same relative velocity bandwidth, 40%.

The CNN filter array designed above was simulated using a fourth order Runge-Kutta algorithm modified from [26]. Fig. 13 shows the responses of the two filter arrays to inputs consisting of bars 10 pixels wide translating at different velocities. The filter arrays consist of 50 cells long. Fig. 14 shows the corresponding motion energies. As intended, the filters show velocity bandpass behavior with a peak response for stimuli moving at 1 pixel/s. Notice also that these filters are direction sensitive.

VII. EXTENSION TO TWO DIMENSIONS

These filters can be used to detect motion in two dimensions. Assume that we wish to detect motion at 1 pixel/s in the horizontal direction on the image plane. Other directions can be selected by rotation of the filter array. The corresponding filter array consists of rows of one-dimensional filters aligned in the horizontal direction. If these one-dimensional filter arrays are not interconnected, the two-dimensional CNN filter array is sensitive to any motion in the image plane which has a horizontal component of motion as long as the gradient of the image intensity has a component in the horizontal direction.

Given information only within a local neighborhood, the only component of the velocity vector of the optical flow which can be recovered is the component perpendicular to an edge. This is the well known aperture problem [27]. We can construct filters tuned to motion perpendicular to local edges, by adding feedback connections between cells which are vertically aligned. The resulting $A$ template is

\[
A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\ 0 & -0.637 & 0.606 & -0.96 & 2g & -1.194 & 0 & -0.2 & 0 \\ 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\]

The value of $g$ controls the width of the convolution kernel in the direction perpendicular to the tuned velocity. For higher values of $g$, the filter is more sharply tuned to the motion of edges that are perpendicular to the tuned direction. See Fig. 15. In an implementation of this array with capacitors and transconductance amplifiers, the feedback interconnections
Fig. 15. A contour plot of the cross-section of the spatio-temporal frequency response of the two-dimensional filter with feedback coefficients linking adjacent rows in the plane \( \omega_x = -\omega_y \) shows that the filter is tuned to vertically oriented spatial edges.

correspond to resistors with conductance \( g \) linking neighboring capacitors in adjacent one-dimensional filter arrays.

VIII. SUMMARY AND CONCLUSION

Based upon the CNN paradigm, we have introduced a new architecture for spatio-temporal image filtering called the CNN filter array. Because they are linear, these filter arrays have well-defined spatio-temporal transfer functions and frequency responses that characterize the mapping from their input to their output. Simple graphical Nyquist type stability criteria exist for these arrays, based upon the notion of determining the locations of the poles of the spatio-temporal transfer function. Using the fairly general results here, we have demonstrated the systematic design of CNN filtering arrays for motion sensitive filtering. This approach has significant advantages over previous approaches to motion sensitive filtering, such as the Gabor filter, as well as other CNN approaches to image motion analysis.

REFERENCES

[1] C. L. Baker, Jr., “Spatial- and temporal-frequency selectivity as a basis for velocity preference in cat striate cortex neurons,” Visual Neuroscience, vol. 4, pp. 101–113, 1990.
[2] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, Principles of Neural Science. New York: Elsevier, 1991.
[3] K. Nakayama, “Biological image motion processing: A review,” Vision Research, vol. 25, pp. 625–660, 1985.
[4] E. H. Adelson and J. R. Bergen, “Spatiotemporal energy models for the perception of motion,” J. Optical Society of America A, Vol. 2, pp. 284–299, Feb. 1985.
[5] ———, “The extraction of spatio-temporal energy in human and machine vision,” in Proc. IEEE Workshop on Motion: Representation and Analysis, pp. 151–155, May 1986.
[6] A. B. Watson and J. A. J. Ahumada, “Model of human visual-motion sensing,” J. Optical Society of America A, vol. 2, pp. 322–342, Feb. 1985.
[7] D. J. Heeger, “Model for the extraction of image flow,” J. Optical Society of America A, vol. 4, pp. 1455–1471, Aug. 1987.
[8] ———, “Optical flow from spatio-temporal filters,” in Proc. Int. Conf. Computer Vision, pp. 181–190 1987.
[9] N. M. Grzywacz and A. L. Yuille, “A model for the estimate of local image velocity by cells in the visual cortex,” in Proc. R. Soc. Lond. B, vol. 259, pp. 129–161, 1990.
[10] D. J. Fleet, Measurement of Image Velocity. Boston: Kluwer, 1992.
[11] L. O. Chua and L. Yang, “Cellular neural networks: Applications,” IEEE Trans. Circuits Syst., vol. 35, Oct. 1988.
[12] ———, “Cellular neural networks: Theory,” IEEE Trans. Circuits Syst., vol. 35, Oct. 1988.
[13] D. J. Fleet and A. D. Jeppson, “Hierarchical construction of orientation and velocity selective filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, pp. 315–325, Mar. 1989.
[14] T. Roska, T. Boros, P. Thiran, and L. O. Chua, “Detecting simple motion using cellular neural networks,” in Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications, pp. 127–138, 1990.
[15] D. Gabor, “Theory of communications,” J. IEEE London, vol. 93, pp. 429–457, 1946.
[16] S. Marcelja, “Mathematical description of the responses of simple cortical cells,” J. Opt. Soc. Am., vol. 70, pp. 1297–1300, Nov. 1980.
[17] J. G. Daugman, “Six formal properties of two-dimensional anisotropic visual filters: Structural principles and frequency-orientation selectivity,” IEEE Trans. Syst. Man, Cybern., vol. SMC-13, pp. 882–887, 1983.
[18] ———, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” J. Optical Society of America A, vol. 2, pp. 1160–1169, July 1985.
[19] C. Guezulis and L. O. Chua, “Stability analysis of generalized cellular neural networks,” Tech. Rep. UCB/ERL M97/23, Univ. California at Berkeley Electronics Research Laboratory, Mar. 1991.
[20] T. Roska and L. O. Chua, “Cellular neural networks with nonlinear and delay-type template elements,” in Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications, pp. 12–25, 1990.
[21] P. J. Davis, Circulants Matrices. New York: Wiley, 1979.
[22] M. A. Mahowald and C. Mead, “Silicon retina,” in Analog VLSI and Neural Systems, C. Mead, Ed. Menlo Park, CA: Addison-Wesley, 1989, ch. 15, pp. 257–278.
[23] L. O. Chua and C. W. Wu, “On the universe of stable cellular neural networks,” Int. J. Circuit Theory and Applications, to appear.
[24] L. O. Chua and T. Roska, “Stability of a class of nonreciprocal cellular neural networks,” IEEE Trans. Circuits Syst., vol. 37, pp. 1520–1527, 1990.
[25] C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties. New York: Academic, 1975.
[26] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing. New York: Cambridge University Press, 1988.
[27] D. Marr, Vision. New York: W. H. Freeman and Company, 1982.

Bertram E. Shi received the B.S. and M.S. degrees in electrical engineering from Stanford University in 1987 and 1988, respectively. He is currently studying for his Ph.D. degree in electrical engineering at the University of California, Berkeley. His present research interests lie in the areas of neural networks, signal processing, and computer vision.

Tamás Roska (M’87–SM–90–F’92), for a photograph and biography, please see page 131 of the March issue of this TRANSACTIONS.

Leon O. Chua (S’60–M’62–SM’70–F’74), for a photograph and biography, please see page 173 of the March issue of this TRANSACTIONS.