Gabor-Type Filtering in Space and Time with Cellular Neural Networks

Bertram E. Shi, Member, IEEE

Abstract—Gabor filters are preprocessing stages in image-processing and computer-vision applications. One drawback is that they are computationally intensive on a digital computer. This paper describes the design of cellular neural networks (CNN’s) which compute the outputs of filters similar to Gabor filters. Analog VLSI implementations of these CNN’s might eventually relieve the computational bottleneck associated with Gabor filtering image-processing algorithms. The CNN’s compute both the real and imaginary parts of the filter outputs simultaneously, which is an important feature in applying them in algorithms utilizing the phase of the Gabor output.

Index Terms—Analog circuits, cellular neural networks, filtering, image processing, neural networks.

I. INTRODUCTION

Gabor filters [1] have been used as preprocessors for different tasks in computer vision and image processing. These approaches to image processing and computer vision have been motivated partially by the discovery that the responses of orientation selective cells in the visual cortex can be modeled using Gabor filters [2], [3]. Because image velocity can be considered as an orientation in the space–time domain, three-dimensional (3-D) Gabor filters have been used to model cortical cells’ velocity and directional sensitivity [4]. Initial evidence indicates that approaches based upon Gabor filtering can outperform previously developed approaches [5].

One drawback of Gabor filtering approaches is that they are computationally intensive. Here we describe how to implement filters similar to the Gabor filter using cellular neural networks (CNN’s) [6]–[8]. The advantage of CNN’s is that they can be implemented in analog VLSI alongside photosensors which sense the image [9]–[11]. The filter outputs can be computed in less time than required by serial digital computer implementations and be read off the chip directly, relieving the computational bottleneck of preprocessing with Gabor filters.

The remainder of Section I reviews Gabor filters and defines the class of “Gabor-type” filters which we believe capture the important properties of Gabor filters exploited by many image-processing and computer-vision applications. It concludes with a short review of previous related work. Section II shows that any low-pass spatial filter implemented on a CNN can be transformed into a corresponding Gabor-type filter and gives several illustrative examples of the types of filters which can be achieved on CNN’s. Section III extends the results to spatio–temporal filters. Finally, Section IV summarizes our results and outlines ongoing research in designing and fabricating analog VLSI chips implementing these CNN’s and applying them in computer vision.

In the following, small letters denote space and time waveforms. e.g., \( f(x), f(n), \) or \( f(t) \) where \( x \) represents continuous space, \( n \) represents discrete space, and \( t \) represents continuous time. Capital letters denote Fourier transforms. Transforms of a continuous waveform \( f(x) \) or \( f(t) \) will be written as \( F(\omega_x) \) or \( F(\omega_t) \), while transforms of a discrete waveform \( f(n) \) will be written as \( F(e^{j\omega_n}) \).

A. Gabor Filters

Gabor filters exist for signals of arbitrary dimension where an \( n \)-dimensional signal is defined to be a mapping from \( \mathbb{R}^n \) to \( \mathbb{R} \) or \( \mathbb{C} \). For one-dimensional (1-D) signals, the impulse response \( g(x) \) of a Gabor filter is a complex exponential function modulated by a Gaussian

\[
g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} e^{j\omega_x x} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} (\cos(\omega_x x) + j\sin(\omega_x x)) \tag{1}
\]

where \( \omega_x \) is the angular frequency of the complex exponential and \( \sigma^2 \) is the standard deviation of the Gaussian. Fig. 1(a) and (b) plot the real and imaginary parts of (1). Although the impulse response is defined for continuous \( x \), since the Gabor-type filters implemented by CNN’s are inherently discrete space, we plot the impulse responses at discrete integer values of \( x \) to facilitate comparison with later results.

By the Fourier shift theorem, the frequency response of a Gabor filter is a Gaussian function centered at \( \omega_x = \omega_x^0 \):

\[
G(\omega_x) = \int_{-\infty}^{\infty} g(x) e^{-j\omega_x x} dx = \exp \left( -\frac{\sigma^2(\omega_x - \omega_x^0)^2}{2} \right).
\]

The Gabor filter is a bandpass filter tuned to frequencies near \( \omega_x^0 \) [see Fig. 1(c)]. Generalizing to \( n \)-dimensional signals, the convolution kernel and frequency response are

\[
g(\vec{x}) = \frac{1}{\sqrt{(2\pi\sigma)^n}} e^{-(1/2)\vec{x}^T \Sigma^{-1} \vec{x}} e^{j(\omega_x^T \vec{x})}
\]

\[
G(\vec{\omega}_x) = e^{-(1/2)(\vec{\omega}_x - \vec{\omega}_x^0)^T \Sigma^{-1}(\vec{\omega}_x - \vec{\omega}_x^0)}
\]

where \( \vec{x}, \vec{\omega}_x \in \mathbb{R}^n \), the covariance matrix \( \Sigma \) is \( n \times n \) positive-definite.

Manuscript received September 11, 1996; revised February 26, 1997. This work was supported by the Hong Kong Research Grants Council under Grant HKUST675/9E. This paper was recommended by Associate Editor J. Pineda-de-Gyvez.

The author is with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong.

Publisher Item Identifier S 1057-7122(98)00889-7.
Fig. 1. A comparison of the impulse and frequency responses of the 1-D Gabor filter and Gabor-type filters implementable on a CNN. The filter parameters have been chosen so that the squared errors between the impulse responses are minimized. (a)–(b) The real and imaginary parts of the impulse response of a Gabor filter with $\sigma = 3.32$ and $\omega_{xx} = 0.93$ are sine and cosine functions modulated by a Gaussian (dotted line). The responses are plotted at integer values of $x$ to facilitate comparison with the other plots. (c) The frequency response of the Gabor filter is a Gaussian centered at $\omega_{xx}$. (d)–(e) The real and imaginary parts of the impulse response of the Gabor-type filter in Example 1 for $\lambda = 0.3$ and $\omega_{xx} = 0.93$ are sine and cosine functions modulated by a function which decays exponentially away from the origin (dotted line). (f) The frequency response of the filter in Example 1 is the Fourier transform of the modulating function shifted to $\omega_{xx}$. Since the filter is defined for discrete space, the frequency response is periodic with period $\frac{1}{2\pi}$. (g)–(h) The real and imaginary parts of the Gabor-type filter in Example 2 for $\mu = 0.352$ and $\omega_{xx} = 0.93$ more closely approximate the Gabor. (i) The frequency response of the filter in Example 2 is approximately Gaussian, but is flatter at the peak at $\omega_{xx}$. 

The covariance matrix is often chosen to be a scalar multiple of the identity matrix. Two-dimensional (2-D) Gabor filters are orientation selective and have been used to model receptive fields of orientation selective neurons in the visual cortex. For example, the Gabor filter with impulse response

$$
 g(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x^2+y^2)/(2\sigma^2)} e^{i(\omega_{xx}x + \omega_{yy}y)}
$$

is tuned to spatial frequency $(\omega_{xx}, \omega_{yy})$, This filter responds maximally to edges which are oriented at an angle $\theta = \arctan(\omega_{yy}/\omega_{xx})$ where $\theta$ is defined to be the angle between the horizontal $(x)$ axis and the line perpendicular to the edge (see Fig. 2). 2-D Gabor filters have found applications in computer-vision algorithms for stereo vision [12]–[18], binocular vergence control [19], texture segmentation [20], and face recognition [21].

Three–dimensional spatio–temporal Gabor filters have been used for image-motion analysis [22]–[24]. Image motion can be characterized as an orientation in the space–time domain [4]. The energy spectrum of a 1-D image undergoing uniform translation at velocity is nonzero only along the line $\omega t = -v_x\omega_{xx}$. For 2-D images translating with velocity $\nu = (v_x, v_y)$, the spectrum is nonzero only along the plane $\omega t = -(v_y\omega_{xx} + v_x\omega_{yy})$. A spatio–temporal bandpass filter tuned to frequency $(\omega_{xx}, \omega_{yy}, \omega_t)$ can be considered to be tuned to the velocity

$$
 (v_{xx}, v_{yy}) = \left(\frac{\omega_{xx}}{\omega_t}, \frac{\omega_{yy}}{\omega_t}\right).
$$
Fig. 2. The (a) real and (b) imaginary parts of the impulse response of a 2-D Gabor filter which is tuned to the orientation $\theta = \pi/4$.

Spatio–temporal filtering approaches to estimating the optical flow often compare the outputs of filters tuned to different regions in the spatio–temporal frequency domain. The optical flow is the projection of the 3-D motion vector field resulting from relative motion between a camera and its environment to a 2-D motion vector field in the image plane. Estimation of the optical flow is one of the first steps in many algorithms which extract the shape of surfaces being imaged by a moving camera [25].

For computer-vision applications, the output of a Gabor filter is often expressed in terms of its magnitude and phase. If $y_r(\vec{x})$ and $y_i(\vec{x})$ are the real and imaginary parts of a Gabor filter’s output, its magnitude and phase are

$$
\rho(\vec{x}) = \sqrt{y_r(\vec{x})^2 + y_i(\vec{x})^2},
$$

$$
\phi(\vec{x}) = \arctan(y_i(\vec{x})/y_r(\vec{x})).
$$

The phase of the output at a given pixel is related to the location of edges and other features in the input image near that pixel. Translating the image input results in a phase shift in the Gabor output at a given pixel. This property has motivated the development of “phase-based” approaches to stereo-vision and image-motion analysis. Phase differences between the Gabor filter outputs from two stereo images can be used to estimate disparity [12]–[18] and control binocular vergence in active-vision systems [19]. Fleet [24] showed that the temporal variation of phase is a robust indicator of the local image velocity. Barron et al.’s comparison [5] of algorithms for optical-flow estimation indicates that Fleet’s algorithm using Gabor phase is the most accurate among those tested. Many phase-based algorithms use the amplitude as a confidence measure for the reliability of the phase measurement.

B. Gabor-Type Filters

We extend Gabor filters to Gabor-type filters by allowing modulating functions other than the Gaussian. Formally, define a filter to be Gabor-type if its impulse response $h(\vec{x})$ can be expressed as a complex exponential modulated by a real valued envelope $f(\vec{x})$, which is the impulse response of a low-pass filter

$$
h(\vec{x}) = f(\vec{x})e^{j\omega_{\text{c}} \cdot \vec{x}}.
$$

All Gabor filters are Gabor-type. The frequency response $H(\omega_{\text{c}})$ of a Gabor-type filter is equal to the Fourier transform of the modulating function $F(\omega_{\text{c}})$ shifted to $\omega_{\text{c}}$:

$$
H(\omega_{\text{c}}) = F(\omega_{\text{c}} - \omega_{\text{c0}}).
$$

Gabor-type filters do not necessarily optimize the uncertainty relation between the resolution in space and spatial frequency, a commonly cited advantage of Gabor filters resulting from their Gaussian modulating function. However, they can be designed to satisfy the requirements of filters used for phase-based algorithms in computer vision. In their phase-based disparity study, Westelius et al. [18] examined the use of filters other than the Gabor and found that the overall performance of their algorithm “was not critically dependent upon the type of disparity filter used.” They identified several desirable properties for 1-D filters used to compute phase.

1) The filter has no dc component.
2) The filter is sensitive to positive frequencies only.
3) The filter is insensitive to singular points.
4) The phase of the impulse response does not wrap around (i.e., exceed $\pm \pi$).
5) The filter has small spatial support.
6) The phase of the impulse response is monotonous.
In order for a Gabor-type filter to possess the first three properties, its bandwidth should be small in comparison with the center frequency $\omega_{2\alpha}$, which we assume to be positive. If $F(\omega)$ is identically zero outside the range $-\omega_{2\alpha} \leq \omega \leq \omega_{2\alpha}$, the response of the filter to dc and negative frequencies is identically zero. In general, the frequency response of a Gabor-type filter may have infinite support. However, because of the low-pass nature of $F(\omega)$, it is possible to make the maximum response to dc and negative frequencies smaller than any fixed positive threshold by decreasing the bandwidth sufficiently. In addition, even if the dc response is not zero, an extra processing step can easily be added to remove the dc component. Westelius et al. point out that the sensitivity to singular points also decreases with bandwidth.

On the other hand, the fourth and fifth properties imply a large bandwidth since they constrain the width of the impulse response. The fourth requirement is satisfied by Gabor-type filters if $f(x)$ is identically zero outside the range $-\pi \omega_{2\alpha} \leq x \leq \pi \omega_{2\alpha}$ For filters which do not satisfy this condition, the effects of phase wraparound can be minimized by ensuring that width of $f(x)$ is small enough that the impulse response is close to zero outside this range. In general, the fourth and fifth properties are less critical than the first three. Westelius et al. state that the fourth requirement is “desirable, but not absolutely necessary.” The fifth property, small spatial support, was included since it determines the computational cost of the filter. However, one goal of this paper is to decrease the computational cost of Gabor-type filters by enabling their implementation in analog VLSI.

The sixth property can be satisfied by Gabor-type filters with both large and small bandwidths. Any Gabor-type filter in which $f(x)$ is strictly positive satisfies this property. For filters that do not satisfy this property, the effect of phase nonmonotonicity can be alleviated by ensuring that the absolute value of the low-pass envelope is small when it changes sign.

In summary, a Gabor-type filter can be used effectively in phase-based disparity estimation if its bandwidth is chosen to be small enough and if the modulating function is strictly positive or at least positive over the region where its magnitude is significant. Although Westelius et al.’s study was confined to disparity estimation, we expect that the properties they have identified to be applicable to filters for other phase-based algorithms, such as for image-motion analysis. For example, the Gabor-type filter in Example 1 has been used in a phase-based algorithm for extracting time-to-contact from a moving image taken by a camera translating toward a planar surface [26].

C. Previous Work

Raffo has demonstrated that a resistive network similar to that shown in Fig. 5 followed by an additional processing stage can implement a real valued Gabor-type filter with arbitrary phase [28]. In particular, for 1-D signals the convolution kernel has the form

$$h(x) = e^{-\lambda |x|} \cos(\omega_{2\alpha} n + \phi)$$

where $\phi$ can be chosen arbitrarily. For $\phi = 0$, this is the real part of convolution kernel implemented by the network described in Example 1. For $\phi = \pi/2$, it is the imaginary part.

In some ways, the two networks are similar in complexity and capability. Raffo’s network requires connections to second nearest neighbors. Although the network of Example 1 requires only nearest neighbor interconnections, each pixel requires two nodes. The network of Example 1 computes both real and imaginary parts of the Gabor-type outputs simultaneously, but this could be done by Raffo’s network by adding two additional processing stages rather than one. The arbitrary phase achieved by Raffo’s network could be obtained by a linear combination of the two outputs of the network of Example 1.

One advantage of the network of Example 1 is that the dependency of the conductances and transconductances on the parameters $\lambda$ and $\omega_{2\alpha}$, which determine the shape of the convolution kernel, are simpler than for Raffo’s network. For example, the resistor between nearest neighbor nodes for Raffo’s network has conductance

$$G_1 = 2p(p^2 + 1) \cos \omega_{2\alpha}, \quad \text{where} \quad p = e^{-\lambda}$$

The transconductances in the second processing stage are also complex functions of $p$, $\omega_{2\alpha}$, and $\phi$. On the other hand, $\omega_{2\alpha}$ for the network of Example 1 is uniquely specified by the ratio of the conductances $G_1$ and $G_2$. In addition, if $\omega_{2\alpha} = 2 \arctan (0.5) \approx 0.93$, $G_1$ is $\lambda^2$. The methodology presented here can also be used to implement a wide class of Gabor-type filters, of which Example 1 is only one example.

In addition, this paper discusses not only spatial filtering, but also spatio–temporal filtering. Previous work in this area includes Delbruck’s successful VLSI implementation of a velocity-sensitive filter using a delay line consisting of a cascade of first-order filter stages [29]. Unlike the Gabor-type filters proposed here, the spatio–temporal filters implemented by the delay line are not separable in space and time like the Gabor filters. Space–time separability can be exploited in computing the optical flow [23]. In addition, the delay line does not generate the complex valued filter output critical to phase-based algorithms.

II. SPATIAL FILTERS

This section introduces CNN architectures for Gabor-type spatial filtering. For simplicity, we introduce architectures for 1-D images before generalizing to two dimensions.

A. 1-D Images

The CNN is a neural-network architecture consisting of an array of neurons, called “cells.” Each cell is a first-order continuous-time dynamical system. To filter an $N$-pixel 1-D image, $u(n) \in \mathbb{R}$ where $n \in \{0, 1, \ldots, N-1\}$, we use a 1-D CNN array of $N$ cells where the state at the $n$th cell $v(n) \in \mathbb{C}$ satisfies

$$v(n) = \sum_{k=-r}^{r} a_k v(n+k) + bu(n).$$  \hspace{1cm} (2)
The dot denotes differentiation with respect to time. The
\[ A = [a_k e^{j\omega_0 k}], \quad b \] and \( r \) are complex coefficients called the feedback and feedforward cloning templates and \( r \) is defined to be the connection radius. We represent the feedback cloning template using a \( 1 \times (2r + 1) \) matrix where the center element equals \( a_0 \). For example, for \( r = 2 \), the cloning template matrix is
\[
A = \begin{bmatrix}
a_{-2} & a_{-1} & a_0 & a_1 & a_2
\end{bmatrix}.
\]
This CNN equation is slightly different than that presented in [6] and [7]. The key differences are that here the state is complex rather than real and the bias and nonlinearity are excluded. However, (2) can be considered to be a special case of a two-layer CNN as presented in [6], where: 1) the cells in the first layer represent the real parts of the complex valued state; 2) the cells in the second layer represent the imaginary part; 3) the bias terms are identically zero; and 4) the output always operates in the linear region of the output nonlinearity. Thus, the hardware complexity of the CNN defined here is comparable to that of a two-layer CNN.

Since the dynamics of the CNN in (2) are purely linear, the stability of any template can be evaluated theoretically [30]. If the CNN is stable, then for an input which is constant in time, the state \( v(n) \) settles to a unique equilibrium point which is a spatially filtered version of the input \( u(n) \). The steady-state value of \( v(n) \) is the output of the “computation” performed by this CNN. The length of the transient is the time required to perform the computation. Assume that the CNN array consists of an infinite number of cells indexed by \( -\infty < n < \infty \). Define the discrete-space Fourier transforms of the input and output to be
\[
U(e^{j\omega_0}) = \sum_{n=-\infty}^{\infty} u(n) e^{-j\omega_0 n},
\]
and
\[
V(e^{j\omega_0}) = \sum_{n=-\infty}^{\infty} v(n) e^{-j\omega_0 n}.
\]
The frequency response of the spatial filter is
\[
F(e^{j\omega_0}) = \frac{V(e^{j\omega_0})}{U(e^{j\omega_0})} = \frac{b}{-\sum_{k} a_k e^{j\omega_0 k}}.
\]
A similar result using the discrete Fourier transform (DFT) holds for finite arrays with periodic boundary conditions where the cells at the ends of the array are considered to be nearest neighbors. With other boundary conditions, the analysis here will be approximate, but for stable filters the effect of the boundary conditions decays as the distance from the boundary increases. By changing the values of the cloning template coefficients, different filters, e.g., low-pass, bandpass, and high-pass, can be constructed. Since the CNN consists of a discrete array of cells, the filters are defined only for discrete-space input.

For every low-pass filter implementable on a CNN, the Gabor-type filter obtained by multiplying the impulse response of that low-pass filter by a complex exponential is also implementable on a CNN. Crounse and Chua describe methods for designing CNN’s for low-pass filtering in [31]. The following focuses on mapping a template implementing a low-pass filter to the template which implements the corresponding Gabor-type filter.

Suppose that the cloning templates \( A \) and \( B \) are chosen so that the corresponding CNN implements a low-pass filter with frequency response \( F(e^{j\omega_0}) \). By shifting \( F(e^{j\omega_0}) \) such that it is centered around \( \omega_{2\alpha} \), we obtain the frequency response of the corresponding Gabor-type filter tuned to \( \omega_{2\alpha} \):
\[
H(e^{j\omega_0}) = \frac{b}{-\sum_{k} a_k e^{j\omega_0 k}} = \frac{b}{-\sum_{k} [a_k e^{-jk\omega_0}] e^{j\omega_0 k}}.
\]
This filter can be implemented by a CNN with the same feedforward cloning template and the complex valued feedback cloning template
\[
A = [a_k e^{-jk\omega_0}]_{k=-r}^{r}.
\]
Using the approach in [30], it can be shown that if the original low-pass filter template is stable, then the corresponding Gabor-type template is also stable.

**Example 1:** Consider the low-pass filter implemented by the CNN with cloning templates
\[
A = \begin{bmatrix} 1 & -2 + \lambda^2 & 1 \end{bmatrix}, \quad b = \lambda^2
\]
where \( \lambda \geq 0 \). Substituting into (2)
\[
v(n) = -\lambda^2 v(n) + (v(n+1) - v(n)) + (v(n-1) - v(n)) + \lambda^2 u(n),
\]
Assuming both \( v(n) \) and \( u(n) \) are real, the linear resistive grid [32] in Fig. 3(a) implements this CNN. At steady state, \( v(n) \) is a low-pass filtered version of \( u(n) \). Fig. 3(b) and (c) plots the impulse and frequency responses of the filter which are given by [33]:
\[
f(n) = \sqrt{\frac{\lambda}{4 + \lambda}} e^{-\alpha |n|}, \quad F(e^{j\omega_0}) = \frac{\lambda^2}{2 + \lambda^2 - 2 \alpha \cos \omega_0}
\]
where \( \alpha = \arccosh (2 + \lambda/2) \). This system is a discrete approximation to a continuous space system with impulse and frequency responses
\[
f_c(x) = \frac{\lambda}{2} e^{-|x|}, \quad F_c(\omega_x) = \frac{\lambda^2}{\lambda^2 + \omega_x^2}
\]
where the subscript \( c \) emphasizes that the function is defined on continuous space. The width of the impulse response increases linearly with \( 1/\lambda \). The approximation improves as \( \lambda \) decreases.

By (4), the corresponding Gabor filter [34] is implemented by the CNN with templates
\[
A = [e^{j\omega_0} - (2 + \lambda^2) e^{-j\omega_0}] b = \lambda^2.
\]
Fig. 3. (a) This resistive grid implements the low-pass filtering CNN of Example 1. The resistor labels denote conductances. The (b) impulse and (c) frequency responses of the filter for $\lambda = 0.3$ confirm the expected low-pass behavior.

Fig. 4. A circuit implementation of two cells of a CNN array which implements the Gabor-type filter corresponding to the resistive grid discussed in Example 1. Resistor labels denote conductances. Trapezoidal blocks represent transconductance amplifiers labeled by their gains.

Its impulse and frequency responses, plotted in Fig. 1(d) and (e) are

$$h(n) = f(n)e^{j\omega_{ko}n} = \frac{\lambda}{2} e^{-\lambda n} e^{j\omega_{ko}n}$$

$$H(e^{j\omega}) = F(e^{j(\omega - \omega_{ko})})$$

For a circuit implementation of this network, the complex valued state $\psi(n)$ is represented by the voltages across two capacitors representing its real and imaginary parts $v_r(n)$ and $v_i(n)$. Substituting (6) into (2) and separating the real and imaginary parts, we can express the time evolution of the complex valued state in (2) as an equivalent system where the complex state variable has been replaced by two real valued state variables

$$\begin{bmatrix} v_r(n) \\ v_i(n) \end{bmatrix} = \begin{bmatrix} \cos \omega_{zo} & -\sin \omega_{zo} \\ \sin \omega_{zo} & \cos \omega_{zo} \end{bmatrix} \begin{bmatrix} \psi_r(n) \\ \psi_i(n) \end{bmatrix} + \begin{bmatrix} 2 + \lambda^2 & 0 \\ 0 & 2 + \lambda^2 \end{bmatrix} \begin{bmatrix} v_r(n-1) \\ v_i(n-1) \end{bmatrix}$$

$$+ \begin{bmatrix} \cos \omega_{zo} & \sin \omega_{zo} \\ -\sin \omega_{zo} & \cos \omega_{zo} \end{bmatrix} \begin{bmatrix} \psi_r(n+1) \\ \psi_i(n+1) \end{bmatrix}$$

$$+ \begin{bmatrix} \lambda^2 u(n) \\ 0 \end{bmatrix}.$$
Fig. 5. This resistive network better approximating Gaussian filtering adds negative resistance connections to second nearest neighbors. Only the full connections for node \(n\) are shown. The resistor labels denote conductances.

The circuit in Fig. 4 implements two cells of the CNN and their interconnections with their nearest neighbors. In the following, we refer to each circuit node by its labeled nodal voltage. The voltages across each pair of vertically aligned capacitors represent the real and imaginary parts of the state of a cell. The resistances and transconductance amplifiers implement the interconnection between cells. The input to the network is provided by current sources which supply currents proportional to the input image intensity. By writing KCL at nodes \(v(n)\) and assuming a unit capacitance, one can verify that the time evolution of \(v(n)\) satisfies the top equation in (7) and that \(v(n)\) satisfies the bottom equation. The entire CNN array can be constructed by replicating this circuit to add more cells and connections. At steady state, the voltages across the lower capacitors are the result of convolving the spatial distribution of the input currents with the real part of \(h(n)\). The voltages across the upper capacitors correspond to convolving with the imaginary part. This mixed transconductance amplifier and resistor implementation can be proven to be more robust to parameter variations than implementations based on either resistors or transconductance amplifiers alone [35].

The circuit implementation also gives good intuitive understanding of the CNN’s operation. Assume that the input image is an impulse at pixel \(n\). In the circuit, this corresponds to setting the current source \(\lambda^2 u(n)\) to \(\lambda^2\) amps and setting the remaining current sources to zero. If the gains and conductances were chosen so that \(\lambda = 0.3\) and \(\omega_{2\omega} = 0.3\), then the steady-state voltages across the lower capacitors would follow the spatial distribution shown in Fig. 1(d) where the center peak occurs at cell \(n\) and the voltages across the upper capacitors would follow the distribution shown in Fig. 1(e). To see how this would arise in the circuit, consider the current supplied by the source \(u(n)\). Part of the current flows through the conductance \(G_0\) (which is positive) pushing the voltage \(v_0(n)\) positive. As this voltage increases, the two resistors with conductance \(G_1 = \cos \omega_{2\omega}\) cause a smoothing effect which pulls the voltages \(v_1(n-1)\) and \(v_1(n+1)\) up toward \(v_0(n)\). Current also flows through the diagonal resistor with conductance \(G_2\) with \(\sin \omega_{2\omega}\) pulling \(v_1(n+1)\) positive as well. At the same time, the transconductance amplifier with input \(v_1(n)\) draws current from node \(v_1(n-1)\) pushing \(v_1(n-1)\) negative. The larger \(G_2\), the more the voltages at nodes \(v_1(n-1)\) and \(v_1(n+1)\) are pushed negative and positive. On the other hand, the larger \(G_1\), the greater the smoothing between nodes. Thus, the larger the ratio \(G_2 : G_1 = \sin \omega_{2\omega} : \cos \omega_{2\omega} = \tan \omega_{2\omega}\) the higher the spatial frequency \(\omega_{2\omega}\), at which the impulse response oscillates. This is consistent with our theoretical predictions since \(\tan \omega_{2\omega}\) increases with \(\omega_{2\omega}\).

**Example 2:** Based upon Kobayashi et al.’s resistor network for Gaussian filtering [36], it is possible to design a CNN which implements a filter with an impulse response which is closer to the Gabor, albeit at the price of increased complexity. For 1-D images, the resistor network in Fig. 5 smooths the input current distribution with an approximately Gaussian convolution kernel. The corresponding CNN cloning templates are

\[
A = \begin{bmatrix} -1 & 4 & -(6 + \mu^4) & 4 & -1 \end{bmatrix} \ b = \mu^4.
\]

The filter implemented by this network is a discrete approximation to a continuous space filter with impulse and frequency responses given by [37]

\[
f_c(x) = \frac{\mu^4}{\mu^4 + \omega_{2\omega}^4} \ e^{-\mu^2 \omega_{2\omega}^2 x^2} \ \cos \left( \frac{\mu^2 |x|}{\sqrt{2}} - \frac{\mu^4}{4} \right)
\]

The width of the impulse response increases linearly with \(1/\mu\). The approximation improves as \(\mu\) decreases.

The template for the corresponding Gabor-type filter is

\[
A = \begin{bmatrix} -c e^{2i\omega_{2\omega} x} & 4 e^{i\omega_{2\omega} x} & -(6 + \mu^4) & 4 e^{-i\omega_{2\omega} x} & -c e^{2i\omega_{2\omega} x} \end{bmatrix} \ b = \mu^4.
\]

Fig. 1(g)–(i) plots the impulse and frequency responses of this filter.

To compare how well the filters in the preceding examples approximate Gabor filters, we define the normalized squared error (NSE) to be the total energy in the difference between the Gabor impulse response and the impulse response of the continuous space filter approximated by the CNN filter divided by the total energy in the Gabor impulse response

\[
\text{NSE} = \frac{\int_{-\infty}^{\infty} \left| g(x) - f_c(x) e^{i\omega_{2\omega} x} \right|^2 \ dx}{\int_{-\infty}^{\infty} \left| g(x) \right|^2 \ dx}
\]

where \(g(x)\) is the Gabor impulse response in (1) and \(f_c(x)\) is given either by (5) or (8). The frequencies of the complex exponentials \(\omega_{2\omega}\) in the Gabor and Gabor-type filters are...
chosen to be identical. The results of analyzing the discrete-
space filters will be more complex, but very similar to the
results here, especially as $\lambda$ and $\mu$ approach zero.

Using the fact that $|e^{jx}|^2 = 1$, it can be shown that
the error is independent of $\omega_{20}$ and depends only upon the
difference between the Gaussian envelope of the Gabor filter
and the low-pass envelope $f_c(x)$ of the Gabor-type filter. For
the filter in Example 1, the minimum error of $-15$ dB is achieved for

$$\lambda = k_1/\sigma, \quad k_1 \approx 0.995.$$  

For the filter in Example 2, the minimum error of $-21$ dB is achieved for

$$\mu = k_2/\sigma, \quad k_2 \approx 1.17.$$  

The values of $k_1$ and $k_2$ have been estimated numerically.
Simulation results in Fig. 1 where the parameters chosen minimize the NSE provide visual confirmation that the impulse and frequency responses of the filter in Example 2 are closer to the Gabor filter’s than are those of the filter in Example 1.

To compare the three filters with respect to the desirable
properties for computing phase discussed in Section I-B, we
define the following criteria. The criteria are based upon the
assumption that the input image is white.

**Negative Frequency Rejection (NFR):** This is the ratio be-
tween the energy in the output due to positive frequencies in
the input and the energy due to negative frequencies. If $G(\omega_2)$
is the frequency response of the filter, then

$$\text{NFR} = \frac{\int_0^\infty |G(\omega_2)|^2 d\omega_2}{\int_{-\infty}^0 |G(\omega_2)|^2 d\omega_2}.$$  

**DC Rejection (DCR):** This is a measure of the extent to
which the dc component is suppressed compared with the peak
response. If $G(\omega_2)$ is the frequency response of the filter

$$\text{DCR} = \frac{|G(\omega_{20})|}{|G(0)|}.$$  

**Phase Wraparound Rejection (PWR):** The output at a given
pixel can be considered as the sum of two components: the first (desired) component due to convolving the image with the part
of the impulse response where the phase varies from $-\pi$
to $\pi$ and the second (undesired) component due to convolving
the input image with the part of the impulse response where
the phase has wrapped around. Since we assume the image
input is white, these two components are uncorrelated. We
define the PWR to be the ratio of their energies. If $g(x)$ is the
impulse response of the filter, then

$$\text{PWR} = \frac{\int_{-\pi/\omega_{20}}^{\pi/\omega_{20}} |g(x)|^2 dx}{\int_{-\infty}^{\pi/\omega_{20}} |g(x)|^2 dx + \int_{-\pi/\omega_{20}}^{\infty} |g(x)|^2 dx}.$$  

Table I compares the Gabor filter and the Gabor-type filters in
Examples 1 and 2 with respect to these three criteria as
well as whether or not the phase of the impulse response is
monotonous. The results were estimated numerically using the same filter parameters used to generate the plots. Since the parameters were chosen to minimize the energy in the difference between the filter impulse responses, their spatial support is similar. Since the PWR is defined in terms of the energy in the filters, their PWR is also similar. However, there are large differences between the NFR and DCR. As can be seen from Table I as well as the plots in Fig. 1, the NFR and DCR is largest for the Gabor filter and smallest for the filter in Example 1. The rejection can be improved at the cost of decreased PWR by decreasing the bandwidth of the filters.

Only the filter in Example 2 does not have monotonous phase,
since its low-pass modulating function is not strictly positive.
Although the degree to which the filters exhibit the properties
described in Section I-B varies, we would expect any of the
filters to be effective in phase-based algorithms because these
algorithms have been found to be fairly insensitive to the exact
form of the filter.

### B. 2-D Images

The 2-D generalization of (2) simply extends the summation
to two dimensions:

$$t(m,n) = \sum_{k,l=-\infty}^\infty a_{k,l} t(m+k, n+l) + b_t(m,n).$$  

The feedback cloning template $A = [a_{k,l}]_{k,l=-\infty}^\infty$ can now be represented by a $(2r+1) \times (2r+1)$ matrix, e.g.,

$$A = \begin{bmatrix}
    a_{-1,1} & a_{0,1} & a_{1,1} \\
    a_{-1,0} & a_{0,0} & a_{1,0} \\
    a_{-1,-1} & a_{0,-1} & a_{1,-1}
\end{bmatrix}.$$  

If the CNN filter is stable, its frequency response is

$$F(e^{j\omega_2}, e^{j\omega_1}) = \frac{b}{\sum_{k,l} a_{k,l} e^{-j(k\omega_{20}+l\omega_{10})}}.$$  

As in the 1-D case, for any 2-D CNN low-pass filter there
corresponds a 2-D CNN Gabor filter tuned to frequency
$(\omega_{20}, \omega_{10})$ obtained by replacing the feedback cloning tem-
plate with

$$A = [a_{k,l} e^{-j(k\omega_{20}+l\omega_{10})}]_{k,l=-\infty}^\infty.$$  

Its impulse response is the impulse response of the low-pass
filter modulated by $e^{j(\omega_{20}m + \omega_{10}n)}$. Its frequency response is

$$F(e^{j\omega_2}, e^{j\omega_1}) = \sum_{n=-\infty}^{\infty} F(e^{j\omega_2}, e^{j\omega_1}) e^{j\omega_{10}n}.$$
Example 3: The 2-D extension of the resistive grid of Example 1 is shown in Fig. 6(a). The CNN cloning templates which implement the 2-D grid are

\[
A = \begin{bmatrix}
0 & 1 & 0 \\
1 & -(4+\lambda^2) & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

The corresponding Gabor-type filter tuned to \((\omega_x, \omega_y)\), has cloning templates

\[
A = \begin{bmatrix}
0 & e^{-j\omega_y \mu_o} \\
1 & -(4+\lambda^2) & e^{-j\omega_x \mu_o} \\
0 & e^{j\omega_y \mu_o} & 0
\end{bmatrix}
\]

and frequency response

\[
H(e^{j\omega_x}, e^{j\omega_y}) = \frac{\lambda^2}{4+\lambda^2 - 2\cos(\omega_x - \omega_{xo}) - 2\cos(\omega_y - \omega_{yo})}
\]

which is approximately circularly symmetric around \((\omega_{xo}, \omega_{yo})\). The shape of the passband can be stretched in the directions perpendicular to the \(\omega_x\) and \(\omega_y\) axes by scaling the values of the horizontal and vertical connections. Although they are restricted to nearest neighbors, the additional connections can be quite complex. However, if \(\omega_{yo} = 0\), the array reduces to a set of 1-D filters which are resistively coupled to the rows above and below [see Fig. 6(b)]. With these simpler connections, arbitrary orientations could be obtained by rotating the cell array.

III. SPATIO–TEMPORAL FILTERS

The previous section detailed the construction of CNN spatial filters tuned to arbitrary spatial frequencies \((\omega_x, \omega_y)\). Cascading a CNN spatial filter tuned to \((\omega_x, \omega_y)\) with a temporal filter tuned to \(\omega_{to}\) results in a spatio–temporal filter tuned to \((\omega_x, \omega_y, \omega_{to})\). Since the output of the spatial filter is complex, we must distinguish between positive and negative spatio–temporal frequencies. A temporal filter tuned to \(-\omega_{to}\)
results in a spatio–temporal filter tuned to velocities with the same magnitude in the opposite direction.

CNN spatio–temporal Gabor-type filters constructed in this way are separable in space and time. 3-D spatio–temporal Gabor filters are also space–time separable if their covariance matrices have the form

\[
C = \begin{bmatrix}
C_{xx} & C_{xy} & 0 \\
C_{xy} & C_{yy} & 0 \\
0 & 0 & C_{tt}
\end{bmatrix}.
\]

An important difference between Gabor filters and the CNN filters is that Gabor filters are noncausal in space and time, while the CNN filters are noncausal in space, but causal in time.

In order to ensure space–time separability, the spatial filtering CNN’s must settle much faster than the time scales of the image motion so that their outputs at any time \( t \) are essentially the result of spatial filtering the input at time \( t \). Fortunately, it is easy to design a VLSI implementation of the CNN spatial filters with settling times on the order of microseconds or faster. The time scale of image motion is usually on the order of milliseconds. For example, video frame rates are about one frame every 30–40 ms.

There are several advantages to this spatio–temporal filtering architecture. First, multiple filters tuned to different velocities can be obtained by cascading the same spatio–temporal filtering stage with different temporal filters. Since all of the filters share the same spatial filtering stage, the differences between their outputs are purely a function of the temporal variation of the image, i.e., the motion. Second, the temporal filtering at each pixel is independent of the filtering at the rest of the pixels. Since the output of the spatial frequency stage is bandlimited, it can be subsampled without distortion due to aliasing. Therefore, it is not necessary to build a temporal filter at every pixel of the output. This can result in a significant saving in chip area.

**Example 4:** Denote the output of the spatial filtering stage at a pixel \( \eta \) and time \( t \) by \( \psi(\eta, t) \in \mathbb{C} \). One possible continuous time temporal filter [38] tuned to \( \omega_\theta \) has output \( \psi(\eta, t) \in \mathbb{C} \) which satisfies

\[
\hat{\psi}(\eta, t) = (-\alpha + j\omega_\theta)\hat{\psi}(\eta, t) + \psi(\eta, t)
\]

where \( \alpha > 0 \). It has frequency response

\[
\frac{Y(\eta, \omega_t)}{V(\eta, \omega_t)} = \frac{1}{\alpha + j(\omega_t - \omega_\theta)}.
\]

The combined spatio–temporal frequency response is the product of the spatial and temporal frequency responses. Using the spatial filtering stage of Example 1, the frequency response is

\[
H(e^{j\omega_x}, \omega_t) = \frac{1}{\alpha + j(\omega_t - \omega_\theta)} \cdot \frac{\lambda^2}{\alpha + j(\omega_t - \omega_\theta)}
\]

whose magnitude is plotted in Fig. 7. Since the corresponding impulse response is complex valued, the frequency response is not symmetric with respect to the origin. Due to spatial sampling, the spatio–temporal frequency response is periodic in \( \omega_x \) with period \( 2\pi \). There is no periodicity in \( \omega_t \) since the temporal filter operates in continuous time. Similarly, the impulse response is the product of the spatial and temporal impulse responses

\[
h(n, t) \approx \left\{ \begin{array}{ll}
\frac{\lambda}{2\pi} e^{-\lambda|n|} e^{-\alpha t} e^{j(\omega_\theta n + \omega_\tau t)}, & t \geq 0 \\
0, & t < 0.
\end{array} \right.
\]

Intuitive insight into the operation of the spatio–temporal filter can be gained by considering the effect of the temporal filter on one pixel in isolation since the temporal filtering stage processes each output pixel of the spatial filter independently. Assuming uniform translation, the output of the spatial filtering stage at pixel \( n \) and time \( t = t_0 \) can be written as

\[
\psi(n, t) = R(n, t_0) e^{j\omega_\theta n} e^{j\omega_\tau t} \approx [R(n, t_0) e^{j\omega_\theta n}] e^{-j\omega_\tau t}
\]

where \( t_0 \) is constant and \( R(n, t_0) \in \mathbb{C} \) varies slowly in \( n \) and \( t \). The output of the spatial filter at a fixed pixel \( n \) rotates around the origin of the complex plane with frequency \( -\omega_\tau \). The speed of rotation is proportional to the speed of translation. The direction of rotation depends on the direction of translation. Assuming \( \omega_\tau > 0 \), the direction of rotation is clockwise for positive velocities and counterclockwise for negative (see Fig. 8).
The differential equation satisfied by the temporal filter corresponds to a damped linear oscillator forced by the output of the spatial filter. The derivative along different trajectories of the unforced system are shown in Fig. 9 for $\omega > 0$. If the velocity is negative, the input rotates counterclockwise, facilitating the natural motion of output trajectories and leading to a large response. For positive velocity, the input rotation opposes the natural motion, leading to a small response.

In an analog circuit, the filter can be implemented with two capacitors whose voltages represent the real and imaginary parts of the output. Let $y_n(t)$ and $\tilde{y}_n(t)$ denote the real and imaginary parts of the temporal filter output. These satisfy the following real valued differential equation:

$$\begin{bmatrix} \dot{y}_r(n,t) \\ \dot{\tilde{y}}_r(n,t) \end{bmatrix} = \begin{bmatrix} -\alpha & -\omega \\ \omega & -\alpha \end{bmatrix} \begin{bmatrix} y_r(n,t) \\ \tilde{y}_r(n,t) \end{bmatrix} + \begin{bmatrix} v_r(n,t) \\ \tilde{v}_r(n,t) \end{bmatrix}. \tag{9}$$

Writing KCL at the inverting inputs of the first and third operational amplifiers and assuming unit capacitance, it can readily be shown that the circuit shown in Fig. 10 implements this differential equation. To cascade this temporal filter with the spatial filter in Example 1, the inputs $v_r(n,t)$ and $\tilde{v}_r(n,t)$ should be equal to the voltages $v_p(n)$ and $\tilde{v}(n)$ from the bottom and top rows of Fig. 4.

**IV. CONCLUSION**

This paper has defined a class of Gabor-type spatial and spatio-temporal filters which posses the important properties of filters used in phase-based algorithms for computer vision. Gabor-type filters can be implemented using CNN's. It is possible to convert any low-pass spatial filtering CNN into a corresponding Gabor-type filtering CNN by a simple transformation of the cloning template coefficients. Spatio-temporal
filters tuned to different velocities can be obtained by cascading the spatial filter outputs with appropriately designed temporal bandpass filters.

Ongoing work includes the design and fabrication of the architecture described in Example 1 using the 2.0-μm process provided by MOSIS [39]. Future work includes incorporating this into a chip implementing a spatio–temporal filter, as described in Example 4. We are also investigating the application of these chips to various computer-vision tasks such as binocular stereo vergence control and computation of time-to-contact [26].

REFERENCES


Bertram E. Shi (S’93–M’95) received the B.S. and M.S. degrees in electrical engineering from Stanford University, Stanford, CA, in 1987 and 1988, respectively, and the Ph.D. degree in electrical engineering from the University of California at Berkeley, in 1994. Since 1994, he has been an Assistant Professor in the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. His research interests are in CNN’s, image processing, computer vision, and speech recognition.