METHODS FOR IMAGE PROCESSING AND PATTERN FORMATION IN CELLULAR NEURAL NETWORKS: A TUTORIAL

by

Kenneth R. Crounse and Leon O. Chua

Memorandum No. UCB/ERL M95/45

16 June 1995

METHODS FOR IMAGE PROCESSING AND PATTERN FORMATION IN CELLULAR NEURAL NETWORKS: A TUTORIAL

by

Kenneth R. Crounse and Leon O. Chua

Memorandum No. UCB/ERL M95/45

16 June 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial

Kenneth R. Crounse* Leon O. Chua †

June 16, 1995

Abstract

In this paper we demonstrate that many image processing and pattern formation effects of the simple Cellular Neural Network can be understood by means of a common approach. By examining the dynamics in the frequency domain, when all CNN cells are in the linear region, the mechanisms for IIR spatial filtering, pattern formation, morphogenesis, and synergetics can be shown to be present, even though each cell has only first order dynamics. In addition, the method allows many of the standard CNN templates, such as the nonlinear 'averaging', 'halftoning', and 'diffusion' templates to be explained in a new light. Through many examples, it is shown how generalizations of these templates can be used to design linear and nonlinear filters for image processing tasks such as low-pass filtering, time-varying spatial filtering, and fingerprint enhancement.

1 Introduction

In recent years, there has been much interest in using large scale homogeneous cellular arrays of simple circuits to perform image processing tasks and to demonstrate interesting pattern-forming phenomena. The Cellular Neural Network (CNN), first introduced[1] as an implementable alternative to fully connected neural networks, has evolved into a paradigm for these types of arrays[2].

The motivation for studying the usefulness of such arrays has included the biology of the retina, the systems of morphogenesis, and cellular automata. Examples of these categories include: The silicon retina[3] and the general resistive grids, which have been able to produce many effects of linear and nonlinear spatial filtering and motion sensitive filtering. The systems motivated by the Reaction-Diffusion Equation [4] which can produce effects in fingerprint enhancement, textile fault detection, pattern formation and Synergetics[5]. Finally, the cellular automata based methods which can be used for mathematical morphology[6] and modelling physical systems such as the Ising Spin Glass[7].

In this tutorial we show how many of these phenomena in image processing and pattern formation can be demonstrated by using the original and simplest Cellular Neural Network [1]. The reader interested in designing CNNs for image processing will learn of methods for linear filter design and some nonlinear enhancement techniques. Many standard CNN templates will be explained in this light and can be generalized for other applications. The reader familiar with image processing techniques can hope to learn how many standard approaches in digital image processing can be realized by the analog CNN hardware.

The simplest CNN cell has a single capacitor, giving it first-order dynamics, and is coupled to neighboring cells through non-linear controlled sources. A single transient of the CNN is described by:

$$\frac{d}{dt}x_{i,j}(t) = -x_{i,j}(t) + \sum_{k,l\in\mathcal{N}} A_{k,l}y_{i+k,j+k}(t) + \sum_{k,l\in\mathcal{N}} B_{k,l}u_{i+k,j+k} + I \qquad (1)$$

*Sponsored under the Joint Services Electronics Program, Contract Number F49620-94-C-0038. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation herein.

†The authors are with the Electronics Research Laboratory, University of California at Berkeley, 94720.

with output nonlinearity

$$y(x) = \frac{1}{2}\left[|x-1| - |x+1|\right]$$

as shown in Figure 1. The input, state, and output, represented by $u_{i,j}, x_{i,j}$,and $y_{i,j}$ respectively, are defined on $0 \leq i \leq N_1$ and $0 \leq j \leq N_2$. In a real circuit there are various ways to deal with the boundary. Some of the more analytically convenient will be discussed in Section 2.

Some general results have been obtained regarding the effects of the A template on the behavior of the CNN. First, to guarantee the the CNN will converge to a stable equilibrium it is sufficient to have a symmetric A template, that is $\forall k, l \in \mathcal{N} : A_{k,l} = A_{-k,-l}$. Second, when $A_{0,0} > 1$ then all outputs in steady state will be either ± 1.

The emphasis on the A template can be easily understood by writing Equation 1 in block diagram form, as shown in Figure 2. The correlation sums in the equation can be written as convolutions (denoted by an asterisk) when the templates are symmetric. From the diagram, it can be seen that the B template forms a simple feedforward finite impulse response (FIR) filtered version of the input - which itself can be considered a static input to the rest of the system. On the other hand, the A template is operating in a *feedback* loop along with a nonlinearity - a feature that gives interesting behavior.

Due to implementability concerns, the template neighborhood radius is generally restricted to be as small as possible and the templates are applied in a space-invariant manner. The examples in this tutorial use templates no larger than 5×5, although the theory applies to larger neighborhoods as well. Template symmetry will also be assumed throughout the discussion although many of the results, especially those for stable linear filtering, can be generalized for the asymmetric case.

A typical approach to using such a circuit for image processing, which we use here, is to assume that both the initial state and input are available to hold image 'data' which is to be processed. The task that is performed by letting the transient run is determined by the weights of the A and B templates, the constant bias I, and the time at which the transient is stopped. In other situations, special boundary conditions, time or space-variant templates, special nonlinearities, multiple layers or higher-order cells, and many other properties can be adjusted to get a desired behavior[2]. None of these are discussed in this paper.

The elementary image processing tasks performed on the input data by a single template set can be combined into more complicated operations by an invention known as the CNN Universal Machine[8]. The machine uses the simple CNN in a time-multiplexed fashion, analogous to the ALU of a microprocessor, by controlling the template weights and the source of the data inputs for each operation. The machine supplies memory and register transfers at each cell which allow the outputs of simple CNN operations to be combined and/or supplied to the inputs of the next operation, thereby allowing more complex algorithms to be implemented.

In Section 2 the CNN Central Linear System is introduced as essential background for the following discussions. The presentation is made from a signal processing point of view, notably by the use of the spatial frequency domain to write the time solution of the network. Some boundary conditions are discussed which have the advantage of either being analyzed intuitively, being implementable, or both. In Section 3 CNNs with a stable linearization are discussed in terms of image processing capabilities and pattern formation properties. Particularly it is shown that infinite impulse response (IIR) zero-phase spatial linear filtering can be performed both at the network equilibrium and in a time-varying manner. This interpretation can aid in understanding some conventional templates, such as *diffusion*, as well as offer improvements and design methods for linear filtering on the CNN. Section 4 moves onward to CNNs with unstable linearizations. When the input variable is not used for data, these systems can be considered as spatial frequency generalizations of the classical *averaging* template[9]. One example is the 'fingerprint enhancement' which is discussed in terms of linear pre-filtering followed by morphological constraint. When inputs are used, the process can be understood as a generalization of the *halftoning* template[10]. The input can be thought of as a 'reality' constraint. Finally, Section 5 gives a brief conclusion.

2 Essential Background – The CNN Central Linear System

Because the output nonlinearity applied to the state (see Figure 1) is piecewise linear, the state space can be considered to be the union of regions where the dynamics are exactly linear. The most important region

for our purposes, called the *central state space*, is where all cells have states $-1 \leq x_{i,j} \leq 1$.

In this section, the nonlinearity is ignored and it is assumed that the dynamics in the central state space have been extended to all of state space thereby forming the new central linear system. Later, the nonlinearity will be reintroduced when the effects of saturated cells become unavoidable.

2.1 Spatial Convolution Formulation

To understand the behavior of the CNN central linear system it is convenient to use notation borrowed from image processing. In particular, the state evolution is written in terms of the convolution of two-dimensional infinite spatial sequences.

If all $|x_{i,j}| < 1$, then $y_{i,j} = x_{i,j}$ and the whole system behaves according to the linear system

$$\frac{d}{dt}x_{i,j}(t) = -x_{i,j}(t) + \sum_{k,l \in \mathcal{N}} A_{k+i,l+j}\, x_{k,l}(t) + \sum_{k,l \in \mathcal{N}} B_{k+i,l+j}\, u_{k,l} + I$$

which we now assume to operate over all state space.

To emphasize the point-of-view of sequences undergoing spatial processing which is merely being indexed by time, we now shift notation so that spatial indices are written as arguments and time as a subscript. For simplicity, define the *linearized template masks* as follows:

$$a(n_1, n_2) = \begin{cases} A_{0,0} - 1 & (n_1, n_2) = (0,0) \\ A_{-n_1,-n_2} & \text{for } -n_1, -n_2 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$b(n_1, n_2) = \begin{cases} B_{-n_1,-n_2} & \text{for } -n_1, -n_2 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Note that by specifying either the template or the linearized template mask, the other can be obtained uniquely. For now assume that the sequences $x_0(n_1, n_2)$ and $u(n_1, n_2)$ are defined to be zero for all integers n_1 and n_2 where the supplied data are not defined.

Then the dynamics can be written in convolution form:

$$\frac{d}{dt}x_t(n_1, n_2) = a(n_1, n_2) * x_t(n_1, n_2) + b(n_1, n_2) * u(n_1, n_2) + I \tag{3}$$

2.2 Spatial Frequency Formulation

Because of the spatial mixing introduced by the coupling, writing the time solution of the above equation would not be very intuitive. However, it is straightforward to transform the system by a change of basis into one which is uncoupled, an approach which has a long history in the morphogenesis literature [11]. This will first be shown for the more-intuitive infinite system (although it may not hold in special circumstances). Next, finite arrays will be considered, particularly two cases that give results equivalent to the infinite case.

Throughout this paper we will make use of the two-dimensional Discrete Space Fourier Transform (DSFT) which gives the representation of a sequence in the basis of complex exponentials.

$$\tilde{F}(\omega_1, \omega_2) = \sum_{n_1} \sum_{n_2} f(n_1, n_2) \mathrm{e}^{-j\omega_1 n_1} \mathrm{e}^{-j\omega_2 n_2} \tag{4}$$

and can be reconstructed by the Inverse DSFT:

$$f(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{F}(\omega_1, \omega_2) \mathrm{e}^{j\omega_1 n_1} \mathrm{e}^{j\omega_2 n_2} d\omega_1 d\omega_2 \tag{5}$$

Some of the interesting properties of the DSFT along with some notation which will be used:

- Real, Symmetric $f(n_1, n_2) \leftrightarrow \tilde{F}(\omega_1, \omega_2)$ Real, Symmetric

- $f(0,0) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\tilde{F}(\omega_1,\omega_2)d\omega_1 d\omega_2$ In fact, from the definition it can be seen that $f(0,0)$ adds to the new basis representation equally and can therefore be considered an offset. That is, changing the center element, $f(0,0)$ just shifts the curve $\tilde{F}(\omega_1,\omega_2)$ up or down.
- $f(n_1,n_2) * g(n_1,n_2) \leftrightarrow \tilde{F}(\omega_1,\omega_2)\tilde{G}(\omega_1,\omega_2)$
- $\delta(n_1,n_2)$ is the discrete *delta function* which is zero everywhere except $\delta(0,0) = 1$. $\delta(\omega_1,\omega_2)$ is the continuous support delta function which is zero everywhere except has *area* one at the origin.

2.2.1 Infinite Array

Assuming all the DSFTs exist, the dynamics can be written in the new basis by transforming Equation 3 into:

$$\frac{d}{dt}\tilde{X}_t(\omega_1,\omega_2) = \tilde{A}(\omega_1,\omega_2)\tilde{X}_t(\omega_1,\omega_2) + \tilde{B}(\omega_1,\omega_2)\tilde{U}(\omega_1,\omega_2) + I\delta(\omega_1,\omega_2) \tag{6}$$

which is uncoupled in spatial frequency - that is, for each ω_1,ω_2 this is a single linear first order ordinary differential equation. This equation demonstrates the way that the coefficients of each of the basis functions (modes) of the new basis changes in time. The dynamics of the modes evolve independently for each spatial frequency.

2.2.2 Finite Arrays

The infinite array is convenient for discussion since the basis representation can be discussed independent of array size. However, it can be shown that these results are applicable to the finite size of real circuits. Here we discuss two ways of dealing with the boundaries of a finite array which retains the same features of the above decoupling. These results are related to those found in many other places [11, 12, 13, 14, 15, for example].

Consider the $N_1 \times N_2$ CNN for which $u(n_1,n_2)$ and $x(n_1,n_2)$ are defined on $n_1 \in \{0,1,\ldots,N_1-1\}, n_2 \in \{0,1,\ldots,N_2-1\}$.

First, consider the 'wrap-around' connections which give the topology of a torus. That is, cells on the left and right boundaries 'see' each other as neighbors and cells on the top and bottom 'see' each other as neighbors. In a VLSI implementation this would take quite a bit of extra wiring.

Such a finite system has infinite system representation which has equivalent dynamics on the region of interest. If we tile the plane in a periodic manner with copies of the input and initial conditions, as shown in Figure 3, the resulting dynamics at each cell are identical. An infinite periodic function of this sort can always be written in the basis of the Discrete Fourier Series. That is

$$x(n_1,n_2) = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}\tilde{X}(k_1,k_2)e^{j(2\pi/N_1)k_1n_1}e^{j(2\pi/N_2)k_2n_2}$$

It is easy to show that writing Equation 3 in this basis finally gives:

$$\frac{d}{dt}\tilde{X}_t(k_1,k_2) = \tilde{A}(\frac{2\pi k_1}{N_1},\frac{2\pi k_2}{N_2})\tilde{X}_t(k_1,k_2) + \tilde{B}(\frac{2\pi k_1}{N_1},\frac{2\pi k_2}{N_2})\tilde{U}(k_1,k_2) + I\delta(k_1,k_2)$$

And, in fact,

$$\tilde{X}_t(k_1,k_2) = \tilde{X}(\omega_1,\omega_2)|_{\omega_1=(2\pi/N_1)k_1,\omega_2=(2\pi/N_2)k_2}$$

Therefore Equation 6 can be used, but only a finite number of ω_1,ω_2 are needed.

Secondly, consider the finite array with 'reflected' connections. From the point of view of the cells there is a reflected array at every boundary so that a cell sees itself as its immediate neighbor off the array, and so forth. This would be much easier to implement in circuitry since the connections remain local.

Such an array can be seen to be equivalent to an infinite array where the initial conditions and inputs have been reflected and tiled as shown in Figure 3. Note that this method depends critically on the symmetry of the A-template. An infinite periodic function of this sort can always be written in the basis of the Discrete Cosine Series.

$$x(n_1,n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \tilde{X}'(k_1,k_2) \cos(\frac{\pi k_1(2n_1+1)}{2N_1}) \cos(\frac{\pi k_2(2n_2+1)}{2N_2})$$

And, again, writing Equation 3 in the new basis gives:

$$\frac{d}{dt}\tilde{X}'_t(k_1,k_2) = \tilde{A}(\frac{\pi k_1}{N_1},\frac{\pi k_2}{N_2})\tilde{X}'_t(k_1,k_2) + \tilde{B}(\frac{\pi k_1}{N_1},\frac{\pi k_2}{N_2})\tilde{U}'(k_1,k_2) + I\delta(k_1,k_2)$$

Many other boundary conditions are possible and have been considered [13, for instance]. From an implementation point of view setting the states and inputs off the array to zero is very convenient, but may cause unpleasant effects in image processing due to the abrupt change at the boundary. It should also be mentioned that the various boundary conditions can also be understood in terms of the special forms of state matrices imposed by the boundary type. For instance, the block circulants with circulant blocks[16] are very powerful representation for the wrap-around system[12, 14].

2.3 Time Solution

The discussion on boundary conditions showed that we actually only need to study one differential equation to understand the behavior of any of these systems. The boundaries only affect the spatial frequency of the modes, and the eigenvalues can all be found by appropriate sampling of the DSFT. Returning to the original modal differential equation, Equation 6, the time solution can be written down for each ω_1, ω_2 from elementary linear systems theory as follows:

When $\tilde{A}(\omega_1,\omega_2) \neq 0$:

$$\tilde{X}_t(\omega_1,\omega_2) = e^{\tilde{A}(\omega_1,\omega_2)t}\tilde{X}_0(\omega_1,\omega_2) + \frac{1}{\tilde{A}(\omega_1,\omega_2)}\left[e^{\tilde{A}(\omega_1,\omega_2)t} - 1\right]\tilde{B}(\omega_1,\omega_2)\tilde{U}(\omega_1,\omega_2) \tag{7}$$

When $\tilde{A}(\omega_1,\omega_2) = 0$:

$$\tilde{X}_t(\omega_1,\omega_2) = \tilde{X}_0(\omega_1,\omega_2) + t\tilde{B}(\omega_1,\omega_2)\tilde{U}(\omega_1,\omega_2) \tag{8}$$

For notational simplicity we have dropped the bias term, which can always be considered to be a special case of the input and incorporated into the $\tilde{B}(\omega_1,\omega_2)\tilde{U}(\omega_1,\omega_2)$ term.

The eigenvalues $\tilde{A}(\omega_1,\omega_2)$ determine the stability of the central linear system, while the relationship among the eigenvalues determines the type of processing to be performed. The intimate connection between stability, convergence speed, and processing function through the transformed linearized template mask $\tilde{A}(\omega_1,\omega_2)$ makes template design in the frequency domain very attractive. Note that specifying $\tilde{A}(\omega_1,\omega_2)$ uniquely specifies $a(n_1,n_2)$ which in turn uniquely specifies the A template.

Some observations can be made about the relationship between the A template and $\tilde{A}(\omega_1,\omega_2)$. Template symmetry, which we assume throughout, implies that all the $\tilde{A}(\omega_1,\omega_2)$ will be real - the modes are not phase shifted. Second, templates with a small radius will only use a few term of Equation 4 and therefore $\tilde{A}(\omega_1,\omega_2)$ will be very smooth and slowly varying. In fact, with a single 3×3 mask it is impossible to get band-pass behavior.

3 Stable Central Linear Systems

To help develop an intuitive notion for the behavior of Equation 7 we will make separate studies of its stable and unstable operation. It will be shown that this division leads to qualitatively different image processing and pattern formation behaviors.

3.1 Equilibrium Linear Spatial Filtering

If we have

$$\tilde{A}(\omega_1, \omega_2) < 0$$

then the central linear system is stable and all the exponential terms will tend to zero as time goes to infinity. The stable equilibrium, which is independent of the initial conditions, can easily be found by finding the limit of Equation 7:

$$\tilde{X}_\infty(\omega_1, \omega_2) = \tilde{H}(\omega_1, \omega_2)\tilde{U}(\omega_1, \omega_2)$$

where

$$\tilde{H}(\omega_1, \omega_2) = \frac{-1}{\tilde{A}(\omega_1, \omega_2)}\tilde{B}(\omega_1, \omega_2) \tag{9}$$

The equilibrium state can be thought of as a version of the input which has been spatially filtered by the transfer function $\tilde{H}(\omega_1, \omega_2)$. The filter $\tilde{H}(\omega_1, \omega_2)$ can be shown to have two important properties: 'zero-phase' and 'infinite impulse response' (IIR). Because the A and B templates are real and symmetric, their DSFTs are as well. Therefore, the phase of a spatial sinusoid in the input image is not modified by the transfer functions $\tilde{A}(\omega_1, \omega_2)$ and $\tilde{B}(\omega_1, \omega_2)$. Since $\tilde{H}(\omega_1, \omega_2)$ is a simple function of these transfer characteristic, it will inherit the zero-phase property. And, because the transfer function is made by finding the inverse of the FIR filter $a(n_1, n_2)$ it is, typically, spatially infinite in extent. That is, due to the feedback in the dynamics, the *local* connections of the A template can be used to perform *non-local filtering*.

The simplest way equilibrium linear filtering is performed in the CNN is to only make use of the B template. If the A template has radius zero with $A_{0,0} = \delta < 1$ then the transfer function is given by

$$\tilde{H}(\omega_1, \omega_2) = \frac{-1}{\delta - 1}\tilde{B}(\omega_1, \omega_2)$$

When the obvious choice of $\delta = 0$ is used, the transfer function is equivalent to the FIR filtering of the B-template. Smaller δ will provide faster convergence, but also scales down the output.

Means to incorporate the A template into linear filtering tasks is intuitively less obvious but has been done for several applications[17, 18]. Using the A-template can supply many advantages such as IIR filtering increased frequency shaping for a given template size. Linear spatial filtering is the work horse of image processing algorithms, and the many applications which use linear filtering, such as interpolation, visual modelling, and image compression, could benefit. We now give some examples of approaches to designing such templates.

Example 1 – Low-Pass Filter Design

Good quality low-pass filters have many important uses, such as image interpolation, and therefore provide a speed/performance benchmark for any image processing hardware. A CNN Universal Machine could implement low-pass filters through the serial and/or parallel combination of many linear filtering steps, each of which would happen on the order of a single transient time.

It is interesting to see how close a single CNN template pair can come to performing ideal low-pass filtering of the input by using the equilibrium approach described above. Since the steady state filtering performed by a given template pair can be stated explicitly as a nonlinear function of the template elements, as in Equation 9, a nonlinear constrained error minimization technique can be used.

There are important system constraints on the design process in both the template domain and the frequency domain. Perhaps the two most important are that the template sizes should be small and fixed, and that the gains of $\tilde{A}(\omega_1, \omega_2)$ are strictly negative. Some other concerns are the available range and accuracy of the template elements of a particular CNN implementation and slow convergence speed if the eigenvalues $\tilde{A}(\omega_1, \omega_2)$ are too small.

Since the typical situation is to fix the template size, it is advisable to use use the template elements as the parameters for the minimization process. But, since filter design specifications are usually given in the frequency domain, the error and design constraints should be determined there. For a 5×5 template, this is a 26-dimensional minimization which requires computation of the DSFT to evaluate the error function. We have tried such an approach when reducing the parameter set to 12 by imposing circular symmetry.

Another approach is to use a transformation method similar to that used in FIR filter design[19]. The method will allow us to perform the minimization in the frequency domain while retaining control over template size. In addition, the method reduces the number of parameters to be minimized over by requiring specification ahead of time of the contours of constant gain for the transfer function being designed.

Let $\tilde{C}(\omega_1,\omega_2)$ be a filter with desired contours. Then, if we specify

$$\tilde{A}(\omega_1,\omega_2) = \sum_{r=0}^{R} \alpha_r \tilde{C}^r(\omega_1,\omega_2) \tag{10}$$

$$\tilde{B}(\omega_1,\omega_2) = \sum_{r=0}^{R} \beta_r \tilde{C}^r(\omega_1,\omega_2) \tag{11}$$

Both $\tilde{A}(\omega_1,\omega_2)$ and $\tilde{B}(\omega_1,\omega_2)$ are simple continuous functions of $\tilde{C}(\omega_1,\omega_2)$ and they will both have the same shaped constant contours as $\tilde{C}(\omega_1,\omega_2)$ - and therefore $\tilde{H}(\omega_1,\omega_2)$ will as well. Also of importance, by choosing a $c(n_1,n_2)$ to be nonzero only on a finite support and R appropriately, the size of the support of the nonzero parts of $a(n_1,n_2)$ and $b(n_1,n_2)$ can be controlled.

This method will now be used to design 5×5 A and B templates with the goal of a circularly symmetric low-pass filter at equilibrium with passband extending to 0.4π and stop band starting at 0.5π.

The sequence with the nonzero elements

$$c(n_1,n_2) = \begin{array}{ccc} 0.25 & 0.50 & 0.25 \\ 0.50 & \boxed{1.0} & 0.5 \\ 0.25 & 0.50 & 0.25 \end{array} \tag{12}$$

where the boxed element is at the n_1, n_2 origin, is known to have contours with good circular symmetry and radial monotonicity. Because we want our templates to be 5×5, we can only use $R = 2$ in Equation 11 - since the Inverse DSFT of $\tilde{C}^r(\omega_1,\omega_2)$ is equivalent to $c(n_1,n_2)$ convolved with itself $r-1$ times and each convolution, with this particular mask, increases the template radius by one. For instance,

$$c(n_1,n_2) * c(n_1,n_2) = \begin{array}{ccccc} 0.0625 & 0.2500 & 0.3750 & 0.2500 & 0.0625 \\ 0.2500 & 1.0000 & 1.5000 & 1.0000 & 0.2500 \\ 0.3750 & 1.5000 & \boxed{2.2500} & 1.5000 & 0.3750 \\ 0.2500 & 1.0000 & 1.5000 & 1.0000 & 0.2500 \\ 0.0625 & 0.2500 & 0.3750 & 0.2500 & 0.0625 \end{array} \tag{13}$$

Performing a frequency-weighted minimization with respect to $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2$ returned:

$$\begin{aligned} \tilde{A}(\omega_1,\omega_2) &= -4.8326 + 3.4550\tilde{C}(\omega_1,\omega_2) - 0.7278\tilde{C}^2(\omega_1,\omega_2) \\ \tilde{B}(\omega_1,\omega_2) &= \quad 0.2824 - 0.7382\tilde{C}(\omega_1,\omega_2) + 0.3294\tilde{C}^2(\omega_1,\omega_2) \end{aligned} \tag{14}$$

By taking the Inverse DSFT of Equation 14, recognizing that IDSFT$[\tilde{C}^0(\omega_1,\omega_2)] = \delta(n_1,n_2)$, the sequences $a(n_1,n_2)$ and $b(n_1,n_2)$, and therefore the A and B templates, can easily be found to be:

Low-Pass Filter Templates

$$A = \begin{bmatrix} -0.1137 & -0.4549 & -0.6823 & -0.4549 & -0.1137 \\ -0.4549 & 0.3399 & 1.5896 & 0.3399 & -0.4549 \\ -0.6823 & 1.5896 & -6.5380 & 1.5896 & -0.6823 \\ -0.4549 & 0.3399 & 1.5896 & 0.3399 & -0.4549 \\ -0.1137 & -0.4549 & -0.6823 & -0.4549 & -0.1137 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0515 & 0.2059 & 0.3089 & 0.2059 & 0.0515 \\ 0.2059 & 0.3623 & 0.3127 & 0.3623 & 0.2059 \\ 0.3089 & 0.3127 & 0.7136 & 0.3127 & 0.3089 \\ 0.2059 & 0.3623 & 0.3127 & 0.3623 & 0.2059 \\ 0.0515 & 0.2059 & 0.3089 & 0.2059 & 0.0515 \end{bmatrix}$$

The results are shown in Figure 4. It is instructive to observe the 'strategy' of the minimization algorithm. The B template can be seen to be the main agent of low-pass filtering, although due to its limited size, it does not have a very sharp cutoff. The action of the A template is to provide sharpness by increasing the spatial frequencies that have been dampened by the B template's smooth roll-off.

■

Example 2 – Pattern Formation from Randomness

To demonstrate that patterns can be generated by using the CNN as a purely stable linear system, the following templates were designed using the intuitive approach presented in Example 4.

Aggressive Band-Pass Filtering Templates

$$A = \begin{bmatrix} -0.125 & -0.5 & -0.750 & -0.5 & -0.125 \\ -0.5 & 0.0 & 1.0 & 0.0 & -0.5 \\ -0.750 & 1.0 & -3.75 & 1.0 & -0.750 \\ -0.5 & 0.0 & 1.0 & 0.0 & -0.5 \\ -0.125 & -0.5 & -0.750 & -0.5 & -0.125 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0625 & -0.25 & -0.375 & -0.25 & -0.0625 \\ -0.25 & 0.0 & 0.5 & 0.0 & -0.25 \\ -0.375 & 0.5 & 1.75 & 0.5 & -0.375 \\ -0.25 & 0.0 & 0.5 & 0.0 & -0.25 \\ -0.0625 & -0.25 & -0.375 & -0.25 & -0.0625 \end{bmatrix}$$

The steady state characteristic of these mild-valued templates is that of an aggressive band pass filter, as shown in Figure 5. The input to the system is a low level randomness, but the steady state filter gain at the spatial frequencies with a half-period of two pixels is so great that textures of this feature size dominate in the output. This development of selection among the spatial modes is an order which can be considered a form of pattern formation. However, later we will see that introducing nonlinearity and instability can exhibit more complex pattern forming behavior.

■

3.2 Time-Varying Spatial Filtering

Although the steady-state dynamics are the obvious way to use the stable system, it only makes use of the input control and not the initial conditions. Instead, if the system can be stopped at an arbitrary time, some other interesting filtering effects can be obtained.

One possibility to consider is when at least one of the

$$\tilde{A}(\omega_1, \omega_2) = 0$$

From Equations 7 and 8 it can be seen that this will in general be unstable unless the input is not connected. Making this so, we have

$$\tilde{X}_t(\omega_1, \omega_2) = \tilde{\Gamma}_t(\omega_1, \omega_2)\tilde{X}_0(\omega_1, \omega_2) \tag{15}$$

where

$$\tilde{\Gamma}_t(\omega_1, \omega_2) = e^{\tilde{A}(\omega_1, \omega_2)t} \tag{16}$$

Since the input is not used, the data must be placed in the initial condition. The term $\tilde{\Gamma}_t(\omega_1, \omega_2)$ can be thought as a time-varying transfer function. As time progresses, each mode with negative A-template gain is attenuated until it is eventually completely eliminated. Only the modes with zero gain remain unchanged, and therefore are the only modes remaining at time infinity. If the zero gain modes are chosen at spatial frequencies of interest, then the filtering performed will favor these more and more over time.

Example 3 – Stopped Diffusion

Now we consider the templates known as the Diffusion Template and the Laplace Template because they can be considered discrete space approximations to the Laplacian operator $-\nabla^2$ found in the diffusion and

Laplace partial differential equations. The intuition is that the data is supplied to the initial conditions and then are allowed to diffuse for some convenient amount of time, therefore smoothing out the data.

The filtering done can be understood in terms of Equation 15. Consider the following 'Laplace' template:

$$\begin{array}{ll} \text{'Laplace'} \\ \text{Template} \end{array} \qquad A = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & -3.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$$

The eigenvalues, $\tilde{A}(\omega_1, \omega_2)$ are shown in Figure 6. There is one zero eigenvalue at $\omega_1, \omega_2 = 0, 0$ or 'DC' which can be easily seen since the sum of the linear template mask corresponding to the A template is zero. All other eigenvalues are negative. Using Equation 16 we see that as time proceeds only the $\omega_1, \omega_2 = 0, 0$ mode (the average value) of the initial condition remains unchanged since $\tilde{\Gamma}_t(0,0) = 1$ for all time where as all other $\tilde{\Gamma}_{t \to \infty}(\omega_1, \omega_2) \to 0$.

Figure 6 shows the equivalent spatial filtering transfer functions when stopping the transient at various times. The input is progressively more strictly low-passed. It is interesting to note that the equivalent filter is not very circularly symmetric when waiting only a short time. This could perhaps be improved by using the analogous template

$$\begin{array}{ll} \text{Circular} \\ \text{'Diffusion'} \\ \text{Template} \end{array} \qquad A = \begin{bmatrix} 0.5 & 1.0 & 0.5 \\ 1.0 & -5.0 & 1.0 \\ 0.5 & 1.0 & 0.5 \end{bmatrix}$$

which is based on the $c(n_1, n_2)$ mask of Example 1. Finally, the impulse response of the equivalent filter achieved at time $t = 10$ is shown in Figure 6 to be significant on a 22×22 support demonstrating the tradeoff between the time taken to process and the spatial extent of effect. If very strong filtering is required, the speed of convergence can be increased simply by multiplying the linear template mask $a(n_1, n_2)$ by a constant greater than one. Note that this is not exactly equivalent to multiplying the A-template by a constant K, but the new center element must be found by

$$A'_{0,0} = K \times (A_{0,0} - 1) + 1$$

Of course, this method will only work as long as larger template values are available on the particular CNN implementation.

■

Time-varying filtering can be done in a robust manner as well using both the initial condition and input for data. In this situation, $\tilde{\Gamma}_t(\omega_1, \omega_2)$ varies smoothly from 1 to 0 as time goes from 0 to ∞.

Then Equation 7 can be written as

$$\tilde{X}_t(\omega_1, \omega_2) = \left[\tilde{\Gamma}_t(\omega_1, \omega_2)\right] \tilde{X}_0(\omega_1, \omega_2) + \left[1 - \tilde{\Gamma}_t(\omega_1, \omega_2)\right] \tilde{H}(\omega_1, \omega_2)\tilde{U}(\omega_1, \omega_2) \tag{17}$$

which shows that at any given time, the time trajectory of the state provides a 'homotopy' between the initial condition and the filtered input. That is, the spatial frequency content in the state image at any given time is a normalized weighted combination (convex sum) between the spatial frequency content initial state and the filtered input. Note that the weights of the combination at a given time depend, in general, on the spatial frequency being considered.

The simplest case is when both templates have radius zero, so that their related DSFTs are constant for all spatial frequency. Then, the $\tilde{\Gamma}_t(\omega_1, \omega_2)$ does not depend on spatial frequency and simple linear combinations of two data images supplied to the initial conditions and input can be formed by stopping the system at the appropriate moment.

Many other schemes are possible. Using filters so that $\tilde{A}(\omega_1, \omega_2) = \tilde{B}(\omega_1, \omega_2)$ may be useful to get results that have the content from the initial conditions at certain desired spatial frequencies and from the input at the remaining spatial frequencies. Some other possibilities are to give zero initial conditions so that only the input is processed, or to supply both input and initial condition with the same data image.

Example 4 – Robust Time-Varying Filter Design.

We now consider the design of a time-varying filter which has similar behavior to the Diffusion Templates. That is, we want to have the filter to smoothly transit from an all-pass at time zero to an 'extreme' characteristic as time gets large.

The design procedure requires a prototype linear mask $p(n_1, n_2)$, to be called the 'midpoint' filter, which is non-zero only for the spatial radius we want the final template to be defined, and has desirable frequency domain characteristics such as the filter type (e.g. low-pass) and the shape of the constant gain contours. We then will show how to design templates to achieve at steady-state an 'extreme' version of the prototype filter. Finally, it will be shown that by supplying the data to be filtered to both the input and initial condition, the time-varying filtering performed will smoothly transit from an all-pass filter to (approximately) the midpoint filter and finally converging to the extreme filter – allowing access to a whole class of filters of increasing strictness in time.

For this example, we chose a normalized version of the Laplace template, which has been shifted to have all positive eigenvalues, as our 'midpoint' transfer function. This corresponds to a design sequence of

$$p(n_1, n_2) = \begin{array}{ccc} 0.0 & 0.125 & 0.0 \\ 0.125 & \boxed{0.5} & 0.125 \\ 0.0 & 0.125 & 0.0 \end{array} \tag{18}$$

which has a smooth low-pass frequency filter characteristic of the same form as Figure 6(Top).

From this we will consider linearized template masks of the form

$$a(n_1, n_2) = \alpha\,(p(n_1, n_2) - \delta(n_1, n_2)) - \beta\delta(n_1, n_2)$$

$$b(n_1, n_2) = \beta p(n_1, n_2)$$

where $\alpha, \beta > 0$. Note that the non-zero support remains the same as for the prototype sequence. For our example, we chose $\alpha = 2, \beta = 1$ which gives the templates:

Robust 'Laplace' Template

$$A = \begin{bmatrix} 0.0 & 0.25 & 0.0 \\ 0.25 & -1.0 & 0.25 \\ 0.0 & 0.25 & 0.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0 & 0.125 & 0.0 \\ 0.125 & 0.5 & 0.125 \\ 0.0 & 0.125 & 0.0 \end{bmatrix}$$

The action of these templates can be graphically understood more intuitively in the frequency domain:

$$\tilde{A}(\omega_1, \omega_2) = \alpha(\tilde{P}(\omega_1, \omega_2) - 1) - \beta$$

$$\tilde{B}(\omega_1, \omega_2) = \beta\tilde{P}(\omega_1, \omega_2)$$

giving the equilibrium filter

$$\tilde{H}(\omega_1, \omega_2) = \frac{-\beta\tilde{P}(\omega_1, \omega_2)}{\alpha\tilde{P}(\omega_1, \omega_2) - \alpha - \beta}$$

which can be confirmed to be an 'extreme' version of $\tilde{P}(\omega_1, \omega_2)$. Spatial frequencies for which $\tilde{P}(\omega_1, \omega_2) = 1$ will have $\tilde{H}(\omega_1, \omega_2) = 1$ but all others will be further attenuated. To see the transformation from prototype to the equilibrium extreme version for the example filter, see Figure 7.

In theory, α and β can be chosen to make the equilibrium filter arbitrarily extreme, as in the Diffusion case. However, note that β represents a stability margin and α modifies the convergence speed. Either increasing α or decreasing β will increase the extremeness of the final equilibrium filter, but sensitivity will increase since the ratio of maximum to minimum eigenvalues is $\approx \frac{\alpha}{\beta}$.

Now we will show how the templates can be used for time-varying filtering. Setting the initial condition and input equal to the same data gives us the time-varying filter form

$$\tilde{X}_t(\omega_1,\omega_2) = \left[\tilde{\Gamma}_t(\omega_1,\omega_2) + \left(1 - \tilde{\Gamma}_t(\omega_1,\omega_2)\right)\tilde{H}(\omega_1,\omega_2)\right]\tilde{U}(\omega_1,\omega_2) \tag{19}$$

Since the system is stable by construction, the homotopy principle is valid and the system can be thought of as a family of filters smoothly varying from all-pass at $t = 0$ to the 'extreme' filter $\tilde{H}(\omega_1,\omega_2)$ at equilibrium. The time-varying filtering operation will have the same contours as the design filter. Note that when $\tilde{P}(\omega_1,\omega_2) = 1$, then $\tilde{H}(\omega_1,\omega_2) = 1$ and therefore these spatial frequencies remain unchanged for all time – exactly as in the case of the Diffusion technique. For our example, the equivalent spatial filter is shown for various time in Figure 8.

Now, to understand that the 'midpoint' filter does in fact live up to its name, we use the approximation that

$$\tilde{\Gamma}_t(\omega_1,\omega_2) = e^{\tilde{A}(\omega_1,\omega_2)t} \approx 1 + \tilde{A}(\omega_1,\omega_2)t \tag{20}$$

for small enough t. Substituting into Equation 19 we find

$$\tilde{X}_t(\omega_1,\omega_2) \approx \left[1 - (\alpha+\beta)t + (\alpha+\beta)t\tilde{P}(\omega_1,\omega_2)\right]\tilde{U}(\omega_1,\omega_2) \tag{21}$$

So choosing $t = \frac{1}{\alpha+\beta}$ gives the desired result

$$\tilde{X}_{t=\frac{1}{\alpha+\beta}}(\omega_1,\omega_2) \approx \tilde{P}(\omega_1,\omega_2)\tilde{U}(\omega_1,\omega_2) \tag{22}$$

■

3.3 Design Issues

There are many pitfalls in designing CNN linear filters. Since convergence to the steady state depends on the spatial frequency – in a most unpleasant way since the spatial frequencies with the most gain move the slowest – it may be wise to design the filters time t' where this is the longest time you're willing to wait for the answer. In addition, minimization techniques tend to choose very small eigenvalues to achieve sharpness. There are robustness issues with implementation. Having very small eigenvalues could be shifted to instability. In addition, spatial variations in template implementation could affect the noise quality of the output. In general these things depend on the specific implementation [20].

3.4 Saturated Behavior

Until now, the saturation nonlinearity has been ignored. There are two ways that it may disrupt the stable linear operation of the CNN. First, the stable equilibrium may lie outside of the central linear state space. This can always be remedied by scaling the templates so that the solution is smaller by an appropriate amount. Unfortunately it may not be known ahead of time if this will occur for a particular input. In other cases it may be beneficial, as demonstrated in the next example.

Second, even though the the stable equilibrium may lie in the central linear space, the trajectory may leave this space and find a stable solution with saturated cells and never return. We have not observed this problem and it may in fact not be impossible.

Example 5 – Constrained Inverse Filtering

When a source image $w(n_1,n_2)$ has been blurred by a known convolution mask $f(n_1,n_2)$, it is very tempting to use Equation 9 to perform the inverse filtering. For instance, if $f(n_1,n_2)$ can reasonably be represented by a mask with small non-zero support, and $\tilde{F}(\omega_1,\omega_2)$ is strictly positive, then choosing $a(n_1,n_2) = -f(n_1,n_2)$ and $b(n_1,n_2) = \delta(n_1,n_2)$ while setting the input $u(n_1,n_2)$ equal to the acquired

image $f(n_1, n_2) * w(n_1, n_2)$ the exact inverse is given in steady state. This can be seen by substituting the appropriate DSFTs into Equation 9 as follows:

$$\tilde{X}_\infty(\omega_1, \omega_2) = \frac{-1}{-\tilde{F}(\omega_1, \omega_2)} \tilde{F}(\omega_1, \omega_2)\tilde{W}(\omega_1, \omega_2) = \tilde{W}(\omega_1, \omega_2)$$

The problem with this approach is noise. Spatial frequencies which have been greatly attenuated by small values of $\tilde{F}(\omega_1, \omega_2)$ will be likely to contain as much noise as signal. When this is inverted by multiplying by the large number $1/\tilde{F}(\omega_1, \omega_2)$ it will produce a large spurious component in the output. [1] Since a potentially large new random signal is being added to the output, the range of intensities in the output is likely to be much larger than the source image $w(n_1, n_2)$.

To demonstrate this, an image of boxes with a known intensity range was blurred by a known convolution mask, which then had spatially uncorrelated noise added to it, as shown in Figure 9. The direct inverse was then performed. The noise amplification dominates, and the image had to be rescaled for printing, resulting in a overall loss of contrast.

One way to get around this is to constrain the inversion process to have values which are in the known intensity range of the source image $w(n_1, n_2)$. The CNN saturation nonlinearity provides a natural way to do this. The same stable templates are applied, but the input is scaled appropriately so that the equilibrium is not with in the central linear state space, thereby using the saturation nonlinearity. The results, as seen in the Figure, are much improved. The exact effects are not well understood, but the saturated CNN appears to do more than produce an output which is just a thresholded version of the purely linear system.

■

4 Unstable Central Linear System

The central linear system with input is unstable if any of the

$$\tilde{A}(\omega_1, \omega_2) \geq 0$$

In the unstable system, the effects of the initial condition do not eventually die away, there is no steady state linear solution, and therefore the previous 'homotopy' interpretation of Equation 17 no longer holds. Because the states of the central linear system will keep growing, the nonlinearity is critical to supply overall stability and cannot be ignored.

4.1 Autonomous Behavior

To help develop a general intuitive understanding of the CNN with an unstable central linear system, we first consider the homogeneous version, $\tilde{U}(\omega_1, \omega_2) = 0$ and $I = 0$. Such a system can be considered to be pattern forming in the 'Turing sense' since any small perturbation around the unstable equilibrium $\tilde{X}_0(\omega_1, \omega_2) = 0$ will cause the unstable modes to grow until saturation is reached. From a CNN image processing point of view, these templates can be considered frequency domain generalizations of the 'averaging' template which perform a signal-dependent nonlinear spatial filtering operation. Such processing can be roughly decomposed into two stages – a linear pre-filtering operation where some spatial frequencies are emphasized and others de-emphasized, followed by a 'morphological constraint' filtering stage where the cellular automata-like behavior of the saturated system causes certain neighborhood configuration to be chosen and others rejected.

4.1.1 Linear Pre-filtering

The linear filtering regime will be valid until some time t' when the first cell becomes saturated. The system stopped at this time can be considered to have performed the linear filtering

$$\tilde{X}_{t'}(\omega_1, \omega_2) = e^{\tilde{A}(\omega_1, \omega_2)t'} \tilde{X}_0(\omega_1, \omega_2)$$

[1] There are many conventional ways to deal with this problem in image processing [19], some of which have been applied to the CNN [18].

It is important to note that for a given template, the time t' depends on the size of the initial conditions, and therefore the pre-filtering operation performed does also. If the initial conditions are very large, almost no filtering will take place. On the other hand, with very small initial conditions the filtering will strongly favor only the most unstable modes. Approximately speaking, the initial state image can be broken down into regions of different signal amplitude which will then undergo differing amounts of pre-filtering. One interpretation is that the part of the initial condition formed from the unstable spatial modes is normalized in each of the regions.

4.1.2 Morphological Constraint

The result of the linear pre-filtering can be arbitrary combinations of (mostly) unstable modes. However, once the saturation region is reached a spatial frequency/phase separation takes place to meet the new constraints. If the band of unstable modes is reasonably large, most of the cells will be saturated in steady state. In fact, if the average value of $\tilde{A}(\omega_1, \omega_2)$ is positive, which is equivalent to $A_{0,0} > 1$, all cells will be saturated. However, only certain neighborhoods of binary cells will be accepted, depending on the template.

The types of accepted global morphologies are difficult to determine and seems to depend on the form of the nonlinearity [15, 4]. It is easy to enumerate all possible stable steady state neighborhood configurations of $y(n_1, n_2) = \pm 1$ by using the necessary condition for stability at cell n_1, n_2,

$$\text{sgn}\,[a(n_1, n_2) * y(n_1, n_2)] = y(n_1, n_2) \tag{23}$$

For a particular cell, for simplicity choose (0,0), the left hand side can be written as a boolean function of the outputs in its neighborhood,

$$g[y(-1,-1), y(-1,0), y(-1,1), y(0,-1), y(0,0), y(0,1), y(1,-1), y(1,0), y(1,1)]$$

since this term is a threshold logic function. Then, the condition for the possibility of a stable neighborhood configuration can be written as the boolean equation

$$g \oplus y(0,0) = 0$$

which can be evaluated for every neighborhood configuration. However, the global morphology eludes us since these possible binary configurations must fit together in an overlapping manner.

In some cases, more can be said about the possible stable patterns[21]. For instance, the averaging template will support arbitrarily large black or white patches of constant output. However, the boundaries between these constant regions may or may not be stable. In general, it can be seen that increasing the center element will make the condition of Equation 23 easier to satisfy thereby allowing more possible neighborhood configurations and stable boundaries. (In the extreme case, the output is a thresholded version of the initial state.) Decreasing the center element too much will make the constraint difficult to meet causing the system to seek a long time for a globally stable output configuration, which may result in an output which is largely unrelated to the initial condition.

Example 6 – First Order Turing Patterns and Synergetics in CNN

The following template is an unstable band-pass filter. It is based on the same mask as Example 2 but has been made unstable by increasing the center element. The band of unstable modes is an origin-centered annulus with peak instability at radial frequency of $\pi/2$, which has a half-period of two pixels.

Unstable Band-Pass Template

$$A = \begin{bmatrix} -0.0625 & -0.25 & -0.375 & -0.25 & -0.0625 \\ -0.25 & 0.0 & 0.5 & 0.0 & -0.25 \\ -0.375 & 0.5 & 0.0 & 0.5 & -0.375 \\ -0.25 & 0.0 & 0.5 & 0.0 & -0.25 \\ -0.0625 & -0.25 & -0.375 & -0.25 & -0.0625 \end{bmatrix}$$

Figure 10 (Left) shows the time-sequence which follows from initial conditions perturbed around the unstable equilibrium $\forall i, j : x_{i,j} = 0$. These initial conditions are the same as those used in Example 2. This system is not driven externally, and the symmetry breaking which occurs is due to the spatial-frequency

selective instability. By $t = 2$ some cells are near saturation and the linear pattern shows a strong resemblance to that formed with the stable driven system of Figure 5. However, as the instability begins to drive cells into the saturation region, some more complicated behavior begins to take place. Arbitrary combinations of modes are no longer able to be maintained and a constraint on the local structure, or morphology, is imposed. In this example, even though blobs of size two pixels appear in the linear pattern, they are not accepted by the saturated system. By time $t = 40$ the blobs have been completely eliminated and only connected squiggles of width two remain.

The width two squiggles in random orientations can intuitively be considered to define a *class* of inherent patterns for the system. Under this framework, the output is the nonlinear projection (as defined by the dynamical system) of the input onto this class. In our example, the CNN found the best squiggles it could in the randomness of the initial condition.

It is interesting to see what happens when the initial conditions are biased with a clear feature found in the inherent pattern class. The time evolution of such an example is shown in Figure 10[(Right)]. The same random initial conditions have been modified so that a three adjacent width-two rows of cells have been set to $-0.1, +0.1, -0.1$ from top to bottom respectively. As time proceeds, it is seen that the feature grows quickly and has a strong effect on its relatively featureless neighbors. This is known as the synergistic effect [5] between the system and the choice of initial conditions. By the time steady state is reached, more than half of the array has aligned itself sympathetically with the feature in the initial condition.

■

Example 7 – Fingerprint Enhancement

To demonstrate the generalized averaging template idea we present an approach to fingerprint enhancement. The example has been used in other places as a potential application for reaction-diffusion type systems [4, 22]. Here we give some design methodologies which will hopefully help to pinpoint some important practical issues.

Figure 11 shows a fingerprint image which is in desperate need of enhancement. Many of the ridges have been interrupted and the valleys smudged with ink. The spatial frequency content of the image is shown to have most of its energy near the DC modes, which corresponds to uneven ink distribution, and in an annulus which corresponds to the typical ridge widths.

The goal is to extract the ridges and valleys in the image. To do this, we designed a generalized 'averaging' template

Fingerprint Averaging Template

$$A = \begin{bmatrix} -0.0205 & -0.1198 & -0.1546 & -0.1198 & -0.0205 \\ -0.1198 & 0.1542 & 0.2947 & 0.1542 & -0.1198 \\ -0.1546 & 0.2947 & A_{0,0} & 0.2947 & -0.1546 \\ -0.1198 & 0.1542 & 0.2947 & 0.1542 & -0.1198 \\ -0.0205 & -0.1198 & -0.1546 & -0.1198 & -0.0205 \end{bmatrix}$$

which peaks in the midst of the spatial frequencies of interest and therefore with the right choice of $A_{0,0}$ will have unstable modes there. The constant contours of gain for $\tilde{A}(\omega_1, \omega_2)$ when $A_{0,0} = 0.25$ are shown superimposed on the fingerprint DFT in Figure 11 showing good agreement. Since the template is similar to that of Example 6 we can expect that the final morphology will be that of squiggles, which is appropriate for the fingerprint.

Changing the center element will have the effect of narrowing or widening the annulus of unstable modes. This will have two effects: First, in the linear region it will change the relative strengths of filtering found in the pre-filtered image. This is also a function of the size of the initial conditions. Second, reducing the band of unstable modes changes the morphological constraint. Particularly, decreasing the band of unstable modes will decrease the number of accepted neighborhood configurations until perhaps nothing useful is accepted. Some other parameters which can also be controlled are the size of initial conditions, which determines the amount of pre-filtering, and the stopping time, which can be used to limit the effects of the morphologic constraint.

The fingerprint image was scaled to fit in $[-0.5, 0.5]$ and supplied to the initial conditions. Three different center elements were tried giving different width bands of unstable modes with the result shown in Figure 12. The most restricted band was with $A_{0,0} = 0$ The results at $t = 150$ show nicely shaped plausible ridges which,

unfortunately, have little to do with the underlying image. This is a demonstration of the problem of having too tight of a morphological constraint. Since the state is free to adjust itself to meet the morphological constraint, it does not need to obey any realities of the initial condition - too much synergy and not enough reality. The opposite problem is evident when $A_{0,0} = 1.1$ which has produced many broken lines and unaligned regions in the output which have not benefited from the synergy effect. Choosing the center element $A_{0,0} = 0.5$ shows a good balance between these examples. The results are more visually pleasing with less jagged lines when the system is not allowed to reach equilibrium, as shown.

■

4.2 Non-autonomous Behavior

The CNN with unstable central linear system which has input or bias follows the same two stages of filtering as the autonomous version: linear pre-filtering and a boolean morphological constraint. However, the interaction between the data supplied to the input or bias and the data supplied to the initial conditions makes overall intuitive understanding difficult.

When the initial state and input are somehow being used in conjunction, the linear pre-filtering can be extended to both the initial state and input. However, the actual pre-filtering done is a complex combination of the templates and all the data.

In saturation, similar reasoning to the previous section can be used to write a necessary condition for a particular cell to be stable in its neighborhood:

$$\text{sgn}\left[a(n_1, n_2) * y(n_1, n_2) + b(n_1, n_2) * u(n_1, n_2) + I\right] = y(n_1, n_2)$$

Now, the possible stable neighborhood configurations is informed by both the A template and the input. If the $b(n_1, n_2) * u(n_1, n_2)$ term is locally very large, the output is likely to follow it. In the extreme, zero initial conditions and an A template of radius zero with $A_{0,0} > 1$ gives a thresholded version of $b(n_1, n_2) * u(n_1, n_2)$. This is a widely used class of CNN templates known as the Linear Threshold Class [23]. On the other hand, when the input is small, then a morphologically constrained version of the pre-filtered state is more likely to appear. In regions where the input is moderate, some combination of these will be accepted in a way that contain some of the spectral favoring by the A template and some of the input. This is the mechanism used by the halftoning template, where noise in the initial condition is spectrally shaped and amplified to generate high-frequency patterns which are then biased by the input to appear at the right densities.

Example 8 – From Stripes to Spots

To understand the behavior of the non-autonomous system, we give an example of the simple case – the input is a simple spatially constant bias. By doing this, two prototypical patterns of pattern forming systems can be observed: stripes and spots. For demonstration purposes, the same template as Example 7 was used except for the application of a slowly spatially varying input which will demonstrate the behavior of a range of biases from $I = 0$ on the left to $I = 1$ on the right.

The output is shown in Figure 13. Three distinct regions can be observed. For biases $I < 0.25$ the 'fingerprint' stripes are formed, as above. However, roughly for $0.25 < I < 0.8$ 'leopard' spots are manifested reflecting a change in allowed morphologies. Note that the CNN dynamics could be re-written so that the bias term is absorbed into the output nonlinearity and therefore the allowed morphology could be understood in terms of the type of output nonlinearity. Finally, when $I > 0.8$ the bias dominates and only constant output can be supported.

■

Example 9 – Fingerprint Enhancement Revisited

The result of the previous example can be used to design a generalized halftoning template for fingerprint enhancement. Two problems from the 'averaging' template approached can be resolved. First, when there is a large smudge, we would like to replace it with a reasonable fingerprint structure based on the neighboring ridges. Second, the topologies of the ridges should be left intact even when they do not match the topological constraint.

By observation of figure 13 it is seen that if smudges could be made to supply zero input, then ridges would appear. These ridges would match adjacent ridges if the center element was small enough. Also, if the

very fine structure was made large enough, it would force the output to locally follow the input independent of the neighborhood configuration. Both of these can be achieved by supply a high-pass filtered version of the fingerprint to the input. As an example, the following template pair was chosen.

Fingerprint Enhancement Template

$$A = \begin{bmatrix} -0.0205 & -0.1198 & -0.1546 & -0.1198 & -0.0205 \\ -0.1198 & 0.1542 & 0.2947 & 0.1542 & -0.1198 \\ -0.1546 & 0.2947 & 0.0 & 0.2947 & -0.1546 \\ -0.1198 & 0.1542 & 0.2947 & 0.1542 & -0.1198 \\ -0.0205 & -0.1198 & -0.1546 & -0.1198 & -0.0205 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.25 & -0.5 & -0.25 \\ -0.5 & 3.0 & -0.5 \\ -0.25 & -0.5 & -0.25 \end{bmatrix}$$

The small $A_{0,0}$ term will ensure that the synergy is high when it is needed. The resulting output can be seen in Figure 14.

■

5 Conclusion

We have given many examples and design procedures which take advantage of a modal representation of the linearized CNN dynamics. Some classic results from the field of pattern formation have been demonstrated in the simple CNN with first order cells. In addition, some approaches to designing CNN templates for both linear and nonlinear spatial filtering were given – which we hope will prove useful for the design of processing steps in image processing algorithms for the CNN Universal Machine.

References

[1] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, vol. 32, Oct. 1988.

[2] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Transactions on Circuits and Systems - I*, vol. 40, pp. 147–156, Mar. 1993.

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

[4] C. B. Price, *Computing With Active Media: Pattern Formation and Analysis by Reaction-Diffusion Systems, Neural Fields and Coupled Map Lattices*. PhD thesis, Katholieke Universiteit Leuven, June 1993.

[5] H. Haken, "Synergetics: From pattern formation to pattern analysis and pattern recognition," *International Journal of Bifurcation and Chaos*, vol. 4, no. 5, pp. 1069–1083, 1994.

[6] K. Preston Jr. and M. J. B. Duff, *Modern Cellular Automata: Theory and Applications*. New York: Plenum, 1984.

[7] T. Toffoli, *Cellular Automata Machines*. Cambridge, MA: The MIT Press, 1987.

[8] T. Roska and L. O. Chua, "The CNN Universal Machine: An analogic array computer," *IEEE Transactions on Circuits and Systems - II*, vol. 40, pp. 163–173, Mar. 1993.

[9] L. O. Chua and L. Yang, "Cellular Neural Networks: Applications," *IEEE Transactions on Circuits and Systems*, vol. 32, Oct. 1988.

[10] K. R. Crounse, T. Roska, and L. O. Chua, "Image halftoning with Cellular Neural Networks," *IEEE Transactions on Circuits and Systems - II*, vol. 40, pp. 267–283, Apr. 1993.

[11] A. M. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society of London*, vol. 237 B, pp. 37–72, 1952.

[12] D. G. Kelly, "Stability in contractive nonlinear neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 37, pp. 231–241, Mar. 1990.

[13] J. L. White and A. A. Abidi, "Active resistor networks as 2-D sampled data filters," *IEEE Transactions on Circuits and Systems - I*, vol. 39, pp. 724–733, Sept. 1992.

[14] B. E. Shi and L. O. Chua, "Resistive grid image filtering: Input/output analysis via the CNN framework," *IEEE Transactions on Circuits and Systems - I*, vol. 39, pp. 531–548, July 1992.

[15] L. Goras and L. O. Chua, "Turing patterns in CNNs: Part 2," *IEEE Transactions on Circuits and Systems - II*, Oct. 1995. To appear.

[16] P. J. Davis, *Circulant Matrices*. New York, NY: John Wiley and Sons, 1979.

[17] B. Shi, T. Roska, and L. O. Chua, "Design of linear Cellular Neural Networks for motion sensitive filtering," *IEEE Transactions on Circuits and Systems - II*, vol. 40, pp. 320–331, May 1993.

[18] J. P. Miller, T. Roska, T. Szirányi, K. R. Crounse, L. O. Chua, and L. Nemes, "Deblurring of images by Cellular Neural Networks with applications to microscopy," in *Proceedings of the Third IEEE International Workshop on Cellular Neural Networks and Their Applications*, pp. 237 - 242, Dec. 1994.

[19] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: P T R Prentice Hall, 1990.

[20] B. E. Shi, *Spatio-temporal Image Filtering with Cellular Neural Networks*. PhD thesis, University of California at Berkeley, Mar. 1994.

[21] P. Thiran, K. R. Crounse, L. O. Chua, and M. Hasler, "Pattern formation properties of autonomous Cellular Neural Networks," *IEEE Transactions on Circuits and Systems - II*, Oct. 1995. To appear.

[22] T. Roska, L. O. Chua, D. Wolff, T. Kozek, R. Tetzlaff, and F. Puffer, "Simulating nonlinear waves and partial differential equations via CNN: Part ii," *IEEE Transactions on Circuits and Systems - II*, Oct. 1995. To appear.

[23] L. O. Chua and B. E. Shi, "Exploiting Cellular automata in the design of Cellular Neural Networks for binary image processing," Memorandum UCB/ERL M89/130, University of California at Berkeley Electronics Research Laboratory, Nov. 1989.

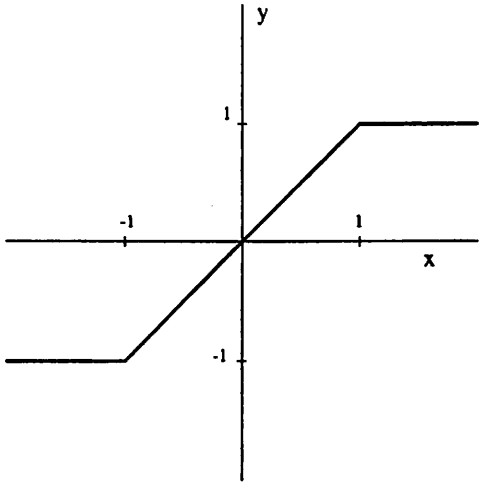


Figure 1: The CNN Output Nonlinearity.

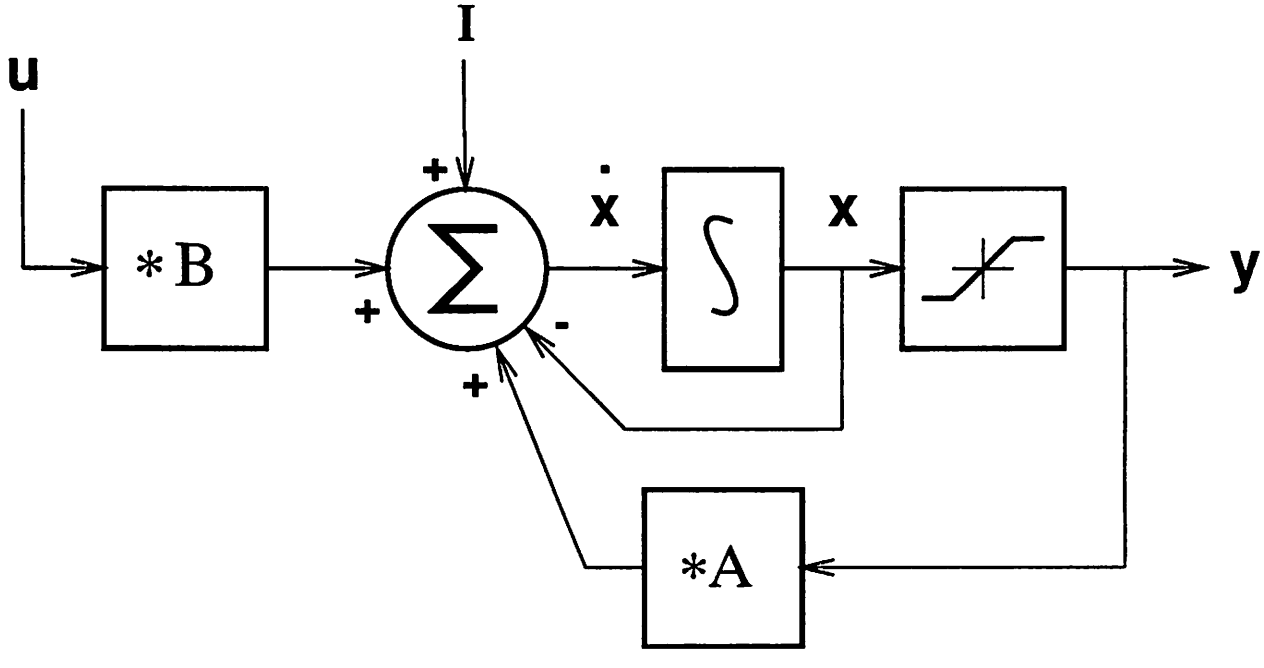
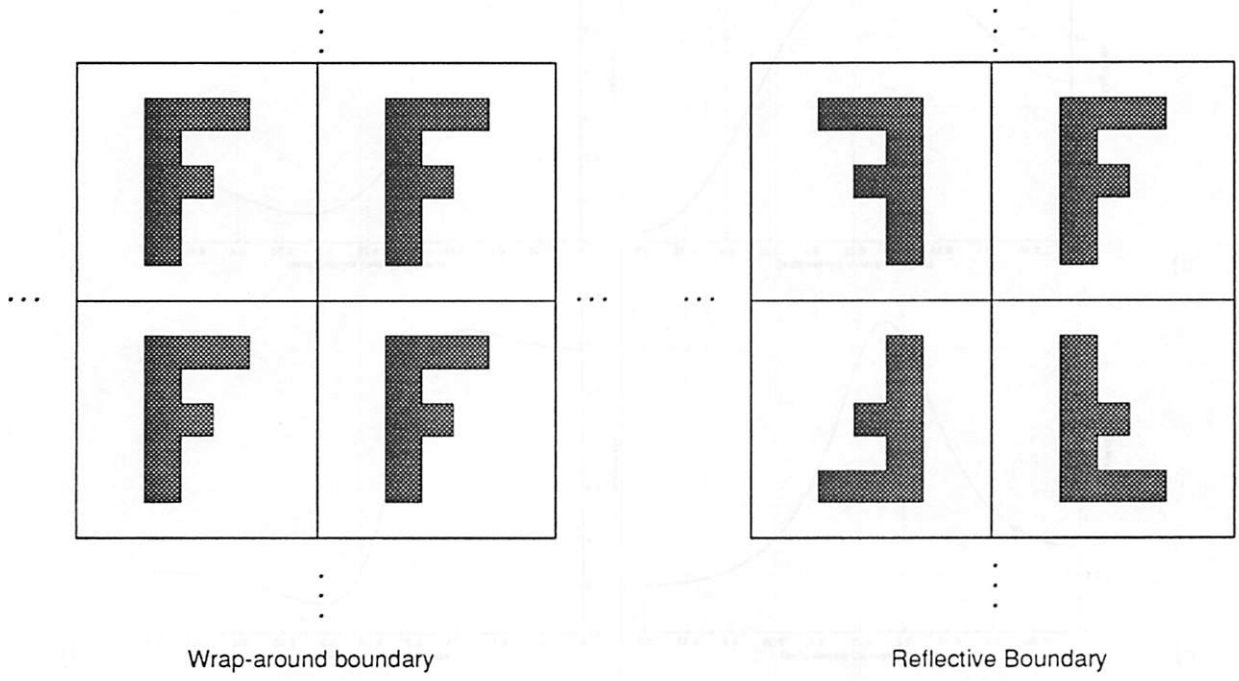


Figure 2: A Block Diagram Representation of the Standard CNN.



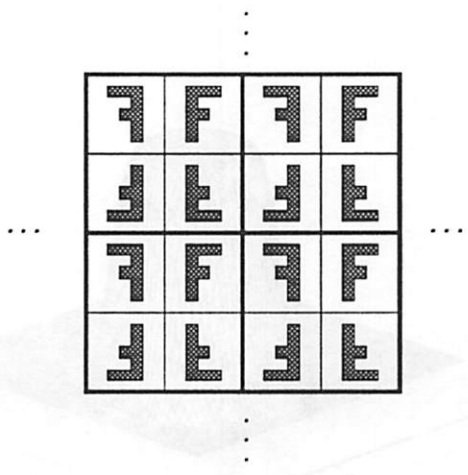
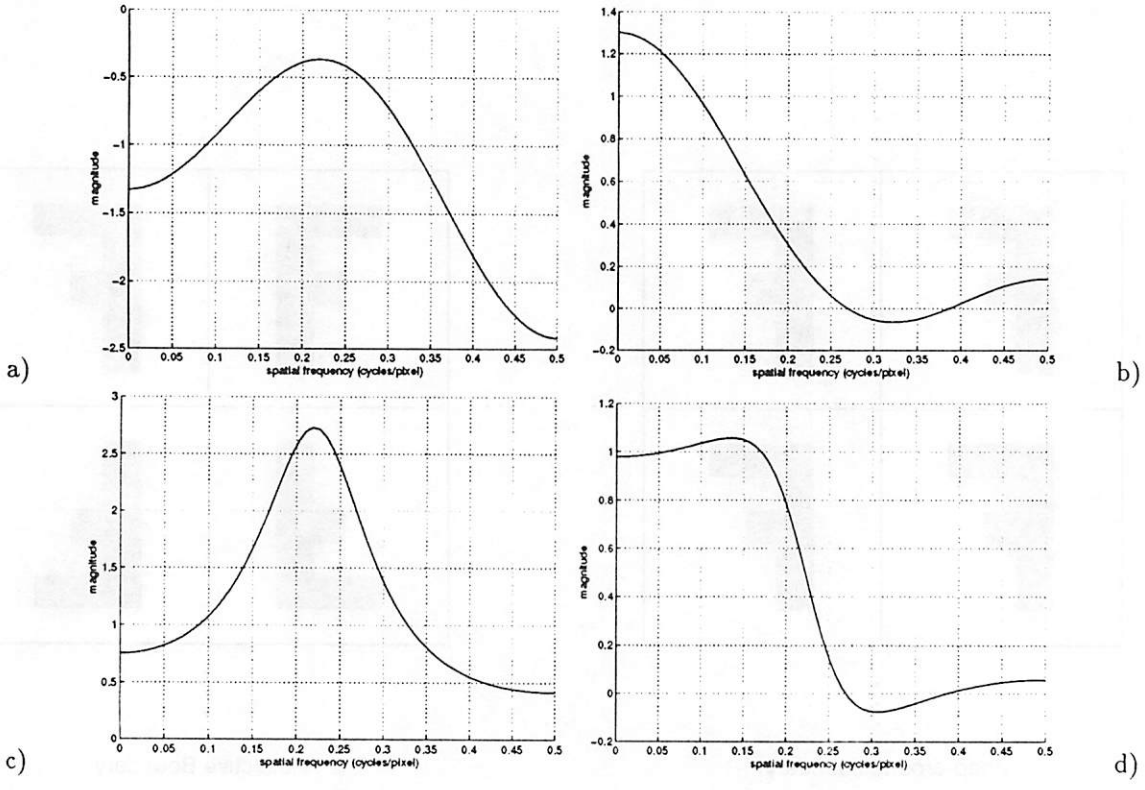


Figure 3: Two Methods to Handle the Boundary. Wrap-around or toroidal boundary conditions and reflective boundary conditions are equivalent to an infinite periodic system as shown (Top). In fact, a reflective system is equivalent to a toroidal one of four times the size (Bottom).



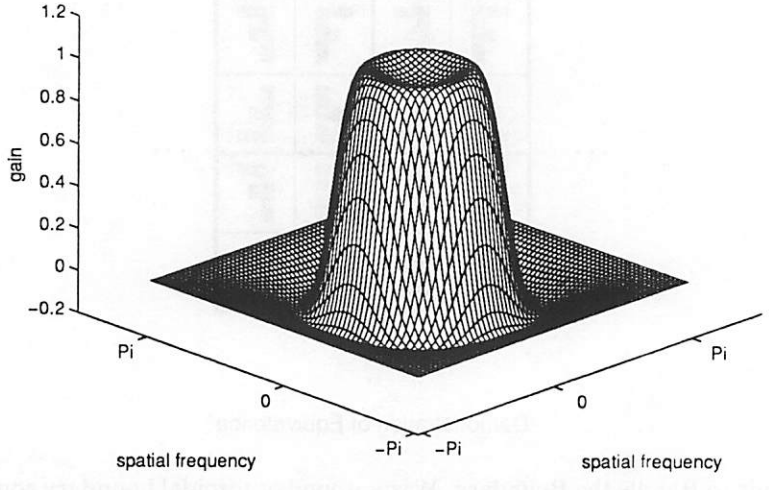


Figure 4: CNN Low Pass Filter Design. The system eigenvalues, which are the $\tilde{A}(\omega_1, \omega_2)$ are shown in a) along one frequency radius. The filtering action of the B template, $\tilde{B}(\omega_1, \omega_2)$ is shown in b) to be a smooth roll-off low-pass filter. The filtering action of the A template, $\frac{-1}{\tilde{A}(\omega_1, \omega_2)}$, is shown in c) which gets multiplied by graph b) to produce the system transfer function shown in d). The filter characteristic is shown (lower) to have good circular symmetry and a reasonably steep cutoff and little ripple.

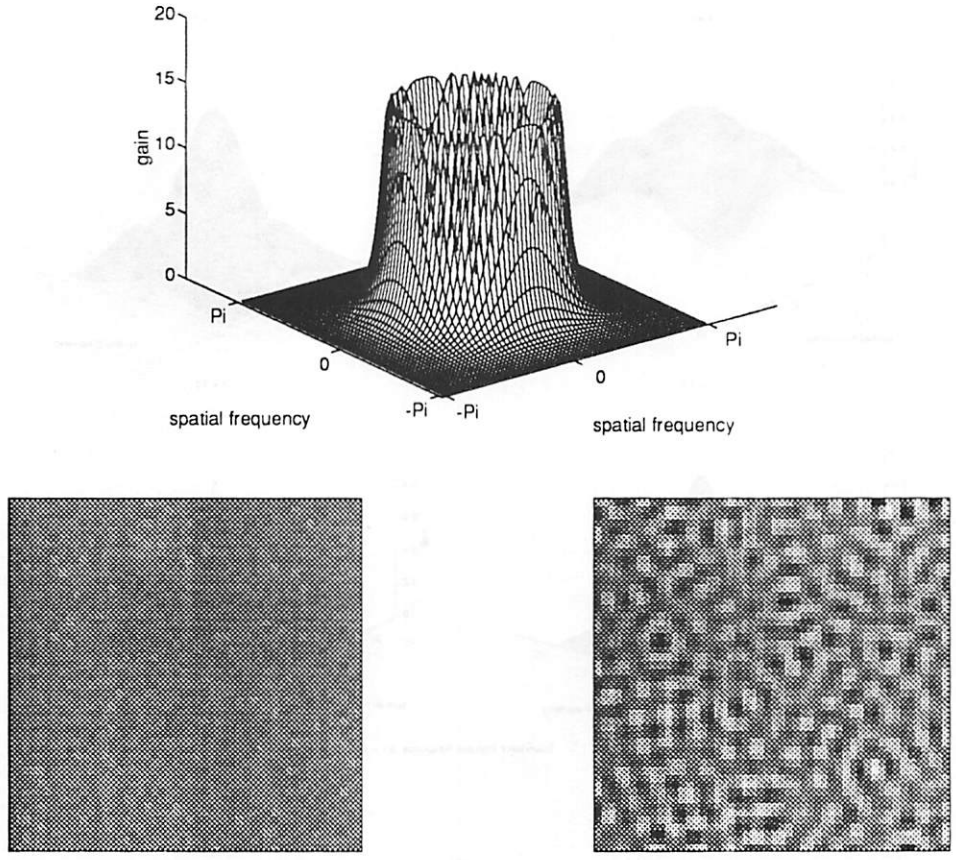


Figure 5: Pattern Formation in a Stable Linear CNN. The steady state transfer function $\tilde{H}(\omega_1, \omega_2)$ (Top) shows an aggressive band pass characteristic. When the input (Left) is set to independent random values on $[-0.1, 0.1]$ the state output, shown at $t = 200$ (Right), converges to a apparent texture.

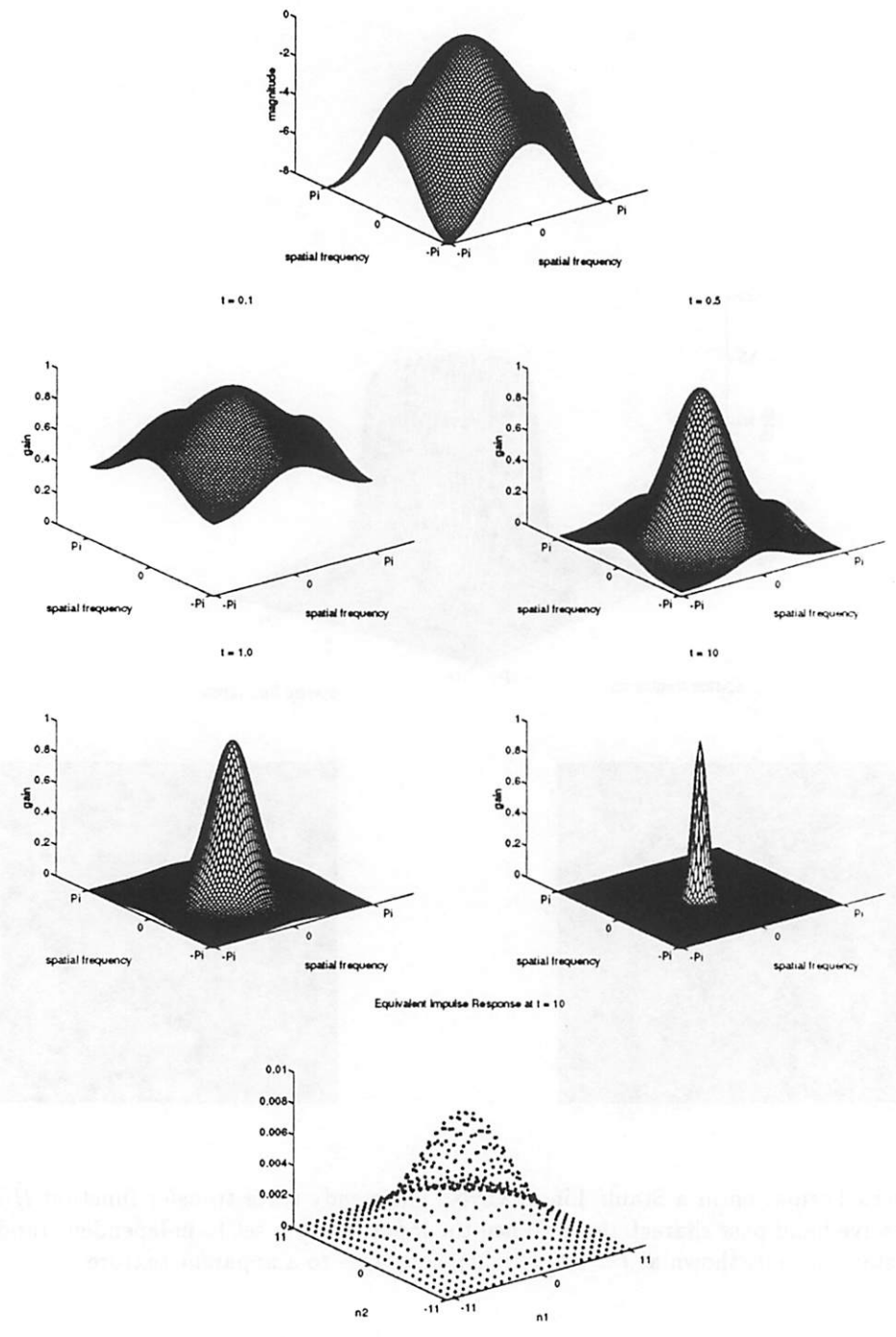


Figure 6: 'Laplace' Template as Time-Varying Filter. The eigenvalues $\tilde{A}(\omega_1, \omega_2)$ (Top) are all negative except for the one corresponding to the average value. The equivalent spatial frequency filtering operation is shown (Middle) for various times becoming increasingly narrow. The equivalent impulse response at $t = 10$ (Bottom) is quite broad.

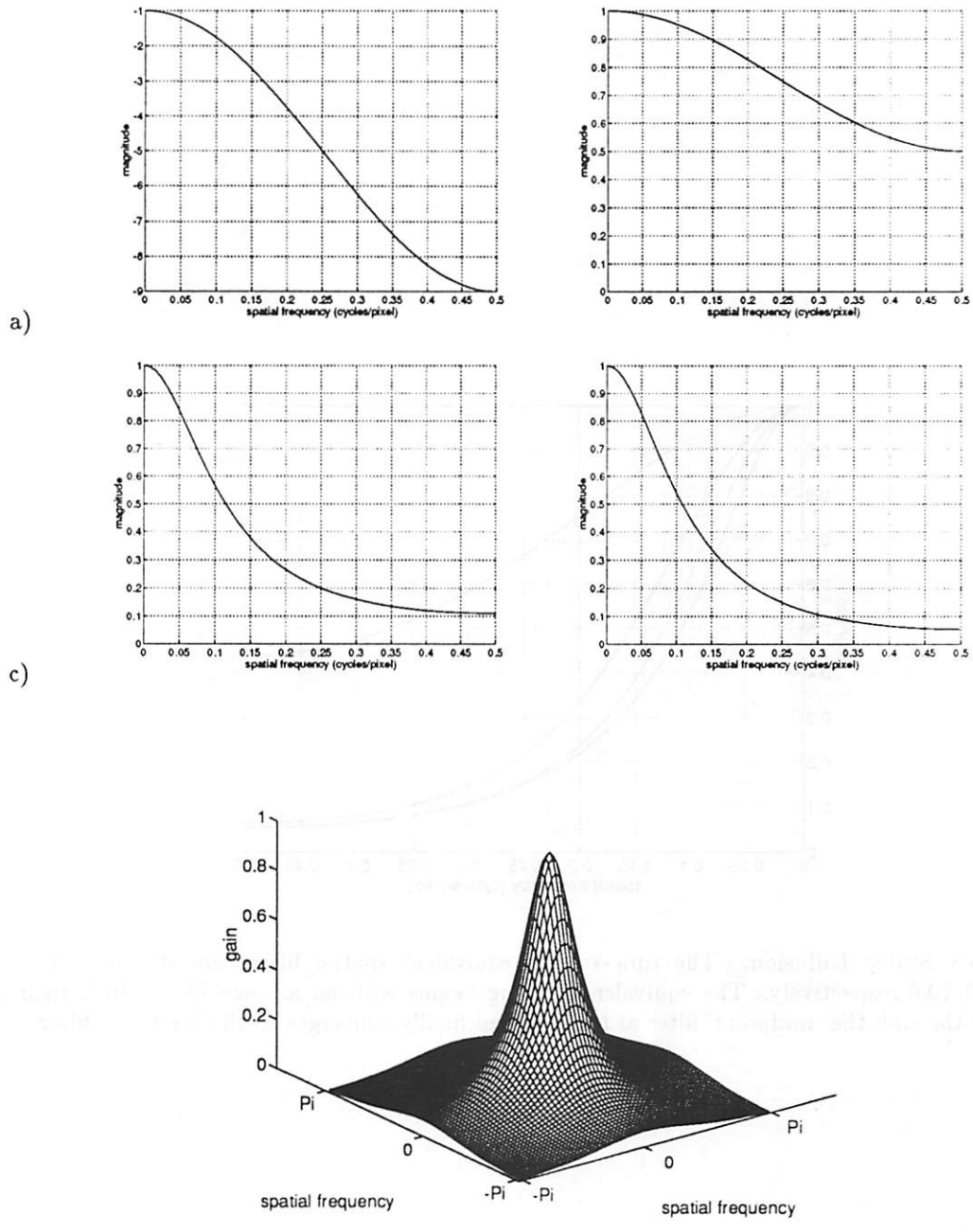


Figure 7: CNN Extreme Filter Design. The system eigenvalues, $\tilde{A}(\omega_1, \omega_2)$, which are scaled and shifted version of the prototype filter, are shown in a) along the frequency radius $\omega_2 = 0$. The filtering action of the B template, $\tilde{B}(\omega_1, \omega_2)$, in b) is equal to that of the prototype filter. The filtering action of the A template, $\frac{-1}{\tilde{A}(\omega_1, \omega_2)}$, is shown in c) which gets multiplied by graph b) to produce the steady state transfer function shown in d). The filter characteristic is shown (lower) to be an extreme version of the prototype filter.

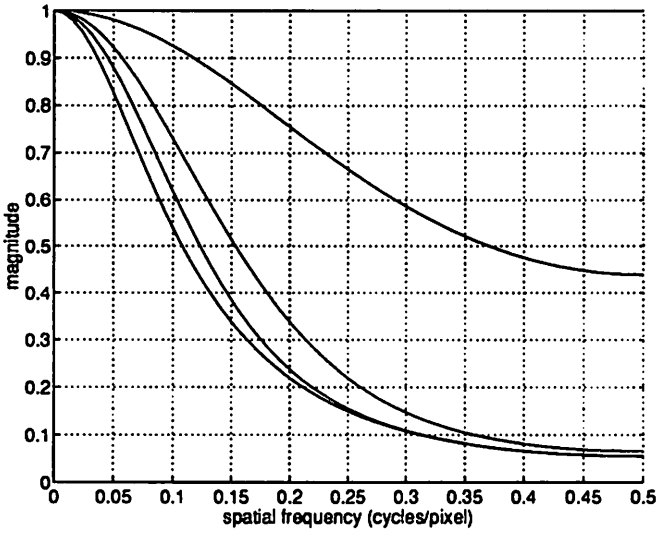


Figure 8: CNN Stable Diffusion. The time-varying equivalent spatial filters are shown at times $t = 0.0, 0.1, 0.5, 1.0, 10.0$ respectively. The equivalent filtering begins with an all-pass filter which then passes approximately through the 'midpoint' filter at $t = 0.33$ and finally converges to the 'extreme' filter.

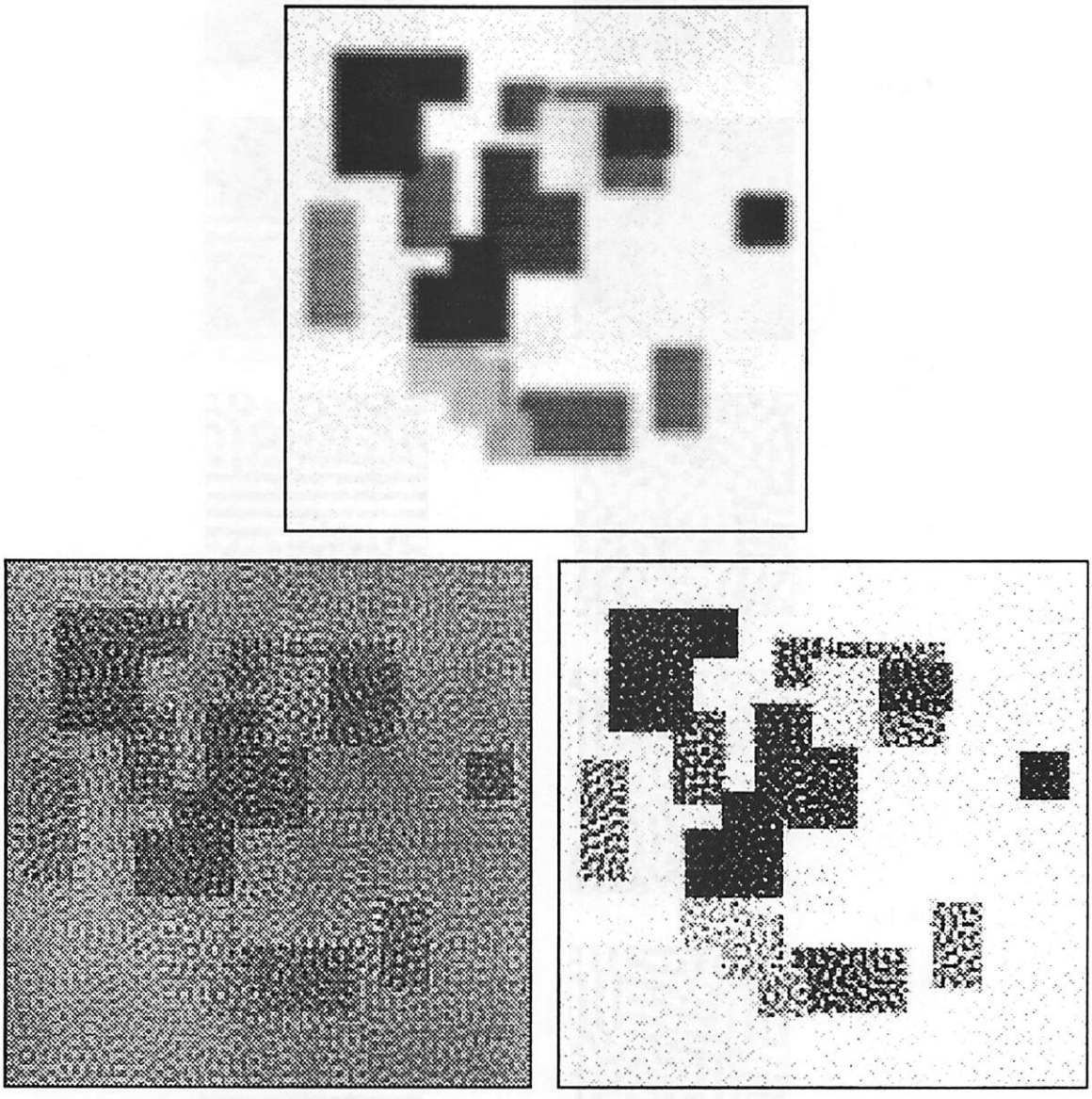


Figure 9: Constrained Inverse Filtering. A blurred and noisy version of the source image (Top) has been directly inverse filtered by the CNN at equilibrium resulting in a noise-dominated image (Left). When the CNN saturation region is used, the result is constrained to reasonable values and the noise is reduced (Right).

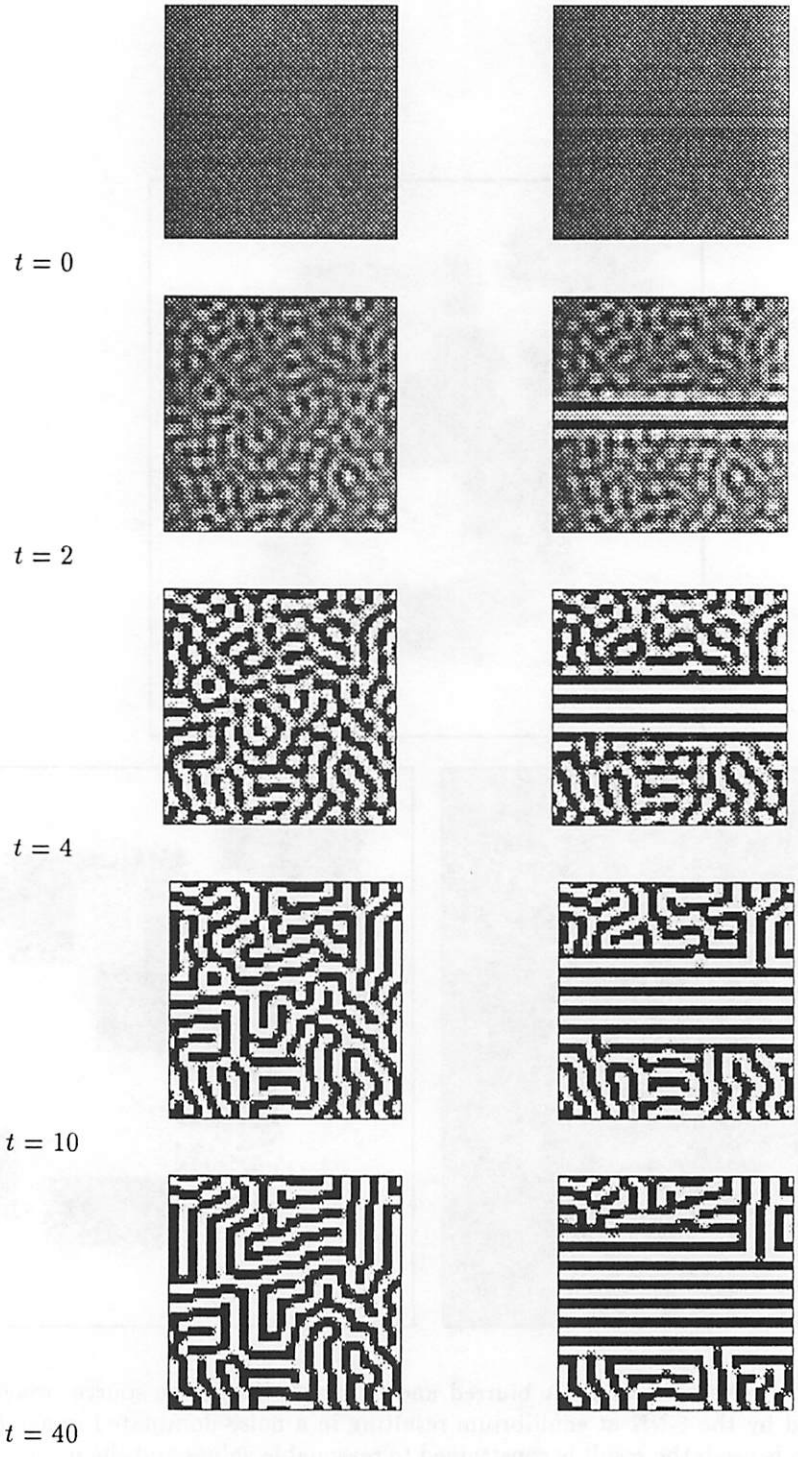


Figure 10: Pattern Formation and Synergetics in the CNN. Started from random initial conditions, the unstable autonomous system produces a typical inherent pattern by means of linear pre-filtering and morphological constraints (Left). When the initial condition is biased by a feature favored by the CNN system, a synergy is demonstrated producing an output aligned to the feature (Right).

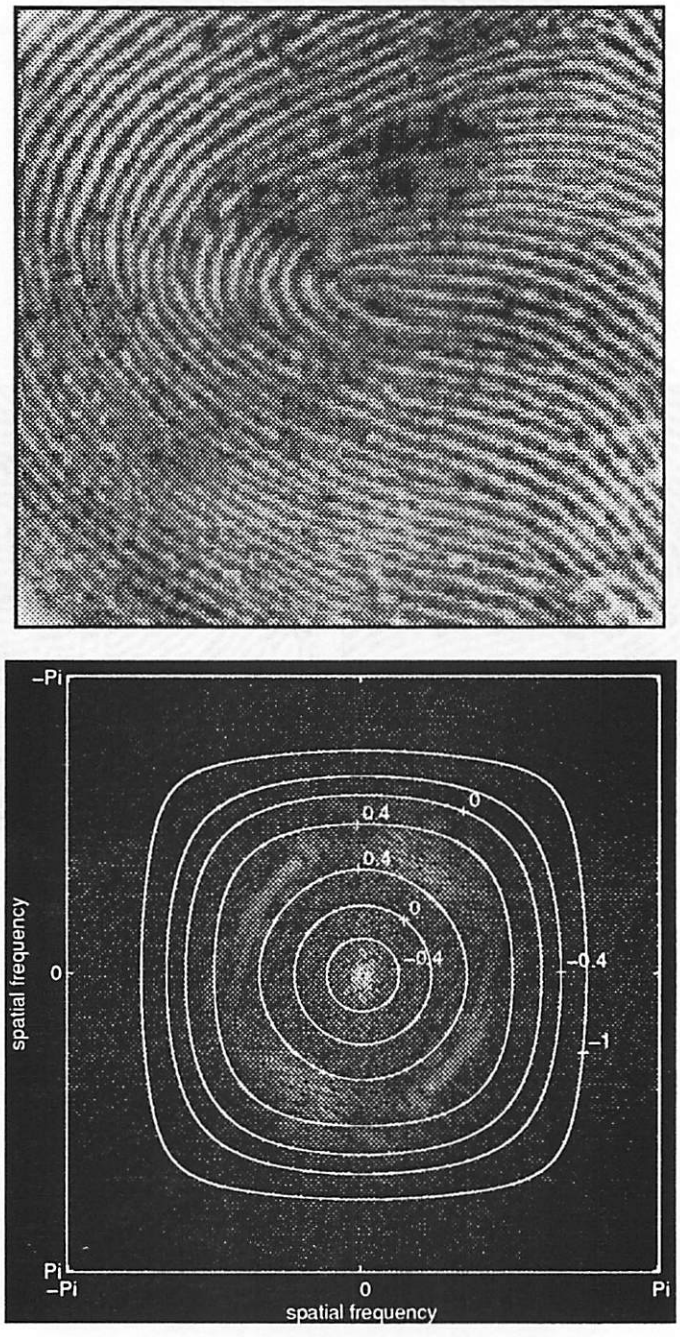


Figure 11: Fingerprint 'Averaging' Template Design. The degraded fingerprint (Top) image (courtesy of K. Lotz) used as the input data to the initial state. The fingerprint DFT (Bottom: gray scale image) was used to design the frequency domain characteristics $\tilde{A}(\omega_1, \omega_2)$ of the linear template mask. (Bottom: contours)

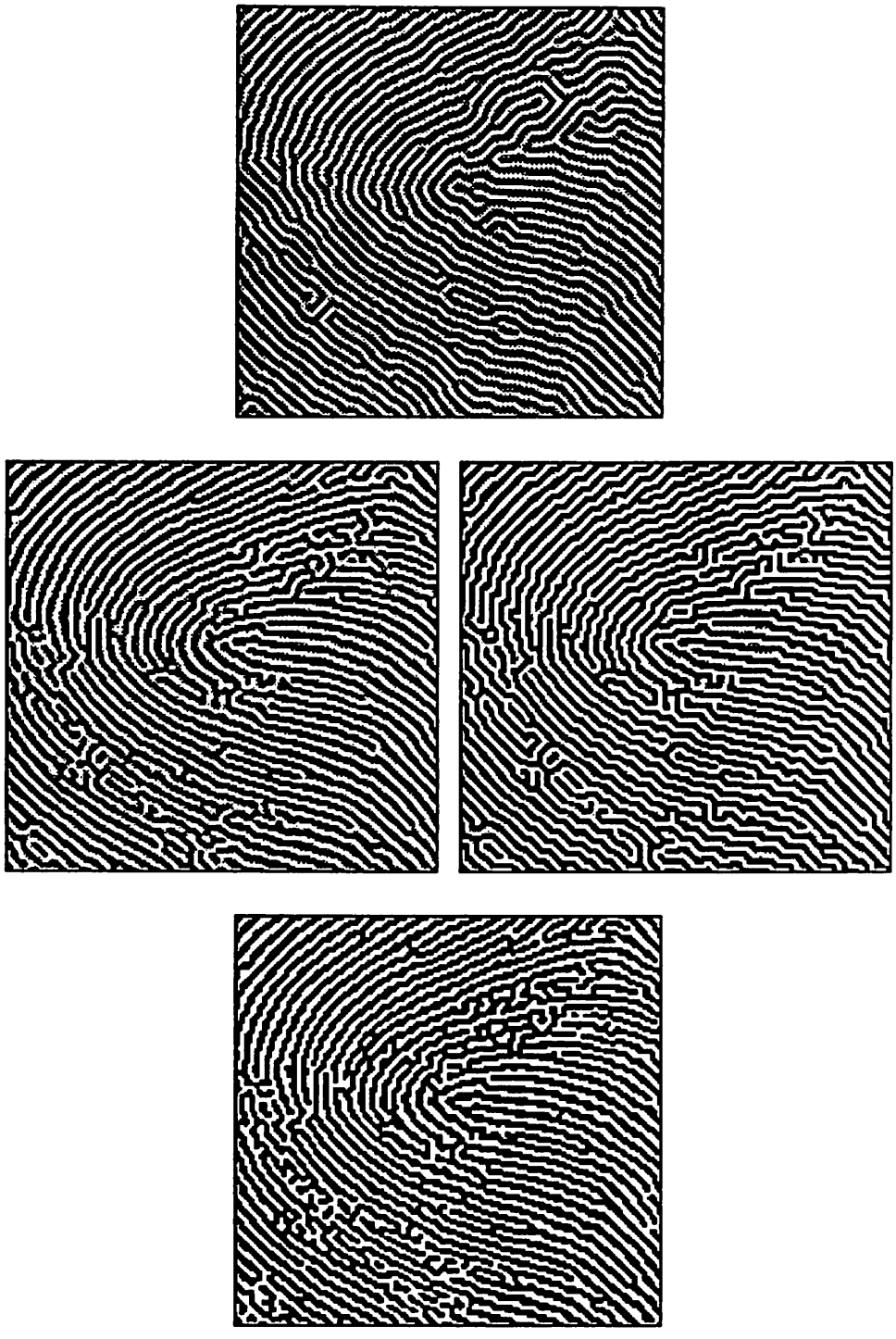


Figure 12: Fingerprint 'Averaging' Template Results. Results are shown for three different values of the center element, which controls the width of the unstable band. The tradeoff between synergy and reality in the output can be seen for the three values of $A_{0,0} = 0.0, 0.5, 1.1$ from Top to Bottom respectively. The system with $A_{0,0} = 0.5$ was stopped at $t = 11$ (Left) and $t = 150$ (Right) showing the advantage of an early stopping time. The other two were stopped at $t = 150$.

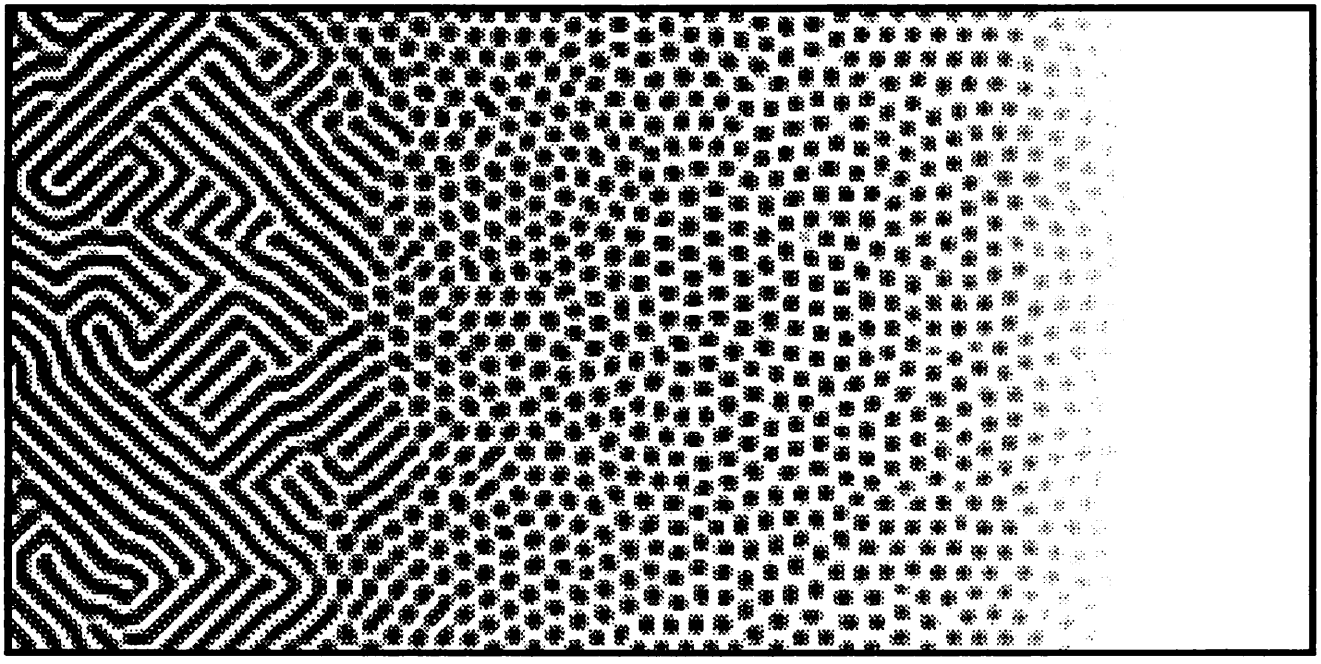


Figure 13: Stripes and Spots in the simple biased CNN. The figure is the steady state output of the system when the input was small and random. The input is a constant bias which slowly varies from 0 on the left and 1 on the right.

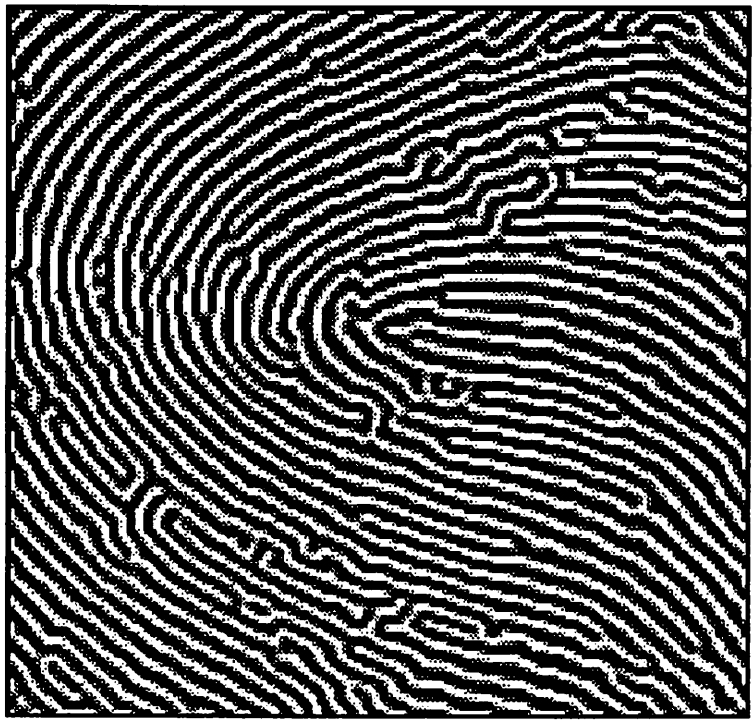


Figure 14: Fingerprint Enhanced by CNN Generalized Halftoning. A high-passed version of the fingerprint was supplied to the input as a reality constraint on the dynamics. The output shows good preservation of ridge topology and, except in the most difficult places, a plausible reconstruction.