IMAGE PROCESSING AND SELF-ORGANIZING CNN

MAKOTO ITOH
Department of Information and Communication Engineering,
Fukuoka Institute of Technology,
Fukuoka 811-0295, Japan

LEON O. CHUA
Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley,
Berkeley, CA 94720, USA

Received March 4, 2005; Revised April 15, 2005

CNN templates for image processing and pattern formation are derived from neural field equations, advection equations and reaction-diffusion equations by discretizing spatial integrals and derivatives. Many useful CNN templates are derived by this approach. Furthermore, self-organization is investigated from the viewpoint of divergence of vector fields.

Keywords: CNN; neural field equation; self-organization; divergence; complexity; numerical integration; numeral differentiation.

1. Introduction

CNN is a dynamic nonlinear system defined by coupling only identical simple dynamical systems called cells located within a prescribed sphere of influence, such as nearest neighbors [Chua, 1998]. Because of its simplicity, and ease for chip (hardware) implementation, the CNN has found numerous applications in image and video signal processing, in robotic and biological visions, and in higher brain functions. CNN has parameters, which must be derived analytically or numerically from given functions. Recently, a method for designing these CNN parameters was proposed [Itoh & Chua, 2003]. In this design, one has to choose a candidate from a set of basic CNN templates, and tune its parameters to achieve the desired functions. However, a fundamental question has not yet been addressed; namely, how are these basic CNN templates obtained or derived?

Recently, research on Emergence, Edge of Chaos, and Complexity has provided new insights for addressing the above fundamental problem. The terminology “edge of chaos” sometimes refers to the idea that many complex systems seem to evolve towards a regime between order and chaos. If the behavior of a system is too ordered, there is not enough variability or novelty to carry out interesting and useful functions. On the other hand, if the behavior of a system is too disordered, there is too much noise to sustain robust functions. This is why the dynamical memories of Star CNNs are designed to operate in the parameter domain of “edge of chaos”. The output pattern of dynamic memories wanders randomly from one stored pattern to another stored pattern, one after another. Furthermore, average of the a flash of new pattern (spurious pattern) sometimes emerges which suggests new applications. The change of patterns is not periodic, but chaotic [Itoh & Chua, 2004]. In Reaction-Diffusion CNNs, the terminology “edge of chaos” refers to the fact that when the cell parameters are chosen within some subset of the locally active (not passive) parameter domain where the equilibrium state of the decoupled cell is stable, the
probability of the emergence of complex nonhomogeneous static as well as dynamic patterns is greatly enhanced over a wide range of coupling parameters [Dogaru & Chua, 1998]. The appearance of this higher-level collective properties and behaviors of a system is called Emergence.

Emergence is the process of germinating new and coherent structures, patterns and properties in a complex system. Emergent phenomena occur due to the local interactions between the elements of a system. Emergent phenomena are often unexpected, nontrivial results of interactions of relatively simple components. Self-organization refers to a process in which the internal organization of a system increases automatically without being guided or managed by an outside source. Self-organizing systems typically display emergent properties.

The divergence of a vector field is a measure of the expansion rate of dynamic flows. It is positive when the flow is expanding, and negative when the flow is compressing. In physical terms, the divergence of a vector field measures the extent to which a vector field flow behaves like a source, or a sink, at a given point. Furthermore, the average of the divergence of vector fields is equal to the sum of Lyapunov exponents which are a measure of chaos. Thus, the divergence of the flow is closely related to the phase transition of physical systems, an example, par excellence, of self-organization. However, self-organization has not been fully investigated from the viewpoint of the divergence of vector fields.

In this paper, we derive basic CNN templates for image processing and pattern formation from neural field equations, advection equations, and reaction–diffusion equations by discretizing spatial integrals and derivatives. For this purpose, numerical integration techniques are used to calculate spatial integrals (Riemann sums, trapezoidal rule, Simpson’s rule), and finite difference approximation are used to calculate spatial derivatives (forward difference, backward difference, central difference). We will show many useful templates emerge from this approach. Finally, we investigate the divergence property of vector fields of discretized CNN cells and explore their emergence of self-organization phenomena.

2. Neural Field Equations and CNN

The neural field is a mathematical model of the cortex. The neural field equation represents highly dense cortical neurons as a spatially continuous field. The dynamics of one-dimensional neural field equations given by

\[
\frac{\partial u(x,t)}{\partial t} = -u(x,t) + \int a(x,x')f(u(x',t))dx' + \int b(x,x')s(x')dx' + z(x),
\]

where \(u(x,t)\) is the average membrane potential of neuron at position \(x\) at time \(t\), \(a(x,x')\) and \(b(x,x')\) are the connectivity functions which represent the average intensity of connections from neurons at place \(x'\) to neurons at place \(x\), \(f(u)\) is the output function which determines the firing rate of the neuron as a function of its membrane potential, \(s(x')\) is the input stimulus externally applied to neurons at position \(x'\), and \(z(x)\) is the threshold [Amari, 1978; Kubota & Aihara, 2004].

The discretized version of Eq. (1) can be obtained by using the method of [Kubota & Aihara, 2004]. Divide the domain of field \([x_{\text{min}}, x_{\text{max}}]\) into \(N\) intervals \([x_{i-1}, x_i]\) \((i = 1, 2, \ldots, N)\) with length \(\Delta x\), where

\[
x_i = x_{\text{min}} + i\Delta x \quad (i = 1, 2, \ldots, N)
\]

and

\[
\Delta x = \frac{x_{\text{max}} - x_{\text{min}}}{N}.
\]

Let \(u_i\) be the average membrane potential of neurons in the interval \([x_{i-1}, x_i]\), \(\overline{a}_{ij}\) and \(\overline{b}_{ij}\) be the average intensity of connections from neurons in the interval \([x_{j-1}, x_j]\) to neurons in the interval \([x_{i-1}, x_i]\), \(s_i\) be the input stimulus externally applied to neurons in the interval \([x_{j-1}, x_j]\). If we substitute the approximations

\[
\begin{align*}
  u(x) &\rightarrow u_i, \quad u(x') \rightarrow u_j, \quad f(u_j) \rightarrow p_j, \\
  a(x, x') &\rightarrow \overline{a}_{ij}, \quad b(x, x') \rightarrow \overline{b}_{ij}, \\
  s(x') &\rightarrow s_j, \quad z(x) \rightarrow z_i,
\end{align*}
\]

for each term in Eq. (1) and replace the integral by a summation, we would obtain a one-dimensional fully-connected CNN equation [Chua, 1998]

\[
\frac{du_i}{dt} = -u_i + \sum_{j=1}^{N} \left(\overline{a}_{ij}p_j + \overline{b}_{ij}s_j\right)\Delta x + z_i
\]

\[
= -u_i + \sum_{j=1}^{N} \left(a_{ij}p_j + b_{ij}s_j\right) + z_i,
\]

where \(b_{ij} = \overline{b}_{ij}\Delta x\), \(a_{ij} = \overline{a}_{ij}\Delta x\). If each cell (neurons in the interval \([x_{i-1}, x_i]\)) \(C_i\) is coupled locally only to neighbor cells inside some neighborhood.
(sphere of influence) \( N_i \), then Eq. (5) can be written as

\[
\frac{du_i}{dt} = -u_i + \sum_{j \in N_i} \left(a_{ij}p_j + b_{ij}s_j\right) + z_i,
\]

(6)

where \( N_i \) denotes the \( r \)-neighborhood of cell \( C_i \).

Consider next the two-dimensional neural field equation

\[
\frac{\partial u(w,t)}{\partial t} = -u(w,t) + \int \int a(w,w') f(u(w',t)) \, dx' \, dy' + \int b(w,w') s(w') \, dx' \, dy' + z(w),
\]

(7)

where \( w = (x,y) \in \mathbb{R}^2 \) and \( w' = (x',y') \in \mathbb{R}^2 \). In this case, the domain \([y_{\min}, y_{\max}]\) is also divided into \( M \) intervals \([y_{i-1}, y_i]\) \( (i = 1, 2, \ldots, M) \) with length \( \Delta y \). Thus, the domain \([x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]\) is divided into \( MN \) subdomains. Let \( u_{ij} \) be the average of \( u(x,t) \) in the region \([x_{i-1}, x_i] \times [y_{j-1}, y_j]\). Then, the following standard CNN equation [Chua, 1998] can be obtained from Eq. (7)

\[
\frac{du_{ij}}{dt} = -u_{ij} + \sum_{k,l \in N_{ij}} \left(a_{kl}p_{kl} + b_{kl}s_{kl}\right) + z_{ij},
\]

(8)

\((i,j) \in \{1, \ldots, M\} \times \{1, \ldots, N\}\)

where \( u_{ij} \) denotes the state of cell \( C_{ij} \), \( p_{kl} \) and \( s_{kl} \) denote the output and input of cell \( C_{kl} \), \( N_{ij} \) denotes the \( r \)-neighborhood of cell \( C_{ij} \), and \( a_{kl}, b_{kl}, \) and \( z_{ij} \) denote the feedback, control and threshold template parameters, respectively. The matrices \( A = [a_{kl}] \) and \( B = [b_{kl}] \) are referred to as the feedback template \( A \) and the feedforward (input) template \( B \), respectively [Chua & Roska, 2002]. The output \( p_{ij} \) and the state \( u_{ij} \) of each cell are usually related through the piecewise-linear saturation function

\[
p_{ij} = f(u_{ij}) = \frac{1}{2} \left( |u_{ij} + 1| - |u_{ij} - 1| \right).
\]

(9)

If we restrict the neighborhood radius of every cell to \( r = 1 \) and assume that \( z_{ij} \) is the same for the whole network, then the template \( \{A, B, z\} \) is fully specified by 19 parameters, which are the elements of the \( 3 \times 3 \) matrices \( A \) and \( B \), namely

\[
A = \begin{bmatrix}
a_{-1,-1} & a_{-1,0} & a_{-1,1} \\
a_{0,-1} & a_{0,0} & a_{0,1} \\
a_{1,-1} & a_{1,0} & a_{1,1}
\end{bmatrix}, \quad B = \begin{bmatrix}
b_{-1,-1} & b_{-1,0} & b_{-1,1} \\
b_{0,-1} & b_{0,0} & b_{0,1} \\
b_{1,-1} & b_{1,0} & b_{1,1}
\end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix}.
\]

(10)

and the threshold \( z \). Thus, the CNN is characterized by the following templates.

**Two-dimensional CNN template:**

\[
A = \begin{bmatrix}
a_{-1,-1} & a_{-1,0} & a_{-1,1} \\
a_{0,-1} & a_{0,0} & a_{0,1} \\
a_{1,-1} & a_{1,0} & a_{1,1}
\end{bmatrix}, \quad B = \begin{bmatrix}
b_{-1,-1} & b_{-1,0} & b_{-1,1} \\
b_{0,-1} & b_{0,0} & b_{0,1} \\
b_{1,-1} & b_{1,0} & b_{1,1}
\end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix}.
\]

(11)

**One-dimensional CNN template:**

\[
A = \begin{bmatrix} a_{0,-1} & a_{0,0} & a_{0,1} \end{bmatrix}, \quad B = \begin{bmatrix} b_{0,-1} & b_{0,0} & b_{0,1} \end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix}.
\]

(12)

3. Numerical Integration and Differentiation Formulas

In order to discretize the neural field equations, advection equations, and reaction–diffusion equations, we apply the numerical integration and differentiation algorithms to their spatial integrals and derivatives, respectively. In this section, we recall some well-known numerical integration and numerical differentiation algorithms.

3.1. Numerical integration formulas

The formulas for the Riemann sum, the trapezoidal rule, and the Simpson's rule are described
as follows:

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Riemann sum</td>
<td>[ \int_a^b k(x)dx \approx \left( k(c - h) + k(c) \right)h, ]</td>
</tr>
<tr>
<td></td>
<td>where ( c = \frac{a + b}{2}, h = \frac{b - a}{2}, a = c - h. )</td>
</tr>
<tr>
<td>Right Riemann sum</td>
<td>[ \int_a^b k(x)dx \approx \left( k(c) + k(c + h) \right)h, ]</td>
</tr>
<tr>
<td></td>
<td>where ( c = \frac{a + b}{2}, h = \frac{b - a}{2}, b = c + h. )</td>
</tr>
<tr>
<td>Simpson’s rule</td>
<td>[ \int_a^b k(x)dx \approx \frac{h}{3} \left( k(c - h) + 4k(c) + k(c + h) \right), ]</td>
</tr>
<tr>
<td></td>
<td>where ( c = \frac{a + b}{2}, h = \frac{b - a}{2}. )</td>
</tr>
<tr>
<td>Trapezoidal rule</td>
<td>[ \int_a^b k(x)dx \approx \frac{h}{2} \left( k(c - h) + 2k(c) + k(c + h) \right), ]</td>
</tr>
<tr>
<td></td>
<td>where ( c = \frac{a + b}{2}, h = \frac{b - a}{2}. )</td>
</tr>
<tr>
<td>Iterated trapezoidal rule</td>
<td>[ \int_a^b \int_d^c g(x, y)dydx \approx \frac{hk}{2} \left( g(c - h, f) + g(c + h, f) + g(c, f - h) + g(c, f + h) + 4g(c, f) \right), ]</td>
</tr>
<tr>
<td></td>
<td>where ( c = \frac{a + b}{2}, f = \frac{d + c}{2}, h = \frac{b - a}{2}, k = \frac{d - e}{2}. )</td>
</tr>
</tbody>
</table>

### 3.2. Numerical differentiation formulas

The formulas for forward difference, backward difference, central difference, and second derivative approximations are described as follows:

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward difference method</td>
<td>[ \frac{dk(x_0)}{dx} \approx \frac{k(x_0 + h) - k(x_0)}{h}, ]</td>
</tr>
<tr>
<td></td>
<td>where ( h &gt; 0. )</td>
</tr>
</tbody>
</table>
The derivative is approximated by the formula
\[
\frac{dk(x_0)}{dx} \approx \frac{k(x_0) - k(x_0 - h)}{h},
\]
where \(h > 0\).

The derivative is approximated by the formula
\[
\frac{dk(x_0)}{dx} \approx \frac{k(x_0 - h) - k(x_0 + h)}{2h},
\]
where \(h > 0\).

The second derivative is approximated by the formula
\[
\frac{dk^2(x_0)}{dx^2} \approx \frac{k(x_0 + h) - 2k(x_0) + k(x_0 - h)}{h^2},
\]
where \(h > 0\).

The second derivative is approximated by the formula
\[
\frac{dg^2(x_0, y_0)}{dx^2} + \frac{dg^2(x_0, y_0)}{dy^2} \\
\approx \frac{g(x_0 + h, y_0) + g(x_0 - h, y_0) + g(x_0, y_0 + h) + g(x_0, y_0 - h) - 4g(x_0, y_0)}{h^2},
\]
where \(h > 0\).

4. CNN Templates for Neural Field Equations

In this section, we discretize the neural field equations by using the numerical integration techniques described in Sec. 3.

4.1. Neural field equation A

Let us assume the intensity of output connections and the threshold are uniform, and assume there are no connections between input stimulus, that is,
\[
\int a(x, x')f(u(x', t))dx' \to a \int f(u(x', t))dx',
\]
\[
\int b(x, x')s(x')dx' \to bs(x),
\]
\[
z(x) \to z.
\]

Then, Eq. (13) assumes the simplified form
\[
\frac{\partial u(x, t)}{\partial t} = -u(x, t) + a \int f(u(x', t))dx' + bs(x) + z,
\]
where \(a, b\) and \(z\) are constants.

Let us replace \(u(x, t)\) with \(u_i\), which is the average membrane potential of neurons in the interval \([x_{i-1}, x_i]\) (namely, the cell \(C_i\)), and assume each cell is coupled locally only to cells inside a neighborhood of radius \(r = 1\).

4.1.1. Left Riemann sum

If we replace the integral in Eq. (14) by a left Riemann sum, we would obtain
\[
\frac{du_i}{dt} = -u_i + a\left(f(u_{i-1}) + f(u_i)\right)\Delta x + bs_i + z,
\]
which is characterized by the CNN template
\[
A = \bar{a} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \quad B = b \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} 1 \end{bmatrix},
\]
where \(\bar{a} = a\Delta x\).

- If we set \(\bar{a} = 2, b = 2, z = 0\), we would obtain a Shadow Projection CNN, which projects onto the right, the shadow of all objects illuminated from the left.

Observe that any one-dimensional template can be used for two-dimensional image processing by
inducing a pseudo two-dimensional template:
\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad A = \bar{a}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad B = b
\begin{array}{ccc}
\end{array}
\quad z = z.
\]
(17)

4.1.2. Right Riemann sum

If we replace the integral in Eq. (14) by a right Riemann sum, we would obtain
\[
\frac{du_i}{dt} = -u_i + a \left( f(u_{i}) + f(u_{i+1}) \right) \Delta x + bs_i + z,
\]
which is characterized by the CNN template
\[
\begin{array}{ccc}
0 & 1 & 1 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad A = \bar{a}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}
\quad B = b
\begin{array}{ccc}
\end{array}
\quad z = z,
\]
(19)

where \( \bar{a} = a \Delta x \).

- If we set \( \bar{a} = 2, b = 2, z = 0 \), we would obtain a Shadow Projection CNN, which projects onto the left, the shadow of all objects illuminated from the right.

4.1.3. Simpson’s rule

If we replace the integral in Eq. (14) by a Simpson’s rule, we would obtain
\[
\frac{du_i}{dt} = -u_i + \frac{a}{3} \left( f(u_{i-1}) + 4f(u_{i}) + f(u_{i+1}) \right) \Delta x + bs_i + z,
\]
which is characterized by the CNN template
\[
\begin{array}{ccc}
1 & 4 & 1 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad A = \bar{a}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}
\quad B = b
\begin{array}{ccc}
\end{array}
\quad z = z,
\]
(21)

where \( \bar{a} = (a/3)\Delta x \).

- If we set \( \bar{a} = 0.25, b = 0, z = 0 \), we would obtain a Horizontal Line Detection CNN, which extracts all horizontal lines.

4.1.4. Trapezoidal rule

If we replace the integral in Eq. (14) by a Trapezoidal rule, we would obtain
\[
\frac{du_i}{dt} = -u_i + \frac{a}{2} \left( f(u_{i-1}) + 2f(u_{i}) + f(u_{i+1}) \right) \Delta x + bs_i + z,
\]
which is characterized by the CNN template
\[
\begin{array}{ccc}
1 & 2 & 1 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad A = \bar{a}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad B = b
\begin{array}{ccc}
\end{array}
\quad z = z,
\]
(23)

where \( \bar{a} = (a/2)\Delta x \).

- If we set \( \bar{a} = 1, b = 0, z = 0 \), we would obtain a Horizontal Line Detection CNN, which extracts all horizontal lines.

- If we set \( \bar{a} = 1, b = 0, z = 2 \), we would obtain a Horizontal Shadow CNN, which projects onto both the left and the right shadow of all objects [Roska et al., 1999].

Applying the iterated trapezoidal rule to the two-dimensional neural field equations
\[
\frac{\partial u(w, t)}{\partial t} = -u(w, t) + a \int f(u(w', t)) dx' dy' + bs(w) + z,
\]
(24)

we obtain
\[
\frac{du_{ij}}{dt} = -u_{ij} + \frac{a}{2} \left( f(u_{i,j-1}) + f(u_{i,j+1}) + f(u_{i-1,j}) + f(u_{i+1,j}) \right) \Delta x \Delta y + bs_{ij} + z,
\]
(25)

which is characterized by the following CNN template
\[
\begin{array}{ccc}
0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{array}
\quad A = \bar{a}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}
\quad B = b
\begin{array}{ccc}
\end{array}
\quad z = z,
\]
(26)

where \( \bar{a} = (a/2)\Delta x\Delta y \). The following useful functions can be realized by this CNN template (see [Chua, 1998; Roska et al., 1999; Itoh & Chua, 2003b; Hänggi et al., 1999])

1. Hole Filling CNN \((\bar{a} = 1, b = 6, z = 0)\)
2. Filled-Contour Extraction CNN \((\bar{a} = 1, b = 2, z = -4)\)
3. Selected Objects Extraction CNN \((\bar{a} = 1, b = 7, z = -1)\)
4. Face-vease-illusion CNN \((\bar{a} = 1, b = -6, z = 0)\)
5. Solid Black Framed Area Extraction CNN \((\bar{a} = 1, b = -0.4, z = -2)\)
6. Noise Removal CNN \((\bar{a} = 1, b = 0, z = 0)\)

If we include also connections between the diagonal cells and the center cell, we would obtain
the following CNN template from Eq. (24)

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
1 & 8 & 1 \\
1 & 1 & 1
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad z = \left[ \begin{array}{c}
\end{array} \right].
\]

(27)

- If we set \( \alpha = 1, \beta = 8, z = 6 \), we would obtain a Figure Reconstruction CNN, which reconstructs a given input image [Roska et al., 1999].
- If we set \( \alpha = 1, \beta = 1, z = 9 \), we would obtain a Concave-filling CNN, which fills all concavities in the given image [Roska et al., 1999; Itoh & Chua, 2003b].

4.2. Neural field equation B

Let us assume that there are no connections between the outputs, and assume the intensity of the input connections and the threshold is uniform, that is,

\[
\int a(x, x') f(u(x', t)) dx' \rightarrow a f(u(x, t)),
\]

\[
\int b(x, x') s(x') dx' \rightarrow b \int s(x') dx', \quad z(x) \rightarrow z.
\]

(28)

Then, Eq. (1) assumes the simplified form

\[
\frac{du(x, t)}{dt} = -u(x, t) + af\left(u(x, t)\right) + b \int s(x') dx' + z,
\]

(29)

where \( a, b \) and \( z \) are constants.

Let \( u_i \) denote the average membrane potential of Cell \( C_i \) (neurons in the interval \([x_{i-1}, x_i]\)) and assume each cell is coupled locally only to cells inside a neighborhood of radius \( r = 1 \).

4.2.1. Left Riemann sum

If we replace the integral in Eq. (29) by a left Riemann sum, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) + b(s_{i-1} + s_i) \Delta x + z,
\]

which is characterized by the CNN template

\[
A = a \left[ \begin{array}{c}
0 & 1 & 0 \\
\end{array} \right], \quad B = \beta \left[ \begin{array}{c}
1 & 1 & 0 \\
\end{array} \right], \quad z = \left[ \begin{array}{c}
\end{array} \right],
\]

(31)

where \( \beta = b \Delta x \).

- If we set \( a = 1, \beta = 1, z = -1 \), we would obtain a left-peeling CNN, which peels one pixel from the left [Roska et al., 1999].

4.2.2. Right Riemann sum

If we replace the integral in Eq. (29) by a right Riemann sum, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) + b(s_i + s_{i+1}) \Delta x + z,
\]

which is characterized by the CNN template

\[
A = a \left[ \begin{array}{c}
0 & 1 & 0 \\
\end{array} \right], \quad B = \bar{b} \left[ \begin{array}{c}
0 & 1 & 1 \\
\end{array} \right], \quad z = \left[ \begin{array}{c}
\end{array} \right],
\]

(33)

where \( \bar{b} = b \Delta x \).

- If we set \( a = 1, \bar{b} = 1, z = -1 \), we would obtain a right-peeling CNN, which peels one pixel from the right [Roska et al., 1999].

4.2.3. Simpson’s rule

If we replace the integral in Eq. (29) by a Simpson’s rule, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) + \frac{b}{3} (s_{i-1} + 4s_i + s_{i+1}) \Delta x + z,
\]

which is characterized by the CNN template

\[
A = a \left[ \begin{array}{c}
0 & 1 & 0 \\
\end{array} \right], \quad B = \bar{b} \left[ \begin{array}{c}
1 & 4 & 1 \\
\end{array} \right], \quad z = \left[ \begin{array}{c}
\end{array} \right],
\]

(35)

where \( \bar{b} = (b/3) \Delta x \).

- If we set \( a = 2, \bar{b} = 0.25, z = -2.5 \) or \( a = 2, \bar{b} = 1, z = -6 \), we would obtain a peeling CNN, which peels one pixel from the right, and from the left.

4.2.4. Trapezoidal rule

If we replace the integral in Eq. (29) by a Trapezoidal rule, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) + \frac{b}{2} (s_{i-1} + 2s_i + s_{i+1}) \Delta x + z,
\]

which is characterized by the CNN template

\[
A = a \left[ \begin{array}{c}
0 & 1 & 0 \\
\end{array} \right], \quad B = \bar{b} \left[ \begin{array}{c}
1 & 2 & 1 \\
\end{array} \right], \quad z = \left[ \begin{array}{c}
\end{array} \right],
\]

(37)

where \( \bar{b} = (b/2) \Delta x \).
• If we set \( a = 2, \tilde{b} = 1, z = -4 \), we would also obtain a peeling CNN, which peels one pixel from the right, and from the left.

From the two-dimensional field equations

\[
\frac{\partial u(w, t)}{\partial t} = -u(w, t) + a f(u(w, t)) + b \int s(w') dx' dy' + z, \tag{38}
\]

we have the following CNN template

\[
A = a \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \tilde{b} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} 0 \end{bmatrix}, \tag{39}
\]

where \( \tilde{b} = (b/2)\Delta x \Delta y \).

• If we set \( a = 0, \tilde{b} = 0.125, z = -1 \), we would obtain an Inverse Half-Toning CNN [Chua, 1998; Itoh & Chua, 2003b].

From Eq. (38), we also obtain the following CNN template

\[
A = a \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \tilde{b} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} 0 \end{bmatrix}, \tag{40}
\]

where \( \tilde{b} = (b/4)\Delta x \).

• If we set \( a = 1, \tilde{b} = 1, z = -1 \), we would obtain a Point Removal CNN [Chua, 1998].

• If we set \( a = 0, \tilde{b} = 0.625, z = -1 \), we would obtain an Inverse Half-Toning CNN [Chua, 1998; Itoh & Chua, 2003b].

4.3. Neural field equation C

Let us assume that the intensity of the input connections, the output connections, and the threshold \( z(x) \) are uniform, that is,

\[
\begin{align*}
\int a(x, x') f(u(x', t)) dx' & \rightarrow a \int f(u(x, t)) dx', \\
\int b(x, x') s(x') dx' & \rightarrow b \int s(x') dx',
\end{align*}
\]

\( z(x) \rightarrow z \). \tag{41}

Then, Eq. (1) assumes the simplified form

\[
\frac{\partial u(x, t)}{\partial t} = -u(x, t) + a \int f(u(x', t)) dx' + b \int s(x') dx' + z. \tag{42}
\]

If we apply the left and the right Riemann sums, the trapezoidal rules, and the Simpson’s rules, to Eq. (42), we would obtain the following CNN templates:

\[
A = D_1 \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad D_1 \begin{bmatrix} 0 & 1 & 1 \\ 1 & 4 & 1 \end{bmatrix}, \quad D_1 \begin{bmatrix} 1 & 4 & 1 \end{bmatrix}, \tag{43}
\]

where \( D_1 \) and \( D_2 \) are constants.

• If we set \( D_1 = D_2 = 1, z = 0 \), we would obtain a Horizontal Line Detection CNN.

Applying the iterated trapezoidal rule to the two-dimensional neural field equation

\[
\begin{align*}
\frac{\partial u(w, t)}{\partial t} & = -u(w, t) + a \int \int f(u(w', t)) dx' dy' \\
& + b \int \int s(w') dx' dy' + z,
\end{align*}
\]

we obtain

\[
A = D_1 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \end{bmatrix}, \quad D_1 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \end{bmatrix}, \quad D_1 \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}. \tag{44}
\]

• If we set \( D_1 = D_2 = 1, z = 0 \), we would obtain a Horizontal and Vertical Line Detection CNN, which detects both horizontal and vertical lines.

We can also obtain the following CNN template from Eq. (44).

\[
A = D_1 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \end{bmatrix}, \quad B = D_2 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \end{bmatrix}, \quad z = \begin{bmatrix} 1 \end{bmatrix}. \tag{46}
\]
If we set \( D_1 = D_2 = 1, \ z = 0 \), we would obtain a Figure Correction CNN, which corrects the output \( f(u_i) \) if \( u_i(0) \neq s_i(u_i(0) = \pm1, s_i = \pm1) \), that is, if the initial state is different from the input image.

5. CNN Templates for Nonhomogeneous Advection Equations

The wave equation is closely related to the advection equation

\[
\frac{\partial u(x, t)}{\partial t} = -D \frac{\partial u(x, t)}{\partial x},
\]

(47)

This equation describes the passive advection of some scalar field carried along by a flow of constant speed. This advection equation has the solution

\[
u(x, t) = \phi(x - D t), \]

(48)

where \( \phi \) is an arbitrary function. This solution moves from the left to the right with speed \( D \) without changing shape.

5.1. Nonhomogeneous advection equations

Let us consider the following nonhomogeneous advection equation

\[
\frac{\partial u(x, t)}{\partial t} = g(x, u) - D \frac{\partial u(x, t)}{\partial x},
\]

(49)

where \( g(x, u) \) is a continuous function and \( D \) is a constant. Define the characteristic curve of Eq. (49) by

\[
x = x_0 + D t,
\]

(50)

where \( x_0 \) is a constant. Then, Eq. (49) has the form

\[
\frac{\partial u(x_0 + D t, t)}{\partial t} = g(x_0 + D t, u) - D \frac{\partial u(x_0 + D t, t)}{\partial x}.
\]

(51)

If we set \( v(t) = u(x_0 + D t, t) \), we obtain

\[
\frac{dv(t)}{dt} = \frac{du(x_0 + D t, t)}{dt} = \frac{\partial u(x_0 + D t, t)}{\partial t} + \frac{\partial u(x_0 + D t, t)}{\partial x} \frac{d(x_0 + D t)}{dt}
\]

\[
m = \frac{\partial u(x_0 + D t, t)}{\partial t} + D \frac{\partial u(x_0 + D t, t)}{\partial x}
\]

\[
g(x_0 + D t, u(x_0 + D t, t))
\]

\[
g(x_0 + D t, v(t)).
\]

(52)

Therefore, Eq. (49) is transformed into the form

\[
\frac{dv(t)}{dt} = g(x_0 + D t, v(t)).
\]

(53)

Let \( v(t) = \phi(x_0, v_0, t) \) be a solution of Eq. (53), which satisfies

\[
v_0 = v(0) = u(x_0, 0) \Delta \equiv h(x_0).
\]

(54)

Substituting this and \( x_0 = x - D t \) into \( \phi(x_0, v_0, t) \), we get

\[
\phi(x_0, v_0, t) = \phi(x_0, h(x_0), t)
\]

\[
= \phi(x - D t, h(x - D t), t),
\]

(55)

and

\[
\phi(x_0, v_0, t) = v(t) = u(x_0 + D t, t) = u(x, t).
\]

(56)

Thus, solutions of Eq. (49) can be written as

\[
u(x, t) = \phi(x - D t, h(x - D t), t).
\]

(57)

Note that \( u(x, t) \) moves from the left to the right with the speed \( dx/dt = D \) while retaining its shape, if \( \phi(x_0, v_0, t) \) can be written as

\[
\phi(x_0, v_0, t) = \psi(x_0, v_0),
\]

(58)

where \( \psi(x_0, v_0) \) is a continuous function of \( x_0 \) and \( v_0 \).

5.2. Discretization of partial differential equation

Consider next the following partial differential equation

\[
\frac{\partial u(x, t)}{\partial t} = -u(x, t) + af\left(u(x, t)\right)
\]

\[-D \frac{\partial f\left(u(x, t)\right)}{\partial x} + bs(x) + z.
\]

(59)

where

\[
f(u) = \frac{1}{2}\left(|u + 1| - |u - 1|\right).
\]

(60)

If \( g(x, u) \) in Eq. (49) satisfies

\[
g(x, u) = -u + af(u) + bs(x) + z,
\]

(61)
then Eqs. (49) and (59) are equivalent in the region \(|u| \leq 1\). In order to discretize Eq. (59), we divide the domain \([x_{\text{min}}, x_{\text{max}}]\) into \(N\) intervals \([x_{i-1}, x_i]\) \((i = 1, 2, \ldots, N)\) with length \(\Delta x\). Furthermore, let \(u_i\) be the average of \(u(x, t)\) in the interval \([x_{i-1}, x_i]\).

### 5.2.1. Forward difference method
If we replace the derivative in Eq. (59) by a forward difference method, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) - \frac{D(-f(u_{i-1}) + f(u_{i+1}))}{\Delta x} + bs_i + z
\]

\[
= -u_i + af(u_i) + \frac{D(f(u_{i-1}) - f(u_{i+1}))}{\Delta x} + bs_i + z,
\]

which is characterized by the template

\[
A = \overline{D} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}, \quad B = b \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix},
\]

where \(\overline{D} = D/\Delta x\) and \(a = 2\overline{D}\).

- If we set \(\overline{D} = 2, b = 2\), we would obtain a Shadow Projection CNN, which projects onto the right, the shadow of all objects illuminated from the left [Chua, 1998].

### 5.2.2. Backward difference method
If we replace the derivative in Eq. (59) by a backward difference method, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) - \frac{D(-f(u_i) + f(u_{i+1}))}{\Delta x}
\]

\[
= -u_i + af(u_i) + \frac{D(f(u_{i-1}) - f(u_{i+1}))}{\Delta x} + bs_i,
\]

which is characterized by the template

\[
A = \overline{D} \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, \quad B = b \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix},
\]

where \(\overline{D} = D/\Delta x\) and \(a = 2\overline{D}\).

- If we set \(\overline{D} = 2, b = 2\), we would obtain a Shadow Projection CNN, which projects onto the left, the shadow of all objects illuminated from the right [Chua, 1998].

### 5.2.3. Central difference method
If we replace the derivative in Eq. (59) by a central difference method, we would obtain

\[
\frac{du_i}{dt} = -u_i + af(u_i) - \frac{D(-f(u_{i-1}) + f(u_{i+1}))}{\Delta x}
\]

\[
= -u_i + af(u_i) + \overline{D}(f(u_{i-1}) - f(u_{i+1})) + bs_i.
\]

Equation (66) is characterized by the template

\[
A = \overline{D} \begin{bmatrix} 1 & 2 & -1 \end{bmatrix}, \quad B = b \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} z \end{bmatrix},
\]

where \(\overline{D} = D/\Delta x\) and \(a = 2\overline{D}\).

- If we set \(\overline{D} = 1, b = 0\), we would obtain a Horizontal Hole Detection CNN, which detects the number of horizontal holes [Chua, 1998].

### 6. CNN Templates for Reaction–Diffusion Equations

Reaction–diffusion is a process in which two or more chemical species diffuse over a surface and react with one another to produce patterns. This process can produce a variety of patterns. Consider a one-dimensional reaction–diffusion equation

\[
\frac{\partial u(x,t)}{\partial t} = g(u(x,t)) + D \frac{\partial^2 u(x,t)}{\partial x^2},
\]

where \(g(u, x)\) is a scalar function of \(u\) and \(x\), and \(D\) is the diffusion coefficient. Divide the domain \([x_{\text{min}}, x_{\text{max}}]\) into \(N\) intervals \([x_{i-1}, x_i]\) \((i = 1, 2, \ldots, N)\) with length \(\Delta x\). Let \(u_i\) be the average of \(u(x, t)\) in the interval \([x_{i-1}, x_i]\). Then, the discretized equation can be described by

\[
\frac{du_i}{dt} = g(u_i) + \frac{D(u_{i-1} - 2u_i + u_{i+1})}{\Delta x^2}
\]

\[
= g(u_i) + \overline{D}(u_{i-1} - 2u_i + u_{i+1}),
\]

where \(\overline{D} = D/\Delta x^2\).

Consider next the following partial differential equation with spatial integral

\[
\frac{\partial u(x,t)}{\partial t} = -u(x,t) + D \frac{\partial^2 f(u(x,t))}{\partial x^2} + b \int s(y)dy + z,
\]

where \(s(y)\) is a spatial function and \(b\) and \(z\) are coefficients.
where
\[ f(u) = \frac{1}{2}(|u + 1| - |u - 1|). \] (71)

Equation (70) is obtained from Eq. (1) by replacing the integral term
\[ \int a(x, y)f(u(y, t)) dy \] (72)
with the diffusion term
\[ \frac{D}{\partial x^2} \frac{\partial^2 f(u(x, t))}{\partial x^2}. \] (73)

If we apply the second derivative approximation to Eq. (70) [Chua, 1995, 1999], we would obtain
\[
\frac{du_i}{dt} = -u_i + \frac{D}{\Delta x^2} \left( f(u_{i-1}) - 2f(u_i) + f(u_{i+1}) \right)
+ b(s_{i-1} + 2s_i + s_{i+1}) + z
= -u_i + \overline{D} \left( f(u_{i-1}) - 2f(u_i) + f(u_{i+1}) \right)
+ b(s_{i-1} + 2s_i + s_{i+1}) + z. \] (74)

Thus, Eq. (70) is characterized by the CNN template
\[ A = \overline{D} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \quad B = b \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \quad z = \begin{bmatrix} \cdot \end{bmatrix}. \] (75)

- If we set \( D = -1, b = 1, z = 0 \), we would obtain a half-toning CNN, which transforms a given gray-scale image into a “half-tone” binary image [Chua, 1998; Roska et al., 1999; Itoh & Chua, 2003b].

Consider the following two-dimensional partial differential equation with spatial integral
\[
\frac{\partial u(w, t)}{\partial t} = -u(x, t)
+ D \left( \frac{\partial^2 f(u(w, t))}{\partial x^2} + \frac{\partial^2 f(u(w, t))}{\partial y^2} \right)
+ b \int s(w')dx'dy' + z. \] (76)

where \( w = (x, y) \) and \( w' = (x', y') \). Applying the trapezoidal rule to Eq. (76), we obtain
\[
\frac{du_{ij}}{dt} = -u_{ij} + \frac{D}{\Delta x^2} \left( f(u_{i-1,j}) + f(u_{i+1,j}) - 2f(u_{i,j}) \right)
+ \frac{D}{\Delta y^2} \left( f(u_{i,j-1}) + f(u_{i,j+1}) - 2f(u_{i,j}) \right)
+ b \left( s_{i,j-1} + s_{i,j+1} + s_{i-1,j} + s_{i+1,j} \right) \Delta x\Delta y + z. \] (77)

If \( \Delta x = \Delta y \), then we obtain
\[
\frac{du_{ij}}{dt} = -u_{ij} + \left( D(f(u_{i,j-1}) + f(u_{i,j+1}) + f(u_{i-1,j}) + f(u_{i+1,j}) - 4f(u_{i,j})) \right) / \Delta x^2
+ b \left( s_{i-1} + s_{i+1,j} + s_{i,j-1} + s_{i,j+1} \right) \Delta x^2 + z, \] (78)

which is characterized by the following CNN template
\[ A = \overline{D} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = b \begin{bmatrix} 1 & 4 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} \cdot \end{bmatrix}. \] (79)

where \( \overline{D} = D / \Delta x^2 \), and \( \overline{b} = b \Delta x^2 \). We can also obtain the CNN template
\[ A = \overline{D} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad B = b \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad z = \begin{bmatrix} \cdot \end{bmatrix}. \] (80)

- If we set \( \overline{D} = -1, \overline{b} = 1, z = 0 \), we would also obtain a half-toning CNN, which transforms a given gray-scale image into a “half-tone” binary image [Chua, 1998; Roska et al., 1999; Itoh & Chua, 2003b].

7. Classification of CNNs
Many CNNs are classified into the following groups from the results of Sec. 6.
- Neural Field Equation A
\[ \frac{\partial u(x,t)}{\partial t} = -u(x,t) + a \int f(u(x',t)) dx' + bs(x) + z \]

- Discretized CNNs
  - Shadow Projection CNN (Riemann sum)
  - Horizontal Line Detection CNN (Simpson’s rule)
  - Horizontal Shadow CNN (Trapezoidal rule)
  - Hole Filling CNN (Trapezoidal rule)
  - Filled Contour Extraction CNN (Trapezoidal rule)
  - Selected Objects Extraction CNN (Trapezoidal rule)
  - Solid Black Framed Area Extraction CNN (Trapezoidal rule)
  - Noise Removal CNN (Trapezoidal rule)

- Neural Field Equation B
\[ \frac{\partial u(x,t)}{\partial t} = -u(x,t) + af(u(x,t)) + b \int s(x')dx' + z \]

- Discretized CNNs
  - Peeling CNN (Riemann sum, Simpson’s rule, Trapezoidal rule)
  - Inverse Half-Toning CNN (Trapezoidal rule)
  - Point Removal CNN (Trapezoidal rule)

- Neural Field Equation C
\[ \frac{\partial u(x,t)}{\partial t} = -u(x,t) + a \int f(u(x',t)) dx' + b \int s(x')dx' + z \]

- Discretized CNNs
  - Horizontal Line Detection CNN (Riemann sum, Simpson’s rule, Trapezoidal rule)
  - Horizontal and Vertical Line Detection CNN (Trapezoidal rule)
  - Figure Correction CNN (Trapezoidal rule)

- Nonhomogeneous Advection Equation
\[ \frac{\partial u(x,t)}{\partial t} = -u(x,t) + af(u(x,t)) + D \frac{\partial f(u(x,t))}{\partial x} + bs(x) + z \]

- Discretized CNNs
  - Shadow Projection CNN (Forward difference, Backward difference)
  - Horizontal Hole Detection CNN (Central difference)
• Reaction–Diffusion Equation
\[
\frac{\partial u(x,t)}{\partial t} = -u(x,t) + D \frac{\partial^2 f(u(x,t))}{\partial x^2} + b \int s(x')dx' + z
\]
• Discretized CNN
  — Half-toning CNN (Second derivative approximation)

The CNN basic templates without scale parameters are listed as follows:

![Templates for image processing](image)

We note that the following two template are used to design some useful CNNs in [Itoh & Chua, 2003]

![Templates for image processing](image)

where \( \gamma \) is a constant. These templates are not derived in this paper, since we assumed that the couplings between cells are uniform. However, by assuming that the self-input stimulus is weaker or stronger than others, we can derive the templates (82).

8. Self-Organization

In many complex systems, there is a large number of subsystems whose individual dynamics are relatively simple. However, when these subsystems are strongly coupled, their dynamics can be complex and self-organized behavior can emerge. In this section, we study the self-organization of the dynamical systems from the viewpoint of the divergence property of vector fields.

Consider the following classes of partial differential equations (without inputs):

1. Neural field equation
\[
\frac{\partial u(w,t)}{\partial t} = -u(w,t) + a \int \int f(u(w',t))dx'dy',
\]
(83)

where \( w = (x,y) \in \mathbb{R}^2 \) and \( w' = (x',y') \in \mathbb{R}^2 \), and
\[
f(u) = \frac{1}{2} \left( |u + 1| - |u - 1| \right).
\]
(84)

2. One-dimensional advection equation
\[
\frac{\partial u(x,t)}{\partial t} = g(u) - D \frac{\partial u(x,t)}{\partial x}.
\]
(85)
3. First-order reaction–diffusion equation
\[
\frac{\partial u(w, t)}{\partial t} = g(u(w)) + D_u \left( \frac{\partial^2 u(w, t)}{\partial x^2} + \frac{\partial^2 u(w, t)}{\partial y^2} \right).
\] (86)

4. Second-order reaction–diffusion equation
\[
\frac{\partial u(w, t)}{\partial t} = h(u(w), v(w)) + D_u \left( \frac{\partial^2 u(w, t)}{\partial x^2} + \frac{\partial^2 u(w, t)}{\partial y^2} \right),
\]
\[
\frac{\partial v(w, t)}{\partial t} = k(u(w), v(w)) + D_v \left( \frac{\partial^2 v(w, t)}{\partial x^2} + \frac{\partial^2 v(w, t)}{\partial y^2} \right).
\] (87)

5. Second-order conservative system
\[
\frac{\partial u(w, t)}{\partial t} = v(w, t),
\]
\[
\frac{\partial v(w, t)}{\partial t} = h(u(w)) + D_u \left( \frac{\partial^2 u(w, t)}{\partial x^2} \right).
\] (88)

Assume that the parameters \(a, D, D_u\) and \(D_v\) satisfy
\[
a > 0, \quad D > 0, \quad D_u > 0, \quad D_v > 0.
\]

From Eqs. (83)–(88), we obtain the following corresponding discretized differential equations:

1. Discretized neural field equation
\[
\frac{du_{i,j}}{dt} = -u_{i,j} + \bar{a}\left(f(u_{i,j-1}) + f(u_{i,j+1}) + f(u_{i-1,j}) + f(u_{i+1,j}) + 4f(u_{i,j})\right).
\] (89)

2. Discretized one-dimensional advection equation
\[
\frac{du_i}{dt} = g(u_i) + \bar{D}(u_{i-1} - u_i).
\] (90)

3. Discretized first-order reaction–diffusion equation
\[
\frac{du_{i,j}}{dt} = g(u_{i,j}) + \bar{D}_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}).
\] (91)

4. Discretized second-order reaction–diffusion equation
\[
\frac{du_{i,j}}{dt} = h(u_{i,j}, v_{i,j}) + \bar{D}_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}),
\]
\[
\frac{dv_{i,j}}{dt} = k(u_{i,j}, v_{i,j}) + \bar{D}_v(v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j} - 4v_{i,j}).
\] (92)

5. Discretized second-order conservative system
\[
\frac{du_{i,j}}{dt} = u_{i,j},
\]
\[
\frac{dv_{i,j}}{dt} = h(u_{i,j}) + \bar{D}_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}),
\] (93)

where \(\bar{a} = a\Delta x^2\), \(\bar{D} = D/\Delta x\), \(\bar{D}_u = D_u/\Delta x^2\) and \(\bar{D}_v = D_v/\Delta y^2\).

A dynamical system defined by a vector field
\[
\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n), \quad i = 1, 2, \ldots, n
\] (94)

is said to be dissipative, if the phase space volume contracts along trajectories in the sense that the divergence
\[
\text{div} X \triangleq \sum_i \frac{\partial f_i}{\partial x_i} < 0,
\] (95)

for all \(x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n\). If the divergence
\[
\text{div} X = \sum_i \frac{\partial f_i}{\partial x_i} = 0,
\] (96)

for all \(x \in \mathbb{R}^n\), then the dynamical system is said to be conservative. The phase space of a dissipative dynamical system is continually shrinking onto smaller regions of the phase space, called the attracting set (fixed points, periodic orbits or chaotic attractors). In contrast, the phase space volume of conservative dynamical systems is invariant. The divergence of the vector field (89)–(93) can be written as follows:

1. Discretized neural field equation
\[
\text{div}(X) = -1 + 4\bar{a} \left( \frac{df(u_{i,j})}{du_{i,j}} \right)
\]
\[
= \begin{cases} 
-1 + 4\bar{a} & \text{if } |u_{i,j}| \leq 1, \\
-1 & \text{if } |u_{i,j}| \geq 1.
\end{cases}
\] (97)
2. Discretized one-dimensional advection equation
\[
\text{div}(\mathbf{X}) = \frac{dg(u_i)}{du_i} - D. \quad (98)
\]
3. Discretized first-order reaction–diffusion equation
\[
\text{div}(\mathbf{X}) = \frac{dg(u_{i,j})}{du_{i,j}} - 4D_u. \quad (99)
\]
4. Discretized second-order reaction–diffusion equation
\[
\text{div}(\mathbf{X}) = \frac{\partial h(u_{i,j}, v_{i,j})}{\partial u_{i,j}} + \frac{\partial k(u_{i,j}, v_{i,j})}{\partial v_{i,j}} - 4(D_u + D_v). \quad (100)
\]
5. Discretized second-order conservative system
\[
\text{div}(\mathbf{X}) = 0. \quad (101)
\]

It follows that if the local cells satisfy the following conditions
\[
4\bar{a} < 1, \quad \frac{dg(u_i)}{du_i} < D, \quad \frac{dg(u_{i,j})}{du_{i,j}} < 4D_u, \quad \frac{\partial h(u_{i,j}, v_{i,j})}{\partial u_{i,j}} + \frac{\partial k(u_{i,j}, v_{i,j})}{\partial v_{i,j}} < 4(D_u + D_v),
\]
then the systems (89)–(92) are dissipative.

Let us decompose the divergence of Eqs. (97)–(101) into two components; namely, a component originating from the uncoupled cell, and a component originating from the coupling between cells. They are listed in the following table:

| Equation | \(\text{div}(\mathbf{X})|_{\text{cell}}\) | \(\text{div}(\mathbf{X})|_{\text{coupling}}\) |
|----------|--------------------------------|------------------|
| Eq. (97) | \(-1 < 0\) | \(4\bar{a} \left( \frac{df(u_{i,j})}{du_{i,j}} \right) = \begin{cases} 4\bar{a} & \text{if } |u_{i,j}| \leq 1, \\ 0 & \text{if } |u_{i,j}| > 1. \end{cases}\) |
| Eq. (98) | \(\frac{dg(u_i)}{du_i}\) | \(-\bar{D} < 0\) |
| Eq. (99) | \(\frac{dg(u_{i,j})}{du_{i,j}}\) | \(-4\bar{D}_u < 0\) |
| Eq. (100) | \(\frac{\partial h(u_{i,j}, v_{i,j})}{\partial u_{i,j}} + \frac{\partial k(u_{i,j}, v_{i,j})}{\partial v_{i,j}}\) | \(-4(\bar{D}_u + \bar{D}_v) < 0\) |
| Eq. (101) | 0 | 0 |

Self-organization usually requires a continuous exchange of energy between the cells. It is usually associated with more complex, nonlinear phenomena. Any nonhomogeneous static or dynamic structure which emerges from the interactions between the cells is usually observed only when the system attractor is sufficiently far from equilibrium. In order to investigate the self-organization phenomena of Eqs. (89)–(93), let us study the dynamics of the uncoupled local cells, which are described as follows:

1. Discretized neural field equation
\[
\frac{du}{dt} = -u. \quad (103)
\]
2. Discretized one-dimensional advection equation
\[
\frac{du}{dt} = g(u). \quad (104)
\]
3. Discretized first-order reaction–diffusion equation
\[
\frac{du}{dt} = g(u). \quad (105)
\]
4. Discretized second-order reaction–diffusion equation
\[
\frac{du}{dt} = h(u, v), \quad \frac{dv}{dt} = k(u, v). \quad (106)
\]
5. Discretized second-order conservative system

\[ \frac{du}{dt} = v, \quad \frac{dv}{dt} = h(u). \]  \hspace{1cm} (107)

Let \( \rho_i \) denote the characteristic roots of the equilibrium points of these cells. Assume that all equilibrium points have at least one characteristic root \( \rho_j \) satisfying

\[ \begin{align*} 
\text{Re}(\rho_j) &> 0, \quad \text{if div}(X)|_{\text{coupling}} < 0, \\
\text{Re}(\rho_j) &< 0, \quad \text{if div}(X)|_{\text{coupling}} > 0,
\end{align*} \]  \hspace{1cm} (108)

where \( \text{Re}(x) \) indicates the real part of \( x \). Furthermore, assume that there is at least one equilibrium point \( e_k \) satisfying

\[ \begin{align*} 
div(X)|_{\text{equilibrium point } e_k} &> 0, \\
&\quad \text{if div}(X)|_{\text{coupling}} < 0, \\
div(X)|_{\text{equilibrium point } e_k} &< 0, \\
&\quad \text{if div}(X)|_{\text{coupling}} > 0.
\end{align*} \]  \hspace{1cm} (109)

Then, we can expect a continuous interaction between the cells, that is, a continuous exchange of energy between the cells. Furthermore, the attractor corresponding to any nonhomogeneous structure is expected to be sufficiently far from equilibrium. At equilibrium and near equilibrium there is only one steady state. In far from equilibrium, there may be several steady states.

The conservative system (93) cannot satisfy these inequalities because the divergence of the vector field is equal to 0. The Sine–Gordon Equation corresponds to this class of equations, which can have soliton solutions [Chua, 1998]. In contrast, the systems (90)–(92) may satisfy these inequalities, as demonstrated in the following examples.

### 8.1. Discretized neural field equation

Consider the discretized neural field equation (first-order CNN with the trapezoidal template)

\[ \begin{align*} 
\frac{du_{i,j}}{dt} &= -u_{i,j} + a \left( f(u_{i,j-1}) + f(u_{i,j+1}) \\
&\quad + f(u_{i-1,j}) + f(u_{i+1,j}) + 4f(u_{i,j}) \right). \hspace{1cm} (110)
\end{align*} \]

The dynamics of the uncoupled cell is defined by:

\[ \frac{du}{dt} = -u. \]  \hspace{1cm} (111)

Since the characteristic root \( \rho \) is equal to \(-1\), the origin is characterized by a sink. If we set \( a > 0 \), then we obtain

\[ \text{div}(X)|_{\text{coupling}} = 4a \left( \frac{df(u_{i,j})}{du_{i,j}} \right) = \begin{cases} 
4a > 0 & \text{if } |u_{i,j}| \leq 1, \\
0 & \text{if } |u_{i,j}| > 1.
\end{cases} \]  \hspace{1cm} (112)

Thus, Eq. (108) is not satisfied in the region \( |u_{i,j}| > 1 \). In our computer simulations, the emergence of dynamic patterns was not observed. The final states converge to static patterns as shown in Fig. 26 of [Itoh & Chua, 2003a].

### 8.2. Discretized one-dimensional advection equation

Consider the discretized one-dimensional advection equation

\[ \frac{du_{i,j}}{dt} = -u_{i,j} + 2f(u_{i,j}) + a(u_{i,j-1} + u_{i,j+1}) \\
+ u_{i-1,j} + u_{i+1,j} - 4u_{i,j}, \]  \hspace{1cm} (113)

where \( a = 1 \) and

\[ f(u_{i,j}) = \frac{1}{2} \left( |u_{i,j} + 1| - |u_{i,j} - 1| \right). \]  \hspace{1cm} (114)

The dynamics of the uncoupled cell is defined by:

\[ \frac{du}{dt} = -u + 2f(u). \]  \hspace{1cm} (115)

Equation (115) has three equilibrium points \( u = 0, \pm 2 \).

The equilibrium point \( u = 0 \) satisfies

\[ \rho = \text{div}(X)|_{u=0} = 1 > 0, \]  \hspace{1cm} (116)

which is a source. The equilibrium points \( u = \pm 2 \) satisfy

\[ \rho = \text{div}(X)|_{u=\pm 2} = -1 < 0, \]  \hspace{1cm} (117)

which is a sink. Furthermore, we get the inequality

\[ \text{div}(X)|_{\text{coupling}} = -4a = -4 < 0. \]  \hspace{1cm} (118)

Thus, Eq. (109) is not satisfied at the equilibrium point \( u = \pm 2 \). In our computer simulations, the emergence of dynamic patterns was not observed. The final states converge to a static pattern.
8.3. Discretized first-order reaction–diffusion equation

Consider the first-order reaction–diffusion equation [Yang & Chua, 2001]
\[
\frac{du_{i,j}}{dt} = -u_{i,j}^3 + \alpha u_{i,j} + \bar{a}(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}),
\]
where \( \alpha \) is a constant. The dynamics of the cell is written as
\[
\frac{du}{dt} = -u^3 + \alpha u.
\]
(120)

If we set \( \alpha = 2, \bar{a} = 1 \), Eq. (120) has three equilibrium points \( u = 0, \pm \sqrt{2} \).

The equilibrium point \( u = 0 \) satisfies
\[
\rho = \text{div}(X)|_{u=0} = -3u^2 + \alpha = 2 > 0.
\]
(121)

which is a source. The equilibrium points \( u = \pm \sqrt{2} \) satisfy
\[
\rho = \text{div}(X)|_{u=\pm \sqrt{2}} = -3u^2 + \alpha = -6 < 0
\]
(122)

which is a sink. Furthermore, we get the inequality
\[
\text{div}(X)|_{\text{coupling}} = -4\alpha = -4 < 0.
\]
(123)

Thus, Eq. (108) is not satisfied at the equilibrium points \( u = \pm \sqrt{2} \). In our computer simulations, the emergence of dynamic patterns was not observed. The final states converge to a static pattern as shown in Fig. 2 of [Yang & Chua, 2001].

8.4. Second-order reaction–diffusion CNN

8.4.1. One-port and second-order CNN

Consider the one-port and second-order CNN proposed by [Yang & Chua, 2001]
\[
\frac{du_{i,j}}{dt} = u_{i,j}^3 + (0.5\alpha - b)u_{i,j} - v_{i,j}^3
+ (1.5\alpha + b)v_{i,j} + \bar{D}_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}),
\]
(124)
\[
\frac{dv_{i,j}}{dt} = u_{i,j} - v_{i,j}.
\]
The dynamics of the uncoupled cell is defined by:
\[
\frac{du}{dt} = u^3 + (0.5\alpha - b)u - v^3 + (1.5\alpha + b)v,
\]
\[
\frac{dv}{dt} = u - v.
\]
(125)

\[\textbf{Case 1}. \quad \alpha = -1, \quad b = -10, \quad \bar{D}_u = 2.\]

In this case Eq. (125) has the origin as its only equilibrium point. It follows that
\[
\rho_1 + \rho_2 = \text{div}(X)|_{u,v=(0,0)} = 0.5\alpha - b - 1 = 8.5 > 0,
\]
\[
\rho_1\rho_2 = -2\alpha = 2 > 0,
\]
\[
\text{div}(X)|_{\text{coupling}} = -4\bar{D}_u = -8 < 0.
\]

Since \( \text{Re}(\rho_i) > 0 \) \( i = 1, 2 \), this equilibrium point is characterized by a source. Thus, Eqs. (108) and (109) are satisfied. In our computer simulations, we found the emergence of spiral waves as shown in Fig. 8 of [Yang & Chua, 2001].

\[\textbf{Case 2}. \quad \alpha = -1, \quad b = -1, \quad \bar{D}_u = 1.\]

In this case, Eq. (125) has the origin as its only one equilibrium point, which satisfies the relation
\[
\rho_1 + \rho_2 = \text{div}(X)|_{u,v=(0,0)} = 0.5\alpha - b - 1 = -0.5 < 0.
\]
\[
\rho_1\rho_2 = -2\alpha = 2 > 0,
\]
\[
\text{div}(X)|_{\text{coupling}} = -4\bar{D}_u = -4 < 0.
\]

Since \( \rho_1 < 0 \) and \( \rho_2 < 0 \), this equilibrium point is characterized by a sink. In this case, Eq. (108) is not satisfied, and we cannot find the emergence of any dynamic patterns. The final states converge to static patterns as shown in Fig. 6 of [Yang & Chua, 2001].

\[\textbf{Case 3}. \quad \alpha = 1, \quad b = -1, \quad \bar{D}_u = 1.\]

In this case, Eq. (125) has three equilibrium points \( \{(0,0), (1,1), (-1,-1)\} \). The origin \( (u,v) = (0,0) \) satisfies
\[
\rho_1 + \rho_2 = \text{div}(X)|_{u,v=(0,0)} = 0.5\alpha - b - 1 = 0.5 > 0,
\]
\[
\rho_1\rho_2 = -2\alpha = -2 < 0,
\]
which is a saddle.

The equilibrium points \( (u,v) = \{(1,1), (-1,-1)\} \) satisfy
\[
\rho_1 + \rho_2 = \text{div}(X)|_{u,v=\pm(1,1)} = -2.5\alpha - b - 1 = -2.5 < 0,
\]
\[
\rho_1\rho_2 = 6 - 2\alpha = 4 > 0.
\]
which are sinks. Furthermore, we get the inequality
\[ \text{div}(X)|_{\text{coupling}} = -4\overline{D}_u = -4 < 0. \] (130)
Thus, Eq. (108) is not satisfied, and we cannot find the emergence of dynamic patterns. The final states converge to a static pattern as shown in Fig. 5 of [Yang & Chua, 2001].

### 8.4.2. Fitzhugh–Nagumo CNN

Consider the Fitzhugh–Nagumo CNN [Dogaru & Chua, 1998]
\[
\frac{du_{i,j}}{dt} = cu_{i,j} - u_{i,j}^3 - u_{i,j} + \overline{D}_u (u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}),
\]
(131)
\[
\frac{dv_{i,j}}{dt} = -\epsilon (u_{i,j} - bv_{i,j} + a),
\]
where \(a, b, c\) and \(\epsilon\) are positive constants. The dynamics of the uncoupled cell is defined by:
\[
\frac{du}{dt} = cu - u^3 - v,
\]
(132)
\[
\frac{dv}{dt} = -\epsilon (u - bv + a).
\]

- **Case 1.** \(a = 0, b = 1.2, c = 1, \epsilon = -0.1, \overline{D}_u = 1\).

In this case, Eq. (132) has three equilibrium points \((u, v) = (0, 0), (1/\sqrt{2}, 5/6\sqrt{2}), (-1/\sqrt{2}, -5/6\sqrt{2})\). The origin \((u, v) = (0, 0)\) satisfies the relation
\[
\rho_1 + \rho_2 = \text{div}(X)|_{(u,v)=(0,0)} = c - u^2 + eb \approx 0.88 > 0,
\]
\[
\rho_1 \rho_2 = (c - u^2)eb - \epsilon \approx -0.02 < 0,
\]
which is a saddle.

The equilibrium points \((1/\sqrt{2}, 5/6\sqrt{2})\) and \((-1/\sqrt{2}, -5/6\sqrt{2})\) satisfy
\[
\rho_1 + \rho_2 = \text{div}(X)|_{(u,v)=(1/\sqrt{2}, 5/6\sqrt{2})} = c - u^2 + eb = 0.38 > 0,
\]
\[
\rho_1 \rho_2 = (c - u^2)eb - \epsilon \approx 0.004 > 0,
\]
which are sources. Furthermore, we get the inequality
\[ \text{div}(X)|_{\text{coupling}} = -4\overline{D}_u = -4 < 0. \] (135)
Thus, Eqs. (108) and (109) are satisfied. In our computer simulations, we found the emergence of spiral waves.

- **Case 2.** \(a = 0.1, b = 1.4, c = 1, \epsilon = -0.1, \overline{D}_u = 0.1\).

In this case, Eq. (132) has three equilibrium points
\[
(u, v) = (-1.03, 0.664), (0.274, 0.267), (0.758, 0.613).
\]
The first equilibrium point \((u, v) = (-1.03, 0.664)\) satisfies the relation
\[
\rho_1 + \rho_2 = \text{div}(X)|_{(u,v)=(-1.03,0.664)} = c - u^2 + eb \approx -0.204 < 0,
\]
\[
\rho_1 \rho_2 = (c - v^2)eb - \epsilon \approx 0.109 > 0.
\]
which is a sink.

The second equilibrium point \((u, v) = (0.274, 0.267)\) satisfies the relation
\[
\rho_1 + \rho_2 = \text{div}(X)|_{(u,v)=(0.274,0.267)} = c - u^2 + eb \approx 0.785 > 0,
\]
\[
\rho_1 \rho_2 = (c - v^2)eb - \epsilon \approx -0.0295 < 0.
\]
which is a saddle.

The third equilibrium point \((u, v) = (0.758, 0.613)\) satisfies the relation
\[
\rho_1 + \rho_2 = \text{div}(X)|_{(u,v)=(0.758,0.613)} = c - u^2 + eb \approx 0.285 > 0,
\]
\[
\rho_1 \rho_2 = (c - v^2)eb - \epsilon \approx 0.0405 > 0.
\]
which is a source.

Furthermore, we get the inequality
\[ \text{div}(X)|_{\text{coupling}} = -4\overline{D}_u = -0.4 < 0. \] (139)
Thus, Eq. (108) is not satisfied at the first equilibrium point. In our computer simulations, the spiral waves are not sustained for a long time if we set \(\overline{D}_u = 0.1\) (see Fig. 15 of [Dogaru & Chua, 1998]). However, the spiral waves continue to emerge if we set \(\overline{D}_u = 0.08\). Note that there is a possibility of self-organization even though the system does not satisfy Eqs. (108) and (109). In other words, the conditions given in the preceding section are not necessary conditions for the emergence of self-organization.

### 9. Emergence of Patterns

The emergence of dynamic and static patterns for the discretized CNNs considered in Sec. 8 is
summarized as follows:

<table>
<thead>
<tr>
<th>Discretized one-dimensional advection equation</th>
</tr>
</thead>
</table>
| \[
\frac{du_{i,j}}{dt} = -u_{i,j} + \bar{a}\left(f(u_{i,j-1}) + f(u_{i,j+1}) + f(u_{i-1,j}) + f(u_{i+1,j}) + 4f(u_{i,j})\right)
\] |
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Eqs. (108) and (109)</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{a} = 1$</td>
<td>not satisfied</td>
<td>static patterns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discretized first-order reaction–diffusion equation</th>
</tr>
</thead>
</table>
| \[
\frac{du_{i,j}}{dt} = -u_{i,j}^3 + \alpha u_{i,j} + \bar{a}(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j})
\] |
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Eqs. (108) and (109)</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 2$, $\bar{a} = 1$</td>
<td>not satisfied</td>
<td>static patterns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>One-port and second-order CNN</th>
</tr>
</thead>
</table>
| \[
\frac{du_{i,j}}{dt} = u_{i,j}^3 + (0.5\alpha - b)u_{i,j} - v_{i,j}^3 + (1.5\alpha + b)v_{i,j} + D_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j})
\]
| \[
\frac{dv_{i,j}}{dt} = u_{i,j} - v_{i,j}.
\] |
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Eqs. (108) and (109)</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = -1$, $b = -10$, $D_u = 2$</td>
<td>satisfied</td>
<td>spiral waves</td>
</tr>
<tr>
<td>$\alpha = -1$, $b = -1$, $D_u = 1$</td>
<td>not satisfied</td>
<td>static patterns</td>
</tr>
<tr>
<td>$\alpha = 1$, $b = -1$, $D_u = 1$</td>
<td>not satisfied</td>
<td>static pattern</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FitzHugh–Nagumo CNN</th>
</tr>
</thead>
</table>
| \[
\frac{du_{i,j}}{dt} = cu_{i,j} - u_{i,j}^3 - v_{i,j} + D_u(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j})
\]
| \[
\frac{dv_{i,j}}{dt} = -\epsilon(u_{i,j} - bv_{i,j} + a).
\] |
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Eqs. (108) and (109)</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 0.0$, $b = 1.2$, $c = 1$, $\epsilon = -0.1$, $D_u = 1.0$</td>
<td>satisfied</td>
<td>spiral waves</td>
</tr>
<tr>
<td>$a = 0.1$, $b = 1.4$, $c = 1$, $\epsilon = -0.1$, $D_u = 0.1$</td>
<td>not satisfied</td>
<td>static patterns</td>
</tr>
<tr>
<td>$a = 0.1$, $b = 1.4$, $c = 1$, $\epsilon = -0.1$, $D_u = 0.08$</td>
<td>not satisfied</td>
<td>spiral waves</td>
</tr>
</tbody>
</table>
10. Conclusion

We have derived CNN templates for image processing and pattern formation from neural field equations, advection equations and reaction–diffusion equations by discretizing spatial integral and derivative. Many useful templates for image processing are derived from this method. Furthermore, we investigated the emergence of self-organization in discretized CNNs. In all cases where a static or dynamic nonhomogeneous pattern or structure emerges, the uncoupled cell must operate in the locally active regime [Chua, 1998]. In most cases, static and dynamic patterns emerge when the uncoupled cells are operating in or near an edge of chaos regime [Dogaru & Chua, 1998; Yang & Chua, 2001].

Acknowledgments

This work is supported in part by the MURI contract no. N00014-03-1-0698, the DURINT contract no. N00014-01-0741, and the MARCO FENA Award no. 442521/WK57015.

References