A systematic design methodology for finding CNN parameters with prescribed functions is proposed. A given function (task) is translated into several local operations, and they are realized as stable states of the CNN system. Many CNN parameters (CNN genes) with the same functions can be easily derived by using this design methodology. A genetic algorithm based CNN gene design methodology is also proposed. Two new genetic “activation and inactivation” operations are introduced to generate CNN genes effectively. Many useful CNN genes can be obtained systematically from known genes by using these genetic operations. Furthermore, the signal propagation property for activated and inactivated CNN genes is studied.

Keywords: Cellular neural networks; CNN; CNN genes; genetic algorithm.

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, CNN has found numerous applications in image and video signal processing, in robotic and biological visions, and in higher brain functions.

Every CNN has parameters, which must be tuned to achieve certain functions (these parameters determine the CNN genes). That is, these parameters must be derived analytically or numerically from given functions. Many methods for deriving the CNN parameters have been proposed [Zarandy, 1999], and several software libraries have been published [Roska et al., 1999]. We note that these parameters represent only a typical choice since many other parameters within some neighborhood of the listed parameter will give the same result. In other words, the gene for a particular task is not unique, and one can optimize the gene parameters to enhance robustness.

In this paper, we propose a systematic approach for finding CNN parameters with prescribed functions (task). The CNN architecture realizes global tasks by using local operations. In order to design the CNN genes, we have to decompose a global task into several local operations. Then, we can realize these local operations as stable states of a CNN system via a “driving-point plot analysis” [Chua, 1998]. By using this method, we can easily design many CNN parameters (CNN genes) with the same functions. Next, we propose a new method based on genetic algorithms. In this method, two new operations, “activation and inactivation” were introduced. Many useful CNN genes can be derived systematically from known CNN genes by using these genetic operations. Finally, we show that both activated and inactivated CNN genes have several interesting properties on signal propagations.
2. Dynamics of the CNN Equations

The dynamics of a standard cellular neural network with a neighborhood of radius \( r \) are governed by a system of \( n = MN \) differential equations [Chua, 1998]

\[
\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{k,l \in N_{ij}} (a_{k-l,j}y_{kl} + b_{k-l,j}u_{kl}) + z_{ij},
\]

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

where \( N_{ij} \) 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 \( B = [b_{kl}] \) formed by the “coupling” coefficients are referred to as the feedforward template \( A \) and the feedforward (input) template \( B \), respectively. The output \( y_{ij} \) and the state \( x_{ij} \) of each cell are related through the piecewise-linear saturation function

\[
y_{ij} = f(x_{ij}) = \frac{1}{2}(|x_{ij} + 1| - |x_{ij} - 1|),
\]

whose graph is shown in Fig. 1.

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

\[
A = \begin{bmatrix}
a_{-1,-1} & a_{-1,0} & a_{-1,1} \\
a_{0,-1} & a_{0,0} & a_{0,1} \\
a_{1,-1} & a_{1,0} & a_{1,1}
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
b_{-1,-1} & b_{-1,0} & b_{-1,1} \\
b_{0,-1} & b_{0,0} & b_{0,1} \\
b_{1,-1} & b_{1,0} & b_{1,1}
\end{bmatrix},
\]

and the element \( z \). The corresponding CNN gene is defined by

\[
\mathcal{G} : a_{-1,-1} a_{-1,0} a_{-1,1} \cdots a_{1,0} a_{1,1} b_{-1,-1} b_{-1,0} b_{-1,1} \cdots b_{1,0} b_{1,1} z.
\]

It is sometimes convenient to decompose \( \mathcal{G} \) into three substrings:

\[
\text{A-gene : } a_{-1,-1} a_{-1,0} a_{-1,1} a_{0,-1} a_{0,0} a_{0,1} a_{1,-1} a_{1,0} a_{1,1},
\]

\[
\text{B-gene : } b_{-1,-1} b_{-1,0} b_{-1,1} b_{0,-1} b_{0,0} b_{0,1} b_{1,-1} b_{1,0} b_{1,1},
\]

\[
\text{z-gene : } z.
\]

In this paper, the following palettes are applied to the state \( x_{ij} \) and the output \( y_{ij} \):

<table>
<thead>
<tr>
<th>state ( x_{ij} )</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 &lt; x_{ij} )</td>
<td>red</td>
</tr>
<tr>
<td>( x_{ij} = 1 )</td>
<td>black</td>
</tr>
<tr>
<td>( -1 &lt; x_{ij} &lt; 1 )</td>
<td>gray scale</td>
</tr>
<tr>
<td>( x_{ij} = -1 )</td>
<td>white</td>
</tr>
<tr>
<td>( x_{ij} &lt; -1 )</td>
<td>green</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>output ( y_{ij} )</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_{ij} = 1 )</td>
<td>black</td>
</tr>
<tr>
<td>( -1 &lt; y_{ij} &lt; 1 )</td>
<td>gray scale</td>
</tr>
<tr>
<td>( y_{ij} = -1 )</td>
<td>white</td>
</tr>
</tbody>
</table>

The system equation (1) can be transformed into the form

\[
\frac{dx_{ij}}{dt} = -x_{ij} + a_{00}f(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} (a_{k-l,j}y_{kl} + b_{k-l,j}u_{kl} + z_{ij})
\]

where \( g(x_{ij}) = x_{ij} + a_{00}f(x_{ij}) \). The last two terms
in Eq. (8) indicate the influence of neighbor cells. The nonlinear function \( g(x_{ij}) \) satisfy the following

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - a_{00}, & x_{ij} \leq -1 \\
(a_{00} - 1)x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + a_{00}, & x_{ij} \geq 1.
\end{cases}
\]  \hspace{1cm} (9)

Therefore, if \( a_{00} > 1 \), the nonlinear function \( g(x_{ij}) \) has a local minimum: \(-a_{00} + 1\) at \( x_{ij} = -1 \) and a local maximum: \( a_{00} - 1 \) at \( x_{ij} = 1 \) as shown in Fig. 2.

In the case of an isolated CNN cell, Eq. (8) can be written as

\[
\frac{dx_{ij}}{dt} = -x_{ij} + a_{00} f(x_{ij}) + b_{00} u_{ij} + z_{ij} = g(x_{ij}) + w_{ij},
\]  \hspace{1cm} (10)

where \( w_{ij} = b_{00} u_{ij} + z_{ij} \) is called the offset level. Assume that \( a_{00} > 1 \). Then, Eq. (10) has the following steady states:

\[
w_{ij} < -a_{00} + 1 \quad \text{unstable equilibrium point} \quad \frac{w_{ij}}{a_{00} - 1} \quad -a_{00} - w_{ij}
\]  
\[-a_{00} + 1 < w_{ij} < a_{00} - 1 \quad \text{stable equilibrium point} \quad a_{00} - w_{ij}
\]  
\[a_{00} - 1 < w_{ij} \quad \text{stable equilibrium point} \quad a_{00} - w_{ij}
\]  \hspace{1cm} (11)

The role of \( w_{ij} \) is simply to shift up or down the value of the stable equilibrium points (see Fig. 3).

We also have the following relation

\[
w_{ij} \quad x_{ij}(\infty) \quad y_{ij}(\infty)
\]  
\[w_{ij} < -a_{00} + 1 \quad x_{ij}(\infty) = -a_{00} - w_{ij} \quad y_{ij}(\infty) = -1
\]  
\[-a_{00} + 1 < w_{ij} < a_{00} - 1 \quad x_{ij}(\infty) = -a_{00} - w_{ij} \quad y_{ij}(\infty) = -1 \quad \text{if } x_{ij}(0) < 0
\]  
\[x_{ij}(\infty) = a_{00} - w_{ij} \quad y_{ij}(\infty) = 1 \quad \text{if } x_{ij}(0) > 0
\]  \hspace{1cm} (12)

where \( x_{ij}(\infty) = \lim_{t \to \infty} x_{ij}(t) \) and \( y_{ij}(\infty) = \lim_{t \to \infty} y_{ij}(t) \). When coupling is added (that is, in the case of coupled CNN cells), \( w_{ij} \) is no longer time invariant but shift up and down continuously until equilibrium is reached. We will often refer to the equilibrium state analysis as the “driving-point plot analysis” [Chua, 1998].
3. CNN Genes with Same Image Processing Property

The CNN gene specifying a given function is in general not unique. That is, there are CNN genes which have distinct parameters but have the same functions. For example, the following five genes can all be used for figure reconstruction\(^1\) [Roska et al., 1999].

\(^1\)If two binary images \(P_1\) (mask) and \(P_2\) (marker) are given (\(P_2\) contains just a part of \(P_1\) (\(P_2 \subset P_1\)), then the output consists of a binary image showing only those objects of \(P_1\) marked by \(P_2\).
In the following sections, we propose a method for finding CNN parameters for implementing a prescribed function. We can easily derive many CNN genes with same functions by using this method.
4. Basic CNN Genes

Consider the following set of templates:

\[
\begin{align*}
A_0 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & A_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} & A_2 &= \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
A_3 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} & A_4 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
B_0 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & B_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} & B_2 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & b & 1 \\ 0 & 1 & 0 \end{bmatrix} \\
B_3 &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & b & 1 \\ 0 & -1 & 0 \end{bmatrix} & B_4 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & b & 1 \\ 1 & 1 & 1 \end{bmatrix} & B_5 &= \begin{bmatrix} -1 & -1 & -1 \\ -1 & b & -1 \\ -1 & -1 & -1 \end{bmatrix}
\end{align*}
\]

\[z = \begin{bmatrix} c \end{bmatrix}\]

The parameters \(b\) and \(c\) must be tuned to achieve a particular function. In this paper, the template \(A_j\) \((j = 2, 3, 4)\) are called *trapezoidal templates*, since they are related to the trapezoidal rule for approximating an integral. The templates \(B_2\) (for \(b = -4\)) and \(B_4\) (for \(b = -8\)) are called *Laplacian templates*, since they are related to discrete Laplacian operators.

Their corresponding CNN genes are given by

\[
\begin{align*}
\text{A-gene} & : \\
A_0\text{-gene} & : \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
A_1\text{-gene} & : \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \\
A_2\text{-gene} & : \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \end{bmatrix} \\
A_3\text{-gene} & : \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 \end{bmatrix} \\
A_4\text{-gene} & : \begin{bmatrix} 1 & 1 & 1 & 1 & 8 & 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]
Designing CNN Genes

5. Design of CNN Genes

The CNN architecture realizes global tasks by using local operations. In order to design the CNN genes, we have to decompose the global task into local operations. We then realize these local operations as stable states of Eq. (1). Since the local operations are not always equivalent to the global task, we have to examine whether the designed gene can indeed implement the global task. This validation step is generally carried out by extensive numerical simulations. In the subsequent sections, we give examples of CNN designs for the following tasks:

1. Figure reconstruction
2. Concave-filling
3. Hole-filling
4. Filled contour extraction
5. Selected objects extraction
6. Face-vase-illusion
7. Solid black framed area extraction
8. Global connectivity detection
9. Point removal
10. Point extraction
11. Edge detection
12. Convex corner detection
13. Erosion
14. Dilation
15. Right edge detection
16. Shadow projection
17. Horizontal-hole detection
18. Inverse half-toning and half-toning

See the references [Chua, 1998] and [Roska et al., 1999] for details of these tasks. The subsequent subsections are independent from each other, and hence the reader can start from any subsection.
Figure 5.1: Figure reconstruction CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Figure Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>If two binary images ( P_1 ) (mask) and ( P_2 ) (marker) are given (( P_2 ) contains just a part of ( P_1 ) ( P_2 \subset P_1 )), then the output consists of a binary image representing the objects of ( P_1 ) marked by ( P_2 ). The input image is given by ( P_1 ), and the initial state is given by ( P_2 ). The boundaries are constrained to -1.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input ( u_{ij} ) of cell ( C_{ij} ) is coded in white. Then, the output ( y_{ij} ) is printed in white. (b) Assume that the input ( u_{ij} ) is coded in black and the “steady state” (equilibrium) output ( y_{kl} ) of all neighbors are colored in white. Then, the output ( y_{ij} ) is printed in white if the cell is not marked. (c) Assume that the input ( u_{ij} ) is coded in black and the “steady state” (equilibrium) output ( y_{kl} ) of at least one neighbor is colored in black. Then, the output ( y_{ij} ) is printed in black even if the cell is not marked.</td>
</tr>
</tbody>
</table>

### Example

**Design A:**

Choose \( A_4 \)-gene, \( B_1 \)-gene, and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c, \tag{16}
\]

where

\[g(x_{ij}) = \begin{cases} 
-x_{ij} - 8, & x_{ij} \leq -1 \\
7x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 8, & x_{ij} \geq 1.
\end{cases} \tag{17}
\]

Note that \( g(x) \) has a local maximum (7 at \( x = 1 \)), and a local minimum (-7 at \( x = -1 \)).

First, consider the local operation (a). The situation for this operation is given by

\[\text{input : } u_{ij} = -1, \quad \text{initial state : } x_{ij}(0) = -1, \quad \text{neighbors : } y_{kl} = -1 \text{ or } 1 \left( -8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} \leq 8 \right). \tag{18} \]

Hence, we have the following inequality

\[-8 - b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c \leq 8 - b + c. \tag{19}\]

If the inequality

\[8 - b + c < 7 \tag{20}\]

is satisfied and the output \( y_{kl} \) (\( k, l \in N_{ij}, k \neq i, l \neq j \)) is fixed to \(-1\) or \(1\),\(^2\) then Eq. (16) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from the initial state \( x_{ij}(0) = -1 \)

\(^2\)The output \( y_{kl} \) of every cell at any stable equilibrium point is equal to either \(-1\) or \(1\). Throughout this section, \( y_{kl} \) always mean the output of cell \( C_{kl} \) at equilibrium, i.e. \( t \to \infty \).
tends to this equilibrium point. Thus, we get the following relation
\[
\begin{align*}
    u_{ij} &= -1, \\
    x_{ij}(0) &= -1, \\
    y_{kl} &= -1 \text{ or } 1 \quad (\forall k, l \in N_{ij}, k \neq i, l \neq j)
\end{align*}
\]
which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by
\[
\begin{align*}
\text{input : } & \quad u_{ij} = 1, \\
\text{initial state : } & \quad x_{ij}(0) = -1, \\
\text{neighbors : } & \quad y_{kl} = -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\]
Hence, we have
\[
\sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c = -8 + b + c. \tag{23}
\]
If the inequality
\[
-8 + b + c < 7 \tag{24}
\]
is satisfied and the outputs \(y_{kl} (k, l \in N_{ij}, k \neq i, l \neq j)\) are fixed to -1, then Eq. (16) has a stable equilibrium point in the neighborhood of -1. The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
    u_{ij} &= 1, \\
    x_{ij}(0) &= -1, \\
    y_{kl} &= -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)
\end{align*}
\]
\[
\Rightarrow y_{ij} = -1 \tag{25}
\]
which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by
\[
\begin{align*}
\text{input : } & \quad u_{ij} = 1, \\
\text{initial state : } & \quad x_{ij}(0) = -1 \text{ or } 1, \\
\text{neighbors : } & \quad y_{kl} = -1 \text{ or } 1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)
\end{align*}
\]
Hence, we have
\[
\sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c = 8 + b + c. \tag{27}
\]
If the inequality
\[
7 < -6 + b + c \tag{28}
\]
is satisfied and the outputs $y_{kl}$ ($k \neq i$, $l \neq j$) satisfying Eq. (26) are fixed to $-1$ or $1$, then Eq. (16) has a stable equilibrium point in the neighborhood of 1. The trajectory $x_{ij}(t)$ starting from $x_{ij}(0) = \pm 1$ tends to this equilibrium point. Thus, we get the relation

$$u_{ij} = 1, \quad x_{ij}(0) = -1 \text{ or } 1,$$

$$y_{kl} = -1 \text{ or } 1 \left( -6 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} \leq 8 \right) \Rightarrow y_{ij} = 1$$

which indicates the local operation (c).

From Eqs. (20), (24), and (28), we have

$$13 < b + c < 15, \quad 1 < b - c.$$  \hfill (30)

These inequalities are satisfied by choosing $b = 8$, $c = 6$. Thus, we get the CNN gene #2 in Eq. (13):

<table>
<thead>
<tr>
<th>Figure reconstruction CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_4$-gene : 1 1 1 1 1 8 1 1 1</td>
</tr>
<tr>
<td>$B_1$-gene : 0 0 0 0 8 0 0 0 0</td>
</tr>
<tr>
<td>$z$-gene : 6</td>
</tr>
</tbody>
</table>

(31)

or equivalently

<table>
<thead>
<tr>
<th>Figure reconstruction CNN template</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1 = \begin{bmatrix} 1 &amp; 1 &amp; 1 &amp; 1 &amp; 1 &amp; 8 &amp; 1 &amp; 1 &amp; 1 \ 1 &amp; 8 &amp; 1 &amp; 1 &amp; 1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 &amp; 0 &amp; 0 &amp; 0 &amp; 8 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \end{bmatrix}, \quad z = 6$.</td>
</tr>
</tbody>
</table>

(32)

There are a number of parameters which satisfy the inequalities (20)–(28). This implies that there are many CNN genes which have distinct parameters, but have same functions.

**Design B:**

Choose the following set of CNN genes:

$$\begin{cases} A_5\text{-gene} : 1 1 1 1 4 1 1 1 1, \\ B_1\text{-gene} : 0 0 0 0 0 b 0 0 0, \\ z\text{-gene} : c. \end{cases}$$

(33)

In this case, Eq. (8) has the form

$$\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c,$$  \hfill (34)

where

$$g(x_{ij}) = \begin{cases} -x_{ij} - 4, & x_{ij} \leq -1 \\ 3x_{ij}, & |x_{ij}| \leq 1 \\ -x_{ij} + 4, & x_{ij} \geq 1. \end{cases}$$  \hfill (35)

Note that $g(x)$ has a local maximum (3 at $x = 1$), and a local minimum ($-3$ at $x = -1$). Considering the local operations (a)–(c) for the states (18), (22) and (26), we get the following similar set of inequalities:

$$8 - b + c < 3,$$  \hfill (36)

$$-8 + b + c < 3,$$  \hfill (37)

$$3 < -6 + b + c.$$  \hfill (38)

That is, we have

$$9 < b + c < 11, \quad 5 < b - c.$$  \hfill (39)

These inequalities can be satisfied by choosing $b = 8$, $c = 2$. Thus, we get the CNN gene #4 in Eq. (13):

<table>
<thead>
<tr>
<th>Figure reconstruction CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_5$-gene : 1 1 1 1 1 4 1 1 1</td>
</tr>
<tr>
<td>$B_1$-gene : 0 0 0 0 0 8 0 0 0</td>
</tr>
<tr>
<td>$z$-gene : 2</td>
</tr>
</tbody>
</table>

(40)

or equivalently

<table>
<thead>
<tr>
<th>Figure reconstruction CNN template</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_5 = \begin{bmatrix} 1 &amp; 1 &amp; 1 &amp; 1 &amp; 4 &amp; 1 &amp; 1 &amp; 1 &amp; 1 \ 1 &amp; 4 &amp; 1 &amp; 1 &amp; 1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \ 1 &amp; 1 &amp; 1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 &amp; 0 &amp; 0 \ 0 &amp; 8 &amp; 0 \end{bmatrix}, \quad z = 2$.</td>
</tr>
</tbody>
</table>

(41)

---

3 We have only to replace the constant 7 in Eqs. (20), (24) and (28) with 3.
Similarly, we can obtain the remaining genes in Eq. (13).

Observe that the above choice of parameters satisfy only the conditions for local operations listed above. These are necessary but not sufficient conditions. Whether the resulting CNN genes will indeed implement the prescribed global task must be further analyzed by extensive simulations. All of the CNN genes derived in this paper have undergone such empirical validations. Unfortunately, there is in general no analytical method for proving the validity of a derived gene — except for the very special class of uncoupled genes [Chua & Roska, 2002].

5.2. Concave-filling CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Concave-Filling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Fill the convex location of objects in a binary image $P_1$. The input image is given by $P_1$, and the initial state is also given by $P_1$. The boundaries are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in white and the number of neighbors $y_{kl}$ colored in black is greater than or equal to 4. Then, the output $y_{ij}$ is printed in black. (b) Assume that the input $u_{ij}$ is coded in black. Then, the output $y_{ij}$ is printed in black.</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 5</td>
</tr>
<tr>
<td>Reference</td>
<td>[Roska et al., 1999]</td>
</tr>
</tbody>
</table>

Design A:
Choose $A_4$-gene, $B_1$-gene, and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

$$\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l\in N_{ij}, k\neq i, l\neq j} y_{kl} + bu_{ij} + c, \quad (42)$$

where

$$g(x_{ij}) = \begin{cases} -x_{ij} - 8, & x_{ij} \leq -1 \\ 7x_{ij}, & |x_{ij}| \leq 1 \\ -x_{ij} + 8, & x_{ij} \geq 1. \end{cases} \quad (43)$$

Note that $g(x)$ has a local maximum ($7$ at $x = 1$), and a local minimum ($-7$ at $x = -1$).

(1) input image. (2) initial state. (3) output image. Fig. 5. Concave-filling.
First, consider the local operation (a). The situation for this operation is given by

\[
\text{input: } u_{ij} = -1, \\
\text{initial state: } x_{ij}(0) = -1, \\
\text{neighbors: } y_{kl} = -1 \text{ or } 1 \left( 0 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} \leq 8 \right).
\]

(44)

Hence, we have the following inequality

\[
-b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c \leq 8 - b + c.
\]

(45)

If the inequality

\[
7 < -b + c
\]

(46)
is satisfied and the output \( y_{kl} \) satisfying Eq. (44) is fixed to \(-1\) or \(1\), then Eq. (42) has a stable equilibrium point in the neighborhood of \(1\), and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. Thus, we get the following relation

\[
\begin{align*}
   u_{ij} = -1, \\
x_{ij}(0) = -1,
\end{align*}
\]

\[
\begin{align*}
   y_{kl} = -1 \text{ or } 1 \left( 0 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} \leq 8 \right) \Rightarrow y_{ij} = 1
\end{align*}
\]

(47)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

\[
\text{input: } u_{ij} = 1, \\
\text{initial state: } x_{ij}(0) = 1, \\
\text{neighbors: } y_{kl} = -1 \text{ or } 1 \left( 0 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} \leq 8 \right).
\]

(48)

Hence, we have

\[
-8 + b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} y_{kl} + bu_{ij} + c \leq 8 + b + c.
\]

(49)

If the inequality

\[
-7 < -8 + b + c
\]

(50)
is satisfied and the outputs \( y_{kl} \) are fixed to \(-1\), then Eq. (42) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. Thus, we get the relation

\[
\begin{align*}
   u_{ij} = 1, \\
x_{ij}(0) = 1,
\end{align*}
\]

\[
\begin{align*}
   y_{kl} = -1 \text{ or } 1 \left( k, l \in N_{ij}, k \neq i, l \neq j \right) \Rightarrow y_{ij} = 1
\end{align*}
\]

(51)

which indicates the local operation (b).

From Eqs. (46) and (50), we have

\[
7 < c - b, \quad 1 < c + b.
\]

(52)

These inequalities can be satisfied by choosing \(b = 1\), \(c = 9\). Thus, we get the CNN gene:
or equivalently

\[
\begin{align*}
\text{Concave-filling CNN template} \\
A_4 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & z &= 9.
\end{align*}
\]

The inequalities (52) can also be satisfied if we choose \( b = 0, c = 8 \). In this case, we get the following CNN gene:

\[
\begin{align*}
\text{Concave-filling CNN genes} \\
A_4 &\begin{bmatrix} 1 & 1 & 1 & 1 & 8 & 1 & 1 & 1 \end{bmatrix}, & B_1 &\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, & z &\begin{bmatrix} 8 \end{bmatrix}.
\end{align*}
\]

\[\text{(55)}\]

\[\text{(56)}\]

### 5.3. Hole-filling CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Hole-Filling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Fill the interior of all closed contours in a binary image ( P_1 ). The input image is given by ( P_1 ), and the initial state is given by ( x_{ij}(0) = 1 ). The boundaries are constrained to (-1).</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input ( u_{ij} ) of cell ( C_{ij} ) is coded in black. Then, the output ( y_{ij} ) is printed in black. (b) Assume that the input ( u_{ij} ) is coded in white and all neighbors ( y_{kl} ) are colored in black. Then, the output ( y_{ij} ) is printed in black. (c) Assume that the input ( u_{ij} ) is coded in white and at least one neighbor ( y_{kl} ) is colored in white. Then, the output ( y_{ij} ) is printed in white.</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 6</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998]</td>
</tr>
</tbody>
</table>

**Design A:**

Choose \( A_3 \)-gene, \( B_1 \)-gene, and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left( \sum_{k,l \in N_{ij}, (k-i)(l-j)=0,(k,l)\neq(i,j)} y_{kl} \right) + bu_{ij} + c
\]

\[= g(x_{ij}) + \left( \sum' y_{kl} \right) + bu_{ij} + c, \quad (57)\]

where the summation

\[
\sum' = \sum_{k,l \in N_{ij}, (k-i)(l-j)=0,(k,l)\neq(i,j)}
\]

\[\text{(58)}\]
extends only over the 4 compass positions, and
\[ g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1.
\end{cases} \]  
(59)

Note that \( g(x) \) has a local maximum (3 at \( x = 1 \)), and a local minimum (3 at \( x = -1 \)).

First, consider the local operation (a). In this case, the situation is given by

input : \( u_{ij} = 1 \),
initial state : \( x_{ij}(0) = 1 \),
neighbors : \( y_{kl} = -1 \) or \( 1 \) \( (-4 \leq \sum' y_{kl} \leq 4) \).

\[ u_{ij} = 1, \quad x_{ij}(0) = 1, \]
\[ y_{kl} = 1 \) or \(-1, \quad y_{kl} = -1 \) or \( 1 \) \( (-4 \leq \sum' y_{kl} \leq 4) \) \( \Rightarrow y_{ij} = 1 \)  
(63)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

input : \( u_{ij} = -1 \),
initial state : \( x_{ij}(0) = 1 \),
neighbors : \( y_{kl} = 1 \) \( (k, l \in M_{ij}) \),
where
\[ M_{ij} = \{(k, l) | k, l \in N_{ij}, (k - i)(l - j) = 0, \]
\[ (k, l) \neq (i, j) \} \]  
(65)

\[ (k - i, l - j) \in \{(-1, 0), (0, -1), (0, 1), (1, 0)\} \].

Hence, we have
\[ (\sum' y_{kl}) + bu_{ij} + c = 4 - b + c . \]  
(66)

Hence, we have the following inequality
\[ -4 + b + c \leq (\sum' y_{kl}) + bu_{ij} + c \]
\[ \leq 4 + b + c . \]  
(61)

If the inequality
\[ -3 < -4 + b + c \]  
(62)
is satisfied and the output \( y_{kl} \) satisfying Eq. (60) is fixed to \(-1 \) or \( 1 \), then Eq. (57) has a stable equilibrium point in the neighborhood of \( 1 \), and the trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[ u_{ij} = -1, \quad x_{ij}(0) = 1, \]
\[ y_{kl} = 1 \) \( (k, l \in M_{ij}) \) \( \Rightarrow y_{ij} = 1 \)  
(68)

which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by
input: \( u_{ij} = -1 \),

initial state: \( x_{ij}(0) = 1 \),

eighbors: \( y_{kl} = -1 \) or \( 1 \) \( \left( -4 \leq \sum' y_{kl} \leq 2 \right) \).

Hence, we have

\[
-4 - b + c \leq \left( \sum' y_{kl} \right) + b u_{ij} + c \\
\leq 2 - b + c.
\]  

(70)

If the inequality

\[
2 - b + c < -3
\]

(71)
is satisfied and the outputs \( y_{kl} (k, l \in M_{ij}) \) satisfying Eq. (69) are fixed to \(-1\) or \(1\), then Eq. (57) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. We get the relation

\[
\begin{align*}
&u_{ij} = -1, \quad x_{ij}(0) = 1, \\
y_{kl} = 1 \text{ or } -1 \quad \left( -4 \leq \sum' y_{kl} \leq 2 \right) \Rightarrow y_{ij} = -1
\end{align*}
\]

(72)

which indicates the local operation (c).

From Eqs. (67)–(71), we have

\[
-7 < c - b < -5, \quad 1 < b + c.
\]

(73)

These inequalities can be satisfied by choosing \( b = 5, c = -1 \). Thus, we get the following CNN gene:

\[
\begin{align*}
\text{Hole-filling CNN genes} \\
A_3 \text{-gene}: & \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 \text{-gene}: & \begin{bmatrix} 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z \text{-gene}: & -1
\end{align*}
\]

(74)

or equivalently

\[
\begin{align*}
\text{Hole-filling CNN template} \\
A_3 = & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \\
B_1 = & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
z = & -1
\end{align*}
\]

(75)

The inequalities (73) can also be satisfied if we choose \( b = 6, c = 0 \). In this case, we get the following CNN gene:

\[
\begin{align*}
\text{Hole-filling CNN genes} \\
A_3 \text{-gene}: & \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 \text{-gene}: & \begin{bmatrix} 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z \text{-gene}: & 0
\end{align*}
\]

(76)

or equivalently

\[
\begin{align*}
\text{Hole-filling CNN template} \\
A_3 = & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \\
B_1 = & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
z = & 0
\end{align*}
\]

(77)

5.4. **Filled contour extraction CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Filled Contour Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Extract from a binary image ( P_2 ) (marker) fed to initial states, those regions which contain the boundary and which completely filled the interior of all closed curves of a binary image ( P_1 ) loaded to CNN inputs. The input image is given by ( P_1 ), and the initial state is given by ( P_2 ). The boundaries are constrained to (-1).</td>
</tr>
</tbody>
</table>
Local operations

(a) Assume that the input \( u_{ij} \) of cell \( C_{ij} \) is coded in white and all neighbors \( y_{kl} \) are colored in black. Then the output \( y_{ij} \) is printed in black if the cell is marked.

(b) Assume that the input \( u_{ij} \) is coded in black and the number of neighbors \( y_{kl} \) colored in black is greater than or equal to 2. Then, the output \( y_{ij} \) is printed in black if the cell is marked.

(c) Assume that the input \( u_{ij} \) is coded in black and the number of neighbors \( y_{kl} \) colored in black is smaller than 2. Then, the output \( y_{ij} \) is printed in white.

(d) Assume that the input \( u_{ij} \) is coded in white and at least one neighbor \( y_{kl} \) is colored in white. Then, the output \( y_{ij} \) is printed in white.

Example

Fig. 7

Reference

[Chua, 1998]

Design A:

Choose \( A_3 \)-gene, \( B_1 \)-gene, and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left( \sum' y_{kl} \right) + bu_{ij} + c, \quad (78)
\]

where

\[
\sum' = \sum_{k,l \in N_{ij}, (k-i)(l-j) = 0, (k,l) \neq (i,j)} \quad (79)
\]

and

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1 .
\end{cases} \quad (80)
\]

Note that \( g(x) \) has a local maximum (3 at \( x = 1 \)), and a local minimum (−3 at \( x = -1 \)).

First, consider the local operation (a). The situation for this operation is given by

\[
\begin{align*}
\text{input} : & \quad u_{ij} = -1, \\
\text{initial state} : & \quad x_{ij}(0) = 1, \\
\text{neighbors} : & \quad y_{kl} = 1 \quad (k, l \in M_{ij}),
\end{align*} \quad (81)
\]

where

\[
M_{ij} = \{(k, l)|k, l \in N_{ij}, (k-i)(l-j) = 0, (k, l) \neq (i, j) \}, \quad (82)
\]

\[
(k-i, l-j) \in \{(-1, 0), (0, -1), (0, 1), (1, 0)\}. \quad (82)
\]

Hence, we have the following equation

\[
\left( \sum' y_{kl} \right) + bu_{ij} + c = 4 - b + c. \quad (83)
\]

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{filled_contour.png}
\caption{Filled contour extraction.}
\end{figure}
If the inequality
\[ -3 < 4 - b + c \]  
(84)
is satisfied and the outputs \( y_{kl} (k, l \in M_{ij}) \) are fixed to \(-1\) or \(1\), then Eq. (78) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[ u_{ij} = -1, \; x_{ij}(0) = 1, \; y_{kl} = 1 \quad (k, l \in M_{ij}) \]  
(85)
which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by
\[ \text{input : } u_{ij} = 1, \]
\[ \text{initial state : } x_{ij}(0) = 1, \]
\[ \text{neighbors : } y_{kl} = -1 \text{ or } 1 \quad (0 \leq \sum' y_{kl} \leq 2). \]  
(86)
Hence, we have
\[ b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \leq 2 + b + c. \]  
(87)
If the inequality
\[ -3 < b + c \]  
(88)
is satisfied and the outputs \( y_{kl} (k, l \in M_{ij}) \) are fixed to \(-1\) or \(1\), then Eq. (78) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[ u_{ij} = -1, \; x_{ij}(0) = 1, \; y_{kl} = -1 \text{ or } 1 \quad (0 \leq \sum' y_{kl} \leq 2) \]  
(89)
which indicates the local operation (b).

Next, consider the local operation (c). The situation for this operation is given by
\[ \text{input : } u_{ij} = 1, \]
\[ \text{initial state : } x_{ij}(0) = -1 \text{ or } 1, \]
\[ \text{neighbors : } y_{kl} = -1 \text{ or } 1 \quad (-4 \leq \sum' y_{kl} \leq 2). \]  
(90)
Hence, we have
\[ -4 + b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \]
\[ \leq -2 + b + c. \]  
(91)
If the inequality
\[ -2 + b + c < -3 \]  
(92)
is satisfied and the outputs \( y_{kl} (k, l \in M_{ij}) \) are fixed to \(-1\) or \(1\), then Eq. (78) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[ u_{ij} = 1, \; x_{ij}(0) = 1, \; y_{kl} = -1 \quad (0 \leq \sum' y_{kl} \leq 2) \]  
(93)
which indicates the local operation (c).

Finally, consider the local operation (d). The situation for this operation is given by
\[ \text{input : } u_{ij} = -1, \]
\[ \text{initial state : } x_{ij}(0) = -1 \text{ or } 1, \]
\[ \text{neighbors : } y_{kl} = -1 \text{ or } 1 \quad (-4 \leq \sum' y_{kl} \leq 2). \]  
(94)
Hence, we have
\[ -4 - b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \leq 2 - b + c. \]  
(95)
If the inequality
\[ 2 - b + c < -3 \]  
(96)
is satisfied and the outputs \( y_{kl} (k, l \in M_{ij}) \) satisfying Eq. (90) are fixed to \(-1\) or \(1\), then Eq. (78) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = \pm 1 \) tends to this equilibrium point. We get the relation
\[ u_{ij} = -1, \; x_{ij}(0) = -1 \text{ or } 1, \; y_{kl} = 1 \text{ or } -1 \quad (-4 \leq \sum' y_{kl} \leq 2) \]  
(97)
which indicates the local operation (d).

From Eqs. (88)–(96), we have
\[ -7 < c - b < -5, \quad -3 < b + c < -1. \]  
(98)
These inequalities can be satisfied by choosing \(b = 2, \; c = -4\). Thus we get the CNN gene:

\begin{align}
\text{Filled contour extraction CNN genes} \\
A_3\text{-gene :} & : 0 \; 1 \; 0 \; 1 \; 4 \; 1 \; 0 \; 1 \; 0, \\
B_1\text{-gene :} & : 0 \; 0 \; 0 \; 0 \; 2 \; 0 \; 0 \; 0 \; 0, \\
z\text{-gene :} & : -4, 
\end{align}  
(99)
or equivalently

\[
\begin{align*}
A_4 &= \begin{bmatrix}
0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}, & B_1 &= \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, & z &= -4. \\
\end{align*}
\]  

(100)

The inequalities (98) can also be satisfied if we choose \(b = 2, c = -4.5\). In this case, we get the following CNN gene:

\[
\begin{align*}
A_3 &= \begin{bmatrix}
0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}, & B_1 &= \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, & z &= -4.5. \\
\end{align*}
\]  

(102)

5.5. Selected objects extraction CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Selected Objects Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Extract objects marked by a binary image (P_2) from a binary image (P_1). The input image is given by (P_1), and the initial state is given by (P_2) (marker). The boundaries are constrained to (-1).</td>
</tr>
</tbody>
</table>
| Local operations      | (a) Assume that the input \(u_{ij}\) of cell \(C_{ij}\) is coded in black. Then, the output \(y_{ij}\) for the marked cell is printed in black.  
(b) Assume that the input \(u_{ij}\) is coded in black and at least one neighbor \(y_{kl}\) is colored in black. Then, the output \(y_{ij}\) is printed in black.  
(c) Assume that the input \(u_{ij}\) is coded in black and all neighbors \(y_{kl}\) are colored in white. Then, the output \(y_{ij}\) is printed in white if the cell is not marked.  
(d) Assume that the input \(u_{ij}\) is coded in white. Then, the output \(y_{ij}\) is printed in white. |
| Example               | Fig. 8                                                                                       |
| Reference             | [Chua, 1998]                                                                                 |

**Design A:**

Choose \(A_3\)-gene, \(B_1\)-gene and \(z\)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left( \sum' y_{kl} \right) + bu_{ij} + c, 
\]  

(103)

where

\[
\sum' = \sum_{k,l \in N_{ij}, (k-l)(l-j)=0, (k,l)\neq(i,j)} y_{kl} 
\]  

(104)
and
\[ g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1.
\end{cases} \] \hspace{1cm} (105)

Note that \( g(x) \) has a local maximum (3 at \( x = 1 \)), and a local minimum (\(-3 \) at \( x = -1 \)).

First, consider the local operation (a). The situation for this operation is given by
\[
\begin{align*}
\text{input} & : \quad u_{ij} = 1, \\
\text{initial state} & : \quad x_{ij}(0) = 1, \\
\text{neighbors} & : \quad y_{kl} = -1 \text{ or } 1 \quad \left(-4 \leq \sum' y_{kl} \leq 4\right).
\end{align*}
\] \hspace{1cm} (106)

Hence, we have the following relation
\[
-4 + b + c \leq \left(\sum' y_{kl}\right) + bu_{ij} + c \\
\leq 4 + b + c. \quad (107)
\]

If the inequality
\[
-3 < -4 + b + c \quad (108)
\]
is satisfied and the output \( y_{kl} \) satisfying Eq. (106) is fixed to \(-1 \) or \(1 \), then Eq. (103) has a stable equilibrium point in the neighborhood of \( 1 \). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = \pm 1 \) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
\text{input} & : \quad u_{ij} = 1, \\
\text{initial state} & : \quad x_{ij}(0) = -1 \text{ or } 1, \\
\text{neighbors} & : \quad y_{kl} = -1 \text{ or } 1 \quad \left(-2 \leq \sum' y_{kl} \leq 4\right). \\
\Rightarrow \quad y_{ij} & = 1
\end{align*}
\] \hspace{1cm} (113)

which indicates the local operation (b).

Next, consider the local operation (c). The situation for this operation is given by
\[
\begin{align*}
\text{input} & : \quad u_{ij} = 1, \\
\text{initial state} & : \quad x_{ij}(0) = -1, \\
\text{neighbors} & : \quad y_{kl} = -1 \quad (k, l \in M_{ij}),
\end{align*}
\] \hspace{1cm} (114)

where
\[
M_{ij} = \{(k, l)|k, l \in N_{ij}, (k - i)(l - j) = 0, \quad (k, l) \neq (i, j)\} \quad (115)
\]

\( (k - i, l - j) \in \{(-1, 0), (0, -1), (0, 1), (1, 0)\} \).
Hence, we have
\[
\left( \sum' y_{kl} \right) + bu_{ij} + c = -4 + b + c. \tag{116}
\]
If the inequality
\[
-4 + b + c < 3 \tag{117}
\]
is satisfied and the outputs \( y_{kl} \) (\( k, l \in M_{ij} \)) are fixed to \(-1\), then Eq. (103) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = \pm 1 \) tends to this equilibrium point. Thus, we get the relation
\[
u_{ij} = 1, \quad x_{ij}(0) = -1, \quad y_{kl} = -1 \quad (k, l \in M_{ij}) \implies y_{ij} = 1 \tag{118}
\]
which indicates the local operation (c).

Finally, consider the local operation (d). The situation for this operation is given by
\[
\begin{align*}
\text{input:} & \quad u_{ij} = -1, \\
\text{initial state:} & \quad x_{ij}(0) = -1 \text{ or } 1, \\
\text{neighbors:} & \quad y_{kl} = -1 \text{ or } 1 \quad (-4 \leq \sum' y_{kl} \leq 4).
\end{align*} \tag{119}
\]
Hence, we have
\[
-4 - b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \\
\leq 4 - b + c. \tag{120}
\]
If the inequality
\[
4 - b + c < -3 \tag{121}
\]
is satisfied and the output \( y_{kl} \) satisfying Eq. (119) is fixed to \(-1\) or \(1\), then Eq. (103) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = \pm 1 \) tends to this equilibrium point. We get the relation
\[
u_{ij} = -1, \quad x_{ij}(0) = -1 \text{ or } 1, \quad y_{kl} = 1 \text{ or } -1 \quad \left(-4 \leq \sum' y_{kl} \leq 4\right) \implies y_{ij} = -1 \tag{122}
\]
which indicates the local operation (d).

From Eqs. (112)–(117), we have
\[
c - b < -7, \quad 5 < b + c < 7. \tag{123}
\]
These inequalities can be satisfied by choosing \(b = 7, \ c = -1\). Thus, we get the following CNN gene:

\[
\begin{align*}
\text{Selected objects extraction CNN genes}
A_3 \text{-gene:} & \quad \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1
\end{array} \\
B_1 \text{-gene:} & \quad \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 7 & 0 & 0 & 0
\end{array} \\
z \text{-gene:} & \quad -1
\end{align*} \tag{124}
\]
or equivalently
\[
\begin{align*}
\text{Selected objects extraction CNN template}
A_3 = & \quad \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1
\end{array} \\
B_1 = & \quad \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 7 & 0 & 0 & 0
\end{array} \\
z = & \quad -1
\end{align*} \tag{125}
\]
The inequalities (123) can also be satisfied if we choose \(b = 7.5, \ c = -1.5\). In this case, we get the following CNN gene:

\[
\begin{align*}
\text{Selected objects extraction CNN genes}
A_3 \text{-gene:} & \quad \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1
\end{array} \\
B_1 \text{-gene:} & \quad \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 7.5 & 0 & 0 & 0
\end{array} \\
z \text{-gene:} & \quad -1.5
\end{align*} \tag{126}
\]
or equivalently
\[
\begin{align*}
\text{Selected objects extraction CNN template}
A_3 = & \quad \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1
\end{array} \\
B_1 = & \quad \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 7.5 & 0 & 0 & 0
\end{array} \\
z = & \quad -1.5
\end{align*} \tag{127}
\]
5.6. Face-vase-illusion CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Face-Vase-Illusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Simulate the well-known visual illusion where the input image $P_1$ is perceived either as two symmetric faces, or as a vase, depending on the initial thought or attention, which is simulated by specifying a small patch of black pixels inside the object to be picked out (marked by a binary image $P_2$). The input image is given by $P_1$, and the initial state is given by $P_2$. The boundaries are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in black. Then, the output $y_{ij}$ is printed in white. (The boundary of the input image is colored in white.) (b) Assume that the input $u_{ij}$ is coded in white and at least one neighbor $y_{kl}$ is colored in black. Then, the output $y_{ij}$ is printed in black. (c) Assume that the input $u_{ij}$ is coded in white. Then, the output $y_{ij}$ for a marked cell is printed in black. (d) Assume that the input $u_{ij}$ is coded in white and all neighbors $y_{kl}$ are colored in white. Then, the output $y_{ij}$ is printed in white.</td>
</tr>
</tbody>
</table>

Example: Fig. 9

Reference: [Chua, 1998]

Design A:
Choose $A_3$-gene, $B_1$-gene and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

$$
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left( \sum_{k,l \in N_{ij}, (k-i)(l-j)=0, (k,l) \neq (i,j)} y_{kl} \right) + bu_{ij} + c, \quad (128)
$$

where

$$
\sum' = \sum_{k,l \in N_{ij}, (k-i)(l-j)=0, (k,l) \neq (i,j)} y_{kl} \quad (129)
$$

$$
g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1 .
\end{cases} \quad (130)
$$

Note that $g(x)$ has a local maximum (3 at $x = 1$), and a local minimum ($-3$ at $x = -1$).

First, consider the local operation (a). In this case, the situation is given by

- input: $u_{ij} = 1$, 
- initial state: $x_{ij}(0) = -1$ or $1$, 
- neighbors: $y_{kl} = -1$ or $1 \quad ( -4 \leq \sum' y_{kl} \leq 2) .

Hence, we have the following relation

$$
-4 + b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c 
\leq 2 + b + c .\quad (132)
$$

Assume that the inequality

$$
2 + b + c < -3 \quad (133)
$$

is satisfied and the output $y_{kl}$ satisfying Eq. (131) is fixed to $-1$ or $1$. Then, Eq. (128) has a stable equilibrium point in the neighborhood of $-1$, and the trajectory $x_{ij}(t)$ starting from $x_{ij}(0) = \pm 1$ tends to this equilibrium point. Thus, we get the relation

$$
u_{ij} = -1, \quad x_{ij}(0) = -1 \text{ or } 1, \quad y_{kl} = -1 \text{ or } 1 \quad ( -4 \leq \sum' y_{kl} \leq 2) \Rightarrow y_{ij} = -1 \quad (134)
$$

which indicates the local operation (a).

Next, consider the local operation (b). In this case, the cell situation is given by

- input: $u_{ij} = -1$, 
- initial state: $x_{ij}(0) = -1$ or $1$, 
- neighbors: $y_{kl} = -1$ or $1 \quad ( -4 \leq \sum' y_{kl} \leq 2) .

(135)
neighbors : \( y_{kl} = -1 \) or 1 \((-4 \leq \sum' y_{kl} \leq 2\).

Hence, we have
\[
-2 - b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \\
\leq 4 - b + c. 
\]

If the inequality
\[
3 < -2 - b + c
\]
(136)
is satisfied and the output \( y_{kl} \) is fixed to \(-1\) or 1, then Eq. (128) has a stable equilibrium point in the neighborhood of 1. The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = \pm 1 \) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
    u_{ij} = -1, \\
    x_{ij}(0) = -1 \text{ or } 1, \\
    y_{kl} = -1 \text{ or } 1 \left( -4 \leq \sum' y_{kl} \leq 2 \right) \Rightarrow y_{ij} = 1
\end{align*}
\]
(138)
which indicates the local operation (b).

Next, consider the local operation (c). In this case, the cell situation is given by
\[
\begin{align*}
    \text{input} & : u_{ij} = -1, \\
    \text{initial state} & : x_{ij}(0) = 1, \\
    \text{neighbors} & : y_{kl} = -1 \text{ or } 1 \left( -4 \leq \sum' y_{kl} \leq 4 \right).
\end{align*}
\]
(139)

Hence, we have
\[
-4 - b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \\
\leq 4 - b + c. 
\]
(140)

If the inequality
\[
-3 < -4 - b + c
\]
(141)
is satisfied and the output \( y_{kl} \) satisfying Eq. (139) is fixed to \(-1\) or 1, then Eq. (128) has a stable equilibrium point in the neighborhood of 1. The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. We get the relation
\[
\begin{align*}
    u_{ij} = -1, \\
    x_{ij}(0) = 1, \\
    y_{kl} = 1 \text{ or } -1 \left( -4 \sum' y_{kl} \leq 4 \right) \Rightarrow y_{ij} = 1
\end{align*}
\]
(142)
which indicates the local operation (c).
Next, consider the local operation (d). In this case, the situation is given by

\[
\begin{align*}
\text{input} & : \ u_{ij} = -1, \\
\text{initial state} & : \ x_{ij}(0) = -1, \\
\text{neighbors} & : \ y_{kl} = -1 \quad (k, l \in M_{ij}).
\end{align*}
\]

where

\[
M_{ij} = \{(k, l)|k, l \in N_{ij}, (k - i)(l - j) = 0, (k, l) \neq (i, j)\}
\]

\[
(k - i, l - j) \in \{(-1, 0), (0, -1), (0, 1), (1, 0)\}.
\]

Hence, we have the following equation

\[
(\sum' y_{kl}) + bu_{ij} + c = -4 - b + c.
\]

If the inequality

\[
-4 - b + c < 3
\]

is satisfied and the output \(y_{kl}\) satisfying Eq. (143) is fixed to \(-1\) or \(1\), then Eq. (128) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. We get the relation

\[
\begin{align*}
\{ u_{ij} = -1, \quad x_{ij}(0) = -1, \\
y_{kl} = 1 \text{ or } -1, \quad (k - i)(l - j) = 0 \}
\Rightarrow y_{ij} = 1
\end{align*}
\]

which indicates the local operation (d). From Eqs. (137)–(146), we have

\[
5 < c - b < 7, \quad b + c < -5.
\]

These inequalities can be satisfied by choosing \(b = -6, c = 0\). Thus we get the following CNN gene:

Face-vase illusion CNN genes

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z & = \begin{bmatrix} 0 \end{bmatrix}.
\end{align*}
\]

or equivalently

Face-vase illusion CNN template

\[
\begin{align*}
A_3 = & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \\
B_1 = & \begin{bmatrix} 0 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
z = & \begin{bmatrix} 0 \end{bmatrix}.
\end{align*}
\]

The inequalities (148) can also be satisfied if we choose \(b = -7, c = -1\). In this case, we get the following CNN gene:

Face-vase illusion CNN genes

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 & 0 & -7 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z & = \begin{bmatrix} -1 \end{bmatrix}.
\end{align*}
\]

or equivalently

Face-vase illusion CNN template

\[
\begin{align*}
A_3 = & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \\
B_1 = & \begin{bmatrix} 0 & 0 & 0 \\ 0 & -7 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
z = & \begin{bmatrix} -1 \end{bmatrix}.
\end{align*}
\]

5.7. **Solid black framed area extraction CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Solid Black Framed Area Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global task</strong></td>
<td>Extract binary images representing objects of (P_2) which fit and fill in closed curves of a given binary image (P_1). The CNN input image is given by (P_1), and the initial state is given by (P_2) (marker). The boundaries are constrained to (-1).</td>
</tr>
</tbody>
</table>
Local operations

(a) Assume that the number of neighbors \( y_{kl} \) colored in black is greater than or equal to 2. Then the output \( y_{ij} \) of the marked cell is printed in black.

(b) Assume that the input \( u_{ij} \) is coded in black and the number of neighbors \( y_{kl} \) colored in black is smaller than 2. Then, the output \( y_{ij} \) of the marked cell is printed in white.

(c) Assume that the input \( u_{ij} \) is coded in white. Then, the output \( y_{ij} \) which is not marked by \( P_2 \) is printed in white.

Example

Fig. 10

Reference

[Roska et al., 1999]

Design A:

Choose \( A_4 \)-gene, \( B_1 \)-gene and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left( \sum' y_{kl} \right) + bu_{ij} + c,
\]

where

\[
\sum' = \sum_{k,l \in N_{ij}, (k-i)(l-j)=0, (k,l) \neq (i,j)}
\]

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 8, & x_{ij} \leq -1 \\
7x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 8, & x_{ij} \geq 1.
\end{cases}
\]

Note that \( g(x) \) has a local maximum (7 at \( x = 1 \)), and a local minimum (−7 at \( x = -1 \)).

First, consider the local operation (a). The situation for this operation is given by

input : \( u_{ij} = -1 \) or 1,

initial state : \( x_{ij}(0) = 1 \),

neighbors : \( y_{ij} = -1 \) or 1 \( \left( 0 \leq \sum' y_{kl} \leq 4 \right) \).

Hence, we have the following inequality

\[
b + c \leq \left( \sum' y_{kl} \right) + bu_{ij} + c \leq 4 + b + c.
\]

If the inequality

\[-3 < b + c, \quad -3 < -b + c,\]

(158)
is satisfied and the output \( y_{kl} \) satisfying Eq. (156) is fixed to \(-1\) or \(1\), then Eq. (153) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \) starting from the initial state \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the following relation

\[
\begin{align*}
  u_{ij} = & -1 \text{ or } 1, \quad x_{ij}(0) = 1, \\
y_{kl} = & -1 \text{ or } 1 \quad \left(0 \leq \sum' y_{kl} \leq 4\right) \implies y_{ij} = -1
\end{align*}
\]  

(159)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

\[
\begin{align*}
  \text{input} : & \quad u_{ij} = 1, \\
  \text{initial state} : & \quad x_{ij}(0) = 1, \\
  \text{neighbors} : & \quad y_{kl} = -1 \text{ or } 1 \quad \left(-4 \leq \sum' y_{kl} \leq -2\right).
\end{align*}
\]

(160)

Hence, we have

\[
-4 + b + c \leq \left(\sum' y_{kl}\right) + bu_{ij} + c \\
\leq -2 + b + c.
\]  

(161)

If the inequality

\[
-2 + b + c < -3
\]  

(162)
is satisfied and the output \( y_{kl} \) satisfying Eq. (160) is fixed to \(-1\) or \(1\), then Eq. (153) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation

\[
\begin{align*}
  u_{ij} & = 1, \quad x_{ij}(0) = 1, \\
y_{kl} & = -1 \text{ or } 1 \quad \left(-4 \leq \sum' y_{kl} \leq -2\right) \implies y_{ij} = -1
\end{align*}
\]  

(163)

which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by

\[
\begin{align*}
  \text{input} : & \quad u_{ij} = -1, \\
  \text{initial state} : & \quad x_{ij}(0) = -1, \\
  \text{neighbors} : & \quad y_{kl} = -1 \text{ or } 1 \quad \left(-4 \leq \sum' y_{kl} \leq 4\right).
\end{align*}
\]

(164)

Hence, we have

\[
-4 - b + c \leq \left(\sum' y_{kl}\right) + bu_{ij} + c \\
\leq 4 - b + c.
\]  

(165)

If the inequality

\[
4 - b + c < 3
\]  

(166)
is satisfied and the output \( y_{kl} \) satisfying Eq. (164) is fixed to \(-1\) or \(1\), then Eq. (153) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = -1 \) tends to this equilibrium point. Thus, we get the relation

\[
\begin{align*}
  u_{ij} & = -1, \quad x_{ij}(0) = -1, \\
y_{kl} & = -1 \text{ or } 1 \quad \left(-4 \leq \sum' y_{kl} \leq 4\right) \implies y_{ij} = -1
\end{align*}
\]

(167)

which indicates the local operation (c).

From Eqs. (158), (162), and (166), we have

\[-3 < b + c < -1, \quad -3 < -b + c < -1.\]

(168)

These inequalities can be satisfied by choosing \( b = 0, c = -2 \). Thus, we get the following CNN genes:

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z & = -2.
\end{align*}
\]  

(169)

or equivalently

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0 \end{bmatrix}, \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & 0 \end{bmatrix}, \\
z & = -2.
\end{align*}
\]  

(170)

The inequalities (168) can also be satisfied if we choose \( b = -0.4 \) and \( c = -2 \). In this case, we have the following genes:

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 & -0.4 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
z & = -2.
\end{align*}
\]  

(171)
or equivalently

Solid black framed area extraction CNN template

\[
A_3 = \begin{bmatrix}
  0 & 1 & 0 \\
  1 & 4 & 1 \\
  0 & 1 & 0
\end{bmatrix}, \quad
B_1 = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & -0.4 & 0 \\
  0 & 0 & 0
\end{bmatrix}, \quad
z = -2.
\]

(172)

5.8. **Global connectivity detection CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Global Connectivity Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Determine whether a given geometric pattern ( P_1 ) is globally connected in one contiguous piece, or is it composed of two or more disconnected components. An object or component is any connected set in black color. Delete marked objects in binary image. <strong>An object is marked by changing at least one black pixel into white in the initial state</strong> ( (P_2) ). The output contains unmarked objects only. The input image is given by ( P_1 ), and the initial state is given by ( P_2 ). The boundaries are constrained to (-1).</td>
</tr>
<tr>
<td>Local operations</td>
<td></td>
</tr>
<tr>
<td>(a) Assume that the input ( u_{ij} ) of cell ( C_{ij} ) is coded in white. Then, the output ( y_{ij} ) is printed in white.</td>
<td></td>
</tr>
<tr>
<td>(b) Assume that the input ( u_{ij} ) is coded in black, and the neighbors ( y_{kl} ) satisfy the condition</td>
<td></td>
</tr>
<tr>
<td>[</td>
<td>y_{kl}</td>
</tr>
<tr>
<td>That is, the number of neighbors ( y_{kl} ) which are colored in black is equal to the ones which are coded in black. Then, the output ( y_{ij} ) is printed in black if the cell is marked, and ( y_{ij} ) is printed white if the cell is not marked.</td>
<td></td>
</tr>
<tr>
<td>(c) Assume that the input ( u_{ij} ) is coded in black, and the neighbors ( y_{kl} ) satisfy the condition</td>
<td></td>
</tr>
<tr>
<td>[</td>
<td>y_{kl}</td>
</tr>
<tr>
<td>That is, there is at least one neighbor which is coded in black and colored in white. Then, the output ( y_{ij} ) is printed in white if the cell is marked.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 11</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998]</td>
</tr>
</tbody>
</table>

**Design A:**

Choose \( A_3 \)-gene, \( B_2 \)-gene and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \left[ \sum'(y_{kl} - b_{kl}) \right] + bu_{ij} + c,
\]

(173)
where
\[
\sum' = \sum_{k,l\in\mathcal{N}_{ij}, (k-i)(l-j)=0, (k,l)\neq(i,j)}
\] (174)
and
\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1.
\end{cases}
\] (175)

Note that \(g(x)\) has a local maximum (3 at \(x = 1\)), and a local minimum (\(-3\) at \(x = -1\)).

First, consider the local operation (a). The situation for this operation is given by

- input : \(u_{ij} = -1\),
- initial state : \(x_{ij}(0) = -1\) or \(1\),
- neighbors : \(u_{kl} = -1\) or \(1\) \(\left(-4 \leq \sum' u_{kl} \leq 4\right)\),
- \(y_{kl} = -1\) or \(1\) \(\left(-4 \leq \sum' y_{kl} \leq 4\right)\). (176)

Hence, we have the following equation
\[-8 - b + c \leq \sum'(y_{kl} - b_{kl}) + bu_{ij} + c \leq 8 - b + c.\] (177)

If the inequality
\[8 - b + c < -3\] (178)
is satisfied and the outputs \(y_{kl}\) are fixed to \(-1\), then Eq. (173) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = \pm 1\) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
\text{input} : & \quad u_{ij} = 1, \\
\text{initial state} : & \quad x_{ij}(0) = -1 \text{ or } 1, \\
\text{neighbors} : & \quad u_{kl} = -1 \text{ or } 1 \left(-4 \leq \sum' u_{kl} \leq 4\right), \\
& \quad y_{kl} = -1 \text{ or } 1 \left(-4 \leq \sum' y_{kl} \leq 4\right).
\end{align*}
\] (179)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

- input : \(u_{ij} = 1\),
- initial state : \(x_{ij}(0) = 1\) or \(-1\),
- neighbors : \(u_{kl} = -1\) or \(1\) \(\left(-4 \leq \sum' u_{kl} \leq 4\right)\),
- \(y_{kl} = -1\) or \(1\) \(\left(-4 \leq \sum' y_{kl} \leq 4\right)\). (180)

Hence, we have
\[
\sum'(y_{kl} - u_{kl}) + bu_{ij} + c = b + c.\] (181)
If the inequality
\[-3 < b + c < 3\] (182)
is satisfied and the output \(y_{kl}\) satisfying Eq. (180) is fixed to \(-1\) or \(1\), then Eq. (173) has stable equilibrium points \(x_m\) and \(x_p\) in the neighborhood of \(-1\) and \(1\), respectively. The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) (resp. \(x_{ij}(0) = 1\)) tends to \(x_m\).
(resp. \( x_p \)). Thus, we get the relation
\[
\begin{align*}
  u_{ij} &= 1, \\
  x_{ij}(0) &= 1, \\
  u_{kl}, y_{kl} &= -1 \text{ or } 1 \quad \left( \sum' (y_{kl} - u_{kl}) = 0 \right) \\
\end{align*}
\]
\Rightarrow y_{ij} = 1 \quad (183)

and
\[
\begin{align*}
  u_{ij} &= 1, \\
  x_{ij}(0) &= -1, \\
  u_{kl}, y_{kl} &= -1 \text{ or } 1 \quad \left( \sum' (y_{kl} - u_{kl}) = 0 \right) \\
\end{align*}
\]
\Rightarrow y_{ij} = -1 \quad (184)

which indicate the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by
\[
\begin{align*}
  \text{input} : & \quad u_{ij} = 1, \\
  \text{initial state} : & \quad x_{ij}(0) = 1, \\
  \text{neighbors} : & \quad y_{kl} = -1 \text{ or } 1 \quad \left( \sum' (y_{kl} - u_{kl}) \leq -2 \right). \\
\end{align*}
\]
Hence, we have
\[
\sum' (y_{kl} - u_{kl}) + bu_{ij} + c \leq -2 + b + c. \quad (186)
\]
If the inequality
\[
-2 + b + c < -3 \quad (187)
\]
is satisfied and the output \( y_{kl} \) satisfying Eq. (185) is fixed to \(-1\) or \(1\), then Eq. (173) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
  u_{ij} &= 1, \\
  x_{ij}(0) &= 1, \\
  y_{kl} &= -1 \text{ or } 1 \quad \left( 0 \leq \sum' y_{kl} \leq -2 \right) \\
\end{align*}
\]
\Rightarrow y_{ij} = -1 \quad (188)
which indicates the local operation (c).

From Eqs. (182)–(187), we have
\[
c - b < -11, \quad -3 < b + c < -1. \quad (189)
\]
These inequalities can be satisfied by choosing \( b = 6, c = -8 \). Thus we get the gene:

Global connectivity detection CNN genes
\[
\begin{align*}
  A_3 &- \text{gene} : \begin{array}{cccccccc}
    0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\
  \end{array} \\
  B_1 &- \text{gene} : \begin{array}{cccccccc}
    0 & -1 & 0 & -1 & 6 & -1 & 0 & -1 & 0 \\
  \end{array} \\
  z &- \text{gene} : \begin{array}{c}
    -8 \\
  \end{array}
\end{align*}
\] (190)

or equivalently

Global connectivity detection CNN template
\[
\begin{align*}
  A_3 &= \begin{array}{ccc}
    0 & 1 & 0 \\
    1 & 4 & 1 \\
    0 & 1 & 0 \\
  \end{array}, \quad
  B_1 &= \begin{array}{ccc}
    0 & -1 & 0 \\
    -1 & 6 & -1 \\
    0 & -1 & 0 \\
  \end{array}, \quad
  z &= -8. \quad (191)
\end{align*}
\]
The inequalities (189) can also be satisfied if we choose $b = 5, c = -7$. In this case, we get the following CNN gene:

\[
\begin{align*}
A_3\text{-gene:} & \quad \begin{bmatrix} 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \end{bmatrix}, \\
B_1\text{-gene:} & \quad \begin{bmatrix} 0 & -1 & 0 & -1 & 5 & -1 & 0 & -1 & 0 \end{bmatrix}, \\
z\text{-gene:} & \quad -7.
\end{align*}
\]

or equivalently

\[
\begin{align*}
A_3 & = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, &
B_1 & = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, &
z & = -7.
\end{align*}
\]

5.9. **Point removal CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Point Removal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global task</strong></td>
<td>Delete all isolated black pixels in a binary image $P_1$. The CNN input image is given by $P_1$, and the initial state is also given by $P_1$. The boundaries are constrained to $-1$.</td>
</tr>
</tbody>
</table>
| **Local operations** | (a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in black and the inputs of all neighbors $y_{kl}$ are coded in white. Then, the output $y_{ij}$ is printed in white.  
(b) Assume that the input $u_{ij}$ is coded in black and there is at least one neighbor whose input is coded in black. Then, the output $y_{ij}$ is printed in black.  
(c) Assume that the input $u_{ij}$ is coded in white. Then, the output $y_{ij}$ is printed in white. |
| **Example** | Fig. 12 |
| **Reference** | [Chua, 1998] |

**Design A:**
Choose $A_1$-gene, $B_4$-gene and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c,
\]

where

\[
g(x_{ij}) = \begin{cases} -x_{ij} - 2, & x_{ij} \leq -1 \\ x_{ij}, & |x_{ij}| \leq 1 \\ -x_{ij} + 2, & x_{ij} \geq 1. \end{cases}
\]

Note that $g(x)$ has a local maximum (1 at $x = 1$), and a local minimum ($-1$ at $x = -1$). First,
consider the local operation (a). The situation for this operation is given by

\[
\begin{align*}
\text{input} &: \quad u_{ij} = 1, \\
\text{initial state} &: \quad x_{ij}(0) = 1, \\
\text{inputs of neighbors} &: \quad u_{kl} = -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\] (196)

Hence, we have the following equation

\[
\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c = -8 + b + c. \quad (197)
\]

If the inequality

\[
-8 + b + c < -1 \quad (198)
\]
is satisfied, then Eq. (164) has a stable equilibrium point in the neighborhood of -1, and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. Thus, we get the relation

\[
\begin{align*}
\begin{cases}
&u_{ij} = 1, \quad x_{ij}(0) = 1, \\
&u_{kl} = -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)
\end{cases} \Rightarrow y_{ij} = -1
\end{align*} \quad (199)
\]

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

\[
\begin{align*}
\text{input} &: \quad u_{ij} = 1, \\
\text{initial state} &: \quad x_{ij}(0) = 1, \\
\text{inputs of neighbors} &: \quad u_{kl} = 1 \text{ or } -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\] (200)

Hence, we have

\[
-6 + b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 8 + b + c. \quad (201)
\]

If the inequality

\[
-1 < -6 + b + c \quad (202)
\]
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of 1. The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. We get the relation

\[
\begin{align*}
\begin{cases}
&u_{ij} = 1, \quad x_{ij}(0) = 1, \\
&u_{kl} = -1 \text{ or } 1 \quad \left(-6 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8\right)
\end{cases} \Rightarrow y_{ij} = 1
\end{align*} \quad (203)
\]

which indicates the local operation (b).
Finally, consider the local operation (c). The situation for this operation is given by

input: \( u_{ij} = -1 \),

initial state: \( x_{ij}(0) = -1 \),

inputs of neighbors: \( y_{ij} = -1 \) or 1

\[
\begin{align*}
-8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} & \leq 8 \\
\end{align*}
\]

Hence, we have

\[
-8 - b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 8 - b + c.
\]

If the inequality

\[
8 - b + c < 1
\]

is satisfied, then Eq. (194) has a stable equilibrium point in the neighborhood of -1, and the trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = -1 \) tends to this equilibrium point. Thus, we get the relation

\[
u_{ij} = -1, \quad x_{ij}(0) = -1, \quad u_{kl} = -1 \) or 1, \( (k, l \in N_{ij}, k \neq i, l \neq j) \}
\]

which indicates the local operation (c).

From Eqs. (198) and (206), we have

\[
7 < b - c, \quad 5 < b + c < 7.
\]

These inequalities can be satisfied by choosing \( b = 8, \ c = -2 \). Thus we get the following CNN gene:

<table>
<thead>
<tr>
<th>Point removal CNN genes</th>
</tr>
</thead>
</table>
| \( A_1 \)-gene: \[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\] |
| \( B_1 \)-gene: \[
\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\] |
| \( z \)-gene: \[ -2 \] |

(209)

or equivalently

<table>
<thead>
<tr>
<th>Point removal CNN template</th>
</tr>
</thead>
</table>
| \( A_1 \): \[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\] |
| \( B_1 \): \[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\] |
| \( z \): \[ -2 \] |

(210)

Design B:

Choose \( A_1 \)-gene, \( B_2 \)-gene and \( z \)-gene as a candidate set of CNN genes. Considering the local operations (a)–(c) for the states (196)–(204), we get the following three inequalities:

\[
-4 + b + c < -1,
\]

Note that this gene deletes diagonal lines, since the \( B_2 \) template does not have diagonal elements.
5.10. **Point extraction CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Point Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Extract all isolated black pixels from a binary image $P_1$. The CNN input image</td>
</tr>
<tr>
<td></td>
<td>is given by $P_1$, and the initial state is also given by $P_1$. The boundaries</td>
</tr>
<tr>
<td></td>
<td>are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in black and the</td>
</tr>
<tr>
<td></td>
<td>inputs of all neighbors $y_{kl}$ are coded in white. Then, the output $y_{ij}$ is</td>
</tr>
<tr>
<td></td>
<td>printed in black.</td>
</tr>
<tr>
<td></td>
<td>(b) Assume that the input $u_{ij}$ is coded in black and there is at least one</td>
</tr>
<tr>
<td></td>
<td>neighbor $y_{kl}$ whose input is coded in black. Then, the output $y_{ij}$ is</td>
</tr>
<tr>
<td></td>
<td>printed in white.</td>
</tr>
<tr>
<td></td>
<td>(c) Assume that the input $u_{ij}$ is coded in white. Then, the output $y_{ij}$ is</td>
</tr>
<tr>
<td></td>
<td>printed in white.</td>
</tr>
</tbody>
</table>

**Example**

Fig. 13

**Reference**

[Chua, 1998]

---

**Design A:**

Choose $A_1$-gene, $B_5$-gene and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

$$
\frac{dx_{ij}}{dt} = g(x_{ij}) - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c, 
$$

(217)

where

$$
g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases} \quad (218)
$$

Note that $g(x)$ has a local maximum (1 at $x = 1$), and a local minimum ($-1$ at $x = -1$). First, consider the local operation (a). The situation for this operation is given by

input : $u_{ij} = 1$,  
initial state : $x_{ij}(0) = 1$,  
inputs of neighbors : $u_{kl} = -1, \ (k, l \in N_{ij}, k \neq i, l \neq j)$

(219)

---

![Fig. 13. Point extraction.](image-url)
Hence, we have
\[- \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c = 8 + b + c. \]  \hfill (220)

If the inequality
\[-1 < 8 + b + c \]  \hfill (221)
is satisfied, then Eq. (217) has a stable equilibrium point in the neighborhood of 1, and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
    u_{ij} &= 1, \\
    x_{ij}(0) &= 1, \\
    u_{kl} &= -1 \ (k, l \in N_{ij}, k \neq i, l \neq j) \Rightarrow y_{ij} = 1
\end{align*}
\]  \hfill (222)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by
\[
\begin{align*}
    \text{input : } u_{ij} &= 1, \\
    \text{initial state : } x_{ij}(0) &= 1, \\
    \text{inputs of neighbors : } u_{kl} &= -1 \text{ or } 1 \\
    \left( -6 \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] \leq 8 \right)
\end{align*}
\]  \hfill (223)

Hence, we have the following inequality
\[-8 + b + c \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c \leq 6 + b + c. \]  \hfill (224)

If the inequality
\[6 + b + c < -1 \]  \hfill (225)
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. We get the relation
\[
\begin{align*}
    u_{ij} &= 1, \\
    x_{ij}(0) &= 1, \\
    u_{kl} &= -1 \text{ or } 1 \\
    \left( -6 \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] \leq 8 \right) \Rightarrow y_{ij} = -1
\end{align*}
\]  \hfill (226)

which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by
\[
\begin{align*}
    \text{input : } u_{ij} &= -1, \\
    \text{initial state : } x_{ij}(0) &= -1, \\
    \text{inputs of neighbors : } u_{kl} &= -1 \text{ or } 1 \\
    \left( -8 \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] \leq 8 \right)
\end{align*}
\]  \hfill (227)

Hence, we have
\[-8 - b + c \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c \leq 8 - b + c. \]  \hfill (228)

If the inequality
\[8 - b + c < 1 \]  \hfill (229)
is satisfied, then Eq. (217) has a stable equilibrium point in the neighborhood of $-1$, and the trajectory $x_{ij}(t)$ starting from $x_{ij}(0) = -1$ tends to this equilibrium point. Thus, we get the relation

$$u_{ij} = -1, \quad x_{ij}(0) = -1,$$

$$u_{kl} = -1 \text{ or } 1 \quad \left( -8 \leq - \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right) \Rightarrow y_{ij} = -1$$

which indicates the local operation (c).

From Eqs. (221) and (229), we have

$$7 < b - c, \quad -9 < b + c < -7.$$  \hfill (231)

These inequalities can be satisfied by choosing $b = 1, c = -9$. Thus, we get the following CNN gene:

<table>
<thead>
<tr>
<th>Point extraction CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$-gene: 0 0 0 0 2 0 0 0 0</td>
</tr>
<tr>
<td>$B_3$-gene: -1 -1 -1 -1 1 -1 -1 -1 -1</td>
</tr>
<tr>
<td>$z$-gene: -9</td>
</tr>
</tbody>
</table>

or equivalently

<table>
<thead>
<tr>
<th>Point extraction CNN template</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_3 = \begin{array}{ccc} 0 &amp; 0 &amp; 0 \ 0 &amp; 2 &amp; 0 \ 0 &amp; 0 &amp; 0 \end{array}$, $B_4 = \begin{array}{ccc} -1 &amp; -1 &amp; -1 \ -1 &amp; 1 &amp; -1 \ -1 &amp; -1 &amp; -1 \end{array}$, $z = -9$</td>
</tr>
</tbody>
</table>

**Design B:**

Choose $A_3$-gene, $B_3$-gene and $z$-gene as a candidate set of CNN genes. Considering the local operations (a)–(c) for the states (219)–(227), we get the following three inequalities:

$$-1 < b + 4 + c, \hfill (234)$$

$$2 + b + c < -1, \hfill (235)$$

$$4 - b + c < 1.$$  \hfill (236)

Hence, we have

$$-5 < b + c < -3, \quad 3 < b - c.$$  \hfill (237)

These inequalities can be satisfied by choosing $b = 1, c = -5$. Thus we get the following CNN gene:

<table>
<thead>
<tr>
<th>Point extraction CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$-gene: 0 0 0 0 2 0 0 0 0</td>
</tr>
<tr>
<td>$B_3$-gene: 0 -1 0 -1 1 -1 0 -1 0</td>
</tr>
<tr>
<td>$z$-gene: -5</td>
</tr>
</tbody>
</table>
or equivalently

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B_3 = \begin{bmatrix}
0 & -1 & 0 \\
-1 & 1 & -1 \\
0 & -1 & 0
\end{bmatrix}, \quad z = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}.
\]

(239)

Note that this gene cannot delete diagonal lines, since the \( B_2 \) template does not have diagonal elements.

5.11. Edge detection CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Edge Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Extract edges of objects in a binary image ( P_1 ). The CNN input image is</td>
</tr>
<tr>
<td></td>
<td>given by ( P_1 ), and the initial state is also given by ( P_1 ). The</td>
</tr>
<tr>
<td></td>
<td>boundaries are constrained to (-1).</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input ( u_{ij} ) of cell ( C_{ij} ) is coded in black</td>
</tr>
<tr>
<td></td>
<td>and the inputs of all neighbors are coded in black. Then, the output ( y_{ij} )</td>
</tr>
<tr>
<td></td>
<td>is printed in white.</td>
</tr>
<tr>
<td></td>
<td>(b) Assume that the input ( u_{ij} ) is coded in black and there is at least</td>
</tr>
<tr>
<td></td>
<td>one neighbor whose input is coded in white. Then, the output ( y_{ij} ) is</td>
</tr>
<tr>
<td></td>
<td>printed in black.</td>
</tr>
<tr>
<td></td>
<td>(c) Assume that the input ( u_{ij} ) is coded in white. Then, the output ( y_{ij} )</td>
</tr>
<tr>
<td></td>
<td>is printed in white.</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 14</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998; Roska et al., 1999]</td>
</tr>
</tbody>
</table>

Design A:
Choose \( A_1 \)-gene, \( B_5 \)-gene and \( z \)-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c,
\]

(240)

![Figure 14: Edge detection.](image1.png)

(1) input image. (2) initial state. (3) output image.

Fig. 14. Edge detection.
where
\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases}
\] (241)

Note that \(g(x)\) has a local maximum (1 at \(x = 1\)), and a local minimum (−1 at \(x = -1\)).

First, consider the local operation (a). The situation for this operation is given by

\[
\begin{align*}
\text{input} & : u_{ij} = 1, \\
\text{initial state} & : x_{ij}(0) = 1, \\
\text{inputs of neighbors} & : u_{kl} = 1 \ (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\] (242)

Hence, we have
\[
-8 + b + c < -1
\] (244)

If the inequality
\[
-8 + b + c < -1
\]
is satisfied, then Eq. (240) has a stable equilibrium point in the neighborhood of −1, and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
\text{input} & : u_{ij} = 1, \\
\text{initial state} & : x_{ij}(0) = 1, \\
\text{inputs of neighbors} & : u_{kl} = 1 \ (k, l \in N_{ij}, k \neq i, l \neq j). \\
\end{align*}
\]

Next, consider the local operation (b). The situation for this operation is given by

\[
\begin{align*}
\text{input} & : u_{ij} = 1, \\
\text{initial state} & : x_{ij}(0) = 1, \\
\text{inputs of neighbors} & : u_{kl} = -1 \text{ or } 1 \ (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\] (246)

Hence, we have
\[
-6 + b + c \leq -8 + b + c.
\] (247)

If the inequality
\[
-1 < -6 + b + c
\] (248)
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of 1. The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. We get the relation
\[
\begin{align*}
\text{input} & : u_{ij} = 1, \\
\text{initial state} & : x_{ij}(0) = 1, \\
\text{inputs of neighbors} & : u_{kl} = -1 \text{ or } 1 \ (k, l \in N_{ij}, k \neq i, l \neq j). \\
\end{align*}
\]

Finally, consider the local operation (c). The situation for this operation is given by

\[
\begin{align*}
\text{input} & : u_{ij} = -1, \\
\text{initial state} & : x_{ij}(0) = 1, \\
\text{inputs of neighbors} & : u_{kl} = -1 \text{ or } 1 \ (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}
\] (250)
Hence, we have
\[ -8 - b + c \leq -\left[ \sum_{k, l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c \]
\[ \leq 8 - b + c. \]  
(251)

If the inequality
\[ 8 - b + c < -1 \]  
(252)
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. We get the relation
\[ u_{ij} = 1, \quad x_{ij}(0) = 1, \]
\[ u_{kl} = -1 \text{ or } 1 \left\{ -6 \leq -\left[ \sum_{k, l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] \leq 8 \right\} \Rightarrow y_{ij} = -1 \]  
(253)
which indicates the local operation (c).

From Eqs. (244) and (252), we have
\[ 9 < b - c, \quad 5 < b + c < 7. \]  
(254)
These inequalities can be satisfied by choosing \(b = 8, c = -2\). Thus, we get the following gene:

<table>
<thead>
<tr>
<th>Edge detection CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1)-gene :</td>
</tr>
<tr>
<td>(B_5)-gene :</td>
</tr>
<tr>
<td>(z)-gene :</td>
</tr>
</tbody>
</table>

(255)

or equivalently

<table>
<thead>
<tr>
<th>Edge detection CNN template</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1 = )</td>
</tr>
<tr>
<td>(B_5 = )</td>
</tr>
<tr>
<td>(z = -2)</td>
</tr>
</tbody>
</table>

(256)

**Design B:**

In the previous design, the initial state is given by a binary input image \(P_1\). Here, we change it into \(x_{ij}(0) = -1\). Then, the following three inequalities are obtained from the local operations (a)–(c):
\[ -8 + b + c < 1, \]  
(257)
\[ 1 < -6 + b + c, \]  
(258)

and
\[ 8 - b + c < 1, \]  
(259)
respectively. From Eqs. (257) and (259), we have
\[ 7 < b - c, \quad 7 < b + c < 9. \]  
(260)
These inequalities can be satisfied by choosing \(b = 8, c = -0.5\). Thus, we get the following
gene:

\[
\begin{align*}
\mathcal{A}_1\text{-gene} & : \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}, \\
\mathcal{B}_5\text{-gene} & : \begin{bmatrix} -1 & -1 & -1 & -1 & 8 & -1 & -1 & -1 \end{bmatrix}, \\
z\text{-gene} & : -0.5,
\end{align*}
\]  

or equivalently

\[
\begin{align*}
\text{Edge detection CNN template} & \\
A_1 &= \begin{bmatrix} 0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \end{bmatrix}, & B_5 &= \begin{bmatrix} -1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1 \end{bmatrix}, & z &= -0.5.
\end{align*}
\]

5.12. **Convex corner detection CNN genes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Convex Corner Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global task</strong></td>
<td>Extract convex corners of objects in a binary image ( P_1 ). The CNN input image is given by ( P_1 ), and the initial state is also given by ( P_1 ). The boundaries are constrained to (-1).</td>
</tr>
</tbody>
</table>
| **Local operations**| (a) Assume that the input \( u_{ij} \) is coded in black and the number of neighbors \( u_{kl} \) coded in black is smaller than 4. Then, the output \( y_{ij} \) is printed in black.  
(b) Assume that the input \( u_{ij} \) of cell \( C_{ij} \) is coded in black and the number of neighbors \( u_{kl} \) coded in black is greater than or equal to 4. Then, the output \( y_{ij} \) is printed in white.  
(c) Assume that the input \( u_{ij} \) is coded in white. Then, the output \( y_{ij} \) is printed in white. |
| **Example**         | Fig. 15                                                                                 |
| **Reference**       | [Chua, 1998; Roska et al., 1999]                                                        |

**Design A:**

Choose \( \mathcal{A}_1\text{-gene} \), \( \mathcal{B}_5\text{-gene} \) and \( z\text{-gene} \) as a candidate set of CNN genes. In this case, Eq. (8) has the form

\[
\frac{d x_{ij}}{dt} = g(x_{ij}) - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + b u_{ij} + c,
\]  

where

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases}
\]  

Note that \( g(x) \) has a local maximum (1 at \( x = 1 \)), and a local minimum: \((-1\) at \( x = -1 \)). First, consider the local operation (a). The situation for this operation is given by
input : \( u_{ij} = 1 \),
initial state : \( x_{ij}(0) = 1 \),
inputs of neighbors : \( u_{kl} = -1 \) or \( 1 \)

\[
-8 + b + c \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c \leq b + c - 2 .
\] (266)

If the inequality
\[
b + c - 2 < -1
\] (267)
is satisfied, then Eq. (263) has a stable equilibrium point in the neighborhood of \(-1\), and the trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. Thus, we get the relation
\[
\mathbf{u} = \begin{cases} 1, & x_{ij}(0) = 1, \\ -1 \text{ or } 1, & u_{kl} \end{cases}
\]
\[
\Rightarrow y_{ij} = -1
\] (268)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

input : \( u_{ij} = 1 \),
initial state : \( x_{ij}(0) = 1 \),
inputs of neighbors : \( u_{kl} = -1 \) or \( 1 \)

\[
0 \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] \leq 8 .
\] (269)

Hence, we have
\[
b + c \leq - \left[ \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \right] + bu_{ij} + c \leq 8 + b + c .
\] (270)

If the inequality
\[
-1 < b + c
\] (271)
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of \(1\). The trajectory \( x_{ij}(t) \)
starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. We get the relation

\[
\begin{align*}
    u_{ij} &= 1, \\
    x_{ij}(0) &= 1, \\
    u_{kl} &= -1 \text{ or } 1 \left( 0 \leq -\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right) \implies y_{ij} = 1
\end{align*}
\]  

(272)

which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by

input : \( u_{ij} = -1 \),
initial state : \( x_{ij}(0) = -1 \),
inputs of neighbors : \( u_{kl} = -1 \text{ or } 1 \left( -8 \leq -\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right) \).

Hence, we have

\[
-8 - b + c \leq -\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 8 - b + c.
\]  

(274)

If the inequality

\[
8 - b + c < 1
\]  

(275)

is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = -1 \) tends to this equilibrium point. We get the relation

\[
\begin{align*}
    u_{ij} &= -1, \\
    x_{ij}(0) &= -1, \\
    u_{kl} &= -1 \text{ or } 1 \left( -8 \leq -\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right) \implies y_{ij} = -1
\end{align*}
\]  

(276)

which indicates the local operation (c).

From Eqs. (267) and (275), we have

\[
7 < b - c, \quad -1 < b + c < -1.
\]  

(277)

These inequalities can be satisfied by choosing \( b = 4, c = -4 \). Thus, we get the following CNN gene:

<table>
<thead>
<tr>
<th>Convex corner detection CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 )-gene : 0 0 0 0 0 2 0 0 0</td>
</tr>
<tr>
<td>( z )-gene : -4</td>
</tr>
</tbody>
</table>

(278)

or equivalently

<table>
<thead>
<tr>
<th>Convex corner detection CNN template</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 = \begin{bmatrix} 0 &amp; 0 &amp; 0 \ 0 &amp; 2 &amp; 0 \ 0 &amp; 0 &amp; 0 \end{bmatrix} ), ( B_5 = \begin{bmatrix} -1 &amp; -1 &amp; -1 \ -1 &amp; 4 &amp; -1 \ -1 &amp; -1 &amp; -1 \end{bmatrix} ), ( z = -4 )</td>
</tr>
</tbody>
</table>

(279)
Design B:

In the previous design, the initial state is given by a binary input image $P_1$. Here, we change it into $x_{ij}(0) = -1$. Then, the following three inequalities are obtained from the local operations (a)–(c):

\begin{align*}
  b + c - 2 &< 1, \quad (280) \\
  -1 &< b + c, \quad (281) \\
  b + c - 2 &< 1, \quad (282)
\end{align*}

respectively. From Eqs. (281) and (282), we have

\begin{align*}
  7 &< b - c, \quad -1 < b + c < 3. \quad (283)
\end{align*}

These inequalities can be satisfied by choosing $b = 8$, $c = -8.5$. Thus, we get the following CNN gene:

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & -1 & 8 & -1 & -1 & -1 & -1
\end{bmatrix},
\]

or equivalently

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & -1 & 8 & -1 & -1 & -1 & -1
\end{bmatrix},
\]

or equivalently

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B_5 = \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}, \quad z = -8.5.
\]

or equivalently

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B_5 = \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}, \quad z = -8.5.
\]

5.13. Erosion CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Erosion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global task</strong></td>
<td>Peel off all boundary pixels of a binary image $P_1$. The CNN input image is given by $P_1$, and the initial state is also given by $P_1$. The boundaries are constrained to $-1$.</td>
</tr>
</tbody>
</table>
| **Local operations** | (a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in black and the inputs $u_{kl}$ of all neighbors are coded in black. Then, the output $y_{ij}$ is printed in black.  
(b) Assume that the input $u_{ij}$ is coded in black and there is at least one neighbor $u_{kl}$ whose input is coded in white. Then, the output $y_{ij}$ is printed in white.  
(c) Assume that the input $u_{ij}$ is coded in white. Then, the output $y_{ij}$ is printed in white. |
| **Example**     | Fig. 16                                                                 |
| **Reference**   | [Chua, 1998]                                                             |
Design A: Choose $A_1$-gene, $B_4$-gene and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

$$\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c,$$

(286)

where

$$g(x_{ij}) = \begin{cases} -x_{ij} - 2, & x_{ij} \leq -1 \\ x_{ij}, & |x_{ij}| \leq 1 \\ -x_{ij} + 2, & x_{ij} \geq 1. \end{cases}$$

(287)

Note that $g(x)$ has a local maximum (1 at $x = 1$), and a local minimum ($-1$ at $x = -1$). First, consider the local operation (a). The situation for this operation is given by

$$\begin{align*}
\text{input:} & \quad u_{ij} = 1, \\
\text{initial state:} & \quad x_{ij}(0) = 1, \\
\text{inputs of neighbors:} & \quad u_{kl} = 1 (k, l \in N_{ij}, k \neq i, l \neq j).
\end{align*}$$

(288)

Hence, we have

$$\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c = 8 + b + c.$$  

(289)

If the inequality

$$-1 < 8 + b + c$$

(290)

is satisfied, then Eq. (286) has a stable equilibrium point in the neighborhood of 1, and the trajectory $x_{ij}(t)$ starting from $x_{ij}(0) = 1$ tends to this equilibrium point. Thus, we get the relation

$$\begin{align*}
\{u_{ij} = 1, \quad x_{ij}(0) = 1, \\
\{u_{kl} = 1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)\} \Rightarrow y_{ij} = 1
\end{align*}$$

(291)

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

$$\begin{align*}
\text{input:} & \quad u_{ij} = 1, \\
\text{initial state:} & \quad x_{ij}(0) = 1, \\
\text{inputs of neighbors:} & \quad u_{kl} = -1 \text{ or } 1
\end{align*}$$

(292)

Hence, we have

$$-8 + b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 6 + b + c.$$  

(293)

If the inequality

$$6 + b + c < -1$$

(294)
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of 1. The trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = 1 \) tends to this equilibrium point. We get the relation

\[
\begin{align*}
\begin{cases}
u_{ij} = 1, & x_{ij}(0) = 1, \\
u_{kl} = -1 \text{ or } 1 & \left( -8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 6 \right)
\end{cases}
\Rightarrow y_{ij} = 1
\end{align*}
\]

which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by

\[
\begin{align*}
\text{input : } u_{ij} &= -1, \\
\text{initial state : } x_{ij}(0) &= -1, \\
\text{inputs of neighbors : } u_{kl} &= -1 \text{ or } 1 & \left( -8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right).
\end{align*}
\]

Hence, we have

\[
-8 - b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \\
\leq 8 - b + c.
\]

If the inequality

\[
\begin{align*}
u_{ij} = -1, & \quad x_{ij}(0) = -1, \\
u_{kl} = -1 \text{ or } 1 & \left( -8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right) \\
\Rightarrow y_{ij} = -1
\end{align*}
\]

is satisfied, then Eq. (286) has a stable equilibrium point in the neighborhood of -1, and the trajectory \( x_{ij}(t) \) starting from \( x_{ij}(0) = -1 \) tends to this equilibrium point. Thus, we get the relation

\[
-8 - b + c < 1
\]

These inequalities can be satisfied by choosing \( b = 1, \ c = -9 \). Thus, we get the following CNN gene:

\[
\begin{align*}
\text{Erosion CNN genes} & \\
\begin{cases}
A_1\text{-gene : } & 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \\
B_4\text{-gene : } & 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
z\text{-gene : } & -9
\end{cases}
\end{align*}
\]

or equivalently

\[
\begin{align*}
\text{Erosion CNN template} & \\
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B_4 = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}, \quad z = -9
\end{align*}
\]

![Equation (295)](image-url)
Design B:
Choose $A_1$-gene, $B_2$-gene and $z$-gene as a candidate set of CNN genes. Considering the local operations (a)–(e) for the states $(288)$–$(296)$, we get the following three inequalities:

$$-1 < 4 + b + c,$$
$$2 + b + c < -1,$$
$$4 - b + c < 1.$$  

Hence, we have

$$-5 < b + c < -3, \quad 3 < b - c.$$  

These inequalities can be satisfied by choosing $b = 1$, $c = -5$. Thus we get the following CNN gene:

$$\begin{align*}
\text{Erosion CNN genes} \\
A_1 & = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 
\end{bmatrix}, \\
B_2 & = \begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 
\end{bmatrix}, \\
z & = -5.
\end{align*}$$  

or equivalently

$$\begin{align*}
\text{Erosion CNN template} \\
A_1 &= \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 
\end{bmatrix}, \\
B_2 &= \begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 
\end{bmatrix}, \\
z &= -5.
\end{align*}$$  

Note that this gene cannot completely peel off the corners, since the $B_2$ template does not have diagonal elements.

5.14. Dilation CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Dilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Grow a layer of pixels around objects in a binary image $P_1$.</td>
</tr>
<tr>
<td></td>
<td>The CNN input image is given by $P_1$, and the initial state is also given by $P_1$. The boundaries are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Assume that the input $u_{ij}$ of cell $C_{ij}$ is coded in white and there is at least one neighbor $u_{kl}$ whose input is coded in black. Then, the output $y_{ij}$ is printed in black.</td>
</tr>
<tr>
<td></td>
<td>(b) Assume that the input $u_{ij}$ is coded in white and the inputs $u_{kl}$ of all neighbors are coded in white. Then, the output $y_{ij}$ is printed in white.</td>
</tr>
<tr>
<td></td>
<td>(c) Assume that the input $u_{ij}$ is coded in black. Then, the output $y_{ij}$ is printed in black.</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 17</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998]</td>
</tr>
</tbody>
</table>

Design A:
Choose $A_1$-gene, $B_4$-gene and $z$-gene as a candidate set of CNN genes. In this case, Eq. (8) has the form

$$\frac{dx_{ij}}{dt} = g(x_{ij}) + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c,$$

where

$$g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases}$$  

Note that $g(x)$ has a local maximum (1 at $x = 1$), and a local minimum ($-1$ at $x = -1$). First,
consider the local operation (a). The situation for this operation is given by

\[
\text{input : } u_{ij} = -1, \quad \text{initial state : } x_{ij}(0) = -1,
\]

\[
\text{inputs of neighbors : } u_{kl} = -1 \text{ or } 1 \quad \left( -6 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right).
\]

Hence, we have the following equation

\[
-6 - b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 8 - b + c.
\]

If the inequality

\[
1 < -6 - b + c
\]

is satisfied, then Eq. (309) has a stable equilibrium point in the neighborhood of \(-1\), and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. Thus, we get the relation

\[
\left\{ \begin{array}{l}
\quad u_{ij} = -1, \\
\quad x_{ij}(0) = -1,
\end{array} \right. \quad \left\{ \begin{array}{l}
\quad u_{kl} = -1 \text{ or } 1 \\
\quad \left( -6 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8 \right)
\end{array} \right\} \Rightarrow y_{ij} = 1
\]

which indicates the local operation (a).

Next, consider the local operation (b). The situation for this operation is given by

\[
\text{input : } u_{ij} = -1, \quad \text{initial state : } x_{ij}(0) = -1, \quad \text{inputs of neighbors : } u_{kl} = -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j).
\]

Hence, we have

\[
\sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c = -8 - b + c.
\]

If the inequality

\[
-8 - b + c < 1
\]
is satisfied, then Eq. (16) has a stable equilibrium point in the neighborhood of \(-1\). The trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = -1\) tends to this equilibrium point. We get the relation
\[
\begin{align*}
    u_{ij} &= -1, \quad x_{ij}(0) = -1, \\
    u_{kl} &= -1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)
\end{align*}
\]  
\Rightarrow y_{ij} = -1
\tag{318}
\]
which indicates the local operation (b).

Finally, consider the local operation (c). The situation for this operation is given by
\[
\begin{align*}
    \text{input :} & \quad u_{ij} = 1, \\
    \text{initial state :} & \quad x_{ij}(0) = 1, \\
    \text{inputs of neighbors :} & \quad u_{kl} = -1 \text{ or } 1 \quad (k, l \in N_{ij}, k \neq i, l \neq j)
\end{align*}
\]  
\[
y_{ij} = 1 \quad (318)
\]
Hence, we have
\[
-8 + b + c \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} + bu_{ij} + c \leq 8 + b + c.
\tag{320}
\]
If the inequality
\[
-1 < -8 + b + c
\tag{321}
\]
is satisfied, then Eq. (309) has a stable equilibrium point in the neighborhood of 1, and the trajectory \(x_{ij}(t)\) starting from \(x_{ij}(0) = 1\) tends to this equilibrium point. Thus, we get the relation
\[
\begin{align*}
    u_{ij} &= 1, \quad x_{ij}(0) = 1, \\
    u_{kl} &= -1 \text{ or } 1 \quad \left(-8 \leq \sum_{k,l \in N_{ij}, k \neq i, l \neq j} u_{kl} \leq 8\right) \\
\end{align*}
\]  
\Rightarrow y_{ij} = 1
\tag{322}
\]
which indicates the local operation (c).

From Eqs. (313) and (321), we have
\[
7 < b + c, \quad 7 < c - b < 9.
\tag{323}
\]
These inequalities can be satisfied by choosing \(b = 1, c = 9\). Thus we get the following CNN gene:

<table>
<thead>
<tr>
<th>Dilation CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1)-gene : (0 0 0 2 0 0 0 0)</td>
</tr>
<tr>
<td>(B_4)-gene : (1 1 1 1 1 1 1 1)</td>
</tr>
<tr>
<td>(z)-gene : (9)</td>
</tr>
</tbody>
</table>

or equivalently

<table>
<thead>
<tr>
<th>Dilation CNN template</th>
</tr>
</thead>
</table>
| \(A_1 = \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 2 & 0 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \end{bmatrix}, \quad z = 9\) |

Design B:
Choose \(A_1\)-gene, \(B_2\)-gene and \(z\)-gene as a candidate set of CNN genes. Considering the local operations (a)-(c) for the states (311)-(319), we get the following three inequalities:
\[
1 < -2 + b + c, \quad -4 - b + c < -1, \quad -1 < -4 + b + c.
\tag{326-328}
\]
Hence, we have
\[
3 < c - b < 5, \quad 3 < b + c.
\tag{329}
\]
These inequalities can be satisfied by choosing \(b = 1, c = 9\). Thus we get the following CNN gene:

<table>
<thead>
<tr>
<th>Dilation CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1)-gene : (0 0 0 2 0 0 0 0)</td>
</tr>
<tr>
<td>(B_2)-gene : (0 1 0 1 1 1 0 1)</td>
</tr>
<tr>
<td>(z)-gene : (9)</td>
</tr>
</tbody>
</table>
or equivalently

\[
\text{Dilation CNN template}
\]

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B_2 = \begin{bmatrix}
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix}, \quad z = 5.
\] (331)

Note that this gene cannot grow at the corners completely, since the \(B_2\) template does not have diagonal elements.

### 5.15. Right edge detection CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Right Edge Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Extract right edges of all binary objects which are at least two-pixels wide along the horizontal direction in a given binary image (P_1). The CNN input image is given by (P_1), and the initial state is also given by (P_1). The boundaries are constrained to (-1).</td>
</tr>
<tr>
<td>Local operations</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>(u_{i,j-1})</td>
</tr>
<tr>
<td>(a)</td>
<td>B</td>
</tr>
<tr>
<td>(b)</td>
<td>B</td>
</tr>
<tr>
<td>(c)</td>
<td>B</td>
</tr>
<tr>
<td>(d)</td>
<td>B</td>
</tr>
<tr>
<td>(e)</td>
<td>W</td>
</tr>
<tr>
<td>(f)</td>
<td>W</td>
</tr>
<tr>
<td>(g)</td>
<td>W</td>
</tr>
<tr>
<td>(h)</td>
<td>W</td>
</tr>
</tbody>
</table>

The symbols B and W indicate black and white, respectively.

Example | Fig. 18
Reference | [Chua, 1998]

**Design A:**

We can realize this local operation by using the following new CNN genes:

- \(A_1\)-gene:
  \[
  \begin{bmatrix}
  0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0
  \end{bmatrix}
  \]

- \(B\)-gene:
  \[
  \begin{bmatrix}
  0 & 0 & 0 & p & q & r & 0 & 0 & 0
  \end{bmatrix}
  \]

- \(z\)-gene:
  \[
  \begin{bmatrix}
  s
  \end{bmatrix}
  \]

Their template expression is given by

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
p & q & r \\
0 & 0 & 0
\end{bmatrix}, \quad z = s.
\] (333)

In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + pu_{i,j-1} + qu_{i,j} + ru_{i,j+1} + s,
\] (334)
where

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases}
\]  \tag{335}

Note that \(g(x)\) has a local maximum (1 at \(x = 1\)), and a local minimum (−1 at \(x = -1\)). The local operations can be written as

<table>
<thead>
<tr>
<th>Operation</th>
<th>(u_{i,j-1})</th>
<th>(u_{i,j})</th>
<th>(u_{i,j+1})</th>
<th>(p u_{i,j-1} + q u_{i,j} + r u_{i,j+1} + s)</th>
<th>(x_{ij}(0))</th>
<th>(\Rightarrow)</th>
<th>(y_{ij})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(p + q + r + s)</td>
<td>1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(b)</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>(p + q - r + s)</td>
<td>1</td>
<td>(\rightarrow)</td>
<td>1</td>
</tr>
<tr>
<td>(c)</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>(p - q + r + s)</td>
<td>-1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(d)</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>(p - q - r + s)</td>
<td>-1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(e)</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>(-p + q + r + s)</td>
<td>1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(f)</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>(-p + q - r + s)</td>
<td>1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(g)</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>(-p - q + r + s)</td>
<td>-1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
<tr>
<td>(h)</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>(-p - q - r + s)</td>
<td>-1</td>
<td>(\rightarrow)</td>
<td>-1</td>
</tr>
</tbody>
</table>

From the driving-point plot analysis, we assume that \(p, q, r\) and \(s\) satisfy

\[
\begin{align*}
(a') & \quad p + q + r + s < -1, \\
(b') & \quad p + q - r + s > -1, \\
(c') & \quad p - q + r + s < 1, \\
(d') & \quad p - q - r + s < 1, \\
(e') & \quad -p + q + r + s < -1, \\
(f') & \quad -p + q - r + s < -1, \\
(g') & \quad -p - q + r + s < 1, \\
(h') & \quad -p - q - r + s < 1.
\end{align*}
\]  \tag{337}

These inequalities are satisfied if we choose \(p = 1\), \(r = -1\), \(s = -3\). Thus, we get the genes

\[
\begin{align*}
\text{Right edge detection CNN genes} \\
A_1\text{-gene : } & \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}, \\
B\text{-gene : } & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 \end{bmatrix}, \\
z\text{-gene : } & -3.
\end{align*}
\]  \tag{339}
Their template expression is given by

\begin{equation}
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & -1 \\
0 & 0 & 0
\end{bmatrix}, \quad z = -3.
\tag{340}
\end{equation}

The inequalities (338) can also be satisfied if we choose \( p = 1, r = -2 \) and \( s = -4 \). Thus, we have the gene

\begin{equation}
A_1\text{-gene : } \begin{bmatrix}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0
\end{bmatrix},
\tag{341}
\end{equation}

\begin{equation}
B\text{-gene : } \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & -2 & 0 & 0 & 0
\end{bmatrix},
\end{equation}

\begin{equation}
z\text{-gene : } -4.
\end{equation}

Their template expression is given by

\begin{equation}
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & -2 \\
0 & 0 & 0
\end{bmatrix}, \quad z = -4.
\tag{342}
\end{equation}

### 5.16. Shadow projection CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Shadow Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global task</strong></td>
<td>Project onto the left the shadow of all objects in a binary image ( P_1 ) from the right. The CNN input image is given by ( P_1 ), and the initial state is also given by ( x_{ij}(0) = 1 ). The boundaries are constrained to (-1).</td>
</tr>
<tr>
<td><strong>Local operations</strong></td>
<td>Operation</td>
</tr>
<tr>
<td>(a)</td>
<td>B</td>
</tr>
<tr>
<td>(b)</td>
<td>B</td>
</tr>
<tr>
<td>(c)</td>
<td>W</td>
</tr>
<tr>
<td>(d)</td>
<td>W</td>
</tr>
</tbody>
</table>

The symbols B and W indicate black and white, respectively.

| Example | Fig. 19 |
|Reference | [Chua, 1998] |
Design A:
We can realize this local operation by using the following new CNN genes:

\[
\begin{aligned}
A_1\text{-gene :} & \begin{bmatrix}
0 & 0 & 0 & 0 & 2 & p & 0 & 0 & 0 \\
0 & 2 & p & & & & & & \\
0 & 0 & 0 & & & & & & 
\end{bmatrix}, \\
B_1\text{-gene :} & \begin{bmatrix}
0 & 0 & 0 & 0 & q & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & & & & & & 
\end{bmatrix}, \\
z\text{-gene :} & \begin{bmatrix}
\_ & \\
\_ & \\
\_ & 
\end{bmatrix}
\end{aligned}
\] (343)

Their template expression is given by

\[
A_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & p \\
0 & 0 & 0 
\end{bmatrix}, \quad B_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & q & 0 \\
0 & 0 & 0 
\end{bmatrix}, \quad z = \begin{bmatrix}
\_ \\
\_ \\
\_ 
\end{bmatrix}. 
\] (344)

In this case, Eq. (8) has the form

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + py_{i,j+1} + qu_{i,j} + r, 
\] (345)

where

\[
g(x_{ij}) = \begin{cases}
-x_{ij} - 2, & x_{ij} \leq -1 \\
x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 2, & x_{ij} \geq 1.
\end{cases} 
\] (346)

Note that \( g(x) \) has a local maximum (1 at \( x = 1 \)), and a local minimum (−1 at \( x = -1 \)). The local operations can be written as

<table>
<thead>
<tr>
<th>Operation</th>
<th>( u_{i,j} )</th>
<th>( y_{i,j+1} )</th>
<th>( py_{i,j+1} + qu_{i,j} + r )</th>
<th>( x_{ij}(0) )</th>
<th>( \Rightarrow )</th>
<th>( y_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>1</td>
<td>( p + q + r )</td>
<td>1</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(b)</td>
<td>1</td>
<td>-1</td>
<td>( p - q + r )</td>
<td>1</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(c)</td>
<td>-1</td>
<td>1</td>
<td>( -p + q + r )</td>
<td>-1</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(d)</td>
<td>-1</td>
<td>-1</td>
<td>( -p - q + r )</td>
<td>-1</td>
<td>( \rightarrow )</td>
<td>-1</td>
</tr>
</tbody>
</table>

(347)
From the driving-point plot analysis, we assume that $p$, $q$, $r$, and $s$ satisfy

- **(a')** $p + q + r > -1$,
- **(b')** $p - q + r > -1$,
- **(c')** $-p + q + r > -1$,
- **(d')** $-p - q + r < -1$.

Then, the dynamics of (345) satisfies all of the local operations given in Eq. (347). If we set $r = 2$, then Eq. (348) has the form

$$1 < q + r$$
$$-3 < -q + r < 1$$

These inequalities are satisfied if we choose $q = 2$, $r = 0$. Thus, we get the genes

Shadow projection CNN genes

- **$A_1$-gene**:
  
  \[
  \begin{array}{cccccc}
  0 & 0 & 0 & 0 & 2 & 2 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$B_1$-gene**:
  
  \[
  \begin{array}{cccccc}
  0 & 0 & 0 & 0 & 2 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$z$-gene**:
  
  \[
  0 
  \]

Their template expression is given by

Shadow projection CNN template

- **$A_1$**:
  
  \[
  \begin{array}{cccc}
  0 & 0 & 0 & 0 \\
  0 & 2 & 2 & 0 \\
  0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$B_1$**:
  
  \[
  \begin{array}{cccc}
  0 & 0 & 0 & 2 \\
  0 & 2 & 0 & 0 \\
  0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$z$**:
  
  \[
  0 
  \]

The inequalities (348) can also be satisfied if we choose $p = 1$, $q = 1$ and $r = 0$. Thus, we have the gene

Shadow projection CNN genes

- **$A_1$-gene**:
  
  \[
  \begin{array}{cccccc}
  0 & 0 & 0 & 0 & 2 & 1 \\
  0 & 2 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$B_1$-gene**:
  
  \[
  \begin{array}{cccccc}
  0 & 0 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$z$-gene**:
  
  \[
  0 
  \]

Their template expression is given by

Shadow projection CNN template

- **$A_1$**:
  
  \[
  \begin{array}{cccc}
  0 & 0 & 0 & 0 \\
  0 & 2 & 1 & 0 \\
  0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$B_1$**:
  
  \[
  \begin{array}{cccc}
  0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0 \\
  \end{array}
  \]

- **$z$**:
  
  \[
  0 
  \]

### 5.17. Horizontal hole detection CNN genes

<table>
<thead>
<tr>
<th>Name</th>
<th>Horizontal Hole Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Detect the number of horizontal holes from each horizontal row of a binary image $P_1$ (loaded as an initial state) where objects are coded as black pixels and the background is coded in white. A string of adjacent white pixels in a horizontal row “i” of $P_1$ is called a horizontal hole of horizontal row “i” if each end is terminated by at least one black pixel. The CNN input image is arbitrarily given, and the initial state is also given by $P_1$. The boundaries are constrained to $-1$.</td>
</tr>
</tbody>
</table>
### Local operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>$y_{i,j-1}$</th>
<th>$y_{i,j}$</th>
<th>$y_{i,j+1}$</th>
<th>$\Rightarrow$</th>
<th>$y_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>$\Rightarrow$</td>
<td>B</td>
</tr>
<tr>
<td>(b)</td>
<td>B</td>
<td>B</td>
<td>W</td>
<td>$\Rightarrow$</td>
<td>B</td>
</tr>
<tr>
<td>(c)</td>
<td>B</td>
<td>W</td>
<td>B</td>
<td>$\Rightarrow$</td>
<td>W</td>
</tr>
<tr>
<td>(d)</td>
<td>B</td>
<td>W</td>
<td>W</td>
<td>$\Rightarrow$</td>
<td>B</td>
</tr>
<tr>
<td>(e)</td>
<td>W</td>
<td>B</td>
<td>B</td>
<td>$\Rightarrow$</td>
<td>W</td>
</tr>
<tr>
<td>(f)</td>
<td>W</td>
<td>B</td>
<td>W</td>
<td>$\Rightarrow$</td>
<td>B</td>
</tr>
<tr>
<td>(g)</td>
<td>W</td>
<td>W</td>
<td>B</td>
<td>$\Rightarrow$</td>
<td>W</td>
</tr>
<tr>
<td>(h)</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>$\Rightarrow$</td>
<td>W</td>
</tr>
</tbody>
</table>

The symbols B and W indicate black and white, respectively.

<table>
<thead>
<tr>
<th>Example</th>
<th>Fig. 20</th>
</tr>
</thead>
</table>

| Reference | [Chua, 1998] |

**Design A:**

We can realize this local operation by using the following new CNN genes:

\[
\begin{align*}
A_1\text{-gene} : & \begin{bmatrix} 0 & 0 & 0 & p & 2 & q & 0 & 0 & 0 \end{bmatrix} \\
B\text{-gene} : & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
Z\text{-gene} : & 0
\end{align*}
\]

(354)

Their template expression is given by

\[
A_1 = \begin{bmatrix} 0 & 0 & 0 \\
p & 2 & q \\
0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \end{bmatrix}, \quad z = \begin{bmatrix} 0 \end{bmatrix}
\]

(355)

(1) input image. (2) initial state. (3) output image.

Fig. 20. Horizontal-hole detection.
In this case, Eq. (8) has the form
\[ \frac{dx_{ij}}{dt} = g(x_{ij}) + p y_{i,j-1} + q y_{i,j+1}, \]  
(356)
where
\[ g(x_{ij}) = \begin{cases} 
-x_{ij} - 2, & x_{ij} \leq -1 \\
-x_{ij} + 2, & x_{ij} \geq 1 
\end{cases} \]  
(357)
Note that \( g(x) \) has a local maximum (1 at \( x = 1 \)), and a local minimum (−1 at \( x = -1 \)). The local operations can be written as

<table>
<thead>
<tr>
<th>Operation</th>
<th>( y_{i,j-1} )</th>
<th>( y_{i,j} )</th>
<th>( y_{i,j+1} )</th>
<th>( p y_{i,j-1} + q y_{i,j+1} )</th>
<th>( \Rightarrow )</th>
<th>( y_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( p + q )</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(b)</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>( p - q )</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(c)</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>( p + q )</td>
<td>( \rightarrow )</td>
<td>-1</td>
</tr>
<tr>
<td>(d)</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>( p - q )</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(e)</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>( -p + q )</td>
<td>( \rightarrow )</td>
<td>-1</td>
</tr>
<tr>
<td>(f)</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>( -p - q )</td>
<td>( \rightarrow )</td>
<td>1</td>
</tr>
<tr>
<td>(g)</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>( -p + q )</td>
<td>( \rightarrow )</td>
<td>-1</td>
</tr>
<tr>
<td>(h)</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>( -p - q )</td>
<td>( \rightarrow )</td>
<td>-1</td>
</tr>
</tbody>
</table>

From the driving-point plot analysis, we assume that \( p, q, r \) and \( s \) satisfy
\[ (a') \; p + q > -1, \]  
\[ (b') \; p - q > -1, \]  
\[ (c') \; p + q < 1, \]  
\[ (d') \; p - q > 1, \]  
\[ (e') \; -p + q < -1, \]  
\[ (f') \; -p - q > -1, \]  
\[ (g') \; -p + q < 1, \]  
\[ (h') \; -p - q < 1, \]  
(359)
Then, the dynamics of (356) satisfies all of the local operations given in Eq. (358). Equation (359) can be simplified into
\[ -1 < p + q < 1, \]  
\[ 1 < p - q. \]  
(360)
These inequalities are satisfied if we choose \( p = 1, \) \( q = -1. \) Thus, we get the genes

Horizontal hole detection CNN genes
\[
\begin{align*}
A_1 & = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, & B & = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & z & = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{align*}
\]  
(362)
The inequalities (359) can also be satisfied if we choose \( p = 2, \) \( q = -2. \) Thus, we have the gene

Horizontal hole detection CNN genes
\[
\begin{align*}
A_1 & = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & -2 \\ 0 & 0 & 0 \end{bmatrix}, & B & = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & z & = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{align*}
\]  
(363)
5.18. *Inverse half-toning and half-toning CNN genes*

Half-toning is one of the main image compression techniques used for transmitting graphics over low-bandwidth channels, such as e.g. telephone lines (fax machines). It converts a gray-scale image into a binary image in such a way that the average gray level for corresponding regions of both images is the same. Due to the averaging property of the human visual system, these images will appear to be the same when observed from an appropriate distance. Inverse half-toning transforms a binary image into a gray-scale image in a way which preserves the locally averaged intensities of both images. This operation is complementary to the half-toning example shown in Fig. 22.

<table>
<thead>
<tr>
<th>Name</th>
<th>Inverse Half-Toning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Convert a given binary image $P_1$ into a gray-scale image. The CNN input image and the initial state are given by $P_1$, and the boundaries are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>(a) Averaging inputs in the neighborhood of radius 1.</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 21</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998]</td>
</tr>
</tbody>
</table>

**Design A:**

We can realize this local operation by using the following new CNN genes:

\[
\begin{align*}
A_0\text{-gene} & : \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} , \\
B\text{-gene} & : \begin{bmatrix} q & p & q & p & r & p & q & p & q \end{bmatrix} , \\
z\text{-gene} & : \begin{bmatrix} 0 \end{bmatrix} .
\end{align*}
\]  \hspace{1cm} (365)

(1) input image. (2) initial state. (3) output image.

Fig. 21. Inverse half-toning.
Their template expression is given by

\[
A_0 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 
\end{bmatrix}, \quad B = \begin{bmatrix}
q & p & q \\
p & r & p \\
q & p & q
\end{bmatrix}, \quad z = \begin{bmatrix} 0 \end{bmatrix}
\]  

(366)

Here, the parameters \( p, q, \) and \( r \) satisfy

\[
r = 1 - 4p - 4q, \quad 0 < r < 1, \\
0 < p < 1, \quad 0 < q < 1.
\]  

(367)

Hence, we have

\[
\sum_{k,l \in N_{ij}} b_{kl} = 4p + 4q + r = 1, \\
0 < p < \frac{1}{4}, \quad 0 < q < \frac{1}{4}.
\]  

(368)

Since \( a_{k-i,l-j} = 0 \) \((k, l \in N_{ij})\), Eq. (1) has the form

\[
\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{k,l \in N_{ij}, k \neq i, l \neq j} b_{kl} u_{kl} + r u_{ij}.
\]  

(369)

This system has a stable equilibrium point

\[
x_{ij}^e = \sum_{k,l \in N_{ij}, k \neq i, l \neq j} b_{kl} u_{kl} + r u_{ij},
\]  

(370)

which corresponds to the averaged value of the inputs in the neighborhood of a radius 1 and satisfies the following inequalities

\[
0 \leq x_{ij}^e \leq 1, \quad 0 \leq y_{ij}^e \leq 1 \quad (y_{ij}^e = f(x_{ij}^e)).
\]  

(371)

If we choose \( p = 0.1, \) \( q = 0.07, \) \( r = 0.32, \) we get the following CNN genes:

Inverse half-toning CNN genes

\[
\begin{align*}
A_0\text{-gene} : & \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0, \\
B\text{-gene} : & \quad 0.07 \quad 0.1 \quad 0.07 \quad 0.1 \quad 0.32 \quad 0.1 \quad 0.07 \quad 0.1 \quad 0.07, \\
z\text{-gene} : & \quad 0.
\end{align*}
\]  

(372)

or equivalently

Inverse half-toning CNN template

\[
A_0 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 
\end{bmatrix}, \quad B = \begin{bmatrix}
0.07 & 0.1 & 0.07 \\
0.1 & 0.32 & 0.1 \\
0.07 & 0.1 & 0.07
\end{bmatrix}, \quad z = \begin{bmatrix} 0 \end{bmatrix}
\]  

(373)

In the case where \( p = 0.08, \) \( q = 0.08, \) \( r = 0.36, \) we have the following CNN genes:

Inverse half-toning CNN genes

\[
\begin{align*}
A_0\text{-gene} : & \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0, \\
B\text{-gene} : & \quad 0.08 \quad 0.08 \quad 0.08 \quad 0.08 \quad 0.36 \quad 0.08 \quad 0.08 \quad 0.08 \quad 0.08, \\
z\text{-gene} : & \quad 0.
\end{align*}
\]  

(374)
or equivalently

\[
\begin{array}{c}
\text{Inverse half-toning CNN template} \\
A_0 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
0.08 & 0.08 & 0.08 \\
0.08 & 0.36 & 0.08 \\
0.08 & 0.08 & 0.08 \\
\end{bmatrix}, \quad z = \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}.
\end{array}
\] (375)

**Design B:**

In this design, we use the CNN genes without diagonal elements:

\[
\begin{align*}
\{ A_0 \text{-gene} : & \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \\
\{ B \text{-gene} : & \begin{bmatrix}
p & 0 & p & r & p & 0 & p & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \\
\{ z \text{-gene} : & 0,
\end{align*}
\] (376)

or equivalently

\[
\begin{array}{c}
A_0 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
p & 0 & 0 \\
p & r & p \\
p & 0 & 0 \\
\end{bmatrix}, \quad z = \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}.
\end{array}
\] (377)

The parameters \( p, q \) and \( r \) satisfy

\[ r = 1 - 4p, \quad 0 < r < 1, \quad 0 < p < 1. \] (378)

Hence, we have

\[
\sum_{k,l \in N_{ij}} b_{kl} = 4p + r = 1, \quad 0 < p < \frac{1}{4}.
\] (379)

In this case, the diagonal elements are not used in averaging operations. If we choose \( p = 0.18, \) \( r = 38, \) then we get the following CNN genes:

\[
\begin{array}{c}
\text{Inverse half-toning CNN genes} \\
\{ A_0 \text{-gene} : \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \\
\{ B \text{-gene} : \begin{bmatrix}
0.18 & 0 & 0.18 & 0.38 & 0.18 & 0.18 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \\
\{ z \text{-gene} : \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix},
\end{array}
\] (380)

or equivalently

\[
\begin{array}{c}
\text{Inverse half-toning CNN template} \\
A_0 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
0.18 & 0 & 0.18 \\
0.18 & 0.38 & 0.18 \\
0.18 & 0.18 & 0 \\
\end{bmatrix}, \quad z = \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}.
\end{array}
\] (381)
Let us consider next the design of complementary half-toning genes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Half-Toning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global task</td>
<td>Transform a given gray-scale image $P_1$ into a “half-tone” binary image. The CNN input image and the initial state are given by $P_1$, and the boundaries are constrained to $-1$.</td>
</tr>
<tr>
<td>Local operations</td>
<td>$(a)$ If the averaged value of outputs is greater (resp. smaller) than the averaged value of the inputs, then the output $y_{ij}$ of cell $C_{ij}$ is printed in white (black).</td>
</tr>
<tr>
<td>Example</td>
<td>Fig. 22</td>
</tr>
<tr>
<td>Reference</td>
<td>[Chua, 1998]</td>
</tr>
</tbody>
</table>

**Design A:**
This operation can be realized by using the following CNN genes:

\[
\begin{align*}
\mathcal{A}-\text{gene} : & \quad -q & -p & -q & -p & -r + 4 & -p & -q & -p & -q \\
\mathcal{B}-\text{gene} : & \quad q & p & q & p & r & p & q & p & q \\
\mathcal{Z}-\text{gene} : & \quad 0
\end{align*}
\]
(382)

or equivalently

\[
\begin{align*}
A &= \begin{bmatrix}
-q & -p & -q \\
-p & -r + 4 & -p \\
-q & -p & -q
\end{bmatrix}, \\
B &= \begin{bmatrix}
q & p & q \\
p & r & p \\
q & p & q
\end{bmatrix}, \\
z &= \begin{bmatrix}
0
\end{bmatrix}
\end{align*}
\]
(383)

Here, the parameters $p$, $q$, and $r$ satisfy

\[
r = 1 - 4p - 4q, \quad 0 < r < 1, \quad 0 < p < 1, \quad 0 < q < 1.
\]
(384)

Fig. 20. Horizontal-hole detection.

Fig. 21. Half-toning.

Fig. 22. Half-toning.
Hence, we have

\[ \sum_{k,l \in N_{ij}} b_{kl} = 4p + 4q + r = 1, \]

\[ 0 < p < \frac{1}{4}, \quad 0 < q < \frac{1}{4}. \]  

(385)

If we choose \( p = 0.1, q = 0.07, r = 0.32 \), we can get the following CNN genes:

<table>
<thead>
<tr>
<th>Half-toning CNN genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )-gene :</td>
</tr>
<tr>
<td>\begin{bmatrix} -0.07 &amp; -0.1 &amp; -0.07 &amp; -0.1 &amp; 3.68 &amp; -0.1 &amp; -0.7 &amp; -0.1 &amp; -0.07 \end{bmatrix}</td>
</tr>
<tr>
<td>( B )-gene :</td>
</tr>
<tr>
<td>\begin{bmatrix} 0.07 &amp; 0.1 &amp; 0.07 &amp; 0.1 &amp; 0.32 &amp; 0.1 &amp; 0.07 &amp; 0.1 &amp; 0.07 \end{bmatrix}</td>
</tr>
<tr>
<td>( z )-gene :</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

(386)

or equivalently

\[
\begin{align*}
A &= \begin{bmatrix}
-0.07 & -0.1 & -0.07 \\
-0.1 & 3.68 & -0.1 \\
-0.07 & -0.1 & -0.07
\end{bmatrix}, \\
B &= \begin{bmatrix}
0.07 & 0.1 & 0.07 \\
0.1 & 0.32 & 0.1 \\
0.07 & 0.1 & 0.07
\end{bmatrix}, \\
z &= 0.
\end{align*}
\]  

(387)

**Design B:**

In this design, we use the following CNN genes:

\[
\begin{align*}
A &= \begin{bmatrix}
0 & -p & 0 & -p & -r + 4 & -p & 0 & -p & 0 \\
-p & -r + 4 & -p
\end{bmatrix}, \\
B &= \begin{bmatrix}
0 & p & 0 \\
p & r & p \\
0 & p & 0
\end{bmatrix}, \\
z &= 0.
\end{align*}
\]  

(388)

or equivalently

\[
\begin{align*}
A &= \begin{bmatrix}
0 & -p & 0 & 0 & -p & 0 & -p & 0 & -p & 0 \\
-p & -r + 4 & -p
\end{bmatrix}, \\
B &= \begin{bmatrix}
0 & p & 0 \\
p & r & p \\
0 & p & 0
\end{bmatrix}, \\
z &= 0.
\end{align*}
\]  

(389)

Here, the parameters \( p, q \) and \( r \) satisfy

\[ r = 1 - 4p, \quad 0 < r < 1, \quad 0 < p < 1. \]  

(390)

Hence, we have

\[ \sum_{k,l \in N_{ij}} b_{kl} = 4p + r = 1, \]

\[ 0 < p < \frac{1}{4}. \]  

(391)
In these genes, the diagonal elements are not used in averaging operations. If we choose $p = 0.18$, $r = 38$, then we get the following CNN genes:

\[
\begin{align*}
\text{Half-tonging CNN genes} \\
A\text{-gene} & : 0 \begin{array}{cccccccc}
-0.18 & 0 & -0.18 & 3.72 & -0.18 & 0 & -0.18 & 0
\end{array}, \\
B\text{-gene} & : 0 \begin{array}{cccccccc}
0.18 & 0 & 0.18 & 0.28 & 0.18 & 0 & 0.18 & 0
\end{array}, \\
z\text{-gene} & : 0
\end{align*}
\]

or equivalently

\[
\begin{align*}
\text{Half-tonging CNN template} \\
A = & \begin{array}{cccccccc}
0 & -0.18 & 0 & -0.18 & 3.72 & -0.18 & 0 & -0.18
\end{array}, & B = & \begin{array}{cccccccc}
0 & 0.18 & 0 & 0.18 & 0.28 & 0.18 & 0 & 0.18
\end{array}, & z = & 0
\end{align*}
\]

6. Design of CNN Genes via Genetic Algorithm Approach

It is well known that the genetic algorithm is superior to many methods for solving searching problems. In this section, the genetic algorithm is applied to the design of CNN genes [Itoh & Chua, 2003]. First, let us introduce several genetic operations: a mutation, a crossover, an inversion, a copy, an activation and an inactivation and scaling. The “activation and inactivation” operations are new genetic operations, which are introduced to generate useful CNN genes effectively.

- A mutation operation changes the sign of one element $a_{ij}$

\[
\text{mutation} \begin{array}{cccccccc}
a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1}
\end{array} \begin{array}{cccccccc}
a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & -a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1}
\end{array}
\]

In this example, a mutation occurs at $a_{0,1}$.

- A crossover operation exchanges randomly selected part of CNN genes

\[
\begin{align*}
\text{crossover} \quad \begin{array}{cccccccc}
gene\ C : & c_{-1,-1} & c_{-1,0} & c_{-1,1} & c_{0,-1} & c_{0,0} & c_{0,1} & c_{1,-1} & c_{1,0} & c_{1,1}
gene\ D : & d_{-1,-1} & d_{-1,0} & d_{-1,1} & d_{0,-1} & d_{0,0} & d_{0,1} & d_{1,-1} & d_{1,0} & d_{1,1}
gene\ C' : & c_{-1,-1} & c_{-1,0} & c_{-1,1} & c_{0,-1} & c_{0,0} & c_{0,1} & d_{1,-1} & d_{1,0} & d_{1,1}
gene\ D' : & d_{-1,-1} & d_{-1,0} & d_{-1,1} & d_{0,-1} & d_{0,0} & d_{0,1} & c_{1,-1} & c_{1,0} & c_{1,1}
\end{array}
\end{align*}
\]

Here, a crossover point is denoted by a partition $||$, which is located between $c_{0,1}$ and $c_{1,-1}$ ($d_{0,1}$ and $d_{1,-1}$).
• An inactivation operation changes the element $a_{ij}$ into 0, that is, $a_{ij}$ is switched off

\[
\begin{array}{cccccccc}
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1} \\
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & 0 & a_{1,-1} & a_{1,0} & a_{1,1} \\
\end{array}
\]

(396)

In this example, an inactivation occurs at $a_{0,1}$.

• An activation operation performs the reverse of the “inactivation” operation, that is, $a_{ij}$ is switched on

\[
\begin{array}{cccccccc}
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & 0 & a_{1,-1} & a_{1,0} & a_{1,1} \\
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1} \\
\end{array}
\]

(397)

In this example, an activation occurs at $a_{0,1}$.

• An inversion operation inverts a part of the gene

\[
\begin{array}{cccccccccccc}
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1} \\
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1} \\
\end{array}
\]

(398)

In this example, an inversion occurs between $a_{-1,1}$ and $a_{0,1}$.

• A scaling operation changes the value of an element $a_{ij}$ to other nonzero values

\[
\begin{array}{cccccccccccc}
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{1,-1} & a_{1,0} & a_{1,1} \\
  a_{-1,-1} & a_{-1,0} & a_{-1,1} & b_{0,-1} & b_{0,0} & b_{0,1} & b_{1,-1} & b_{1,0} & b_{1,1} \\
\end{array}
\]

(399)

In this example, a scaling operation occurs at $a_{0,-1}$, and the element $a_{0,-1}$ is changed to $b_{0,-1}$.

• A copy operation reproduces oneself, or a part of it, and joins it to oneself

\[
\begin{array}{cccc}
  a & b & c \\
  a & b & c \\
\end{array}
\]

(400)

\[
\begin{array}{cccc}
  a & b & c & d & e & f \\
  a & b & c & d & e & f \\
\end{array}
\]

(401)

These genetic operations can occur at random locations in CNN genes.
Next, let us illustrate the relation between the basic CNN genes from the viewpoint of genetic algorithms.

Here, the symbol $0$ indicates a zero-dimensional CNN gene, which is the origin of the CNN gene universe.

In the following section, we give some examples demonstrating how the genetic operations can result in new useful genes, which could not be found otherwise. Furthermore, we demonstrate that many useful CNN genes can be obtained systematically from known CNN genes by fewer genetic operations.

6.1. **Mutation of CNN genes**

A mutation operation changes the sign of the element $a_{ij}$ of CNN genes. If we apply this operation to one-dimensional trapezoidal $A_2$-gene, we get the following results.

- **mutation to a Laplacian CNN gene**

\[
\begin{pmatrix}
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0
\end{pmatrix}
\]

\[(402)\]

- **mutation to a horizontal hole detection CNN gene**

\[
\begin{pmatrix}
0 & 0 & 0 & 1 & 2 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & -1 & 0 & 0 & 0
\end{pmatrix}
\]

\[(403)\]

- **mutation to a horizontal hole detection CNN gene**

\[
\begin{pmatrix}
0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 2 & 1 & 0 & 0 & 0
\end{pmatrix}
\]

\[(404)\]

In these examples, a mutation occurred at the element denoted by $|2|$ or $|1|$. The first CNN gene can mutate to a Laplacian CNN gene, and the
next 2 CNN genes can mutate to a *horizontal hole* detection CNN gene. Their template expression is given by

\[
\begin{array}{cccc}
0 & 0 & 0 \\
1 & 2 & 1 \\
0 & 0 & 0 \\
\end{array}
\]

* mutation to a *Laplacian* CNN gene

\[
\begin{array}{cccc}
0 & 0 & 0 \\
1 & -2 & 1 \\
0 & 0 & 0 \\
\end{array}
\]

* mutation to a *horizontal hole detection* CNN gene

\[
\begin{array}{cccc}
0 & 0 & 0 \\
1 & 2 & -1 \\
0 & 0 & 0 \\
\end{array}
\]

In the two-dimensional case, a trapezoidal \( A_2 \)-gene can mutate to a two-dimensional Laplacian template

\[
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{array}
\]

* mutation \[
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{array}
\]

In this example, a mutation operation is applied to the center element denoted by \(||4||\). The two-dimensional *Laplacian* templates can be used in Chua’s reaction–diffusion CNN equations [Chua, 1998]. Its template
expression is given by

\[
\begin{array}{c|c|c}
0 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 0 \\
\hline
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
A_3\text{-gene} : & \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\
\end{array} \\
B_1\text{-gene} : & \begin{array}{cccccccc}
0 & -1 & 0 & -1 & 6 & -1 & 0 & -1 & 0 \\
\end{array} \\
z\text{-gene} : & -8 \\
\end{array}
\]

Next, let us consider the global connectivity detection CNN genes [Chua, 1998]

\[
\begin{array}{c|c|c}
A\text{-gene} : & \begin{array}{cccccccc}
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
\end{array} \\
B_1\text{-gene} : & \begin{array}{cccccccc}
0 & 1 & 0 & 1 & -6 & 1 & 0 & 1 & 0 \\
\end{array} \\
z\text{-gene} : & -8 \\
\end{array}
\]

If we apply the mutation operations to the \(A_3\)-gene and \(B_1\)-gene, then we get the new edge extraction CNN gene

\[
\begin{array}{c|c|c}
A\text{-gene} : & \begin{array}{cccccccc}
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
\end{array} \\
B_1\text{-gene} : & \begin{array}{cccccccc}
0 & 1 & 0 & 1 & -6 & 1 & 0 & 1 & 0 \\
\end{array} \\
z\text{-gene} : & -8 \\
\end{array}
\]

These CNN genes can extract edges corresponding to local extremum points of patterns which are globally connected in one contiguous piece (see Fig. 23). Their template expression is given by

\[
A = \begin{array}{cccc}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0 \\
\end{array}, \quad B_1 = \begin{array}{cccc}
0 & 1 & 0 \\
1 & -6 & 1 \\
0 & 1 & 0 \\
\end{array}, \quad z = -8.
\]

The following inactivated CNN gene has also the same property

\[
A = \begin{array}{cccc}
0 & -1 & 0 \\
0 & 4 & 0 \\
0 & 0 & 0 \\
\end{array}, \quad B_1 = \begin{array}{cccc}
0 & 1 & 0 \\
1 & -6 & 1 \\
0 & 1 & 0 \\
\end{array}, \quad z = -5.
\]
6.2. Inactivation of CNN genes

In this section, we apply the inactivation operations to four kinds of CNN genes, namely, selected objects extraction, hole-filling, filled contour extraction, and face-vase-illusion CNN genes. The inactivation operation introduces some functional disorders to the CNN genes.

Let us consider first the selected objects extraction CNN genes

\[
\begin{align*}
A_3\text{-gene} : & \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B_1\text{-gene} : & \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
z\text{-gene} : & \begin{bmatrix} 0 \end{bmatrix} 
\end{align*}
\]

If we apply the inactivation operation to the \(A_3\)-gene of Eq. (412), we get the following four genes

\[
\begin{align*}
\text{(a)} & \quad A\text{-gene} : \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad B\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad z\text{-gene} : \begin{bmatrix} 0 \end{bmatrix} \\
\text{(b)} & \quad A\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad B\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad z\text{-gene} : \begin{bmatrix} 0 \end{bmatrix} \\
\text{(c)} & \quad A\text{-gene} : \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad B\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad z\text{-gene} : \begin{bmatrix} 0 \end{bmatrix} \\
\text{(d)} & \quad A\text{-gene} : \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 \end{bmatrix} \\
& \quad B\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad z\text{-gene} : \begin{bmatrix} 0 \end{bmatrix} \\
\text{(e)} & \quad A\text{-gene} : \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad B\text{-gene} : \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\
& \quad z\text{-gene} : \begin{bmatrix} 0 \end{bmatrix}
\end{align*}
\]

Here, the symbol \(||0||\) indicates the position where the inactivation occurred. Their template expression is given by

\[
\begin{align*}
A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, & \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \quad z = \begin{bmatrix} 0 \end{bmatrix} \\
A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, & \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \quad z = \begin{bmatrix} 0 \end{bmatrix} \\
A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 1 & 0 \end{bmatrix}, & \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \quad z = \begin{bmatrix} 0 \end{bmatrix}
\end{align*}
\]
Here, the symbol $0$ indicates the position where the inversion occurred. Observe that these CNN genes have interesting behaviors as shown in Fig. 24.

Next, we apply the inversion operation to the $A_3$-gene of Eq. (412)

\[
\text{inversion } \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \\
\end{bmatrix}
\]

In this example, the inversion operations are applied to the following two sections $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. Thus, we get the new selected objects extraction CNN genes with diagonal elements.\(^4\)

\[
\text{\begin{align*}
A_6\text{-gene} : & \begin{bmatrix} 1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \end{bmatrix} \\
B_1\text{-gene} : & \begin{bmatrix} 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \end{bmatrix} \\
z\text{-gene} : & \begin{bmatrix} 0 \end{bmatrix} \\
\end{align*}}
\]

\[
\text{(416)}
\]

Their template expression is given by

\[
\text{\begin{align*}
A_4 & = \begin{bmatrix} 1 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 1 \\
0 & 4 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
\end{bmatrix} \\
B_1 & = \begin{bmatrix} 0 & 0 & 0 \\
0 & 6 & 0 \\
0 & 0 & 0 \\
\end{bmatrix} \\
z & = \begin{bmatrix} 0 \end{bmatrix} \\
\end{align*}}
\]

\[
\text{(417)}
\]

If we apply the inactivation operation to the CNN genes (416), we would obtain the following CNN genes:

\[
\text{\begin{align*}
A\text{-gene} : & \begin{bmatrix} 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \end{bmatrix} \\
z\text{-gene} : & \begin{bmatrix} 0 \end{bmatrix} \\
\end{align*}}
\]

\[
\text{(418)}
\]

Here, the symbol $\big|0\big|$ indicates the position where the inactivation occurred. Their template expression is given by

\[
\text{\begin{align*}
A & = \begin{bmatrix} 0 & 0 & 0 \\
0 & 4 & 0 \\
1 & 0 & 1 \\
\end{bmatrix} \\
B & = \begin{bmatrix} 0 & 0 & 0 \\
0 & 6 & 0 \\
1 & 0 & 1 \\
\end{bmatrix} \\
z & = \begin{bmatrix} 0 \end{bmatrix} \\
\end{align*}}
\]

\[
\text{(419)}
\]

\(^4\)We can get the same CNN gene by applying both the activation and inactivation operations to the gene (413).
Fig. 24. Image processing of the inactivated selected objects extraction CNN genes. See Appendix A.1 for the CNN parameters.
Here, the symbol 0 indicates the position where the inactivation occurred. Observe that these inactivated CNN genes have interesting properties as shown in Fig. 24.

If we change the parameters of the \( B \)-gene and \( z \)-gene in the selected objects extraction CNN genes (412), that is, apply the scaling operations to these CNN genes, we can obtain the hole-filling, filled contour extraction, and face-vase-illusion CNN genes.

\[
A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad z = [0]
\]

selected objects extraction CNN genes \( \implies \) scaling \{ hole-filling CNN genes \}
\{ filled contour extraction CNN genes \}
\{ face-vase-illusion CNN genes \}

where

\[
\begin{align*}
\text{hole-filling CNN genes} & \quad \begin{cases} A_3 \text{-gene} : & [0 \ 1 \ 0 \ 1 \ 4 \ 1 \ 0 \ 1 \ 0], \\ B_1 \text{-gene} : & [0 \ 0 \ 0 \ 0 \ 5 \ 0 \ 0 \ 0 \ 0], \\ z \text{-gene} : & [-1] \end{cases} \\
\text{filled contour extraction CNN genes} & \quad \begin{cases} A_3 \text{-gene} : & [0 \ 1 \ 0 \ 1 \ 4 \ 1 \ 0 \ 1 \ 0], \\ B_1 \text{-gene} : & [0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0], \\ z \text{-gene} : & [-4] \end{cases} \\
\text{face-vase-illusion CNN genes} & \quad \begin{cases} A_3 \text{-gene} : & [0 \ 1 \ 0 \ 1 \ 4 \ 1 \ 0 \ 1 \ 0], \\ B_1 \text{-gene} : & [0 \ 0 \ 0 \ 0 \ -6 \ 0 \ 0 \ 0 \ 0], \\ z \text{-gene} : & [0] \end{cases}
\end{align*}
\]

Observe that the inactivated genes have interesting image processing properties as shown in Figs. 25–27. Although the inactivation operations introduce some functional disorders, they also generate many useful CNN genes. By applying the activation operations or the second (or third) inactivation operations to those CNN genes, we can also obtain other interesting results.

6.3. Signal propagation property of CNN genes

In this section, we study the signal propagation of CNN genes from the viewpoint of gene structures.

6.4. Inactivated CNN genes

Let us consider the selected object extraction CNN
Fig. 25. Image processing of the inactivated hole-filling CNN genes. See Appendix A.2 for the CNN parameters.
Fig. 26. Image processing of the inactivated filled contour extraction CNN genes. See Appendix A.3 for the CNN parameters.
Fig. 27. Image processing of the inactivated face-vase-illusion CNN genes. See Appendix A.4 for the CNN parameters.
Designing CNN Genes

and its inactivated CNN genes

\[
\begin{align*}
\text{(a)} & & \text{A-gene : } & 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\ & & \text{B-gene : } & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ & & \text{z-gene : } & 0 & . \\
\text{(b)} & & \text{A-gene : } & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\ & & \text{B-gene : } & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ & & \text{z-gene : } & 0 & .
\end{align*}
\]

The dynamics of the CNN genes (b) is written as

\[
\frac{dx_{ij}}{dt} = g(x_{ij}) + y_{i,j-1} + y_{i,j+1} + y_{i+1,j} + 6u_{ij},
\]

where

\[
g(x_{ij}) = \begin{cases} 
-x_{ij} - 4, & x_{ij} \leq -1 \\
3x_{ij}, & |x_{ij}| \leq 1 \\
-x_{ij} + 4, & x_{ij} \geq 1.
\end{cases}
\]

The nonlinear function \( g(x) \) has a local maximum (3 at \( x = 1 \)), and a local minimum (–3 at \( x = -1 \)). For \( u_{ij} = 1 \), we have the following relation

\[
3 \leq y_{i,j-1} + y_{i,j+1} + y_{i+1,j} + 6u_{ij} \leq 9. \tag{425}
\]

Recall that the parameters of the selected objects extraction CNN genes (a) were chosen such that they can perform the following operation:

- If the input \( u_{ij} \) is coded in black and at least one neighbor is colored in black, then the output \( y_{ij} \) is printed in black.

However, the genes (b) cannot implement this operation completely. That is, the output \( y_{ij} \) is not printed in black, if the following conditions are satisfied

1. The output \( y_{i-1,j} \) is colored black and other cells are colored in white, that is,
   \[
y_{i-1,j} = 1, \quad y_{i,j-1} = -1, \\
y_{i,j+1} = -1, \quad y_{i+1,j} = -1.
\]
2. The cell \( C_{ij} \) is not marked, and so \( x_{ij}(0) = -1 \).

In this case, we get \( y_{i,j-1} + y_{i,j+1} + y_{i+1,j} + 6u_{ij} = 3 \). From the driving-point plot analysis, we conclude that the trajectory starting from \( x_{ij}(0) = -1 \) stays at \( -1 \), and the output \( y_{ij} \) is printed in white. Therefore, the north cell \( (y_{i-1,j}) \) does not have an influence on the center cells. This means that the signal flow (that is, the color change from white to black) does not move to the south from the north (¶). Similarly, we can obtain the following table

<table>
<thead>
<tr>
<th>CNN genes</th>
<th>possible signal propagation direction</th>
</tr>
</thead>
</table>
| (a) \[
\begin{align*}
\text{A-gene : } & 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\ 
\text{B-gene : } & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 
\text{z-gene : } & 0 & .
\end{align*}
\] | \( \uparrow \) \( \downarrow \) \( \equiv \) \( \Rightarrow \) |
| (b) \[
\begin{align*}
\text{A-gene : } & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\ 
\text{B-gene : } & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 
\text{z-gene : } & 0 & .
\end{align*}
\] | \( \uparrow \) \( \equiv \) \( \Rightarrow \) |

(427)
See Fig. 24 for the signal processing property of these CNN genes. Note that although the activation operations introduce a *functional disorder* to CNN genes, they also generate many useful CNN genes possessing other functionalities. We can obtain similar results for the face-vase-illusion, hole-filling, and filled contour extraction CNN genes, as shown in Figs. 25–27.

The functions of the inactivated selected object extraction CNN genes (in Appendix A.1) can be described as follows:

- **Functions of the inactivated selected object extraction CNN genes**
  - The CNN genes (a) extract marked objects, but do not extract diagonal objects (branches).
  - The CNN genes (b) extract upward objects (branches), but do not extract either downward or diagonal objects (branches).
  - The CNN genes (c) extract right objects (branches), but do not extract either left or diagonal objects (branches).
  - The CNN genes (d) extract downward objects (branches), but do not extract either upward or diagonal objects (branches).
  - The CNN genes (e) extract left objects (branches), but do not extract either right or diagonal objects (branches).

The functions of the inactivated hole-filling CNN genes (a)–(e) in Appendix A.2 and Fig. 25 can be described as follows:

<table>
<thead>
<tr>
<th>CNN genes</th>
<th>possible signal propagation direction</th>
</tr>
</thead>
</table>
| (c)       | [A-gene: \( \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 0 & 0 & 1 \\
\end{array} \) ] | ↑  ↓  ⇒ |
|           | [B-gene: \( \begin{array}{cccccccc}
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
\end{array} \) ] |
|           | [z-gene: \( \begin{array}{cccc}
0 & \\
\end{array} \) ] |
| (d)       | [A-gene: \( \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 0 \\
\end{array} \) ] | ↓  ⇐  ⇒ |
|           | [B-gene: \( \begin{array}{cccccccc}
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
\end{array} \) ] |
|           | [z-gene: \( \begin{array}{cccc}
0 & \\
\end{array} \) ] |
| (e)       | [A-gene: \( \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 0 & 4 & 1 & 0 \\
\end{array} \) ] | ↑  ↓  ⇐ |
|           | [B-gene: \( \begin{array}{cccccccc}
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
\end{array} \) ] |
|           | [z-gene: \( \begin{array}{cccc}
0 & \\
\end{array} \) ] |
Functions of the inactivated hole-filling CNN genes
- The CNN genes (a) fill the interior of closed contours.
- The CNN genes (b) fill the interior of both closed contours and north open contours.
- The CNN genes (c) fill the interior of both closed contours and east open contours.
- The CNN genes (d) fill the interior of both closed contours and south open contours.
- The CNN genes (e) fill the interior of both closed contours and west open contours.

The functions of the inactivated hole-filling CNN genes (a)–(e) in Appendix A.3 and Fig. 26 can be described as follows:

Functions of the inactivated filled contour extraction CNN genes
- The CNN genes (a) extract from an image fed to initial states the regions which fill the interior of all closed curves of an image loaded to CNN inputs.
- The CNN genes (b) extract from an image fed to initial states the regions which fill the interior of all closed curves and the north open contour of an image loaded to CNN inputs.
- The CNN genes (c) extract from an image fed to initial states the regions which filled the interior of all closed curves and the east open contour of an image loaded to CNN inputs.
- The CNN genes (d) extract from an image fed to initial states the regions which fill the interior of all closed curves and the south open contour of an image loaded to CNN inputs.
- The CNN genes (e) extract from an image fed to initial states the regions which fill the interior of all closed curves and the west open contour of an image loaded to CNN inputs.

The functions of the inactivated hole-filling CNN genes (a)–(e) in Appendix A.4 and Fig. 27 can be described as follows:

Functions of the inactivated face-vase-illusion CNN genes
- The CNN genes (a) pick out the pixels which are located inside the marked objects.
- The CNN genes (b) pick out the pixels which are located inside the marked objects and are also located above the pixels which are marked by the initial state.
- The CNN genes (c) pick out the pixels which are located inside the marked objects and are also located on the right side of the pixels which are marked by the initial state.
- The CNN genes (d) pick out the pixels which are located inside the marked objects and are also located below the pixels which are marked by the initial state.
- The CNN genes (e) pick out the pixels which are located inside the marked objects and are also located on the left side of the pixels which are marked by the initial state.
Next we study the signal flow of genes with two inactivation operations. Let us consider the inactivated selected objects extraction CNN genes. Let us consider the activated genes.

In this subsection, we study the activated selected objects extraction CNN genes. Let us consider the activated genes.

\[ \begin{align*}
A\text{-gene} & : \begin{bmatrix}
0 & 0 & 0 & 1 & 4 & 1 & 0 & 1 & 0
\end{bmatrix}, \\
B\text{-gene} & : \begin{bmatrix}
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \\
z\text{-gene} & : 0.
\end{align*} \]

(428)

Here, the inactivation operations are denoted by the symbol \(|0|\). Their template expression is given by

\[ \begin{align*}
(k) & \\
A & = 0 \quad 0 \quad 0 \\
1 & 4 \quad 0 \\
0 & 1 \quad 0
\end{align*} \]

\[ B_1 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 6 & 0 \\
0 & 0 & 0
\end{bmatrix} \]

\[ z = 0. \]

(429)

The dynamics of the above CNN genes \((k)\) is written as

\[ \frac{dx_{ij}}{dt} = g(x_{ij}) + y_{i-1,j-1} + y_{i,j-1} + y_{i, j+1} + 6u_{ij}. \]

(430)

For \(u_{ij} = 1\), we have the following relation

\[ 4 \leq y_{i,j-1} + y_{i+1,j} + 6u_{ij} \leq 8. \]

(431)

Therefore, if the input is coded black, the output is colored in black. This implies that the output image is identical to the input image. That is, the output image becomes the input image.

6.5. Activated CNN genes

In this subsection, we study the activated selected objects extraction CNN genes. Let us consider the activated genes

\[ \begin{align*}
A_1\text{-gene} & : \begin{bmatrix}
1 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0
\end{bmatrix}, \\
B'\text{-gene} & : \begin{bmatrix}
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \\
z\text{-gene} & : 0.
\end{align*} \]

(432)

Here, the activation operation is denoted by the symbol \(|1|\).

The dynamics of the CNN genes \((b')\) is written as

\[ \frac{dx_{ij}}{dt} = g(x_{ij}) + y_{i-1,j-1} + y_{i,j-1} + y_{i,j+1} + 6u_{ij} + y_{i+1,j} + y_{i+1,j} + 6u_{ij}. \]

(433)

Remember that the parameters of the selected objects extraction CNN genes were chosen such that they can perform the following operation:

- If the input \(u_{ij}\) is coded in black and at least one neighbor is colored in black, then the output \(y_{ij}\) is printed in black.

However, the genes \((b')\) cannot perform this operation completely. That is, if \(u_{ij} = 1\), we have the following relation

\[ 3 \leq y_{i,j-1} + y_{i,j-1} + y_{i, j+1} + y_{i+1,j} + 6u_{ij} \leq 9. \]

(434)

Thus, from the driving-point plot analysis, we conclude that the output \(y_{ij}\) is not colored in black even if only one neighbor cell is colored black; for example

\[ y_{i-1,j-1} = 1, \quad y_{i-1,j} = -1, \quad y_{i,j-1} = -1, \quad y_{i,j+1} = -1, \quad y_{i+1,j} = -1. \]

(435)

and hence

\[ y_{i-1,j-1} + y_{i,j-1} + y_{i,j+1} + y_{i+1,j} + 6u_{ij} = 3. \]

(436)

In order to color \(y_{ij}\) in black, we need at least two neighbors colored in black; for example,

\[ y_{i-1,j-1} = 1, \quad y_{i-1,j} = 1, \quad y_{i,j-1} = -1, \quad y_{i,j+1} = -1, \quad y_{i+1,j} = -1. \]

(437)

and so

\[ y_{i-1,j-1} + y_{i,j-1} + y_{i,j+1} + y_{i+1,j} + 6u_{ij} = 5. \]

(438)

Since the third column and third row of the template \(A\) has only one element, the signal flow (color change from white to black) cannot move to the north from the south (\(\downarrow\)), nor to the west from the east (\(\Rightarrow\)). We can obtain similar results for other activated CNN genes. Thus, we have the following table.
The image processing property of the CNN genes \((b')\)–\((e')\) is illustrated in Fig. 28. The functions of the activated selected objects extraction CNN genes \((a')–(e')\) in Appendix A.5 can be described as follows:

- The CNN genes \((b')\) extract the right bases of branches which are located below the pixels marked by the initial state.
- The CNN genes \((c')\) extract the left bases of branches which are located below the pixels marked by the initial state.
- The CNN genes \((d')\) extract the left bases of branches which are located above the pixels marked by the initial state.
- The CNN genes \((e')\) extract the right bases of branches which are located above the pixels marked by the initial state.

The signal flow propagation explains the image processing property of the activated selected objects extraction CNN genes partially. See Figs. 29–31 for the face-vase-illusion, hole-filling, and filled contour extraction CNN genes.
Fig. 28. Image processing of the activated selected objects extraction CNN genes. See Appendix A.5 for the CNN parameters.
(1) input image.  (2) initial state.  (3) CNN genes (a').  
(4) CNN genes (b').  (5) CNN genes (c').  (6) CNN genes (d').  
(7) CNN genes (e').  (8) CNN genes (f').  (9) CNN genes (g').  
(10) CNN genes (h').  (11) CNN genes (i').  (12) CNN genes (j').

Fig. 29. Image processing of the activated hole-filling CNN genes. See Appendix A.6 for the CNN parameters.
Figure 30: Image processing of the activated filled contour extraction CNN genes. See Appendix A.7 for the CNN parameters.

(1) input image. (2) initial state. (3) CNN genes (a'). (4) CNN genes (b'). (5) CNN genes (c'). (6) CNN genes (d'). (7) CNN genes (e'). (8) CNN genes (f'). (9) CNN genes (g'). (10) CNN genes (h'). (11) CNN genes (i'). (12) CNN genes (j').

Fig. 30. Image processing of the activated filled contour extraction CNN genes. See Appendix A.7 for the CNN parameters.
Fig. 31. Image processing of the activated face-vase-illusion CNN genes. See Appendix A.8 for the CNN parameters.
The functions of the activated hole-filling CNN genes $(a')$–$(e')$ in Appendix A.6 and Fig. 29 can be described as follows:

- Functions of the activated hole-filling CNN genes
  - The CNN genes $(a')$ fill the interior of all closed contours.
  - The CNN genes $(b')$ fill the interior of all closed contour, east open contours, and south open contours.
  - The CNN genes $(c')$ fill the interior of all closed contour, west open contours, and south open contours.
  - The CNN genes $(d')$ fill the interior of all closed contour, west open contours, and north open contours.
  - The CNN genes $(e')$ fill the interior of all closed contour and east open contours, and north open contours.

The functions of the activated hole-filling CNN genes $(a')$–$(e')$ in Appendix A.7 and Fig. 30 can be described as follows:

- Functions of the activated filled contour extraction CNN genes
  - The CNN genes $(a')$ extract from an image fed to initial states all those regions which fill completely closed contours of an image loaded to CNN inputs.
  - The CNN genes $(b')$ extract from an image fed to initial states all those regions which fill completely closed contours, east open contours, and south open contours of an image loaded to CNN inputs.
  - The CNN genes $(c')$ extract from an image fed to initial states all those regions which fill completely closed contours, west open contours, and south open contours of an image loaded to CNN inputs.
  - The CNN genes $(d')$ extract from an image fed to initial states all those regions which fill completely closed contour, west open contours, and north open contours of an image loaded to CNN inputs.
  - The CNN genes $(e')$ extract from an image fed to initial states all those regions which fill completely closed contour, east open contours, and north open contours of an input image.

The functions of the activated hole-filling CNN genes $(a')$–$(e')$ in Appendix A.8 and Fig. 31 can be described as follows:

- Functions of the activated face-vase-illusion CNN genes
  - The CNN genes $(a')$ pick out the pixels which are located inside the marked objects.
  - The CNN genes $(b')$ pick out the pixels located inside the specified objects and on the southwest of the pixels marked by the initial state.
  - The CNN genes $(c')$ pick out the pixels located inside the specified objects and on the southwest of the pixels marked by the initial state.
  - The CNN genes $(d')$ pick out the pixels located inside the specified objects and on the northwest of the pixels marked by the initial state.
  - The CNN genes $(e')$ pick out the pixels located inside the specified objects and are located on the northeast of the pixels marked by the initial state.
7. Conclusion

We have proposed a new method for finding CNN parameters with prescribed image processing properties. Many CNN genes with the same functions were derived by using this design methodology. Furthermore, we proposed a method for generating CNN genes from known genes by using genetic algorithms. Two new operations were introduced and added to the conventional genetic algorithms. We obtained many useful CNN genes systematically from the selected objects extraction, face-vase-illusion, hole-filling, and filled contour extraction CNN genes. Furthermore, the signal propagation of inactivated genes was studied.

Acknowledgments

This work was supported in part by the DURINT Contract N00014-01-0741 and the ONR MURI Contract N000-14-03-1-0698.

References


Appendices

A.1. Inactivated selected objects

CNN genes

\[
\begin{align*}
\text{(a) } & \begin{cases} 
A\text{-gene} : & 0 \ 0 \ 0 \ 1 \ 4 \ 1 \ 0 \ 1 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(b) } & \begin{cases} 
A\text{-gene} : & 0 \ 1 \ 0 \ 1 \ 4 \ 1 \ 0 \ 1 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(c) } & \begin{cases} 
A\text{-gene} : & 0 \ 1 \ 0 \ 1 \ 4 \ 1 \ 0 \ 0 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(d) } & \begin{cases} 
A\text{-gene} : & 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0 \ 1 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(e) } & \begin{cases} 
A\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(f) } & \begin{cases} 
A\text{-gene} : & 1 \ 0 \ 1 \ 0 \ 4 \ 0 \ 1 \ 0 \ 1 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(g) } & \begin{cases} 
A\text{-gene} : & 0 \ 0 \ 1 \ 0 \ 4 \ 0 \ 1 \ 0 \ 1 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(h) } & \begin{cases} 
A\text{-gene} : & 1 \ 0 \ 0 \ 0 \ 4 \ 0 \ 1 \ 0 \ 1 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(i) } & \begin{cases} 
A\text{-gene} : & 1 \ 0 \ 1 \ 0 \ 4 \ 0 \ 1 \ 0 \ 0 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases} \\
\text{(j) } & \begin{cases} 
A\text{-gene} : & 1 \ 0 \ 1 \ 0 \ 4 \ 0 \ 0 \ 0 \ 1 \\
B\text{-gene} : & 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \\
z\text{-gene} : & 0 
\end{cases}
\end{align*}
\]
### A.2. Inactivated hole-filling CNN genes

| (a) | \(A\)-gene : 0 1 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -1 |
| (b) | \(A\)-gene : 0 0 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -1 |
| (c) | \(A\)-gene : 0 1 0 1 4 1 0 0 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -1 |
| (d) | \(A\)-gene : 0 1 0 1 4 1 0 0 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -1 |
| (e) | \(A\)-gene : 0 1 0 0 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -1 |

### A.3. Inactivated filled contour CNN genes

| (a) | \(A\)-gene : 1 0 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 5 0 0 0 0 | \(z\)-gene : -4 |
| (b) | \(A\)-gene : 0 0 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 2 0 0 0 0 | \(z\)-gene : -4 |
| (c) | \(A\)-gene : 0 0 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 2 0 0 0 0 | \(z\)-gene : -4 |
| (d) | \(A\)-gene : 0 0 0 1 4 1 0 1 0 | \(B\)-gene : 0 0 0 0 2 0 0 0 0 | \(z\)-gene : -4 |

(A.3)  
(A.4)  
(A.5)
A.4. *Inactivated face-vase illusion*  
*CNN genes*

\[
\begin{align*}
(A.6) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(f) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
1 & 0 & 1 & 0 & 4 & 0 & 1 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(g) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 1 & 0 & 4 & 0 & 1 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(h) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
1 & 0 & 0 & 0 & 4 & 0 & 1 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(i) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
1 & 0 & 1 & 0 & 4 & 0 & 1 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(j) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
1 & 0 & 1 & 0 & 4 & 0 & 0 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : -4
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(A.7) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : 0
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
(A.8) & \quad \begin{cases}
A\text{-gene} & : \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 \\
\end{array} \\
B\text{-gene} & : \begin{array}{cccccccc}
0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 \\
\end{array} \\
z\text{-gene} & : 0
\end{cases} \\
\end{align*}
\]
A.5. Activated selected objects

CNN genes

\[
\begin{align*}
(A.5) &: & A\text{-gene} : & \begin{array}{cccccccc}
1 & 0 & 1 & 0 & 4 & 0 & 0 & 1
\end{array}, \\
B\text{-gene} : & \begin{array}{cccccccc}
0 & 0 & 0 & 0 & -6 & 0 & 0 & 0
\end{array}, \\
z\text{-gene} : & \begin{array}{cccccccc}
0
\end{array}.
\end{align*}
\]

(A.10)

A.6. Activated hole-filling CNN genes

\[
\begin{align*}
(A.6) &: & A\text{-gene} : & \begin{array}{cccccccc}
1 & 1 & 1 & 0 & 4 & 0 & 1 & 0
\end{array}, \\
B\text{-gene} : & \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 6 & 0 & 0 & 0
\end{array}, \\
z\text{-gene} : & \begin{array}{cccccccc}
0
\end{array}.
\end{align*}
\]

(A.11)
### A.7. Activated filled contour CNN genes

<table>
<thead>
<tr>
<th></th>
<th>A-gene</th>
<th>B-gene</th>
<th>z-gene</th>
</tr>
</thead>
<tbody>
<tr>
<td>d‘)</td>
<td>0 1 0 1 4 1 0 1 1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>e‘)</td>
<td>0 1 0 1 4 1 1 1 0</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>f‘)</td>
<td>1 0 1 0 4 0 1 0 1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>g‘)</td>
<td>1 1 1 0 4 0 1 0 1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>h‘)</td>
<td>1 0 1 0 4 1 1 0 1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>i‘)</td>
<td>1 0 1 0 4 0 1 1 1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>j‘)</td>
<td>1 0 1 1 4 0 1 0 1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

(A.12)

<table>
<thead>
<tr>
<th></th>
<th>A-gene</th>
<th>B-gene</th>
<th>z-gene</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a’)</td>
<td>0 1 0 1 4 1 0 1 0</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>(b’)</td>
<td>1 1 0 1 4 1 0 1 0</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>(c’)</td>
<td>0 0 0 0 2 0 0 0 0</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>(d’)</td>
<td>0 1 0 1 4 1 0 1 1</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>(e’)</td>
<td>0 0 0 0 2 0 0 0 0</td>
<td>-4</td>
<td></td>
</tr>
</tbody>
</table>

(A.13) (A.14)
A.8. Activated face-vase-illusion CNN genes

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 0 & 1 & 4 & 1 \\
\hline
(a') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 1 & 0 & 1 & 4 & 1 \\
\hline
(b') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 1 & 1 & 1 & 4 & 1 \\
\hline
(c') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 0 & 1 & 1 & 1 \\
\hline
(d') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 1 & 1 & 1 \\
\hline
(e') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 0 & 1 & 1 \\
\hline
(f') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 1 & 0 & 1 \\
\hline
(g') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 0 & 4 & 1 \\
\hline
(h') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 0 & 1 & 1 \\
\hline
(i') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
& \text{A-gene} & 1 & 0 & 1 & 0 & 4 & 1 \\
\hline
(j') & \text{B-gene} & 0 & 0 & 0 & 0 & -6 & 0 \\
\hline
& \text{z-gene} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]

(A.16)