Genetic algorithms with fuzzy fitness function for object extraction using cellular networks

Sankar K. Pal*, Dinabandhu Bhandari

Machine Intelligence Unit, Indian Statistical Institute, 203 B.B. Road, Calcutta - 700035, India

Received November 1993; revised January 1994

Abstract

Setting up a Cellular Neural Network (CNN) for a particular task needs a proper selection of circuit parameters (cloning template) which determines the dynamics of the network. The present paper provides a methodology, demonstrating the capability of Genetic Algorithms with a fuzzy fitness function, for automatic selection of cloning templates when a CNN is used in extracting object regions from noisy images. Fuzzy geometrical properties of image are used as the basis of fitness function. The proposed method relieves the CNN from using heuristics for the template selection procedure, and performs consistently well in noisy environments.

Key words: Object extraction; Cellular neural networks; Genetic algorithms; Grayness and spatial ambiguity measures

1. Introduction

A cellular neural network (CNN) is an unsupervised neural network proposed by Chua and Yang [1,2]. It is made of a massive aggregate of analog circuit components called cells. Any cell in a cellular neural network is connected only to its neighboring cells and interact only with them. Key features of this network are asynchronous processing, continuous time dynamics and local interaction between the cells (processing elements). Cells not directly connected together affect each other due to propagation effects. The continuous time feature of a CNN imparts it real time signal processing capability while the local interconnections make it amenable to VLSI implementation. Its application in image processing problems, viz., edge detection, noise removal, horizontal and vertical line detection has been reported in [2,7].

Note that setting up a CNN needs a proper selection of circuit parameters of cells. The set of circuit parameters is called cloning template which determines the dynamics of the network and its application domain. There is no standard method for selecting cloning template automatically for a CNN, although some heuristics have been suggested [1,2] for this task.

In this article an attempt is made to develop a methodology for demonstrating an application of genetic algorithms with fuzzy fitness function for performing image processing operations using a CNN.

*Corresponding author. E-mail: sankar@isical.ernet.in

0165-0114/94/$07.00 © 1994—Elsevier Science B.V. All rights reserved
SSDI 0165-0114(94)00008-U
The problem of object extraction has been considered where the required cloning templates are automatically selected using genetic algorithms (GAs). Besides these, some guidelines for selecting heuristically the circuit parameters for the object extraction problem have also been provided. The algorithm uses both spatial (fuzzy compactness and index of area coverage (IOAC)) and grayness (entropy) ambiguity measures as the basis of the fitness function of GAs. This method is unsupervised and unlike the one described in [4], it does not need any prior information of the desired (target) output. The results obtained using the heuristically selected cloning templates are compared with those obtained using the templates selected by GA. It has been found that the performance of the later one is consistently better under different noisy conditions. Although the method developed for selection of cloning template has been used for object extraction problem, it can be applied to other image processing operations.

2. Cellular neural network

An $M \times N$ cellular neural network can be considered as an ordered set $\{C_{ik}\}$ of $MN$ cells which are arranged in a pattern shown in Figure 1. For a two dimensional CNN, $r$-neighborhood ($N'_r$) can be defined as

$$N'_r = \{C_{kl} \mid \max (|k-i|, |l-j|) \leq r, 1 \leq k \leq M; 1 \leq l \leq N\}.$$ 

The state $v_{xi}$ of any cell $C_{ij}$ in a CNN is described by the differential equation

$$C \frac{dv_{xi}(t)}{dt} = - \frac{1}{R_x} v_{xi}(t) + \sum_{C_{kl} \in N'_r} A_{ijkl} v_{ykl}(t) + I_{ij}, \quad 1 \leq i \leq M; 1 \leq j \leq N,$$

(1)

where $A_{ijkl}$ represents the conductance of the link between $C_{ij}$ and $C_{kl}$, $I_{ij}$ is the bias to the cell $C_{ij}$. $R_x$ and $C$ (constants for a CNN) are the resistance and capacitance of the cell. $C \cdot R_x$ is the time constant of the circuit and $R_x$ determines the power dissipation.

The output $v_{yi}(t)$ of the cell $C_{ij}$ is a nonlinear function $f(v_{xi}(t))$ shown in Figure 2. It can be expressed as

$$v_{yi}(t) = f(v_{xi}(t)) = 0.5(v_{xi}(t) + 1) - |v_{xi}(t) - 1|, \quad 1 \leq i \leq M; 1 \leq j \leq N,$$

(2)

with the constraint

$$|v_{xi}(0)| \leq 1, \quad 1 \leq i \leq M; 1 \leq j \leq N.$$

The state of each cell in a CNN is bounded and after the transients settle down, a CNN approaches to a stable state [1, 2]. Moreover, if the circuit parameters satisfy

$$A_{ijkl} > \frac{1}{R_x},$$
then
\[ \lim_{t \to \infty} |v_{xy}(t)| = 1, \quad 1 \leq i \leq M; \quad 1 \leq j \leq N; \]
or equivalently
\[ \lim_{t \to \infty} v_{xy}(t) = \pm 1, \quad 1 \leq i \leq M; \quad 1 \leq j \leq N. \]

Before explaining the methodology for object extraction using a CNN along with automatic selection of the parameters \( A_{ij,kl} \), a brief description of GAs is given in the next section.

3. Genetic algorithms: Basic principles and features

Genetic algorithms [3] are highly parallel and adaptive search and machine learning processes based on the mechanics of natural selection in natural genetic system. They exploit structured information to solve a wide range of complex optimization problems using genetic operators (reproduction/selection, crossover and mutation) on coded solutions (strings/chromosomes) in an iterative fashion. GAs deal simultaneously with multiple points (called, population), not a single point, which helps to find the global optimal solution without getting stuck at local optima. Genetic algorithms are blind, that is, they use only the payoff or penalty (i.e., objective) function and do not need any other auxiliary information.

To solve an optimization problem, genetic algorithms start with the chromosomal (structural) representation of a parameter set. The parameter set is to be coded as a finite length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's. For example, let \( \{a_1, a_2, \ldots, a_p\} \) be a realization of the parameter set and let the binary representation of \( a_1, a_2, \ldots, a_p \) be 10110, 00100, \ldots, 11001 respectively. Then the string 1011000100...11001 is a chromosomal representation of the parameter set \( \{a_1, a_2, \ldots, a_p\} \). Strings are then processed using three simple genetic operations.

Reproduction/selection is a process in which individual strings are copied according to their objective function values, fit, called the fitness function into a mating pool. Therefore, highly fit strings have a higher number of offsprings in the succeeding generation.

The crossover may proceed in two steps. First, members of the reproduced strings in the mating pool
are mated at random. Second, each pair of strings undergoes crossing over as follows: an integer position \( k \) is selected uniformly at random between 1 and \( l - 1 \), where \( l \) is the string length greater than 1. Two new strings are created by swapping all characters from position \( k + 1 \) to \( l \). Let

\[ a = 11000 \quad 10101 \quad 01000 \ldots 01111 \quad 10001, \quad b = 10001 \quad 01110 \quad 11101 \ldots 00110 \quad 10100 \]

be two strings (parents) selected for the cross over operation and the generated random number be 11. Then the newly produced offsprings (swapping all characters after the position 11) will be

\[ a' = 11000 \quad 10101 \quad 01101 \ldots 00110 \quad 10100, \quad b' = 10001 \quad 01110 \quad 11000 \ldots 01111 \quad 10001. \]

In genetic algorithms, mutation is the occasional (with small probability) random alteration of the value of a string position. It helps to prevent the irrecoverable loss of potentially important genetic material. A random bit position of a random string is selected and is replaced by another character from the alphabet. For example, let the third bit of string \( a \), given above, be selected (randomly) for mutation. Then the transformed string after mutation will be

\[ a'' = 11100 \quad 10101 \quad 01000 \ldots 01111 \quad 10001. \]

The effectiveness of this efficient searching technique has been demonstrated in the next section for automatic extraction of objects from images using CNN.

4. CNN for object extraction

Object extraction involves segmenting the whole image into two region types, namely, object region and background region. Two adjacent pixels of an image belong to the same region if they have similar gray value properties. So, both gray level and positional properties play an important role in making a decision whether a pixel belongs to object or background.

In order to explain the principle of object extraction using CNN, let us consider (1) which can be approximated, using (2), by a difference equation as:

\[
v_{xy}(n + 1) = v_{xy}(n) + \frac{h}{C} \left[ -\frac{1}{R_x} v_{xy}(n) + \sum_{l \in N_y} A_{i,j,l} f(v_{sx})(n)) + I_{ij} \right], \quad 1 \leq i \leq M; \quad 1 \leq j \leq N, \tag{3}
\]

where \( h \) is a constant time step and \( v_{xy}(n) \) is the state of the cell \( C_{ij} \) at the \( n \)th instant of time. Eq. (3) can be interpreted as a two-dimensional filter for transforming an image \( v_{x}(n) \) into \( v_{x}(n + 1) \). Usually, the filter is space invariant for image processing [2].

In the above definition \( A_{i,j,k,l} \) is assumed to be position invariant. Now (3) can be written, for an unit time step \( (h = 1) \), as

\[
v_{xy}(n + 1) = v_{xy}(n) - \frac{1}{R_xC} v_{xy}(n) + \frac{1}{C} [T * v_{xy}(n) + I_{ij}] \tag{4}
\]

where

\[
T = \begin{bmatrix}
T_{-r,-r} & \cdots & T_{-r,0} & \cdots & T_{-r,r} \\
\vdots & & \vdots & & \vdots \\
T_{0,-r} & \cdots & T_{0,0} & \cdots & T_{0,r} \\
\vdots & & \vdots & & \vdots \\
T_{r,-r} & \cdots & T_{r,0} & \cdots & T_{r,r}
\end{bmatrix}
\]
is the cloning template and \(*\) is a convolution operator. For any cloning template \(T\), the convolution operator \(*\) is defined as

\[
T \ast v_j = \sum_{k \in N^2} T_{k-1,j} v_{kl}.
\]  

(5)

A CNN processes signals by mapping from one signal space to another one. If the initial state space is \([-1.0, 1.0]^{M \times N}\) and the output space is \([-1, 1]^{M \times N}\), then the dynamic map \(F\) can be defined as

\[
F : [-1.0, 1.0]^{M \times N} \rightarrow [-1, 1]^{M \times N}.
\]  

(6)

Object extraction in an image \(X = \{x_{m,n} : m = 1, 2, \ldots, M; \ n = 1, 2, \ldots, N\}\) of size \(M \times N\) can be considered as a mapping

\[
E : [a, b]^{M \times N} \rightarrow \{A, B\}^{M \times N}
\]  

(7)

where the range of gray levels of the image is \([a, b]\), and \(A, B\) are gray levels of object and background respectively or vice versa. Equations (6) and (7) are analogous. Thus if we can transform the gray level of the input image into \([-1.0, 1.0]\) it is possible to achieve a transformation of the image into \([-1, 1]\) by using a cellular neural network.

4.1. Selection of cloning template

The selection of cloning template \(T\) plays an important role in defining the dynamic rule of a CNN for performing a particular operation. The guidelines for selection of \(T\) for some operations like noise removal, edge detection and line detection are mentioned in [2]. For example, in noise removal of an image, \(T\) can be expressed as an averaging operator. For edge detection, \(T\) is expressed by a Laplacian or a difference operator so that the pixels of homogeneous and boundary regions can possess values \(-1\) and \(+1\) respectively at the stable state of the net. The way one can select \(T\) for an object extraction problem is described below.

As mentioned earlier, in object extraction both gray level and positional properties should be taken into account and one way of achieving this operation is to use an weighted average operator. Therefore one can choose the weighted averaging operator in defining the dynamic rule of CNN in this regard. This operator can be expressed by a cloning template like \(C_1, C_2, C_3\) etc. as shown below (the unit used here is \(10^{-3}\) Q^{-1}):

\[
C_1 = \begin{bmatrix}
0.0 & 1.0 & 0.0 \\
1.0 & 2.0 & 1.0 \\
0.0 & 1.0 & 0.0
\end{bmatrix},
\]  

(8)

\[
C_2 = \begin{bmatrix}
0.5 & 1.0 & 0.5 \\
1.0 & 2.0 & 1.0 \\
0.5 & 1.0 & 0.5
\end{bmatrix},
\]  

(9)

\[
C_3 = \begin{bmatrix}
0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\
0.0 & 1.0 & 2.0 & 1.0 & 0.0 \\
0.5 & 2.0 & 4.0 & 2.0 & 0.5 \\
0.0 & 1.0 & 2.0 & 1.0 & 0.0 \\
0.0 & 0.0 & 0.5 & 0.0 & 0.0
\end{bmatrix}.
\]  

(10)

Let us consider \(C_1\), as an example, for explaining the object extraction operation in terms of the dynamic equation. Here for cell time constant \(C \cdot R_g = 1\) microsecond (\(C = 10^{-9}\) F; \(R_g = 10^3\) Q) and cell bias \(I_i = \) we have

\[
\frac{dv_{yj}(t)}{dt} = 10^9 [-v_{yj}(t) + v_{yj+1}(t) + v_{yj+1}(t) + 2v_{yj}(t) + v_{yj+2}(t) + v_{yj+3}(t)]
\]  

(11)
and

\[ v_{xy}(t) = 0.5(|v_{xy}(t) + 1| - |v_{xy}(t) - 1|), \quad 1 \leq i \leq 8, \quad 1 \leq j \leq 8, \] (12)

Let the initial state of the CNN be the same as the pixel values as shown in Table 1. Then we see that for the pixel at (4,4) position, the derivative \( dv_{xy}/dt \) is positive, whereas the derivative at the point (2,2) is negative. As a result, the pixel values at (4,4) and (2,2) at the next iteration will tend to increase and decrease respectively. After several iterations when the network becomes stable, these values will tend to +1 and -1, denoting object and background, respectively. The dynamic rule (11) is therefore able to detect the homogeneous object and background regions in the image by making their values, +1 or -1 (Table 2). It is only the corner pixels of the object which may possess some intermediate value.

It is therefore clear from the aforesaid discussion that different cloning templates are required for different image processing operations. Again, for a particular operation, the same cloning template may not be applicable for all kinds of images. For example, \( C_1 \) (8) may not be applicable for segmenting thin or noisy image regions.

There is no standard method for selecting automatically a correct set of cloning template parameters in setting up the dynamics of CNN for a particular operation on a given image. In the next section, we develop a methodology using genetic algorithms to provide a solution to this problem. Here, the problem of choosing correct parameters (for a \( r \)-neighborhood cloning template) of a CNN for object extraction has been considered to be equivalent to searching for an appropriate set of parameters \( \{T_{ij} : -r \leq i \leq r, -r \leq j \leq r \} \) from a complex space which gives the best object background classification. (Note that the same framework may be used for other image processing operations.)

### 4.2. Optimum selection of cloning template using genetic algorithms

#### 4.2.1. Fuzziness measures as fitness function

Determination of fitness function of GAs for selecting the optimum parameters of a CNN is an important task. Since image segmentation and object extraction are unsupervised problems, we need an evaluation function for quantifying the desired segmented output so that it can be used for defining the fitness function of GA. The principle of minimization of fuzzy compactness (Comp), index of area coverage (IOAC) and entropy (H) is an established criterion for image enhancement and object

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>An 8 x 8 input image</td>
</tr>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>-0.3</td>
</tr>
<tr>
<td>-0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>-0.3</td>
</tr>
<tr>
<td>-0.7</td>
</tr>
<tr>
<td>-0.5</td>
</tr>
<tr>
<td>-0.2</td>
</tr>
</tbody>
</table>

\[ v_{xy}(t) = 0.5(|v_{xy}(t) + 1| - |v_{xy}(t) - 1|), \quad 1 \leq i \leq 8, \quad 1 \leq j \leq 8, \] (12)

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>The output image when the net reaches to the stable state</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
</tbody>
</table>
extraction [5, 6]. Compactness and index of area coverage of an image reflect geometrical ambiguity whereas, entropy provides a measure of grayness ambiguity in an image. Here we have used these measures and their combinations as the quantitative indices for evaluating picture quality of segmented output. \( H(X) \), \( \text{Comp}(X) \) and \( \text{IOAC}(X) \) of an \( M \times N \) image \( X \), characterized by \( \mu(x_{m,n}) \), are defined as follows [6]:

\[
H(X) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left[ -\mu(x_{m,n}) \ln(\mu(x_{m,n})) - (1 - \mu(x_{m,n})) \ln(1 - \mu(x_{m,n})) \right],
\]

\[
\text{Comp}(X) = \frac{a(X)}{p^2(X)},
\]

and

\[
\text{IOAC}(X) = \frac{\text{length}(X) \cdot \text{breadth}(X)}{p^2(X)};
\]

where

\[
a(X) = \sum_{m=1}^{M} \sum_{n=1}^{N} \mu(x_{m,n}),
\]

\[
p(X) = \sum_{m=1}^{M} \sum_{n=1}^{N} |\mu(x_{m,n}) - \mu(x_{m+1,n})| + \sum_{m=1}^{M} \sum_{n=1}^{N} |\mu(x_{m,n}) - \mu(x_{m,n+1})|,
\]

\[
\text{length}(X) = \max_{n} \left\{ \sum_{n=1}^{N} \mu(x_{m,n}) \right\},
\]

and

\[
\text{breadth}(X) = \max_{m} \left\{ \sum_{n=1}^{N} \mu(x_{m,n}) \right\}.
\]

\( \mu(x_{m,n}) \) denotes the membership value of the \((m,n)\)th pixel to the fuzzy object region.

Entropy of an image \( (X) \) considers the global information and provides an average amount of fuzziness in grayness of \( X \), i.e., the degree of difficulty (ambiguity) in deciding whether a pixel would be treated as black (dark) or white (bright). Compactness and IOAC, on the other hand, take into account the local information and reflect the amount of fuzziness in shape and geometry (spatial domain) of an image. Therefore, the concept of minimization of these ambiguity measures may be considered as the basis of a fitness (evaluation) function of GA for object extraction. One may also use a composite measure (e.g., product of both grayness and spatial ambiguity measures) as the evaluation function so that minimization of this composite measure implies achieving minimum ambiguity (fuzziness) in the resulting segmented image from both points of view.

In the present algorithm we have used the reciprocal of their product as the fitness function, i.e.,

\[
\text{fit}(X) = \frac{1}{\text{Comp}(X) \cdot H(X)} \quad (13)
\]

or

\[
\text{fit}(X) = \frac{1}{\text{IOAC}(X) \cdot H(X)} \quad (14)
\]

so that both grayness and spatial ambiguity of an image are taken care of in the task of object extraction.

4.2.2. Algorithm

As stated earlier, for an \( r \) neighborhood cellular neural network the total number of circuit parameters to be selected in a cloning template is \((2r + 1)^2\). The method of selecting these parameters is written in algorithmic form as follows:

1. Generate \( p = (2r + 1)^2 \) random strings of 0's and 1's each of length \( l' \) corresponding to the parameters \( x_1, x_2, \ldots, x_p \). Each of these strings represent a parameter of the cloning template in a coded form. \( p' \) strings are concatenated to form a chromosomal/string representation of the parameter set. Generate \( 'm' \) such chromosomes to form the initial pool.

2. Input the image \((X)\).

3. Decode the strings representing the parameters of the cloning template and use these to set up a cellular neural network. The gray level range of the input image \((X)\) is scaled to \([-1, 1]\).
4. Compute the fitness function value using (13) or (14) corresponding to the output image \(X'\).

5. If specified number of iterations is reached then STOP.

6. The strings are reproduced or selected to create a new mating pool as described in Section 3.

7. Generate a new population by crossover and mutation.

8. Go to step 3.

Note that unlike the methods developed in [4], this algorithm does not need any prior knowledge of the desired segmented output.

5. Computer simulation and results

The methodologies developed in Sections 4.1 and 4.2 have been experimented on two 64 × 64 synthetic images (Figures 3 and 4) of geometric objects corrupted by Gaussian noise of different standard deviations \(\sigma = 10\) and 20\) and zero mean. (Note that the use of compactness and index of area coverage measures assumes that the input has single object region). The experiment has two parts. In the first part, the effectiveness of CNN has been demonstrated for \(C_1\), \(C_2\) and \(C_3\) cloning templates \((8-10)\) which were selected heuristically. The second part consists of an investigation where we have demonstrated the capability of GAs in selecting automatically the cloning templates. Throughout the experiment, we assumed \(C \cdot R_x\) (cell time constant) = 1 micro second, \(R_x\) (resistance of a cell) = \(10^3 \, \Omega\) and \(I_0\) (cell bias) = 0.

Fig. 3. Image of a squared object with Gaussian noise. (a) \(\sigma = 10\), (b) \(\sigma = 20\).

Fig. 4. Image of a circular object with Gaussian noise. (a) \(\sigma = 10\), (b) \(\sigma = 20\).
Figures 5 and 6 depict the outputs corresponding to the images (Figure 3) with noise levels ($\sigma = 10$ and 20) when $C_1$, $C_2$ and $C_3$ have respectively been used to define the dynamic rule of CNN. The results, as expected, show that the performance of the algorithm deteriorates with increase of $\sigma$ and improves with the increase of size of the cloning template. The template $C_3$ is less noise sensitive and provides better performance as compared to $C_1$ and $C_2$ due to the greater neighborhood effect.

For demonstrating the capability of genetic algorithms with fuzzy fitness function to obtain the optimum parameters of a cloning template, we considered a template of size $3 \times 3$ ($r = 1$). In our experiment, we have considered population size = 10, mutation probability = 0.005, crossover probability = 1 and number of iteration 50. The output obtained for the square images with different noise levels are shown in Figures 7 and 8 when (13) and (14) are respectively used. The deterioration

Fig. 5. Extracted square from its noisy ($\sigma = 10$) version. (a) using $C_1$, (b) using $C_2$, (c) using $C_3$.

Fig. 6. Extracted square from its noisy ($\sigma = 20$) version. (a) using $C_1$, (b) using $C_2$, (c) using $C_3$.

Fig. 7. Extracted square using GA with (13) as fitness function from its noisy versions. (a) $\sigma = 10$, (b) $\sigma = 20$. 
Fig. 8. Extracted square using GA with (14) as fitness function from its noisy versions. (a) $\sigma = 10$, (b) $\sigma = 20$.

Fig. 9. Extracted circle using GA with (13) as fitness function from its noisy versions. (a) $\sigma = 10$, (b) $\sigma = 20$.

Fig. 10. Extracted circle using GA with (14) as fitness function from its noisy versions. (a) $\sigma = 10$, (b) $\sigma = 20$.

The results obtained for the circle images are shown in Figures 9 and 10 respectively when (13) and (14) are used as the fitness function. As expected, for the circle images the outputs obtained using (13) as the fitness function is better as compared to those obtained using (14). The outputs obtained using the heuristically selected template $C_2$ are shown in Figures 11. As in the case of square image, the templates selected for circle image using genetic algorithms perform consistently well in different noise levels as compared to those of heuristic selection. As a typical illustration of the cloning templates obtained by GA after 50 iterations, we list them below (designated by $C_{GA}^1$ and $C_{GA}^2$) for the circle.
image for $\sigma = 10$ when (13) and (14) are used as the fitness function.

$$C_{GA}^1 = \begin{bmatrix} 0.748 & 0.747 & 0.559 \\ 0.958 & 0.964 & 0.909 \\ 0.491 & 0.658 & 0.219 \end{bmatrix} 10^{-3} \Omega^{-1}, \quad C_{GA}^2 = \begin{bmatrix} 0.897 & 0.827 & 0.314 \\ 0.680 & 0.995 & 0.758 \\ 0.249 & 0.749 & 0.192 \end{bmatrix} 10^{-3} \Omega^{-1}.$$

6. Conclusion

The problem of automatic selection of cloning template of cellular neural networks which can perform parallel signal processing in real time has been considered. A solution has been provided by developing a methodology based on Genetic Algorithms with fuzzy fitness function. The grayness (entropy) and spatial ambiguity (compactness and index of area coverage) measures have been used as the basis of the fitness function of GA. The results show that the template selected by GA performs consistently well and is more immune to noise as compared to the heuristically selected template.

As expected, template of larger size is less noise sensitive, but affects thin and elongated regions due to greater neighborhood effect. Again, the cost of realizing the circuit increases with the size of the neighborhood. The framework, based on genetic algorithms, developed here in selecting cloning template for object extraction may also be applicable for some other image processing operations.

Acknowledgment

This work was done while Prof. Sankar K. Pal held a Jawaharlal Nehru Fellowship.

References
