An Evolutionary Autonomous Agents Approach to Image Feature Extraction

Jiming Liu, *Member, IEEE*, Y. Y. Tang, *Senior Member, IEEE*, and Y. C. Cao

***Abstract*—This paper presents a new approach to image feature extraction which utilizes evolutionary autonomous agents. Image features are often mathematically defined in terms of the gray-level intensity at image pixels. The optimality of image feature extraction is to find all the feature pixels from the image. In the proposed approach, the autonomous agents, being distributed computational entities, operate directly in the 2-D lattice of a digital image and exhibit a number of reactive behaviors. To effectively locate the feature pixels, individual agents sense the local stimuli from their image environment by means of evaluating the gray-level intensity of locally connected pixels, and accordingly activate their behaviors. The behavioral repository of the agents consists of: 1) feature-marking at local pixels and self-reproduction of offspring agents in the neighboring regions if the local stimuli are found to satisfy feature conditions, 2) diffusion to adjacent image regions if the feature conditions are not held, or 3) death if the agents exceed their life span. As part of the behavior evolution, the directions in which the agents self-reproduce and/or diffuse are inherited from the directions of their selected high-fitness parents. Here the fitness of a parent agent is defined according to the steps that the agent takes to locate an image feature pixel.**

***Index Terms*— Autonomous agents, diffusion, evolutionary computation, image feature extraction, self-reproduction.**

I. Introduction

IN COMPUTER vision and image processing, image features like edges, lines, curves, corners, and borders may be detected using some mathematically defined operators, such as gradient edge detectors and zero-crossing edge detectors, or using surface fitting methods [12]. Detecting these features can greatly facilitate the interpretation of the scenes.

Many sophisticated techniques and algorithms for image feature extraction have been proposed and applied in recent years [4], [5], [21], [23], [26]. For instance, Liow [21] proposed an extended border tracing technique that combined the operations of region finding and closed contour detection. Alter and Basri [1] applied the so-called Salient Network method for extracting salient curves and noted that this method could suffer the problem of failing to identify any salient curve other than the most salient one (according to their proposed saliency measure). Lee and Kim [20] presented a method of extracting topographic features directly from a gray-level character image, without calculating eigenvalues and eigenvectors of the underlying image intensity surface. The method efficiently computes the directions of principal curvature. Maintz *et al.* [25] investigated the problem of evaluating ridge seeking operators for multimodality medical image matching.

With conventional approaches to image feature extraction, all the possible feature patterns must be carefully enumerated and exhaustively searched. This represents a nontrivial task. Furthermore, the resulting template masks may be sensitive to noise in the image. Another disadvantage is that the complexity of image feature extraction (e.g., a closed border for a region) is determined by the complexity of the image. For instance, in a spiral-like region, the template-based border tracing method [21] can be slowed down simply due to the length of the border to be traversed.

The approach introduced here utilizes evolutionary autonomous agents that can self-reproduce, diffuse, and cease to exist during the course of interacting with a digital image environment. The most distinct characteristics of our approach lie in that it is bottom up, decentralized, and distributed in nature and relies on local agent "processes" whose behaviors are both easy to define and natural for software and hardware implementations. This paper demonstrates such an agent system through illustrative examples in which a class of agents is equipped with the above mentioned behaviors to extract features from the image.

A. Related Work

Evolutionary computation is concerned with applying the computational models of evolutionary processes (e.g., [3]) to either achieving intelligent agent behaviors, where intelligence is measured in terms of the agent ability to contribute to its self-maintenance at genetic, structural, individual, as well as group levels [34], or solving real-life computation-intensive engineering problems, such as numerical optimization. In recent years, researchers in this field have been studying and advancing the methodologies for evolutionary computation as well as their applications in a number of areas, including genetic algorithms, evolution strategies, genetic programming, evolutionary programming, and classifier systems [10], [11], [13], [15], [16], [28], [29], [32]. Fogel [9] has provided a thorough treatment on the foundation and scope of this field.

Evolutionary autonomous agent systems as applied to digital image processing is a newly explored area of research that studies the emergent behaviors in a lattice where agents react to the digital image environment according to a set of behavioral rules. It may be viewed as a further extension to the earlier work on cellular automata [8], [17]–[19], [22], [24]. Cellular

Manuscript received May 1, 1997; revised July 9, 1997. This work was supported in part by a grant from Hong Kong Research Grant Council and in part by a Hong Kong Baptist University Faculty Research Grant.

The authors are with the Department of Computing Studies, Baptist University, Kowloon, Hong Kong.

Publisher Item Identifier S 1089-778X(97)07315-3.

automata, which drew upon Von Neumann's model [31], is concerned primarily with the fixed-point properties in a lattice of finite automata in which cells act locally according to a set of cellular rules [22]. Shanahan [33] investigated a class of cellular automata in which a population of organisms evolves in a microworld of square grid locations by repeatedly executing four local procedures, namely, *cease to exist, move, merge*, and *duplicate*. The goal of his work was to experimentally investigate under what conditions an instance of cellular automata could produce a significant amount of complexity and diversity. Tamayo and Hartman [35] applied computational organisms to model reaction-diffusion systems from which interesting space-time patterns reminiscent of chemical turbulence, solitons, and self-excited oscillations could be constructed and observed.

The proposed autonomous agent model exhibits general cellular behavior characteristics similar to those of cellular automata, but with the following fundamental differences; namely: 1) in the proposed agent model, the automata operate in a gray-level digital image and hence inanimate stimuli are present in the cellular environment, and 2) the behaviors of agents evolve as a result of the interaction with and within the image environment, whereas in cellular automata they are predefined.

B. Organization of the Paper

The remainder of this paper is organized as follows: Section II formally states the global optimality criterion of image feature extraction from the point of view of the evolutionary autonomous agents approach. Section III gives the representation of the proposed autonomous agents, their reactive behaviors, and the algorithm underlying the agent-based image feature extraction. Section IV describes several experiments in which the agent-based approach is effectively applied to extract image edges and multiple features and to follow dynamically moving features. Section V discusses the dynamics of the agent population and examines the issues such as the effects of behavioral parameters on the computation. Finally, Section VI concludes the paper by highlighting the key contributions of this work.

II. Problem Statement

The two-dimensional (2-D) lattice in which the proposed autonomous agents reside is a gray-level image, $\mathcal{I}$, of size $U \times V$ (i.e., an array of U rows by V columns of pixels). Suppose that $\mathcal{I}$ contains a number of pixels whose intensity relative to those of its neighboring pixels satisfies certain mathematically well-defined conditions. These pixels are called feature pixels. The objective of the autonomous agents in $\mathcal{I}$ is to extract all the predefined features of $\mathcal{I}$ by finding and marking at the feature pixels. This is essentially an optimization problem as stated below.

Definition 2.1 (Optimal Feature Extraction): Let M denote the total number of feature pixels in $\mathcal{I}$. If the total number of feature pixels detected and marked by active agents is equal to M, it is said an optimal feature extraction is achieved.

Definition 2.2 (Active Agents): At a certain time t in the 2-D lattice, autonomous agents whose age does not exceed their life span will continue to react to their image environment by way of evaluating the pixel gray-level intensity and selecting accordingly some of their behaviors. Such agents are called the active agents at time t.

From the above notions, we can further define an efficiency measure of the optimal feature extraction during a given period of discrete time t as the average rate of success that active agents find image features during t. Here, the rate of success is defined as the ratio between the total number of extracted feature pixels over the total number of active agents being used during t.

Definition 2.3 (Efficiency of Optimal Feature Extraction): The efficiency measure of an optimal feature extraction by autonomous agents for t time intervals is defined as the following function:

$$Q(t) = \frac{1}{t}\sum_{\tau=1}^{t}\left(\frac{\sum_{k=0}^{\tau} m(k)}{\sum_{k=0}^{\tau} N_{\Phi}(k)}\right) \tag{1}$$

where $m(k)$ is the number of extracted feature pixels found at time k and $N_{\Phi}(k)$ is the number of active autonomous agents Φ in $\mathcal{I}$ at time k.

From Definition 2.3, it is clear that the higher the $Q(t)$ value, the higher the efficiency of feature extraction will become.

III. Evolutionary Autonomous Agents for Image Feature Extraction

The evolutionary nature of the proposed agents approach lies in the way in which the generations of autonomous agents are selected and replicated. This section presents the detailed definitions of evolutionary autonomous agents, including their environment properties, local pixel evaluation function, fitness function, and the evolution of diffusion and asexual self-reproduction behaviors.

A. Two-Dimensional Lattice of an Agent Environment

Autonomous agents operate in a rectangular lattice that corresponds to a digitized image. In the rectangular lattice, each eight-connected grid represents an image pixel, as illustrated in Fig. 1. The grid also signifies a possible location for an autonomous agent to inhabit, either temporarily or permanently.

Definition 3.1 (Neighboring Region of an Agent): The neighboring region of an agent at pixel p is a circular region centered at pixel p with radius κ. The pixels falling inside this region are called the neighboring pixels of the agent. Each of the neighboring pixels is located in one of the eight evenly divided sectors. These sectors are also referred to as the eight directions of the region. Fig. 1 shows an example of the neighboring pixels of an agent from eight directions, respectively, when $\kappa = 1$.

B. Local Stimulus in Two-Dimensional Lattice

An autonomous agent is coded such that it always checks its neighboring region and selects its behavior according to the

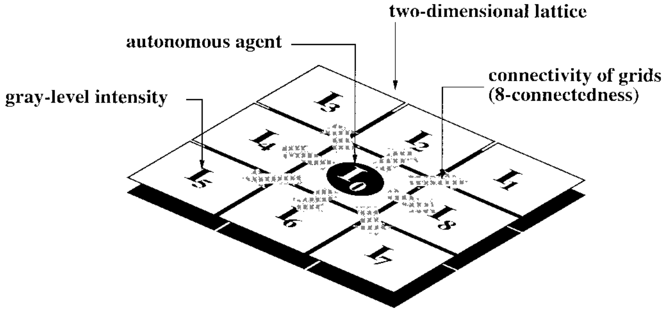


Fig. 1. Each pixel in the 2-D lattice environment is connected to eight neighbors (i.e., eight-connectedness). An autonomous agent can check, self-reproduce, and diffuse to any one of these neighboring locations.

concentration of certain gray-level pixels. If the concentration is within a specific range, the agent will activate its self-reproduction mechanism. Such a concentration is considered as a triggering local stimulus to the agent.

Definition 3.2 (Local Stimulus): The local stimulus that selects and triggers the behaviors of an agent at pixel location (i, j) is computed from the sum of the pixels belonging to a neighboring region which satisfy the following condition: the difference between their gray-level intensity values and the value at (i, j) is less than a threshold. In other words, the stimulus is determined by the density distribution of all the pixels in the neighboring region whose gray-level intensity values are close to the intensity at (i, j), as illustrated in Fig. 2. More specifically, the density distribution is defined as follows:

$$D^{\kappa}_{I(i,j)} = \sum_{s=-\kappa}^{\kappa} \sum_{t=-\kappa}^{\kappa} \{1 | \, \|I(i+s, j+t) - I(i,j)\| < \delta\} \quad (2)$$

where

κ the radius of agent neighboring region, i.e., a circular region centered at (i, j);

s, t the indexes of a pixel belonging to the neighboring region relative to (i, j);

$I(i, j)$ the gray-level value at (i, j);

δ a predefined positive threshold.

C. Agent Behaviors

During the course of evolution, each of the agents in the lattice will exhibit several behaviors, e.g., self-reproducing, randomized/nonrandomized search. These behaviors are triggered by the external stimuli present in the environment, which are computed using (2). As a result of the agent behavior execution, certain patterns, i.e., *markers* left behind by the agents, will emerge, which in turn characterize the features in the digital image environment. This section provides a detailed description of the reactive behaviors of an autonomous agent.

1) Feature Marking: When an agent detects a feature pixel p it will place a fixed marker at p. There may be different kinds of features in an image, hence several kinds of markers may exist. The stimulus for selecting the feature-marking behavior can be defined as follows:

Definition 3.3 (Feature-Marking): Let $\lambda = [u, v]$ be an acceptable range of the pixel count as defined using (2), where $u \leq v$. The agent places a marker at pixel p, if the outcome of its evaluation of the density distribution at p falls inside the λ interval, i.e., $D^{\kappa}_{I(i,j)} \in \lambda$.

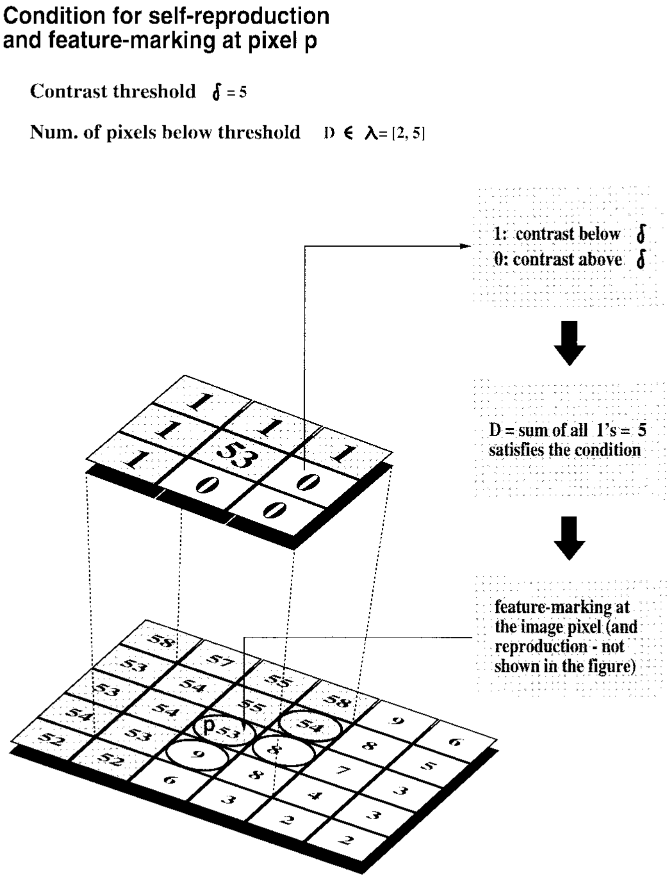

Condition for self-reproduction
and feature-marking at pixel p
Contrast threshold $\delta = 5$
Num. of pixels below threshold $D \in \lambda = [2, 5]$
1: contrast below δ
0: contrast above δ
D = sum of all 1's = 5
satisfies the condition
feature-marking at
the image pixel (and
reproduction - not
shown in the figure)
1
1
1
1
53
1
0
0
0
58
57
53
54
55
53
54
55
58
54
53
p 53
54
9
52
52
9
8
8
6
6
8
7
5
3
4
3
2
3
2

Fig. 2. An illustration of gray-level intensity checking. At each pixel p, an autonomous agent evaluates the outcomes of applying a mathematically well-defined operator. If the specific feature condition is satisfied, the agent leaves a permanent feature-marker at the pixel location.

Fig. 2 presents an example in which circles placed within individual grids denote the markers.

2) Agent Fitness Function: Two of the most important behaviors of an agent are diffusion and self-reproduction. Both can be executed in either a directional or a nondirectional mode. In the directional mode, the agent selects the most effective parent agents among all the previously successful ones and copies the directions as used in their reproduction and diffusion. The selection of the parent agents is based on their fitness function values. What follows defines the agent fitness function.

Definition 3.4 (Agent Fitness Function): Let $F(\phi_i)$ denote the fitness value of an agent ϕ_i. Thus

$$F(\phi_i) = \begin{cases} 1 - \dfrac{N \text{ steps before self-reproduction}}{\texttt{life\_span} \text{ of } \phi_i} \\ \qquad\qquad \text{if } \phi_i \text{ finds triggering stimulus,} \\ -1 \qquad\quad \text{otherwise.} \end{cases} \quad (3)$$

As can be noted from the above definition, the fitness function measures how long it takes the agent to find a feature pixel. The maximum fitness value is equal to one if the agent is exactly placed at the feature pixel at the time of being reproduced.

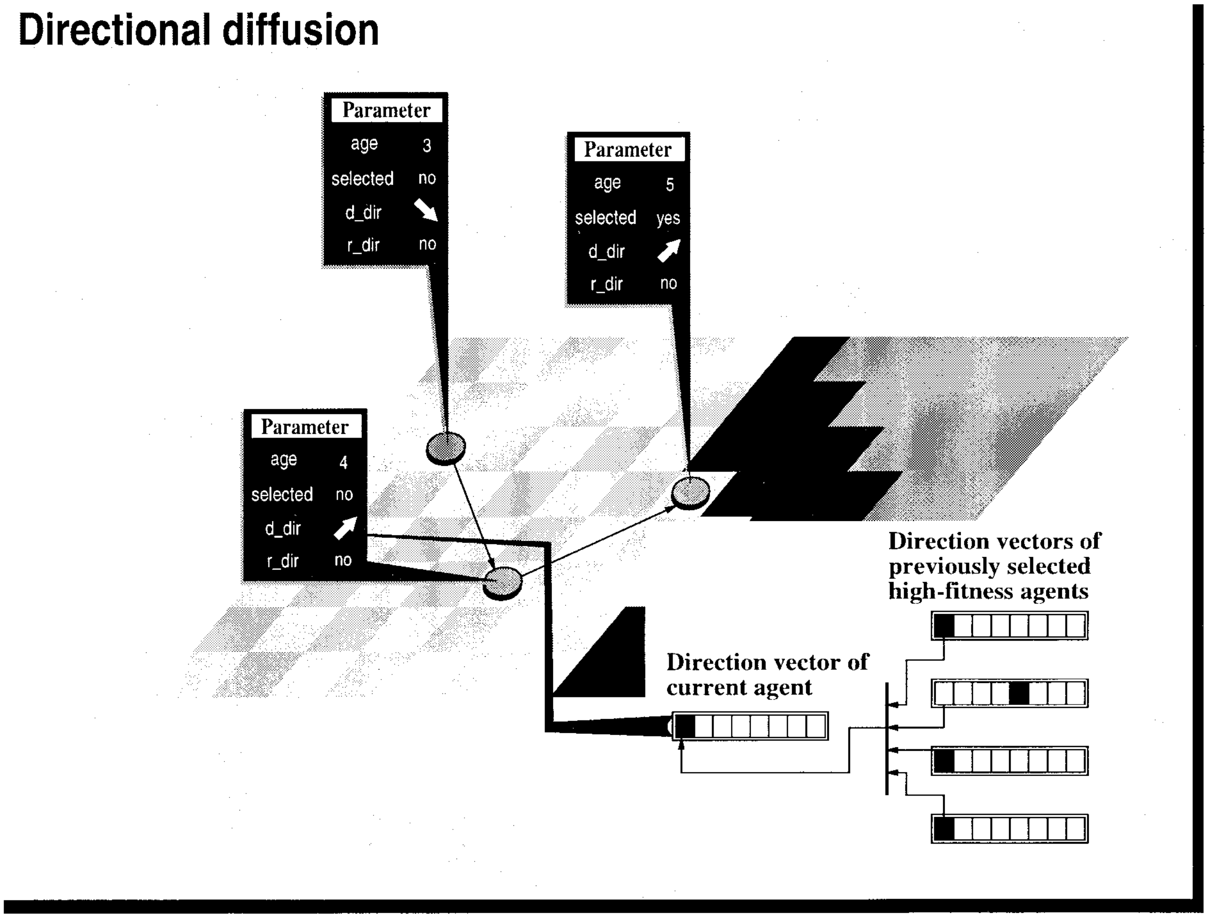


Fig. 3. An illustration of agent diffusion behavior. An autonomous agent of age three diffuses to its neighboring region of the 2-D lattice, in a direction as updated based on the directions of the previously selected agents (see Section III-C5). After each diffusion step, the age of the agent will be incremented by one. The process of diffusion provides a chance for the agent to search image features from the locations of its parent agents.

3) Diffusion According to Definition 2.2, an active agent always evaluates the pixel gray-level intensity of a neighboring region relative to its current location to see whether or not this location can be regarded as a feature pixel, i.e., a pixel that satisfies the condition as stated in Definition 3.3. If the current location is not a feature pixel, the agent will exhibit a diffusion behavior by moving to a new location within its neighboring region in either a directional or a nondirectional (i.e., random-direction) mode. The length of diffusion will be randomly generated. The diffusion behavior plays an important role for the agent to search feature pixels within the 2-D lattice. The specific stimulus that triggers this behavior is given as follows.

Definition 3.5 (Directional Diffusion): Let $\lambda = [u, v]$ be an acceptable range of the pixel count as defined using (2), where $u \leq v$. The agent moves to its adjacent locations whenever the outcome of its evaluation of the density distribution falls outside the λ *interval,* i.e., $D^{\kappa}_{I(i,j)} \notin \lambda$. The direction of the diffusion is selected based on an eight-element probability vector (corresponding to eight evenly divided directional sectors) in which each value indicates the probability of becoming high-fitness if the agent diffuses in the corresponding direction. The length of the diffusion is randomly generated within the diffusion region.

The direction vector of the agent as mentioned in the above definition is updated based on the diffusion directions of its previously selected high-fitness parent agents, as illustrated in Fig. 3. The details on the updating computation are given in Section III-C5.

4) Self-Reproduction: If an agent detects a feature pixel p, it will reproduce a finite number of offspring agents within its neighboring region in either a directional or a nondirectional (i.e., random-direction) mode. The self-reproduction behavior enables the agent to distribute offspring agents near the pixel location that meets the feature definition, and hence increases the likelihood of further feature extraction.

Definition 3.6 (Directional Self-Reproduction): Let $\lambda = [u, v]$ be an acceptable range of the pixel count as defined using (2), where $u \leq v$. The agent self-reproduces a finite number of offspring agents within its neighboring region of radius κ in a direction as computed from its direction probability vector, if the outcome of its evaluation of the density distribution at p falls inside the λ interval, i.e., $D^{\kappa}_{I(i,j)} \in \lambda$. The distance of the offspring from the parent agent inside the self-reproduction region will be randomly generated.

Fig. 4 illustrates the self-reproduction behavior of an autonomous agent. The direction vectors of self-reproduction by

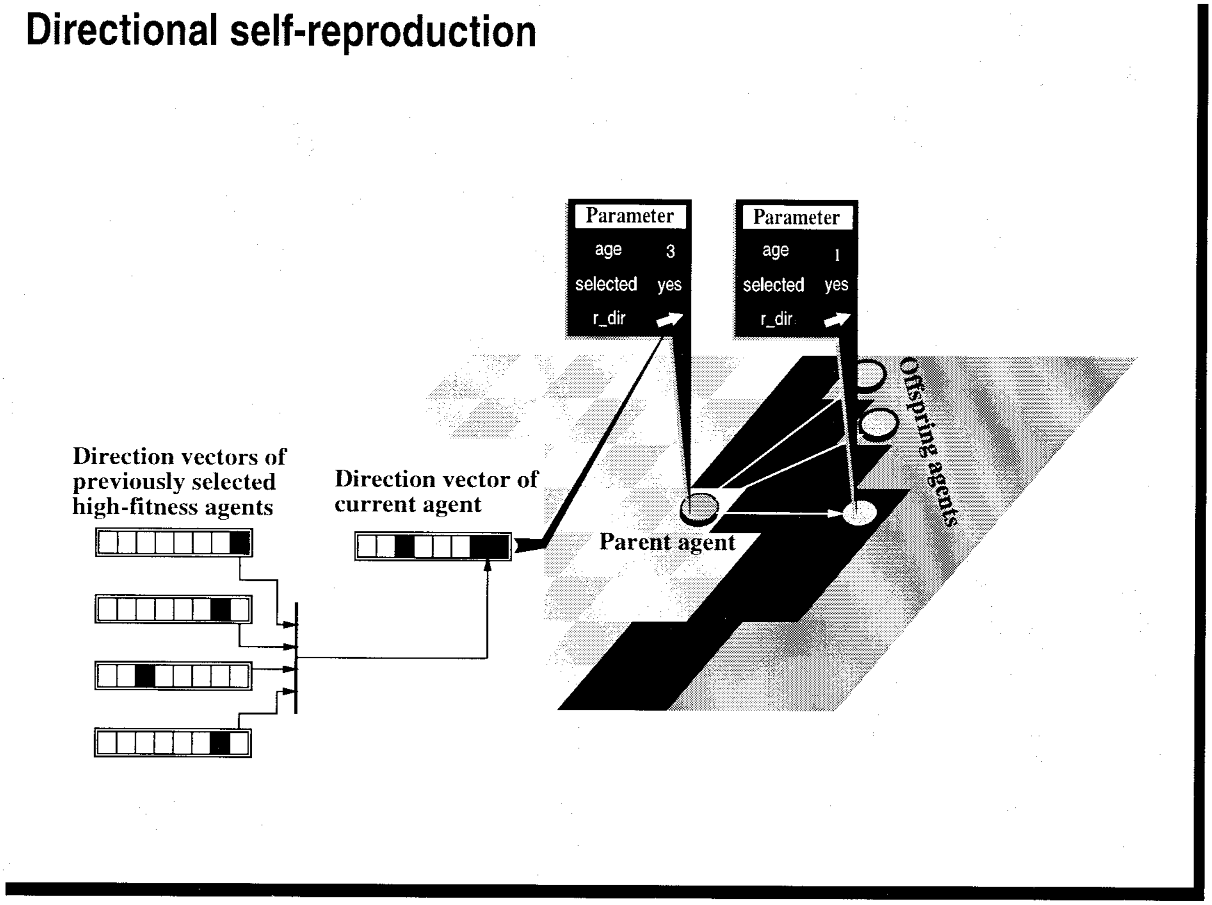

Directional self-reproduction
Parameter
age 3
selected yes
r_dir
Parameter
age 1
selected yes
r_dir
Offspring agents
Direction vectors of previously selected high-fitness agents
Direction vector of current agent
Parent agent

Fig. 4. An illustration of agent self-reproduction behavior. The asexual self-reproduction of an agent is triggered by the external stimuli in the environment as computed from the density distribution of their neighboring pixels of certain gray-level intensity values. The process of diffusion and self-reproduction repeats during the evolution of the autonomous agent population. The direction of self-reproduction is determined by a direction vector as computed from those of previously selected high-fitness parent agents.

the agent (and subsequently by its offspring) depend on an updating mechanism as given in Section III-C5.

5) Direction Vector Updating: Assume that a grandparent agent $\phi_i^{(g-1)}$ of generation $g-1$ produces a set of parent agents $\{\phi_{ij}^{(g)}\}$. This set further produces the offspring of generation $g+1$, as denoted by $\{\phi_{ijk}^{(g+1)}\}$. Thus, the directions of diffusion and self-reproduction by agent $\phi_{ijk}^{(g+1)}$ will be updated using the directions of some selected agents from $\{\phi_{ij}^{(g)}\}$ and $\{\phi_{ijk}^{(g+1)}\}$. The selection of these agents is based on their fitness values as computed using (3).

What follows provides the details on the updating mechanism for an autonomous agent to compute its diffusion and self-reproduction direction vectors. Here, a direction vector specifies the probability of success in locating image features if the corresponding direction is chosen for the respective behavior.

More specifically, the probabilities as associated with directions θ and ω, respectively, for diffusion and self-reproduction by agent $\phi_{ijk}^{(g+1)}$ are derived in the following two steps.

Step 1) Agent Selection: Select all $\phi \in \{\phi_{ij}^{(g)}\}$ and $\{\phi_{ijk}^{(g+1)}\}$, *s.t.* $F(\phi) > 0$;

Step 2) Direction Vector Updating: For all the selected agents, compute

$$p(\theta \in \Theta)_\phi = \frac{N_\theta}{\sum\limits_{\forall i} N_i} \tag{4}$$

and

$$p(\omega \in \Omega)_\phi = \frac{O_\omega}{\sum\limits_{\forall i} O_i} \tag{5}$$

where

Θ the set of possible directions for agent diffusion;

Ω the set of possible directions for agent self-reproduction;

N_i the number of agents that have diffused to the local stimulus from direction i;

O_i the number of agents reproduced by their parents from direction i.

The above formulas generate the probability distributions for the diffusion and self-reproduction directions by way of calculating the percentage of occurrences.

```
input: A digital image of size $U \times V$, in which each pixel has a gray-level value
output: Immobile agents (or markers) over feature pixels

randomly distribute an initial set of agents, $\{\phi_i^{(0)}\}$, over the image
assign the initial agent set to the active agent set: $\Phi \leftarrow \{\phi_i^{(0)}\}$
while $\Phi \neq \emptyset$ do
  for all current $\phi \in \Phi$ do
    if there exists grandparent(s) $\phi'$ of $\phi$ then
    compute the fitness values for all $\phi'$
      select $\phi'$, $s.t.\ F(\phi') > 0$
      update $P(\Theta)_\phi$ and $P(\Omega)_\phi$, using (4) and (5), respectively
    else
      assign $P(\Theta)_\phi$ and $P(\Omega)_\phi$ to uniform distributions
    endif
    if at local triggering stimulus then
      reproduce offspring $\{\phi^{(g+1)}\}$ in direction $\omega \in \Omega$ with $P(\Omega)_\phi$ to a neighboring sector of radius $\kappa$
      $\Phi \leftarrow \Phi \bigcup \{\phi^{(g+1)}\}$
      become immobilized (or leave a marker) at the current location
      $\Phi \leftarrow \Phi - \phi$
    else
      if age$_\phi$ = life_span then
        $\Phi \leftarrow \Phi - \phi$
        remove agent from the image
      else
        diffuse to a neighboring sector of radius $\kappa$, in direction $\theta \in \Theta$ with $P(\Theta)_\phi$
        age$_\phi$ $\leftarrow$ age$_\phi + 1$
      endif
    endif
  endfor
endwhlie
```

Fig. 5. The agent behavioral control and computation algorithm.

TABLE I
THE ATTRIBUTES THAT DETERMINE AGENT BEHAVIORS

Attribute	Description	Value
c	class identifier	char label
δ	intensity contrast threshold	> 0
λ	acceptable range of density value	$[u, v],\ v \geq u > 0$
s	number of offspring self-reproducible	> 0
Δ	life span	$(0, \texttt{life\_span}]$
$P(\Theta)$	direction vector for diffusion	$\Theta = \{1, 2, \cdots, 8\}$
$P(\Omega)$	direction vector for self-reproduction	$\Omega = \{1, 2, \cdots, 8\}$
κ	radius of diffusion and self-reproduction region	$(1, K]$

6) Agent Vanishing ("Death"): When the age of an agent exceeds its life span, the agent will abort further feature-searching movements and vanish from the 2-D lattice environment.

As can readily be noted from the above overview, the reactive behaviors of autonomous agents are parameterized by a set of attributes as summarized in Table I.

D. An Example

Taking border-tracing agents as an example, when an agent of border-sensitive class reaches a border location, it will *permanently* reside at the border and proceed to self-reproduce some offspring within its immediate neighboring region, as shown in Fig. 4. This process is best illustrated in the following reaction scheme:

$$\Phi^{(g)}_{i\in\{i\cap\text{border}\neq\emptyset\}} \xrightarrow{\text{behavioral selection}} \Phi^{(g+1)}_{j\in\{\|i-j\|\leq\kappa\}} \oplus \tilde{\Phi}^{(g+1)}_{i} \tag{6}$$

where $\Phi_i^{(g)}, \Phi_j^{(g+1)}$, and $\tilde{\Phi}_i^{(g+1)}$ denote the agents reaching the border, the reproduced agents within the adjacent neighboring locations, and the agents immobilized at border i (i.e., feature-markers), respectively. $\oplus$ symbolizes that the results are generated from two concurrently selected behaviors.

As the reproduced agents move away from the their current locations, some of them will encounter other parts of the border again, and hence the self-reproduction cycle will repeat itself, while those whose age exceeds their life span will vanish.

E. The Agent Computation Algorithm

The complete algorithm for agent-based image feature extraction is given in Fig. 5.

IV. EXPERIMENTS ON IMAGE FEATURE EXTRACTION

The preceding section has provided a formal model of autonomous agent behaviors. This section further examines how such agents can be applied in digital image environments to extract some interesting image features. In particular, we present some typical image processing experiments on edge/border detection and following.

A. Image Edge Detection

Fig. 6 presents a series of snapshots from an optimal edge detection experiment. The given image here is a 150 × 150 256-gray-level digital image. It was used as the grid lattice for a class of autonomous agents. Initially, a group of 100 agents was randomly distributed in the lattice. Since this is a relatively small number of agents, the majority of them will

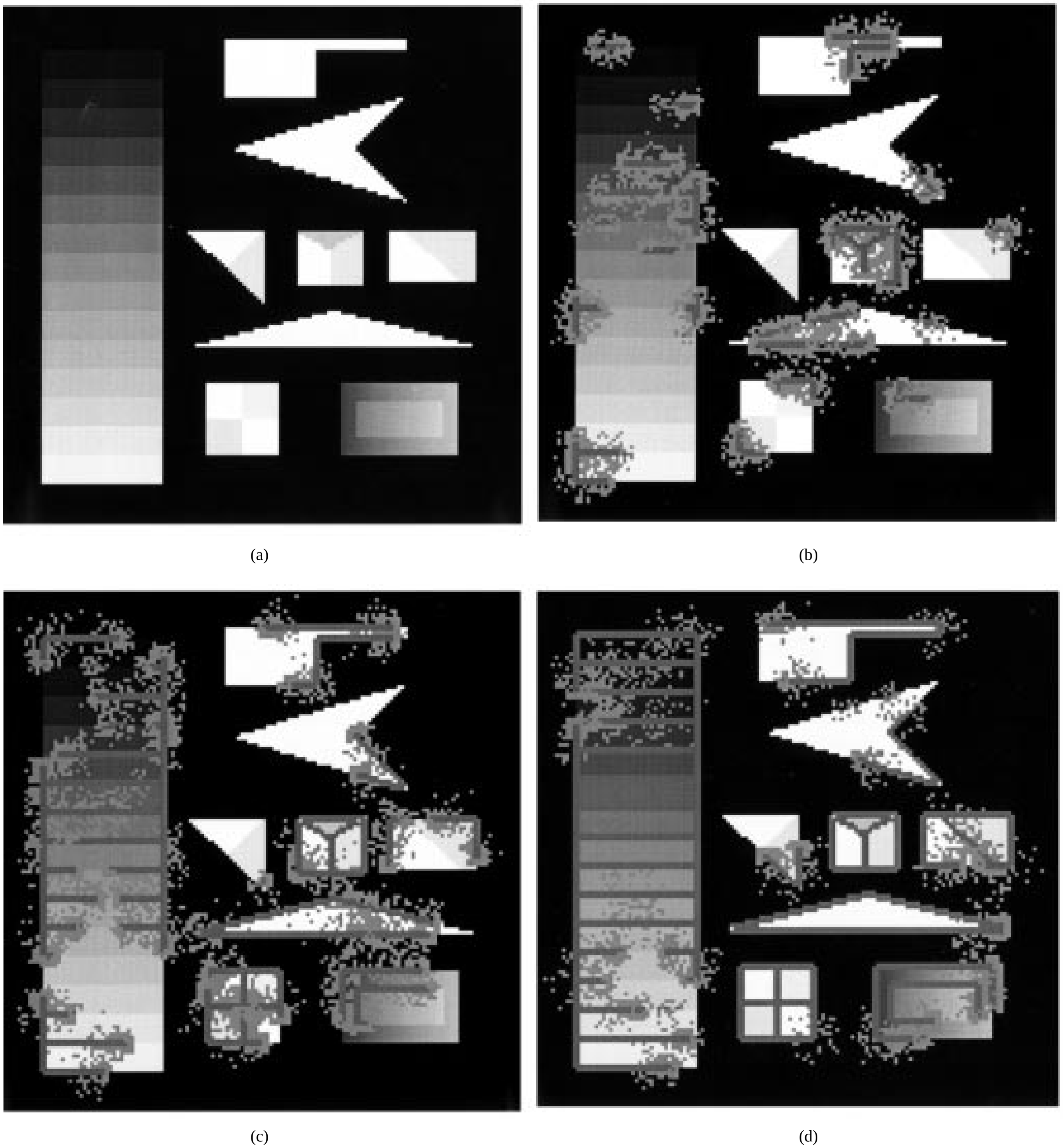
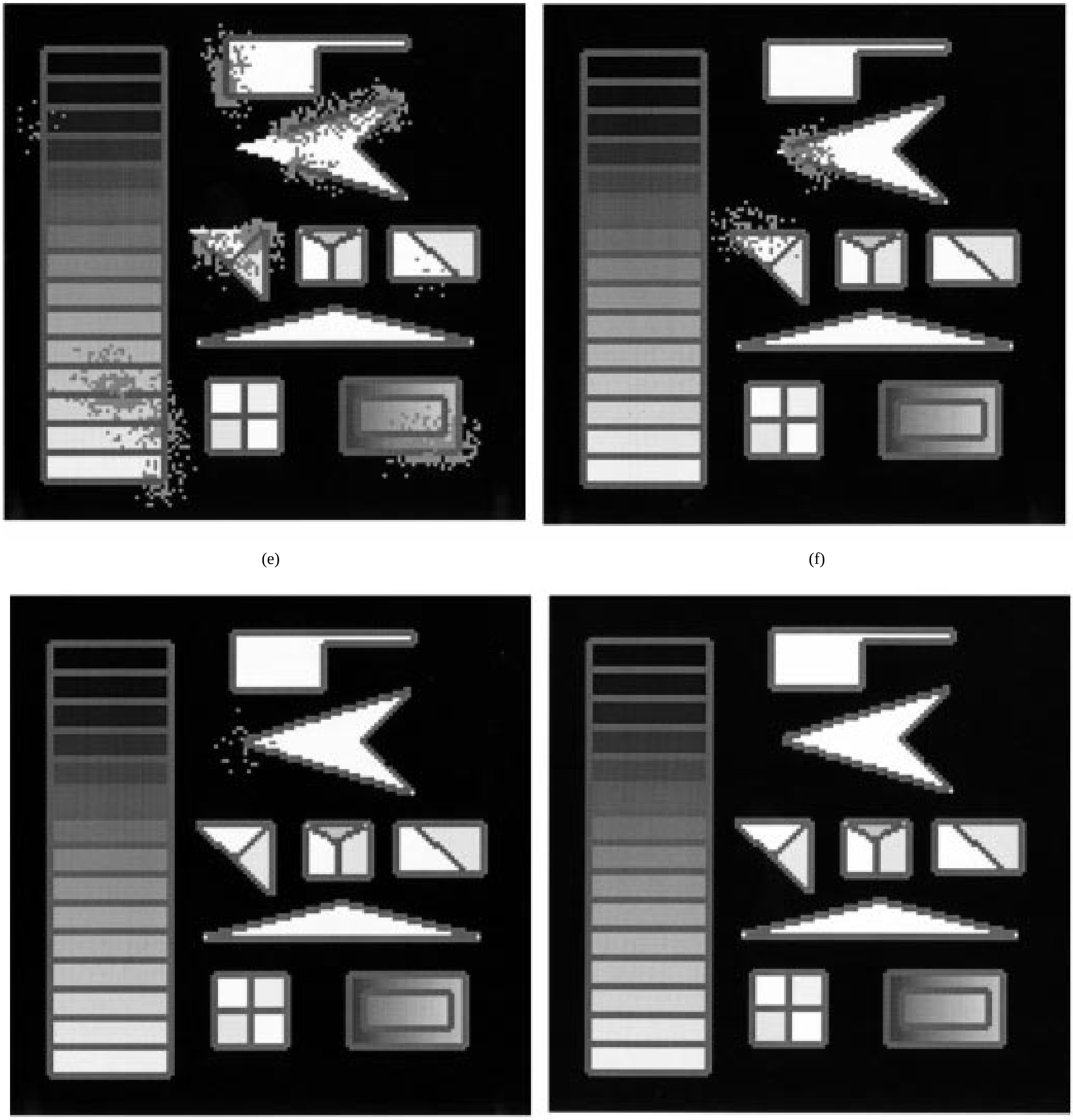


Fig. 6. The extraction of edges from a digital image based on the proposed autonomous agent-based computation model. Note that the triggering condition for the behaviors of an agent is computed from the eight-connected neighbors of the agent. The original input image is the one as labeled $t = 0$. The following images show the evolution of the agent population over the 2-D lattice. (a) $(t = 0)$; (b) $(t = 4)$; (c) $(t = 8)$; (d) $(t = 12)$.

not immediately find the image features, but rather after a few randomized movements as illustrated in Fig. 6 $(t = 4)$. In the snapshots, the clouds of light-grey dots signify the active autonomous agents that are undertaking certain diffusion processes, and the sequences of dark-grey dots are the markers left by the agents once they encounter the feature pixel locations.

By definition, the number of total diffusion movements allowed is determined by the life span of the agents. In this experiment, the life span of the agents was set to three. Therefore, if an agent does not find any features during the interaction with its environment for more than three discrete time steps, it will vanish from the lattice environment. On the other hand, the agents in the environment will asexually self-



(e)

(f)

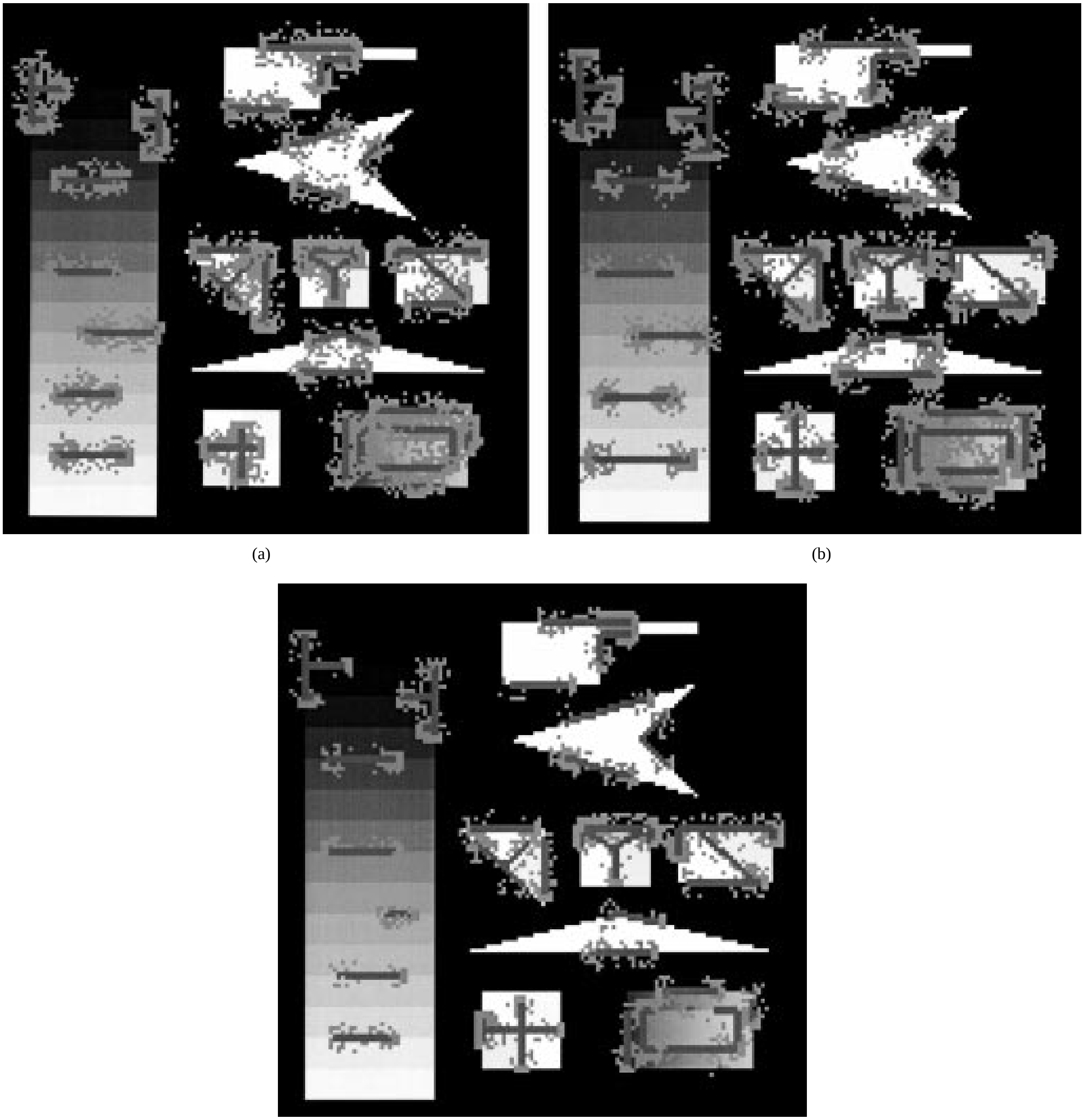
(g)

(h)

Fig. 6. *(Continued.)* The extraction of edges from a digital image based on the proposed autonomous agent-based computation model. At $t = 26$, all image features (i.e., region borders in this case) are found and labeled with markers. (e) $(t = 16)$; (f) $(t = 20)$; (g) $(t = 24)$; (h) $(t = 26)$.

reproduce offspring agents if the triggering condition of certain density-distribution satisfies a given λ interval. In the present experiment, we set $\lambda = [2, 7]$ for a neighboring region of radius $\kappa = 1$, the contrast level difference between the current pixel (at which an agent resides) and its eight neighboring pixels $\delta = 42$, and the number of offspring agents reproduced at a feature pixel $s = 5$. That is to say, an agent will leave a marker if the density distribution of its immediate neighboring pixels, whose gray-level intensity values are not deviated from that of the current location by 42, falls into the specific interval of [2, 7].

1) Effects of Individual Behaviors on Feature Extraction: To examine the effects of individual behaviors on the efficiency of optimal image feature extraction, we further con-



(a)

(b)

(c)

Fig. 7. The snapshots of agent evolution under three different experimental conditions (see text). (a) random reproduction and diffusion, (b) directional diffusion, (c) directional reproduction.

ducted three experiments that corresponded to the following three conditions, respectively.

1) *Mode 1—Random Reproduction and Diffusion:* The directions for the agent to self-reproduce and diffuse within a neighboring region of radius $\kappa = 2$ (i.e., the evenly divided sectors within a 5×5 region for self-reproduction and diffusion) are randomly determined.
2) *Mode 2—Directional Diffusion:* The direction for the agent to self-reproduce is randomly determined whereas the direction for diffusion is determined according to an updated direction vector.
3) *Mode 3—Directional Reproduction:* The directions for self-reproduction and diffusion are determined according to updated direction vectors.

Fig. 7 presents three snapshots of the agent evolution under the above mentioned experimental conditions where the age of the agents was set to three, the number of offspring agents reproduced was set to eight, and $\lambda = [2, 7]$. From the figure,

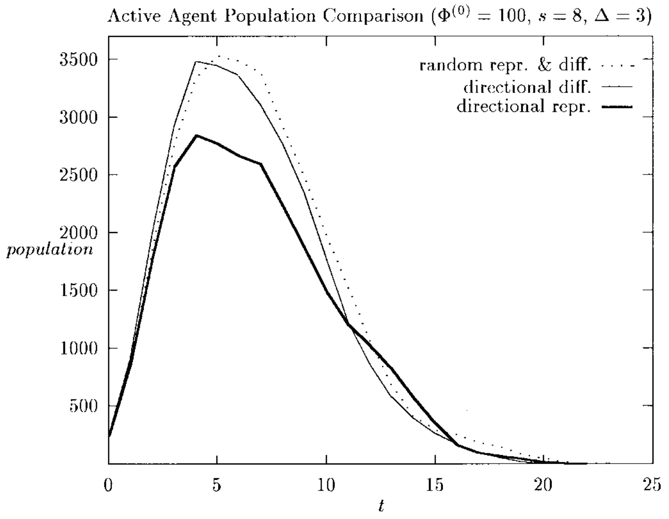


Fig. 8. A comparison of active agents in the 2-D lattice over the entire period of evolution.

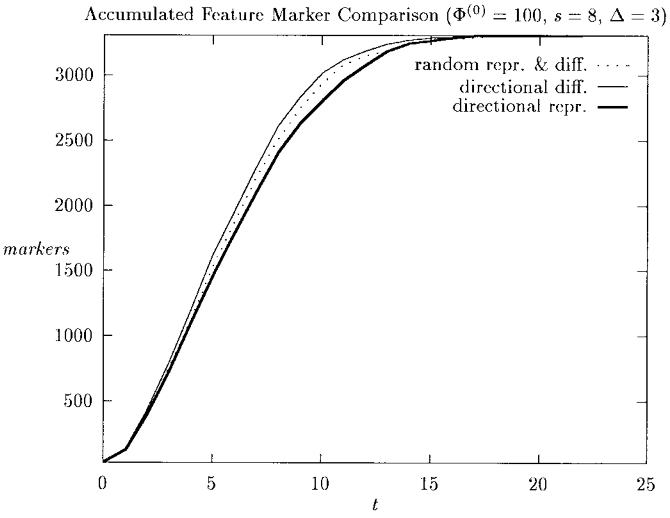


Fig. 9. A comparison of accumulated feature pixels as detected over the entire period of evolution.

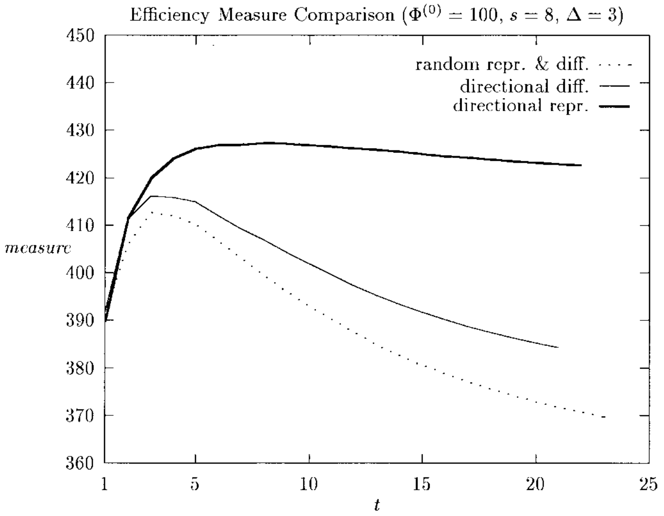


Fig. 10. A comparison of the efficiency measures in optimal feature extraction.

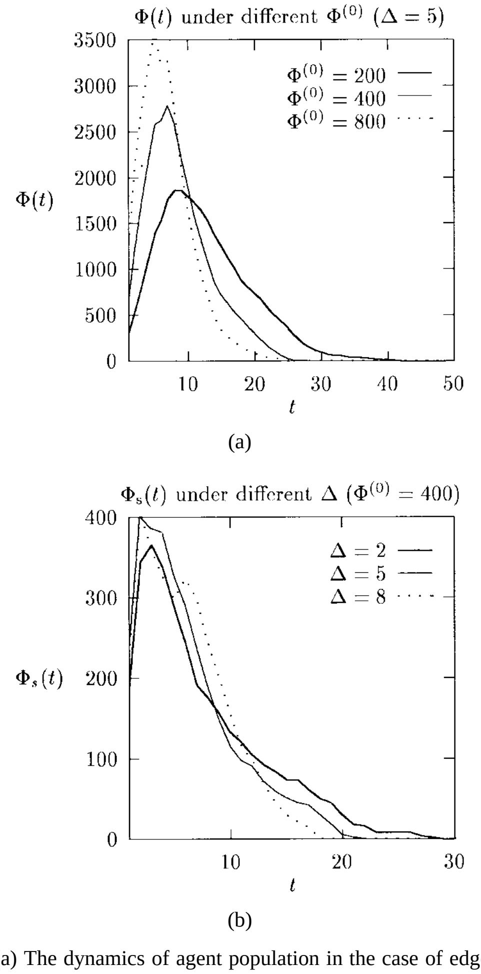
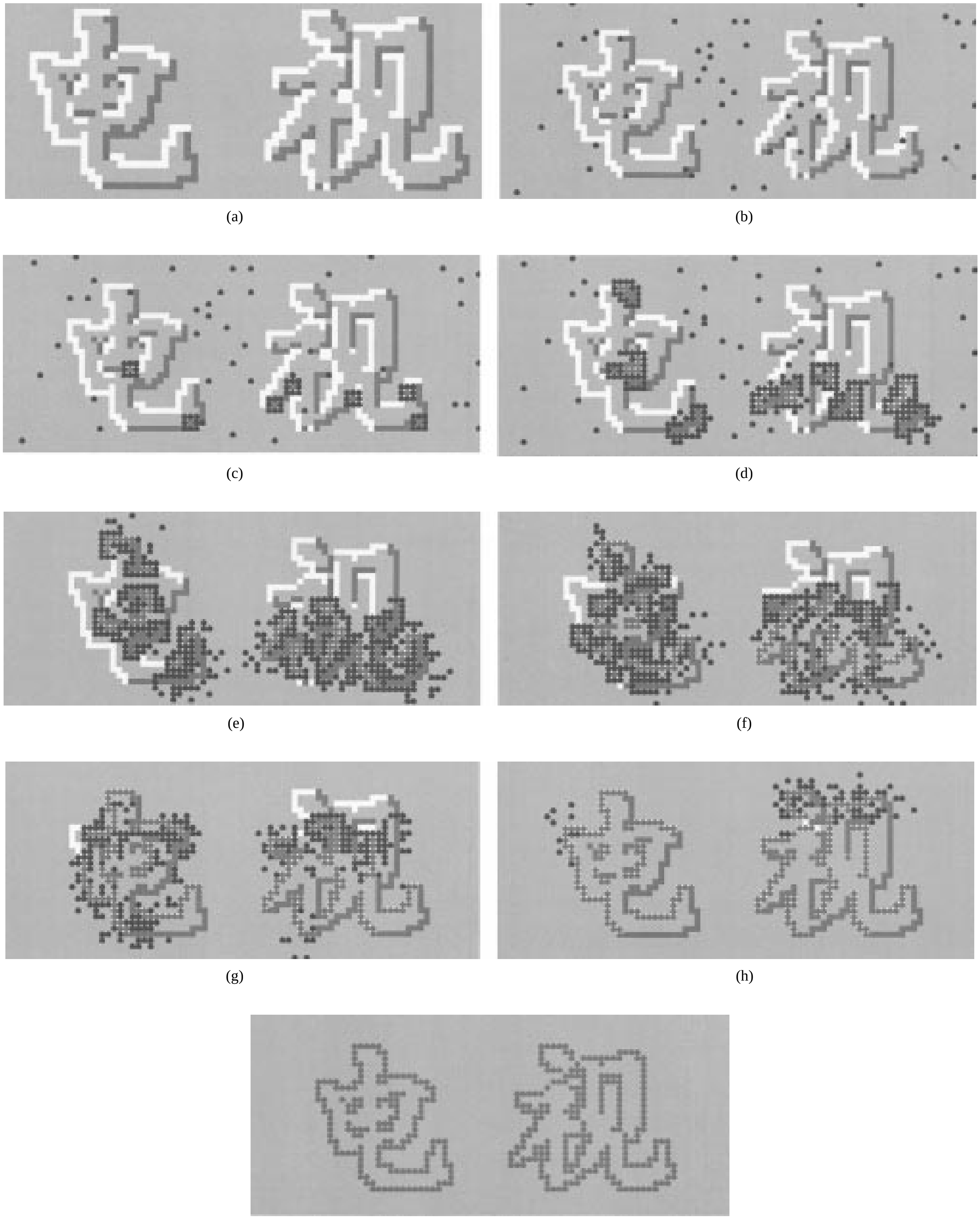


Fig. 11. (a) The dynamics of agent population in the case of edge-detection experiment, under three different initial population sizes, where $\Phi(t)$ curve shows the total number of active agents existing in the lattice. (b) The effect of different life span values on the dynamics of feature extraction, where $\Phi_s(t)$ curve shows the total number of active agents that have found image features during their interaction with the environment.

it can be observed that the shapes of the active agent clouds were different among the three conditions. In particular, Mode 1 produced the largest clouds evenly distributed along the detected feature pixels. While the active agents in both Modes 2 and 3 progressed effectively along the local feature pixels, the layers in the latter case were relative thinner. This is due to the fact that the maximum size of the self-reproduction or diffusion region in both cases was set to 5 × 5. Hence, as only one directional sector was selected in the latter case, the actual reproduction sector became limited to the size of four pixels (i.e., only four out of eight reproduced agents were actually kept).

In addition to these observations, several quantitative comparisons were also conducted, the results of which have been shown in Figs. 8–10. Fig. 8 gives the number of active agents as involved in the region border extraction. Modes 1 and 2 were similar in agent population size, the former used slightly more than the latter. Mode 3 used a smallest number of agents during the initial half of the evolution period and had slightly more agents later for a short period. Fig. 9 shows a comparison of the accumulated feature pixels as detected in



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 12. An example of autonomous agents for extracting borders of two characters. (a) shows the original image, (b)–(h) provide the snapshots of the intermediate steps in border extraction, and (i) shows the feature-markers, representing the found borders. It can be noted that the agents are sensitive to those locations that satisfy the triggering condition, i.e., $\lambda = [1, 10]$ and $\kappa = 2$.

the three modes. Mode 2 was slightly faster than the random mode, while Mode 3 was slightly slower since fewer agents were reproduced each time for reasons as mentioned in the preceding paragraph.

Of greatest interest is Fig. 10 which compares the efficiency measures among the three modes. From this figure, Mode 3 represents by far the most efficient means for optimal feature extraction as its averaged measure is consistently higher than Modes 1 and 2.

2) Parameters Affecting Agent Computation: With respect to the experiment as presented in the preceding subsection (for Mode 1), we further investigated the effect of initial population size on the dynamics of the evolution and found that the total number of active agents involved was not affected by the number of initial agents. The rates of self-reproduction and death, however, can be affected.

In Fig. 11(a), the dynamics of autonomous agent populations with different initial sizes (i.e., 200, 400, and 800) and five-step life span is presented. The areas under the three curves are almost the same, and the curve under higher initial population size converges faster.

Apart from the initial population size, the second factor that is of interest is the life span of the agents. Our experiments showed that if the life span increased, diffusion toward other features would become more likely to occur. This phenomenon can be observed from Fig. 11(b), i.e., the slight differences among the population curves of active agents with three different life span values, respectively.

B. Character Border Searching

Fig. 12(a)–(i) present the evolution of agents in a digital image. The final result of agent evolution, as shown in Fig. 12(i), gives the external borders of two characters. In the experiment, the initial population size was set to 40. The neighboring region, from which the triggering condition was verified, was composed of two consecutive layers from the current location of the agent. Or, in other words, the radius of the triggering region κ was equal to two. The λ interval was set to [1, 10] for the triggering condition. Furthermore, the threshold for the gray-level contrast, δ, was set to zero. Other parameters such as life span were exactly the same as those in the previous experiment.

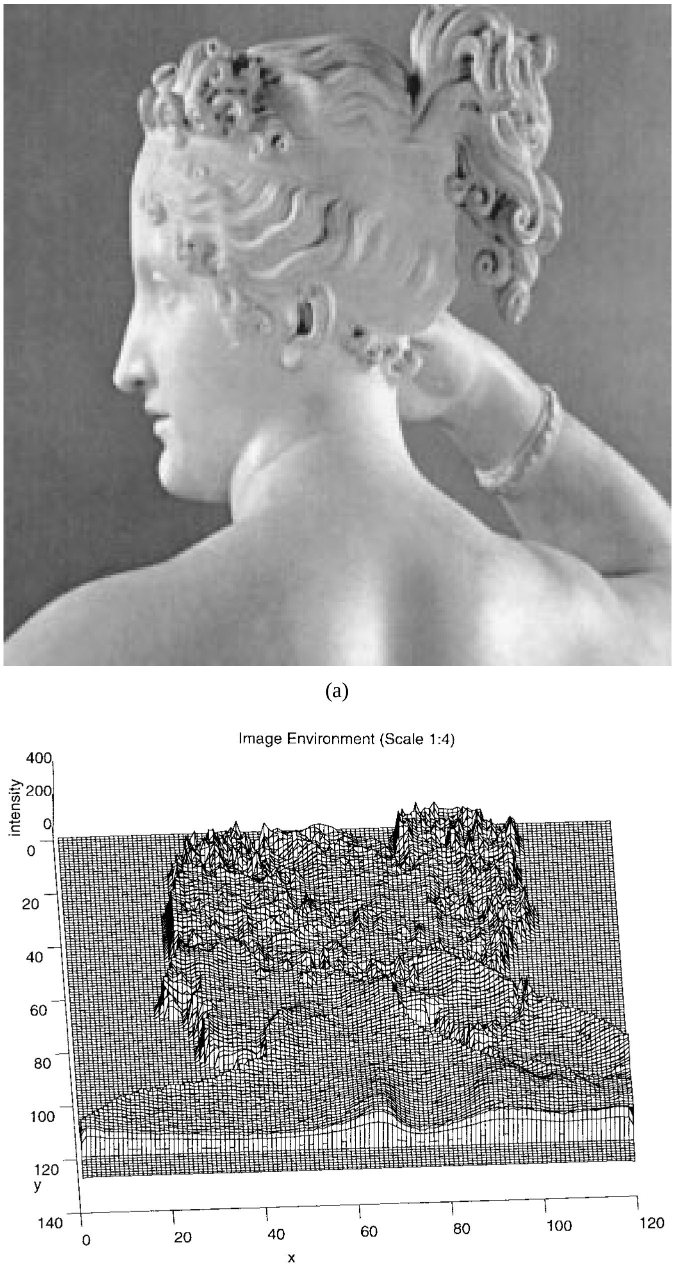
C. Multiple Feature Extraction

All the experiments mentioned so far were concerned only with single-class agents that effectively search and mark their feature pixels. To demonstrate the effects of multiple classes of agents in the simultaneous extraction of significant image features, we have conducted an experiment in which three different classes of agents were designed to extract features from a more complex image as shown in Fig. 13(a). The specifications of the three classes are as follows.

Class 1) $\delta = 50, \lambda = [0, 11], \Phi^{(0)} = 350, s = 8$, and $\Delta = 5$;

Class 2) $\delta = 50, \lambda = [10, 20], \Phi^{(0)} = 350, s = 8$, and $\Delta = 5$;

Class 3) $\delta = 50, \lambda = [20, 23], \Phi^{(0)} = 210, s = 8$, and $\Delta = 5$.

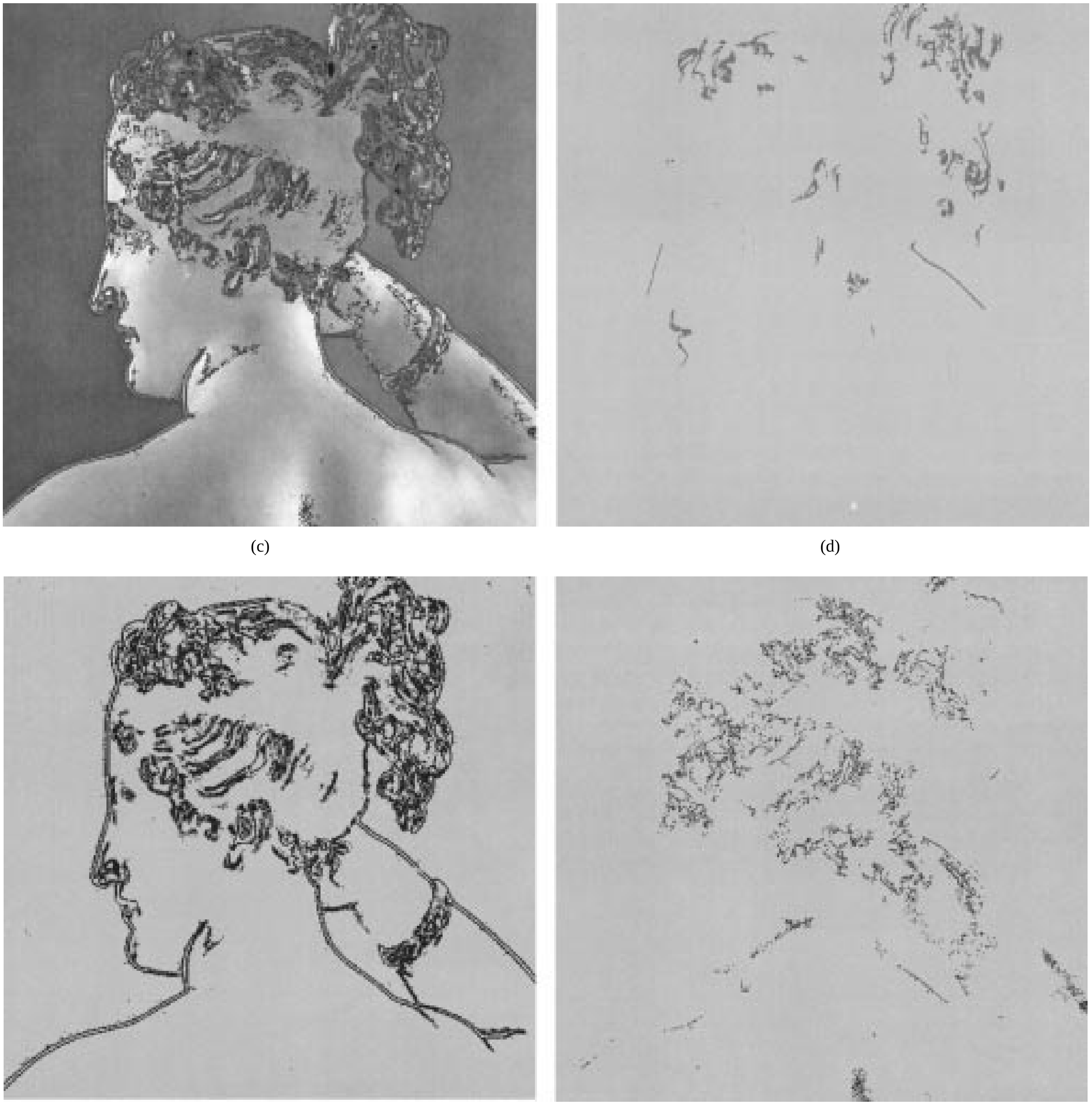


(a)

(b)

Fig. 13. The extraction of face features using three different classes of autonomous agents. (a) The original image is in 256 gray-level intensity of size 520 × 480. (b) The gray-level image is shown in a three-dimensional plot (scale = 1:4) in which the vertical axis represents the intensity levels of pixels. The horizontal dimensions of the plot represent the size of the 2-D lattice in which the autonomous agents reside and evolve.

Initially all three classes of agents were randomly distributed over the 2-D lattice which corresponds to the horizontal grids of the plot in Fig. 13(b). Note that the vertical dimension as defined over the lattice shows the intensity values of the grid pixels. In searching for different image features, the agents of different classes would check and react to their local pixels in a manner as defined in Section III-C. After a number of behavioral evolution steps, the populations of active agents belonging to the three classes would gradually vanish as all the features became emerged. The final results of feature-markers as left by the agents are shown in Fig. 13(c).



(c) (d)

(e) (f)

Fig. 13. *(Continued.)* The extraction of face features using three different classes of autonomous agents. (c) After a period of discrete time, the three classes of the agents completely marked their “territories,” i.e., corresponding image features. (d) The features as marked by Class 1 agents correspond to the sharp-contrast region pixels. (e) The features as marked by Class 2 agents correspond to the significant narrow edges. (f) The features as marked by Class 3 agents correspond to the mild sloping surface regions.

To get a clear view of the features, Fig. 13(c) is further decomposed into three separate figures, one for each type of features as detected by a particular class of agents. More specifically, Fig. 13(d) shows the highest contrast region pixels as found by Class 1 in the hair, ear, and mouth regions. These feature regions identify the most significant corner sketches of the image. Fig. 13(e) shows the result of Class 2 in extracting all the narrowly connected edge pixels. The edge markers give a meaningful outline of the image. Finally, Fig. 13(f) presents the features as marked by Class 3, which correspond to the mild sloping surface regions, i.e., shadows that reflect the overall shape/depth information of the image.

The rate of feature extraction varies from one class to another. Fig. 14 gives the accumulated marker curves for all the three classes. It is noted that Class 2 has the highest extraction rate, whereas Class 3 has the lowest. This is mainly

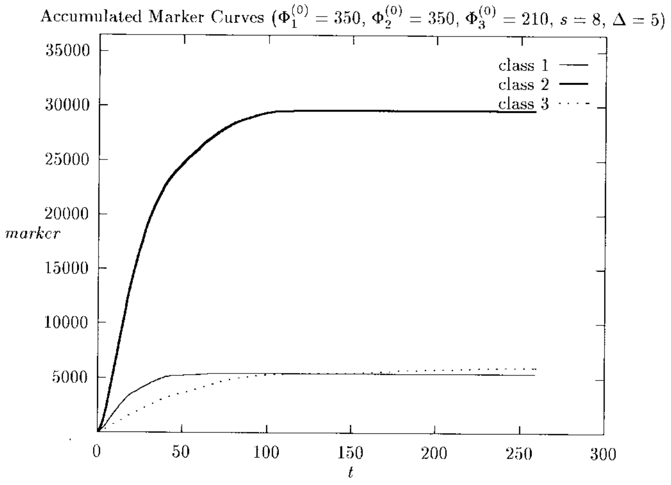


Fig. 14. Accumulated marker curves. The three classes of agents effectively find their corresponding feature pixels after 100 steps.

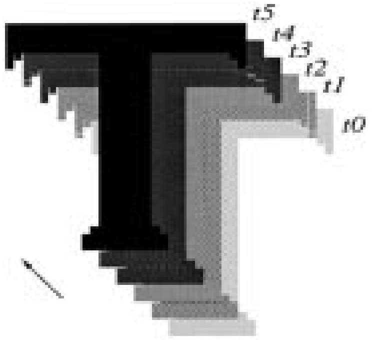


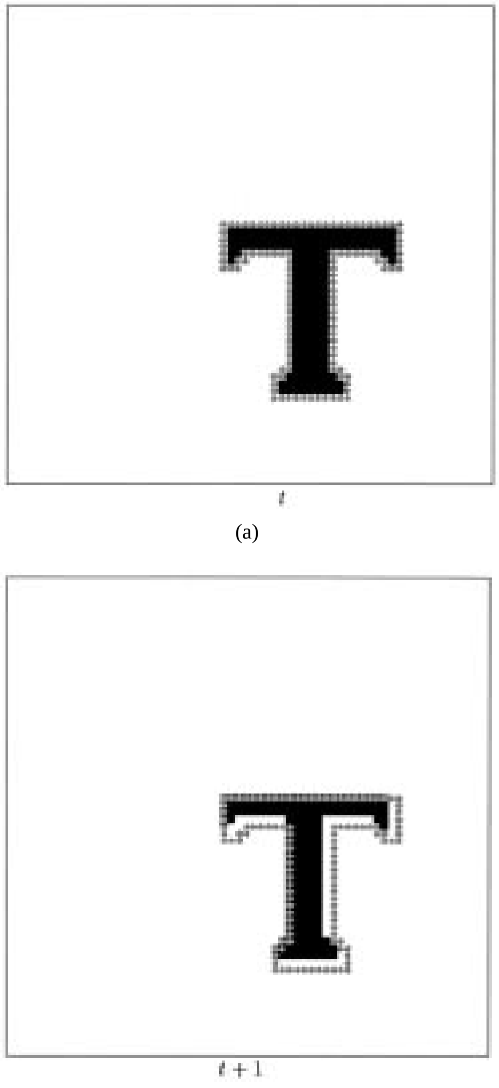
Fig. 15. An example dynamic environment in which a T-shaped object moves in discrete space and time.

due to the fact that the designed agent behaviors, such as diffusion and self-reproduction, are more effective in searching and branching the high-connectivity features.

D. Image Feature Tracking

The previous sections have provided the details on the local mechanism and the global dynamics of agents in *static* digital image environments and shown how the features can be searched as a result of behavioral evolution. As a step further from the static image environments, this section examines the capability of autonomous agents in following detected features in a sequence of digital image frames. This task represents one of the most challenging image-processing problems, namely, visual tracking and motion estimation for robot vision [6], [7].

Fig. 15 shows an overlaid view of multiple digital image frames, each of which gives the location of an object at a certain discrete time. The problem of image feature following in this case can be stated as follows: Suppose that at a specific time, the image features have been identified in a manner as in the static-environment case. Slightly unlike the previous experiments, however, in which the agents marked the feature pixels whenever their search succeeded, this experiment will disable the feature-marking behavior but instead permit the active agents to reside (i.e., become immobilized) at the feature pixels. After some intervals of time, the image features move to new locations, and subsequently, some of the previously successful agents will no longer be at the feature locations, as illustrated in Fig. 16. When such an instance occurs, the agents that previously immobilized at the feature pixels will be activated again as if they were just reproduced. They will adjust their diffusion and self-reproduction directions to *maximize* their fitness in the new environment (i.e., local fitness optimization). This in turn enables the agents to relocate the features. Fig. 17 presents the experimental results concerning the adaptation of the agents in the dynamic environment in which a T-shaped object moves in time, as shown in Fig. 15. The learning curve in Fig. 17 indicates the time as required for the population of the agents to adapt to a new feature movement.



(a)

(b)

Fig. 16. (a) At time t, agents have located the border of a T-shaped object from the environment. (b) At time $t + 1$, the T-shaped object moves to a new location, resulting in previously successful agents to be offset from the feature pixels.

V. Discussions

This section gives some further observations on the proposed evolutionary autonomous agents, concerning their population dynamics and relationships to adaptation.

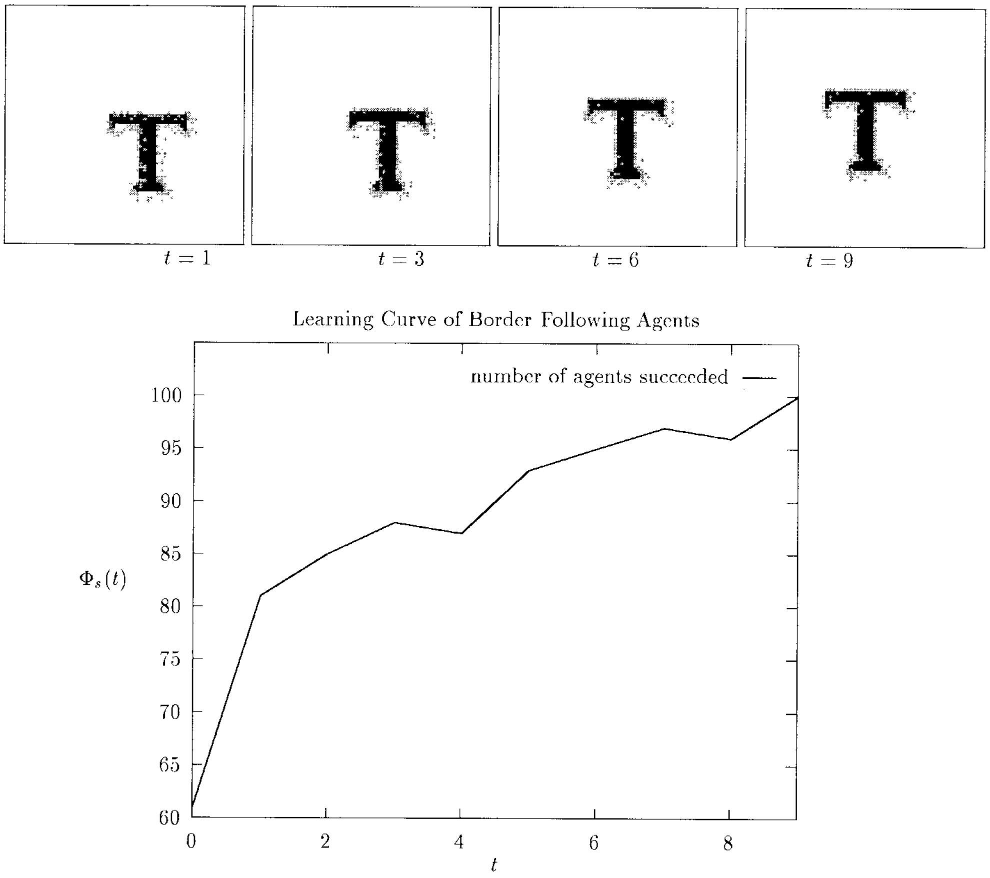


Fig. 17. The snapshots of the feature following agents and their learning curve. After the target has been moved, some of the agents will attempt to move toward the target based on the evolved behaviors. Agents that have found the edge will self-reproduce locally and then become immobilized, and the ones that have not found the edge grid during their life span will vanish from the 2-D lattice.

A. The Dynamics of Evolutionary Autonomous Agents

To enable a better understanding of the empirically obtained results on agent population change as shown earlier, what follows provides a formal description of the agent dynamics.

At a discrete time k, the number of agents that succeed in locating the features can be calculated as follows:

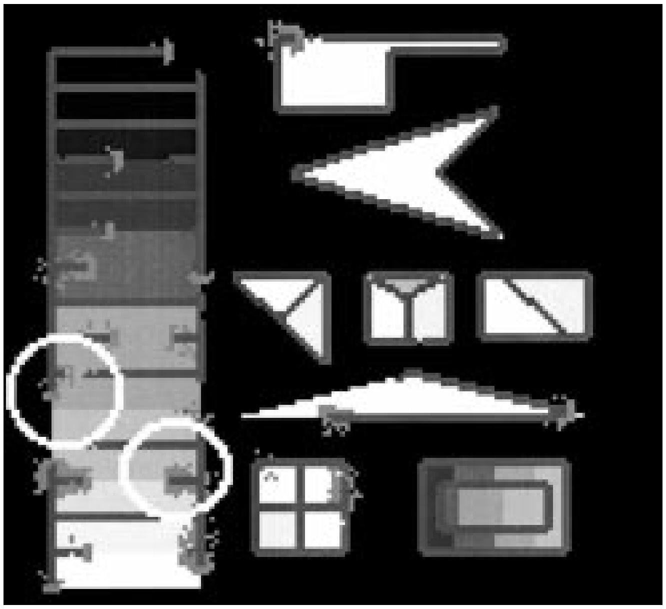
$$R_k = \sum_{i=1}^{\Delta} \psi_k^{k-i} \tag{7}$$

where Δ denotes the `life_span` of the agents, and ψ_k^{k-i} denotes all the agents that were reproduced at time $k - i$ and found the features at time k.

Based on the above definition, we can further derive the equation for computing the agents that vanish at time k, as follows:

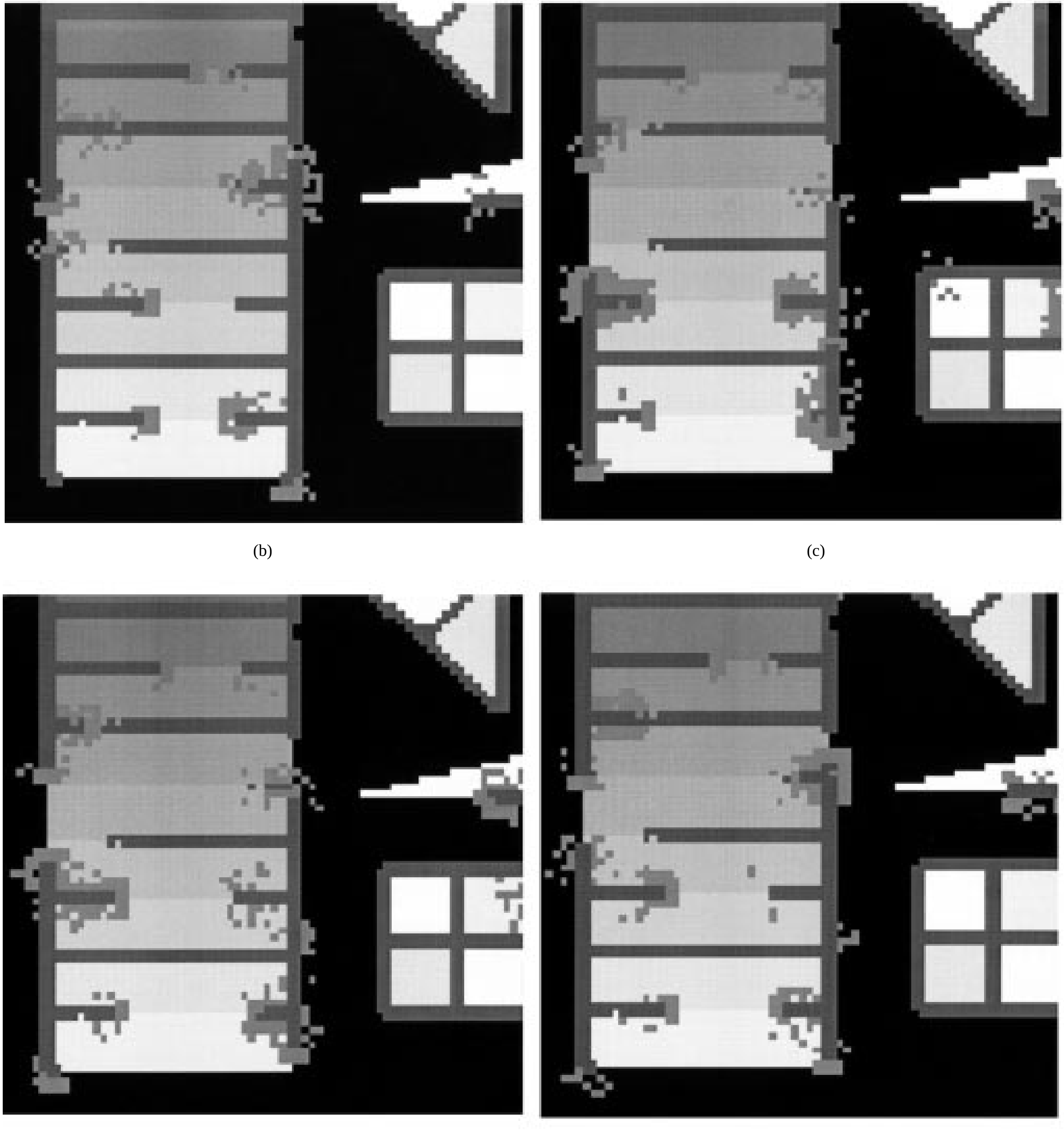
$$W_k = \alpha R_{k-\Delta} - \sum_{i=k-\Delta+1}^{k} \psi_i^{k-\Delta} \tag{8}$$

where $R_{k-\Delta}$ is computed using (7), and α denotes the number of offspring generated asexually by a single self-reproducing agent. The first term of this equation indicates all the agents reproduced at time $k - \Delta$. The second term indicates how many of them have found the feature pixels during a period from time $k - \Delta$ till time k. In other words, this equation expresses that the agents will vanish at time k if they exceed their life span.



(a)

Fig. 18. (a) Two spots are of particular interest, as identified with circles.



(b) (c)

(d) (e)

Fig. 18. *(Continued.)* (b)−(e) give the intermediate steps of agent evolution within the focused regions. It can be noted that the agents at these two spots have failed to branch (upper-left circle) and progress (lower-right circle) to connected feature pixels.

Equations (7) and (8) determine the entire population of active agents at time k, that is

$$U_k = U_0 + \sum_{j=1}^{k}(\alpha R_j) - \sum_{j=\Delta}^{k} W_j \tag{9}$$

where U_0 denotes the number agents initially distributed over the 2-D lattice.

Substituting R_j and W_j with (7) and (8), respectively, yields

$$U_k = U_0 + \alpha \sum_{j=1}^{k}\sum_{n=1}^{\Delta} \psi_j^{j-n} - \sum_{j=\Delta}^{k}\left(\alpha \sum_{n=1}^{\Delta} \psi_{j-\Delta}^{j-\Delta-n} - \sum_{n=j-\Delta+1}^{j} \psi_n^{j-\Delta}\right). \tag{10}$$

It can readily be noted that the growth rate of the active agent population is *positive* if the following is satisfied:

$$\alpha R_k - W_k > 0. \tag{11}$$

Based on (7) and (8), we can rewrite the above condition as follows:

$$\alpha R_k > \alpha R_{k-\Delta} - \sum_{i=k-\Delta+1}^{k} \psi_i^{k-\Delta}. \tag{12}$$

What may also be inferred from the above is that the growth rate of agents will decrease after the markers curve and the agent death curve intersect. The growth rate switches from positive to negative when the death rate is the highest.

B. A Continuous Model of Evolutionary Autonomous Agent Dynamics

The dynamics curves of the agent population as presented in Fig. 11 have offered the behavioral description of a dynamic population system that diffuses in discrete time with a time delay distributed over a specific interval of time. As a matter of fact, the shape of the population distribution as obtained from our experiment fits very well with that of the following continuous integro-differential equation as often used to model a logistic growth with distributed time delay [2], which reads:

$$\frac{1}{N(t)}\,\frac{dN(t)}{dt} = a - b\int_{-\infty}^{t} K(t-z)N(z)\,dz. \tag{13}$$

If let $U = Nb/a, T = at, M = bP, \sigma = s/a,$ and $\Gamma = r/a,$ a special case solution can be written as follows:

$$U = U_0\ \exp[(1-\rho)T + \frac{\rho}{\sigma}(1 - e^{-\sigma T})] \tag{14}$$

where $\rho = \Gamma U_0/\sigma$.

The derivation of (14) can be found in [2]. Some of the mathematical background readings on nonlinear partial differential equations can be found in [27].

C. Balance Between Evolution and Adaptation

In our experimentation, we have noted that there are some cases in which the inherited diffusion and self-reproduction directions could slow down the agents in adapting to their new local stimulus. Fig. 18(a) identifies two spots where this phenomenon is observed: one inside the upper-left circle and the other inside the lower-right circle. As shown in Fig. 18(b)–(e), in the former case the agents failed to branch horizontally to the new feature pixels, whereas in the latter the agents gradually stopped progressing along the preceding feature pixels.

Both cases occurred when the life span (or maximum age) of the agents was set shorter than three. In other words, the agents would have a great inertia to move or self-reproduce in the directions as inherited from the previously selected parent agents. Even though some of the offspring agents encountered new stimuli from other directions, the inherited directions had a strong bias that dominated the agent behaviors. This problem can be remedied if we allow the agents to survive for a longer period of time or we introduce a certain random movement and self-reproduction in the sectors other than the one as updated from the parent agents. In both cases, the agents would have a chance to gradually alter the preceding course of evolution.

The above observation has raised an interesting issue on the balance between the influence from the selected parent agents and the degree of freedom for the current agent to adapt. Our experiments have shown that it is always necessary to avoid the two extreme cases of agent behavioral selection, i.e., 1) completely random diffusion and self-reproduction and 2) complete inheritance.

VI. Conclusion

This paper described an evolutionary autonomous agent-based approach to image feature extraction. While giving the agent computation algorithm, the paper also presented several experimental results to demonstrate how the evolution of the distributed autonomous agents enables the optimal extraction of image features, and discussed the effects of behavioral parameters on the performance.

The advantages of the proposed approach can be summarized as follows:

1) the image feature extraction process is entirely determined by the locality and parallelism of the individual agents;
2) the directions for the diffusion and self-reproduction of the agents are dynamically selected and evolved.

With respect to real-life applications, the proposed approach could have significant impact on difficult image analysis problems, i.e., problems in which conventional edge and contrast enhancement have failed to extract important features. Examples of such applications are:

- identification of pathological foci of early stage cancer and important anatomical features (either not visible or not distinguishable) from ultrasound images of a prostate [30];
- identification of spiculated lesions, microcalcifications, and circumscribed lesions in scanning mammograms for breast cancer [14].

Acknowledgment

The authors wish to thank the anonymous reviewers and D. B. Fogel for their constructive comments and suggestions on both the technical contents and the presentation of this paper. In addition, special thanks should be given to Dr. H. Zhou for his involvement in some of the preliminary experiments and to H. Qin for his help in producing two of the graphical illustrations.

References

[1] T. D. Alter and R. Basri, “Extracting salient curves from images: An analysis of the saliency network,” MIT AI Lab, Cambridge, MA, Memo 1550, 1995.

[2] R. B. Banks, *Growth and Diffusion Phenomena: Mathematical Frameworks and Applications*. Berlin: Springer-Verlag, 1994.

[3] W. Banzhaf and F. H. Eeckman, Eds., *Evolution and Biocomputation: Computational Models of Evolution*. Berlin, Germany: Springer-Verlag, 1995.

[4] M. Barzohar and D. B. Cooper, “Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 7, pp. 707–721, 1996.

[5] S. Beucher, "Segmentation tools in mathematical morphology," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. Singapore: World Scientific, 1993, pp. 443–456.

[6] W. G. Chen, N. Nandhakumar, and W. N. Martin, "Image motion estimation from motion smear—A new computational model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 4, pp. 412–425, 1996.

[7] I. J. Cox and S. L. Hingorani, "An efficient implementation of Ried's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 2, pp. 138–150, 1996.

[8] F. Dellaert and R. D. Beer, "Toward an evolvable model of development for autonomous agent synthesis," in *Artificial Life IV: Proc. 4th Int. Workshop on the Synthesis and Simulation of Living Systems*, R. A. Brooks and P. Maes, Eds. Cambridge, MA: MIT Press, 1994, pp. 246–257.

[9] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 1995.

[10] L. J. Fogel, P. J. Angeline, and T. Bäck, Eds., *Proc. Evolutionary Programming V*. Cambridge, MA: MIT Press, 1996.

[11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, Vol. I*. Reading, MA: Addison-Wesley, 1992.

[13] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[14] W. P. Kegelmeyer, Jr., "Evaluation of stellate lesion detection in a standard mammogram data set," in *State of the Art in Mammographic Image Analysis*, K. Bowyer and S. Astley, Eds. Singapore: World Scientific, 1994, pp. 262–279.

[15] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[16] ——, *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: MIT Press, 1994.

[17] C. G. Langton, "Self-reproduction in cellular automata," *Physica D*, vol. 10, pp. 135–144, 1984.

[18] ——, "Studying artificial life with cellular automata," *Physica D*, vol. 22, pp. 120–140, 1986.

[19] ——, "Artificial life," in *Artificial Life: Proc. Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Redwood City, CA: Addison-Wesley, 1988, pp. 1–47.

[20] S-W. Lee and Y. J. Kim, "Direct extraction of topographic features for gray scale character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 7, pp. 724–729, 1995.

[21] Y. Liow, "A contour trancing algorithm that perserves common boundaries between regions," *CVGIP-Image Understanding*, vol. 53, no. 3, pp. 313–321, 1991.

[22] M. W. Lugowski, "Computational metabolism: Toward biological geometries for computing," in *Artificial Life: Proc. Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Redwood City, CA: Addison-Wesley, 1988, pp. 341–368.

[23] Q-T. Luong, "Color in computer vision," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. Singapore: World Scientific, 1993, pp. 311–368.

[24] P. Maes, "Modeling adaptive autonomous agents," *Artificial Life*, vol. 1, pp. 1–2, 1994.

[25] J. B. A. Maintz, P. A. van den Elsen, and M. A. Viergever, "Evaluation of ridge seeking operators for multimodality medical image matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 4, pp. 353–365, 1996.

[26] N. Merlet and J. Zerubia, "New products in line detection by dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 4, pp. 426–431, 1996.

[27] M. Mesterton-Gibbons, *A Concrete Approach to Mathematical Modeling*. New York: Addison-Wesley, 1989.

[28] Z. Michalewicz, *Genetic Algorithms + Data Structures + Evolution Programs*. Berlin: Springer-Verlag, 1992.

[29] H. Muhlenbein, "How genetic algorithms really work: I. Mutation and hill climbing," in *Parallel Problem Solving from Nature, 2*, R. Manner and B. Manderick, Eds. Amsterdam: North Holland, 1992.

[30] R. E. Muzzolini, Y. H. Yang, and R. A. Pierson, "A multiresolution texture segmentation approach with application to diagnostic ultrasound images," *IEEE Trans. Med. Imag.*, vol. 12, no. 1, pp. 108–123, 1993.

[31] J. Von Neumann, *Theory of Self-Reproducing Automata*. Urbana, IL: Univ. of Illinois Press, 1966.

[32] H-P. Schwefel, *Evolution and Optimum Seeking*. New York: John Wiley, 1995.

[33] M. Shanahan, "Evolutionary automata," in *Artificial Life IV: Proc. 4th Int. Workshop Synthesis and Simulation of Living Systems*, R. A. Brooks and P. Maes, Eds. Cambridge, MA: MIT Press, pp. 387–393, 1994.

[34] L. Steels, "Intelligence—dynamics and representations," in *The Biology and Technology of Intelligent Autonomous Agents*, L. Steels, Ed. Berlin, Germany: Springer-Verlag, 1995, pp. 72–89.

[35] P. Tamayo and H. Hartman, "Cellular automata, reaction-diffusion systems and the origin of life," in *Artificial Life: Proc. Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Redwood City, CA: Addison-Wesley, 1988, pp. 105–124.

Jiming Liu (S'91–M'91) received the M.A. degree in educational technology from Concordia University, Montreal, Canada, and the M.Eng. and Ph.D. degrees, both in electrical engineering, from McGill University, Montreal, Canada.

He is currently an Associate Professor of Computer Science at Hong Kong Baptist University (HKBU). He worked for several years as Software Engineer, Research Associate, Senior Research Agent, and Consultant at Canadian R & D firms and government laboratories, including Canadian Workplace Automation Research Center (Government of Canada), Kentek Inc., Virtual Prototypes, Inc., and Computer Research Institute of Montreal (CRIM), prior to joining HKBU in 1993. His research interests are in the areas of evolutionary computation, autonomous agents and systems, machine learning, and robotics.

Dr. Liu is a member of IEEE Computer Society, IEEE Robotics and Automation Society, and ACM.

Y. Y. Tang (S'88–M'88–SM'96) received the B.S. degree in electrical and computer engineering from Chongqing University, Chongqing, China, the M.Eng. degree in electrical engineering from the Graduate School of Post and Telecommunications, Beijing, China, and the Ph.D. degree in Computer Science from Concordia University, Montreal, Canada.

He is presently an Associate Professor in Department of Computing Studies at Hong Kong Baptist University, Hong Kong, and Senior Researcher in the Centre for Pattern Recognition and Machine Intelligence, Concordia University. He is an Advisory Professor in Beijing University of Posts and Telecommunications, and Chongqing University, China. His current interests include wavelet theory and applications, pattern recognition, image processing, artificial intelligence, and parallel processing.

Y. C. Cao received the B.Eng. degree in computer engineering from Beijing University of Post and Telecommunications, China.

He is currently a Research Assistant in the Autonomous Agents Research Group of Hong Kong Baptist University, Hong Kong. He has served as a Designer as well as a Technical Leader responsible for developing both the standard and the management system for the Chinese IC card payphone. He also heads the System and Network Institute of South China Post and Telecommunications Industry Research Academy in Guang Zhou Post and Telecommunications Equipment Company, China.