

Gustavo Olague · Cynthia B. Pérez · Francisco Fernández
Evelyne Lutton

An artificial life approach to dense stereo disparity

Received: October 27, 2005 / Accepted: September 25, 2008

Abstract This article presents an adaptive approach to improving the infection algorithm that we have used to solve the dense stereo matching problem. The algorithm presented here incorporates two different epidemic automata along a single execution of the infection algorithm. The new algorithm attempts to provide a general behavior of guessing the best correspondence between a pair of images. Our aim is to provide a new strategy inspired by evolutionary computation, which combines the behaviors of both automata into a single correspondence problem. The new algorithm will decide which automata will be used based on the transmission of information and mutation, as well as the attributes, texture, and geometry, of the input images. This article gives details about how the rules used in the infection algorithm are coded. Finally, we show experiments with a real stereo pair, as well as with a standard test bed, to show how the infection algorithm works.

Key words Image matching · Stereo-vision · Infection algorithm · Evolutionary computation

1 Introduction

Artificial life is devoted to the endeavor of understanding the general principles that govern the living state.¹ The field

of artificial life covers a wide range of disciplines, such as biology, physics, robotics, and computer science, to mention but a few. One of the first difficulties when studying artificial life is that there is no generally accepted definition of life. Moreover, an analytical approach to the study of life is impossible. If we attempt to study a living system through a gradual decomposition of its parts, we observe that the initial property of interest is not present anymore. Thus, there appears to be no elementary living thing; life seems to be a property of a collection of components, but not a property of the components themselves. On the other hand, classical scientific fields have proved that the traditional approach of a gradual decomposition of the whole is useful in modeling the physical reality. The aim of this article is to show that the problem of matching the contents of a stereoscopic system could be approached from an artificial life standpoint. Stereo matching is one of the most active research areas in computer vision. It consists of determining which pair of pixels projected on at least two images belongs to the same physical 3D point. The correspondence problem has been approached using sparse, quasidense, and dense stereo-matching algorithms. Sparse matching has normally been based on sparse points of interest to achieve a 3D reconstruction. Unfortunately, most modeling and visualization applications need dense or quasidense reconstruction, rather than sparse point clouds. In order to improve the quality of image reconstruction researchers have turned to the dense surface reconstruction approach.

The correspondence problem has been one of the main subjects in computer vision, and it is clear that these matching tasks can be solved by computer algorithms. Currently, there is no general solution to the problem, and it is also clear that successful matching by computer can have a large impact on computer vision. The matching problem has been considered the hardest and most significant problem in computational stereo. The difficulty is related to the inherent ambiguities being produced during the image acquisition concerning the stereo pair: for example, geometry, noise, lack of texture, occlusions, and saturation. Geometry concerns the shapes and spatial relationships between images and the scene. Noise refers to the inevita-

G. Olague (✉)
CICESE, Research Center, Applied Physics Division, Centro de Investigación Científica y de Educación Superior de Ensenada, B.C. Km. 107 carretera Tijuana-Ensenada 22860, Ensenada, B.C. Mexico
e-mail: olague@cicese.mx

C.B. Pérez
EvoVisión Laboratory, CICESE Research Center, Ensenada, B.C. Mexico

F. Fernández
Universidad de Extremadura, Computer Science Department,
Centro Universitario de Mérida, Mérida, Spain

E. Lutton
APIS Team, INRIA Saclay – Ile-de-France, Parc Orsay Université,
Orsay, France

ble variations of luminosity, which produce errors in the image formation due to a number of sources such as quantization, dark current noise, or electrical processing. Texture refers to the properties that represent the surface or structure of an object. Thus, lack of texture refers to the problem of unambiguously describing those surfaces or objects with similar intensity on gray values. Occlusions can be understood as those areas that appear in only one of the two images due to camera movement. Occlusion is the cause of complicated problems in stereo-matching, especially when there are narrow objects with a large disparity and optical illusions in the scene. Saturation refers to the problem of quantization beyond the dynamic range in which the image sensor normally works. Dense stereo-matching is considered an ill-posed problem. Traditional dense stereo methods are limited to specific precalibrated camera geometries and closely spaced viewpoints. Dense stereo disparity is a simplification of the problem in which the pair is considered to be rectified and the images are taken on a linear path with the optical axis perpendicular to the camera displacement. In this way, the problem of matching two images is evaluated indirectly through a univalued function in disparity space that best describes the shape of the surfaces in the scene. There is one more problem with the approach of studying dense stereo matching using the specific case of dense stereo disparity. In general, the quality of the solution is directly related to the contents of the image pair. In other words, the quality of the algorithms depends on the test images. In our previous work, we have presented a novel matching algorithm based on concepts from artificial life and epidemics that we called an infection algorithm.

The goal of this work is to show that the quality of the algorithm is comparable to the state-of-the-art publications in computer vision literature. We decided to test our algorithm with the test images provided at the Middlebury stereo vision web page.² However, we would like to mention that the problem is very difficult to solve, and the comparison is image-dependent. Moreover, the natural vision system is an example of a system in which the visual experience is a product of a collection of components, but not a property of the components. The infection algorithm presented by Olague et al.³ uses an epidemic automaton that propagates the pixel matches as an infection over the whole image with the purpose of matching the contents of two images. It searches the correspondences between real stereo images following a susceptible–exposed–infected–recovered (SEIR) model that leads to fast labeling. SEIR epidemics refer to diseases with incubation periods and latent infection. The purpose of the algorithm is to show that a set of local rules working over a spatial lattice could achieve the correspondence of two images using a guessing process. The algorithm provides the rendering of 3D information, allowing the visualization of the same scene from novel viewpoints. These new viewpoints are obviously different from the initial photographs. In our past work, we had four different epidemic automata in order to observe and analyze the behavior of the matching process. The best results that we obtained were 47% and 99%. The first case represents a geometrically good image with a moderate

percentage of computational effort saved. The second case represents a high percentage of automatically allocated pixels producing an excellent percentage of computational effort saved, and with an acceptable image quality.

Our current work aims at improving the results based on a new algorithm that uses concepts from epidemic processes in which the transmission of information and mutation is incorporated. We want to combine the best of both epidemic automata in order to obtain a high computational effort saving, with an excellent image quality. Therefore, we are proposing to use knowledge based on geometry and texture in order to decide, during the algorithm, which epidemic automata is better based on the neighborhood information. The benefit of the new algorithm will be shown in Sect. 3 through a comparison with previous results. As well as previous results, the new algorithm uses local information such as zero normalized cross correlation (ZNCC), geometric constraints (epipolar geometry, orientation), and a set of rules applied within the neighborhood. Our algorithm manages global information, which is encapsulated through the epidemic cellular automaton, and information about texture and edges in order to decide which automata it is more appropriate to apply.

This article is organized as follows. The next section describes the nature of the correspondence problem. Section 2 introduces the new algorithm, giving emphasis to an explanation of how the transmission of information was applied to decide between two epidemic automata. Finally, Sect. 3 shows the results of the algorithm, illustrating the behavior, performance, and quality of the evolutionary infection algorithm.

1.1 Problem statement

The problem with computational stereo studies is how to recover the three-dimensional characteristics of a scene from multiple images taken from different viewpoints.^{4–8} A major problem in computational stereo is how to find the corresponding points between a pair of images, which is known as the correspondence problem or stereo-matching. The images are taken by a moving camera in which a unique three-dimensional physical point is projected into a unique pair of image points. A pair of points should match each other in both images. A correlation measure can be used as a similarity criterion between image windows of fixed size. The input is a stereo pair of images, I_l (left) and I_r (right). The correlation metric is used by an algorithm that performs a search process in which the correlation gives the measure used to identify the corresponding pixels on both images. In this work, the infection algorithm attempts to maximize the similarity criterion within a search region. Let p_l with image coordinates (x, y) , and p_r with image coordinates (x', y') , be pixels in the left and right image, respectively, $2W + 1$ the width (in pixels) of the correlation window, $(I_l(x, y))$ and $(I_r(x', y'))$ the mean values of the images in the windows centered on p_l and p_r , respectively, $R(p_r)$ the search region in the right image associated with p_l , and $\phi(I_r, I_l)$ a function of both image windows. The ϕ function is defined

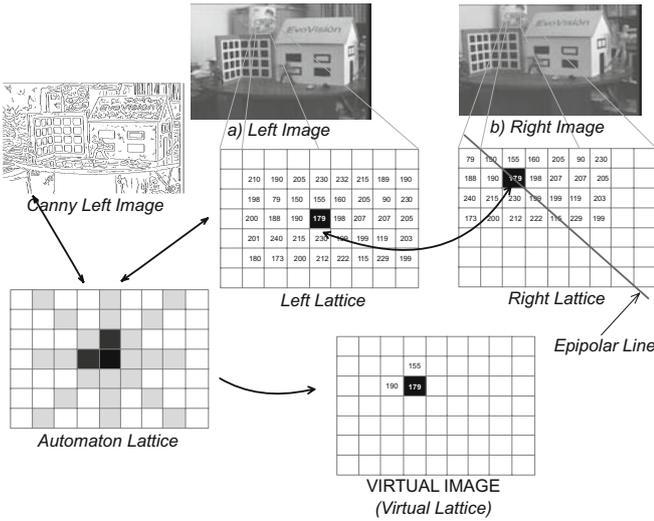


Fig. 1. Relationships between each lattice used by the infection algorithm. A pixel in the left image is related to the right image using a cellular automaton, a canny image, and a virtual image during the correspondence process. The blue color represents the sick (explored) state, and the black color represents the healthy (not explored) state

as the zero-mean normalized cross-correlation (ZNCC) in order to match the contents of both images.

$$\phi(I_l, I_r) = \frac{\sum_{i,j \in [-w,w]} [AB]}{\sqrt{\sum_{i,j \in [-w,w]} A^2 \sum_{i,j \in [-w,w]} B^2}} \quad (1)$$

$$A = (I_l(x+i, y+j) - \bar{I}_l(x, y))$$

$$B = (I_r(x'+i, y'+j) - \bar{I}_r(x', y'))$$

However, stereo matching has many complex aspects that make the problem intractable. In order to solve the problem, a number of constraints and assumptions are exploited which take into account occlusions, lack of texture, saturation, and field of view. Figure 1 shows two images taken at the Evo-Visión laboratory that we used in the experiments. The movement between the images is a translation with a small rotation along the x , y , and z axes, $T_x = 4.91$ mm, $T_y = 114.17$ mm, $T_z = 69.95$ mm, $R_x = 0.84^\circ$, $R_y = 0.16^\circ$, and $R_z = 0.55^\circ$. Figure 1 also shows five lattices that we have used in the evolutionary infection algorithm. The first two lattices correspond to the images acquired by the stereo rig. The third lattice is used by the epidemic cellular automaton in order to process the information that is being computed. The fourth lattice corresponds to the reprojected image, while the fifth lattice (canny image⁹) is used as a database in which we save information related to contours and texture. In this work, we are interested in providing a quantitative result to measure the benefit of using the infection algorithm. We decided to apply our method to the problem of dense two-frame stereo matching. For a comprehensive discussion of the problem, we refer the reader to the survey by Scharstein and Szeliski.² We performed our experiments on the benchmark Middlebury database. This database includes four stereo pairs, named *Tsukuba*, *Venus*, *Teddy*, and *Cones*. It is important to mention that the Middlebury test was limited to specific camera geometries and closely spaced viewpoints. Dense

stereo disparity is a simplification of the problem in which the pair is considered to be rectified, and the images are taken on a linear path with the optical axis perpendicular to the camera displacement. In this way, the problem of matching is evaluated indirectly through a univalued function in disparity space that best describes the shape of the surfaces in the scene. The performance of the infection algorithm has been evaluated according to the proposed methodology by Scharstein and Szeliski, in which the quality is measured by percentages of bad matching. The error is computed as the percentage of pixels far from the true disparity by more than 1 pixel.

2 Infection algorithm with an evolutionary approach

The infection algorithm is based on the concept of a natural virus for searching the correspondences between real stereo images. The purpose is to find all existing corresponding points in stereo images while saving the maximum number of calculations and maintaining the quality of the reconstructed data. The motivation to use what we called an infection algorithm is based on the following: when we observe a scene, we do not observe everything in front of us. Instead, we focus our attention in some regions which keep our interest in the scene. As a result, it is not necessary to analyze each part of the scene in detail. Thus, we pretend to “guess” some parts of the scene through a process of propagation based on artificial epidemics with the purpose of saving computational time.

The search process of the infection algorithm is based on a set of transition rules which are coded as an epidemic cellular automaton. These transition rules allow the development of global behaviors. The mathematical description of the infection algorithm was presented by Olague et al.¹⁰ Nevertheless, this article introduces the idea of evolution within the infection algorithm using the concepts of inheritance (transmission of information between individual cells) and mutation in order to achieve a balance between exploration and exploitation.¹¹ As can be seen, the idea of evolution is rather different from the traditional genetic algorithms. Concepts like an evolving population are not included in the evolutionary infection algorithm. Instead, we incorporate aspects such as inheritance and mutation to develop a dynamic matching process. Inheritance is understood here in the sense of information sharing and transmission between individual cells. Thus, a group of cells will decide which of two automata will be applied based on the probability of success computed from the local neighborhood using contour and texture information. In order to introduce the new algorithm, we now define some notations.

Definition 1: Cellular automata. A cellular automaton is a continuous map $G: S^L \rightarrow S^L$ which commutes with δ_i ($1 \leq i \leq d$).

However, this definition is not useful for computations. Therefore we consider an alternative characterization.

Given S , a finite set, and a d -dimensional shift space S^L , consider a finite set of transformations $N \subseteq L$. Given a function $f: S^N \rightarrow S$, called a local rule, the global cellular automaton map is given by

$$G_f(c)_v = f(c_v + N)$$

where $v \in L$, $c \in S^Z$, and $v + N$ consists of the set of translations of v by elements in N .

Definition 2: Epidemic cellular automata. Our epidemic cellular automaton can be formally introduced as a quadruple $E = (S, d, N, f)$, where $S = 5$ is a finite set composed of 4 states and the wild card (*), $d = 2$ a positive integer, $N \subset Z^d$ a finite set, and $f_i: S^N \rightarrow S$ an arbitrary set of (local) functions, where $i = \{1, \dots, 14\}$. The global function $G_f: S^L \rightarrow S^L$ is defined by $G_f(c)_v = f(c_v + N)$. It is also, useful to mention that S is defined by the following sets:

- $S = \{\alpha_1, \varphi_2, \beta_3, \varepsilon_0, *\}$ a finite alphabet;
- $S_f = \{\alpha_1, \beta_3\}$ is the set of final output states;
- $S_0 = \{\varepsilon_0\}$ is called the initial input state.

Our epidemic cellular automata have four states that are defined below. Let α_1 be the explored (sick) state, which represents the cells that have been infected by the virus (it refers to the pixels that have been computed in order to find their matches), ε_0 be the not-explored (healthy) state, which represents the cells which have not been infected by the virus (it refers to the pixels which remain in the initial state), ε_0 be the automatically allocated (immune) state, which represents the cells which cannot be infected by the virus. This state represents the cells which are immune to the disease (it refers to the pixels which have been confirmed by the algorithm in order to automatically allocate a pixel match). Finally, let φ_2 be the proposed (infected) state, which represents the cells which have acquired the virus with a probability of recovering from the disease (it refers to the pixels which have been guessed by the algorithm in order to decide later on the better match based on local information).

In previous work, we defined four different epidemic cellular automata from which we detect two epidemic graphs that provide singular results in our experiments (see Fig. 3). One epidemic cellular automaton produces a 47% saving of effort, while the other produces a 99% saving of effort. These automata use a set of transformations expressed by a set of rules grouped within a single graph. Each automaton transforms a pattern of discrete values over a spatial lattice. A whole different set of behaviors is achieved by changing the relationships between the four states using the same set of rules. Each rule represents a relationship which produces a transition based on local information. These rules are used by the epidemic graph to control the global behavior of the algorithm. In fact, the evolution of cellular automata is typically governed not by a function expressed in closed form, but by a “rule table” consisting of a list of the discrete states that occur in an automaton together with the values to which these states are to be mapped in one iteration of the algorithm.¹²

The goal of the search process is to achieve a good balance between two rather different epidemic cellular automata in order to combine the benefits of each automaton. Hence, our algorithm not only finds a match within the stereo pair, but also provides an efficient and general process using geometric and texture information. It is efficient because the final image combines the best of each partial image within the same amount of time. It is also general because the algorithm could be used with any pair of images with little additional effort to adapt it.

Our algorithm attempts to provide a remarkable balance between the exploration and exploitation of the matching process. Two cellular automata were selected because each one provides a particular characteristic from the exploration and exploitation standpoint. The 47% epidemic cellular automaton, which we called A, provides a strategy which exploits the best solution. Here, the best solution refers to areas where matching is easier to find. The 99% epidemic cellular automaton, which we called B, provides a strategy which explores the search space when the matching is hard to achieve.

The pseudocode for the evolutionary infection algorithm is depicted in Fig. 2. The first step consists of calibrating both cameras. Knowing the calibration for each camera, it is possible to compute the spatial relationship between two cameras. Then, two sets of rules that correspond to 47% and 99% are coded in order to decide which set of rules will be used during the execution of the algorithm. The sets of rules contain information about the configuration of the pixels in the neighborhood. Next, we built a lattice with the contour and texture information that we called the canny left image. Thus, we iterate the algorithm as long as the number of pixels with immune and sick states is different between times t and $t + 1$. Each pixel is evaluated according to a decision that is made based on the three following criteria.

1. The decision to use A or B is weighted by considering the current evaluated pixels in the neighborhood. In this way, the inheritance or transmission of information is incorporated within the algorithm through a combination of the set of rules that correspond to 47% and 99% automata. Each cell propagates to its neighbors which of the two automata it is best to apply in order to achieve a balance between exploration and exploitation.
2. The decision is also made based on the current local information (texture and contour). Thus, environmental adaptation is contemplated as a driving force in the dynamic matching process.
3. The probability of mutation which could change the decision of using A or B is computed. This provides the system with the capability of stochastically adapting the dynamic search process. This probability criterion depends on the local information of the neighborhood in order to decide which set of rules to use in the matching process. If the local information represents an image contour, there is more probability of using A than B. Otherwise, the probability of mutation is really smaller.

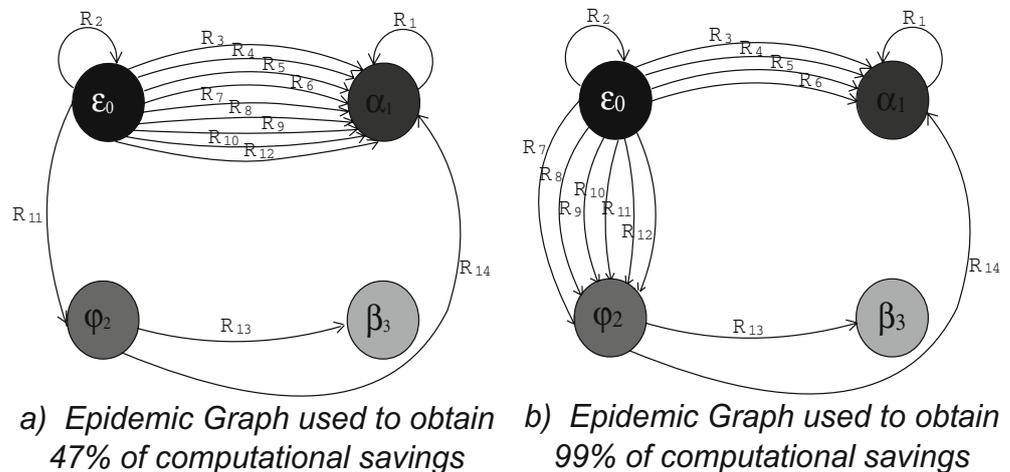
Fig. 2. Pseudocode for the evolutionary infection algorithm

```

1. BEGIN
2. Calculate projection matrices MM1 and MM2 of the left and right image.
3. Calculate fundamental matrix (F) left to right.
4. Coding of rules matrices 47% and 99%.
5. Process left image with the canny edge detector.
6. Initiate epidemic automata to Healthy (Not-Explored) state.
7. Corner detection of left image with canny edge detector.
8. Obtain the left image with canny edge detector.
9. While the number of Immune (Automatic) and Sick (Explored) states be
   different at time t and t+1 step.
10. DO
11.   FOR col=6 to N
12.     FOR row=6 to M
13.       neighbors=countKindNeighbor(row,col)
14.       pos=statesPosition(row,col)
15.       stateCell=automata[row][col].cell
16.       contEdge=findEdgeProbability(row,col)
17.       IF (contEdge > threshold)
18.         action= findRule(stateCell,neighbors,pos,Mrules47)
19.       ELSE
20.         action= findRule(stateCell,neighbors,pos,Mrules99)
21.       ENDFIF
22.       SWITCH(action)
23.         1:EXPLORED
24.         2:PROPOSED
25.         3:Analyse the inclination of the epipolar lines.
26.           IF the inclination is at least equal to the
27.             inclination of three neighbors.
28.             AUTOMATICALLY ALLOCATED
29.           ELSE
30.             EXPLORED
31.           ENDFIF
32.         ENDSWITCH
33.       ENDFOR
34.     ENDDO
35. Virtual image is obtained.
36. END

```

Fig. 3. Evolutionary epidemic graphs used in the infection algorithm



Thus, while the number of immune and sick cells do not change between times t and $t + 1$, the algorithm searches for the set of rules which better match the constraints. An action is then activated which produces a path and sequence around the initial cells. When the algorithm needs to execute a rule to evaluate a pixel, it calculates the corresponding epipolar line using the fundamental matrix information. The correlation window is defined and centered with respect to the epipolar line when the search process is started. This search process provides a nice balance between exploration and exploitation. The exploration process occurs when the epidemic cellular automata analyze the neighborhood around the current cell in order to decide where there is a good match. Once we find a good match, a process of

exploitation occurs in order to guess as many point matches as possible. Our algorithm not only executes the matching process, but also takes advantage of the geometric and texture information in order to achieve a balance between the results of the 47% and 99% epidemic cellular automata. This allows a better result in texture quality, as well as the geometric shape, and also saves computational effort. Figure 3 shows the two epidemics graphs for 47% and 99%, in which we can appreciate that the differences between the graphs are made by changing the relationships between the four states. Each relationship is represented as a transition based on a local rule, which as a set is able to control the global behavior of the algorithm. The next section explains how each rule works according to the above classification.

Fig. 4. Table summarizing the 14 rules we have used in the infection algorithm

$R_1: CC(\alpha_1) \rightarrow ACCION(\alpha_1)$
$R_2: CC(\epsilon_0) \wedge LU(\epsilon_0) \wedge CU(\epsilon_0) \wedge RU(\epsilon_0) \wedge LR(\epsilon_0) \wedge RR(\epsilon_0) \wedge LD(\epsilon_0) \wedge CD(\epsilon_0) \wedge RD(\epsilon_0) \rightarrow ACCION(\epsilon_0)$
$R_3: CC(\epsilon_0) \wedge LR(\epsilon_0) \wedge RR(\epsilon_0) \wedge LD(\epsilon_0) \wedge CD(\epsilon_0) \wedge RD(\alpha_1) \rightarrow ACCION(\alpha_1)$
$R_4: CC(\epsilon_0) \wedge CU(\epsilon_0) \wedge RU(\alpha_1) \wedge RR(\alpha_1) \wedge CD(\epsilon_0) \rightarrow ACCION(\alpha_1)$
$R_5: CC(\epsilon_0) \wedge LU(\alpha_1) \wedge CU(\epsilon_0) \wedge LR(\alpha_1) \wedge CD(\epsilon_0) \rightarrow ACCION(\alpha_1)$
$R_6: CC(\epsilon_0) \wedge LU(\alpha_1) \wedge CU(\alpha_1) \wedge RU(\alpha_1) \wedge LR(\epsilon_0) \wedge RR(\epsilon_0) \rightarrow ACCION(\alpha_1)$
$R_7: CC(\epsilon_0) \wedge EXP(3) \wedge EXT(3) \rightarrow ACCION(\alpha_1)$
$R_8: CC(\epsilon_0) \wedge EXP(3) \wedge AUT(3) \rightarrow ACCION(\alpha_1)$
$R_9: CC(\epsilon_0) \wedge CU(\epsilon_0) \wedge LR(\alpha_1) \wedge RR(\epsilon_0) \wedge LD(\alpha_1) \wedge CD(\varphi_2) \rightarrow ACCION(\alpha_1)$
$R_{10}: CC(\epsilon_0) \wedge CU(\epsilon_0) \wedge LR(\epsilon_0) \wedge RR(\alpha_1) \wedge CD(\varphi_2) \wedge RD(\alpha_1) \rightarrow ACCION(\alpha_1)$
$R_{11}: CC(\epsilon_0) \wedge EXP(3) \rightarrow ACCION(\varphi_2)$
$R_{12}: CC(\epsilon_0) \wedge PROP(3) \rightarrow ACCION(\alpha_1)$
$R_{13}: CC(\varphi_2) \wedge EXP(3) \rightarrow ACCION(\beta_3)$
$R_{14}: CC(\varphi_2) \wedge PROP(3) \rightarrow ACCION(\alpha_1)$

LU(), CU(), RU(), LR(), RR(), LD(), CD() and RD(): These functions evaluate the corresponding location in the neighborhood

ACCION(): This function provides the output state of the rule

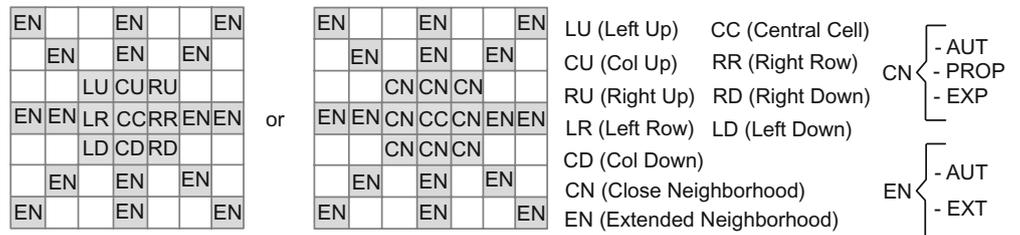
EXP(): This function represents the number of explored pixels in the close neighborhood

PROP(): This function represents the number of proposed pixels in the close neighborhood

AUT(): This function represents the number of automatically allocated pixels in the close and extended neighborhoods

EXT(): This function represents the number of automatically allocated pixels in the extended neighborhood.

Fig. 5. The layout of the neighborhood used by the cellular automata



2.1 Transitions of our epidemic automata

Each epidemic graph has 14 transition rules that we divide into three classes: basic rules, initial structure rules, and complex structure rules. Each rule could be represented as a predicate that encapsulates an action allowing a change of state on the current cell based on the neighborhood information. The basic rules relate the obvious information between the initial and explored states. The initial structure rules consider only the spatial set of relationships between close neighborhoods. The complex structure rules consider the spatial set of relationships between the close neighborhood and also those within the extended neighborhood. The transitions of our epidemic automata are based on a set of rules (Fig. 4). These rules are coded according to the neighborhood that is shown in Fig. 5. The basic rules correspond to rules 1 and 2. The initial structure is formed by rules 3–6. Finally, the complex structure rules correspond to the rest of the rules. Until this moment, the basic and initial structure rules have not been changed, and they

remain without changes. The rest of the rules are easily modified in order to produce different behaviors and a certain percentage saving of computational effort. The 14 epidemic rules related to the case of 47% are explained below.

- **Rules 1 and 2.** The epidemic transitions of these rules represent two obvious actions. First, if it lacks information in the close neighborhood, then no change is made in the current cell. Second, if the central cell was already sick (Explored), then no change is produced in the current cell (Figs. 6 and 7).
- **Rules 3–6.** The infection algorithm begins the process with the nucleus of infection around the whole image. The purpose of creating an initial structure in the matching process is to explore the search space in such a way that the information is distributed in several processes. Thus, the propagation of the matching is realized in a higher number of directions from the central cell. We use these rules only during the beginning of the process (Fig. 8).

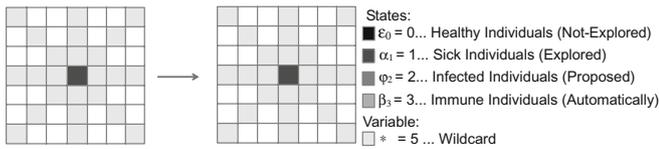


Fig. 6. Rule 1. The central cell state does not change if the pixel has already been evaluated

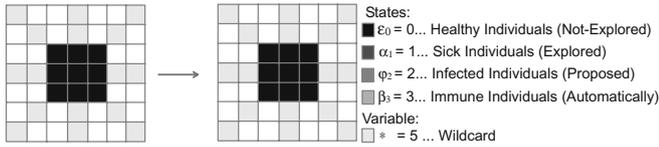


Fig. 7. Rule 2. The central cell state does not change if there is a lack of information in the neighborhood

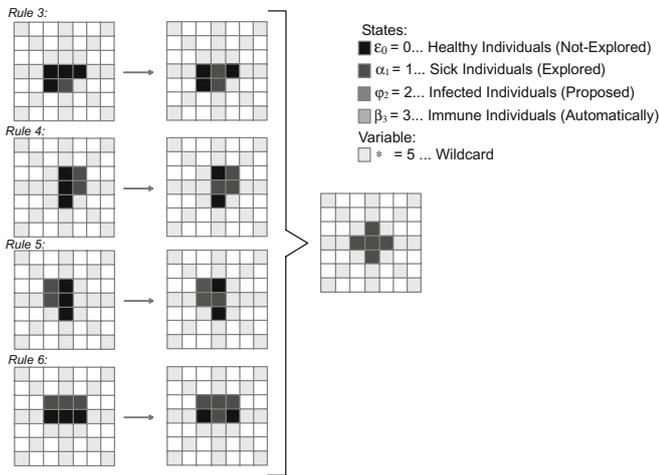


Fig. 8. Transition of rules 3–6. These four rules create the initial structure for the propagation

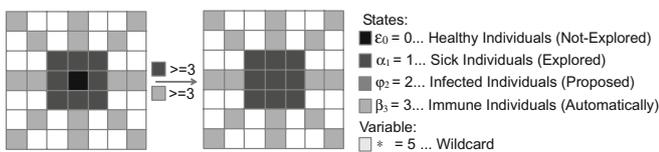


Fig. 9. Rule 7. This epidemic transition indicates that it is necessary to have at least three sick (explored) individuals in the close neighborhood, and at least three immune (automatically allocated) individuals in the extended neighborhood in order to change the central cell

- **Rules 7 and 8.** These rules ensure the evaluation of the pixels in a region where immune (automatically allocated) individuals exist (Fig. 9). The figure of rule 8 is similar to that of rule 7. The main purpose of these rules is to control the quantity of immune individuals within a set of regions (Fig. 10).
- **Rules 9 and 10.** These transition rules avoid the linear propagation of infected (proposed) individuals. Rules 9 and 10 take into account the information of the close neighborhood and one cell of the extended neighborhood (Fig. 11).

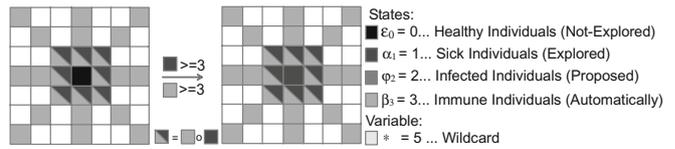


Fig. 10. Rule 8. This epidemic transition indicates that it is necessary to have at least three sick (explored) pixels in the close neighborhood, and at least three immune pixels in the whole neighborhood in order to change the central cell

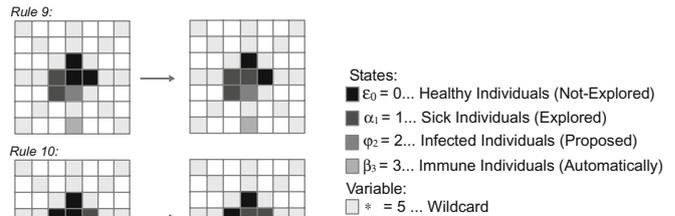


Fig. 11. Rules 9 and 10. These rules avoid the linear propagation of the infected (proposed) pixels

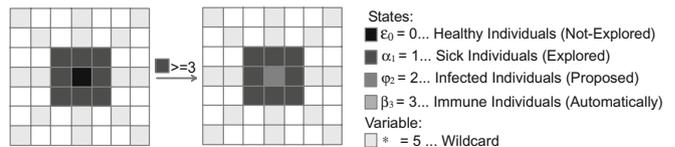


Fig. 12. Rule 11. This epidemic transition rule represents the quantity of infected (proposed) individuals needed in order to obtain the immune (automatically allocated) individuals. In this case, if there are three sick (explored) individuals in the close neighborhood, then the central cell changes to an infected (proposed) state

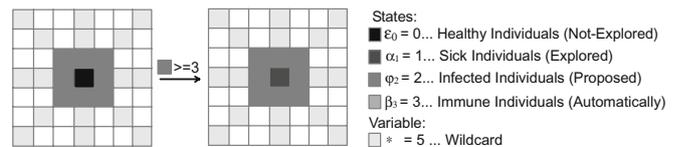


Fig. 13. Rule 12. This transition controls the infected (proposed) individuals within a region. It requires at least three infected (proposed) pixels around the central cell

- **Rule 11.** This rule generates infected (proposed) individuals in order to obtain a higher number of immune (automatically allocated) individuals later. If the central cell is in a healthy state (not explored) and there are at least three sick individuals (explored) in the close neighborhood, then the central cell is infected (proposed) (Fig. 12).
- **Rules 12 and 14.** The reason for these transitions is to control the infected (proposed) individuals. If we have at least three infected (proposed) individuals in the close neighborhood, the central cell is evaluated (Figs. 13 and 14).
- **Rule 13.** This rule is one of the most important epidemic transition rules because it indicates the saving in

computational effort of individual pixels during the matching process. Rule 13 can be summarized as follows: if the central cell is infected (proposed) and there are at least three sick (explored) individuals in the close neighborhood, then we automatically guess the corresponding pixel in the right image without any computation. The number of sick (explored) individuals can be changed according to the desired percentage of computational savings (Fig. 15).

3 Experimental results

We have tested an infection algorithm with an evolutionary approach on a real stereo pair of images. The infection algorithm was implemented under the Linux operating system on an Intel Pentium 4 at 2.0 Ghz with 256 Mb of RAM. We have used libraries programmed in C++, designed

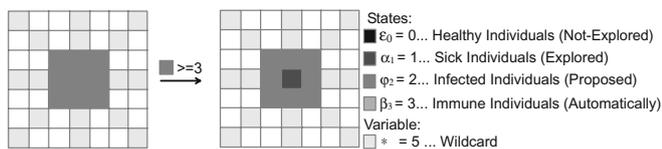


Fig. 14. Rule 14. This epidemic transition controls the infected (proposed) individuals in different small regions of the image. If the central cell is in an infected (proposed) state, then the central cell is evaluated

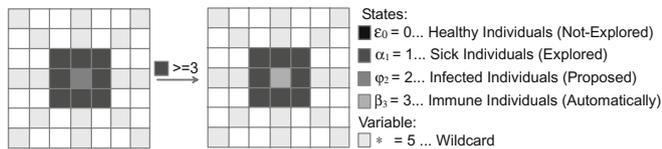


Fig. 15. Rule 13. This transition indicates when the infected (proposed) individuals will change to immune (automatically allocated) individuals

Fig. 16. Results of different experiments in which the rules were changed to contrast with the epidemic cellular automata



a) Final view with 47% savings.



b) Final view with 70% savings.



c) Final view with 99% savings.



d) Final view with evolution of 47% and 99% epidemic automata.

especially for computer vision, called VxL (Vision x Libraries). The runtime of the evolutionary infection algorithm with a 99% saving of operations is about 2:16 min compared with 17:18 min for a traditional exhaustive search algorithm. That means the time saving is about 87% compared with the exhaustive search. We have proposed to improve the results obtained by the infection algorithm through the implementation of an evolutionary approach using inheritance and mutation operations. The idea was to combine the best of both the 47% and 99% of the epidemic automata, in order to obtain a high saving of computational effort together with an excellent image quality. During the correspondence process, we used knowledge based on geometry and texture in order to decide which epidemic automaton it was better to apply during the evolution of the algorithm. Figure 16 shows a set of experiments where the epidemic cellular automaton was changed in order to modify the behavior of the algorithm and to obtain a better virtual image. Figure 16a is the result of obtaining a 47% savings in computational effort, while Fig. 16b is the result of obtaining a 70% savings in computational effort, and Fig. 16c shows the result of obtaining a 99% savings in computational effort. Figure 16d presents the result we obtain with the new algorithm. Clearly, the final image shows how the algorithm combines both epidemic cellular automata. We can see that the geometry is preserved with a nice texture reconstruction. We can also see that the new algorithm takes about the same time as the 70% epidemic cellular automaton with a slightly better texture result. Figure 17a shows the behavior of the evolutionary infection algorithm that corresponds to the final result in Fig. 16d. Figure 17b shows the behavior of the two epidemic cellular automata during the execution of the correspondence process.

We decided to test the infection algorithm with the standard test bed used in the computer vision community.² Scharstein and Szeliski have set up test data along with the ground truth available on their website to use it as, test bed for the quantitative evaluation and comparison of different

Fig. 17. Evolution of the epidemic cellular automata to solve dense correspondence matching

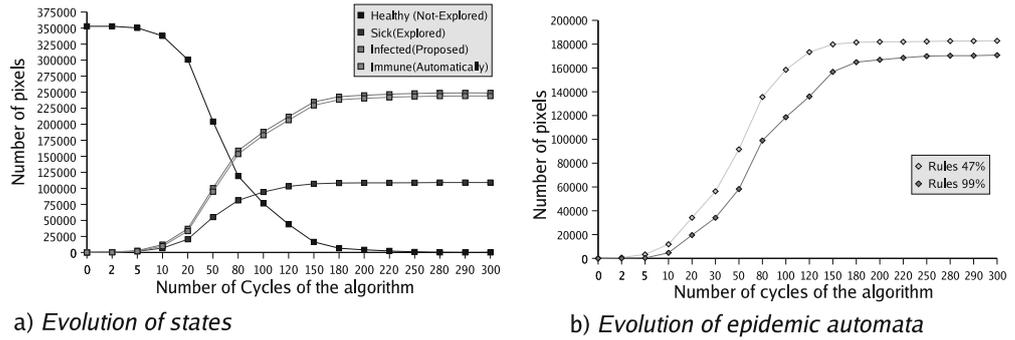


Table 1. Our result on the Middlebury database. The numbers represent the percentage of “bad” pixels computed over different regions ($B_{\bar{O}}$, $B_{\bar{T}}$, and $B_{\bar{D}}$), and the number in parentheses represents the algorithm ranking in the Middlebury stereo vision webpage

Infection algorithm	Tsukuba			Venus		
	$B_{\bar{O}}$	$B_{\bar{T}}$	$B_{\bar{D}}$	$B_{\bar{O}}$	$B_{\bar{T}}$	$B_{\bar{D}}$
Original 1st Interpretation	8.90 7.95 (24)	10.5 9.54 (23)	41.6 28.9 (24)	5.33 4.4 (21)	6.42 5.53 (21)	42.5 31.7 (23)
	Teddy			Cones		
	$B_{\bar{O}}$	$B_{\bar{T}}$	$B_{\bar{D}}$	$B_{\bar{O}}$	$B_{\bar{T}}$	$B_{\bar{D}}$
Original 1st Interpretation	19.0 17.7 (23)	25.3 25.1 (23)	45.6 44.4 (24)	15.2 14.3 (24)	21.9 21.3 (23)	39.3 38.0 (24)

stereo algorithms. They use two quality measures, the root-mean-square (RMS) error, and the percentage of bad matching pixels. These metrics are based on known ground truth data in order to evaluate the performance of a stereo algorithm.

The following quality metric is used to represent the percentage of “bad” pixels in three common problem areas for stereo algorithms: $B_{\bar{O}}$, bad pixels in nonoccluded areas; $B_{\bar{T}}$, bad pixels in textureless areas; $B_{\bar{D}}$, bad pixels in areas near depth discontinuities.

$$B = \frac{1}{N} \sum (|d_c(x, y) - d_r(x, y)| > \delta_d) \quad (2)$$

where B represent the total matching error, and is used for $B_{\bar{O}}$, $B_{\bar{T}}$, or $B_{\bar{D}}$ depending on the area to be analyzed. N is the total pixels of the area, $d_c(x, y)$ is the disparity map values, $d_r(x, y)$ is the ground truth map values, and δ_d is the disparity error tolerance used (i.e., 1.0). Note that this metric was designed to measure algorithms from a correctly matched standpoint.

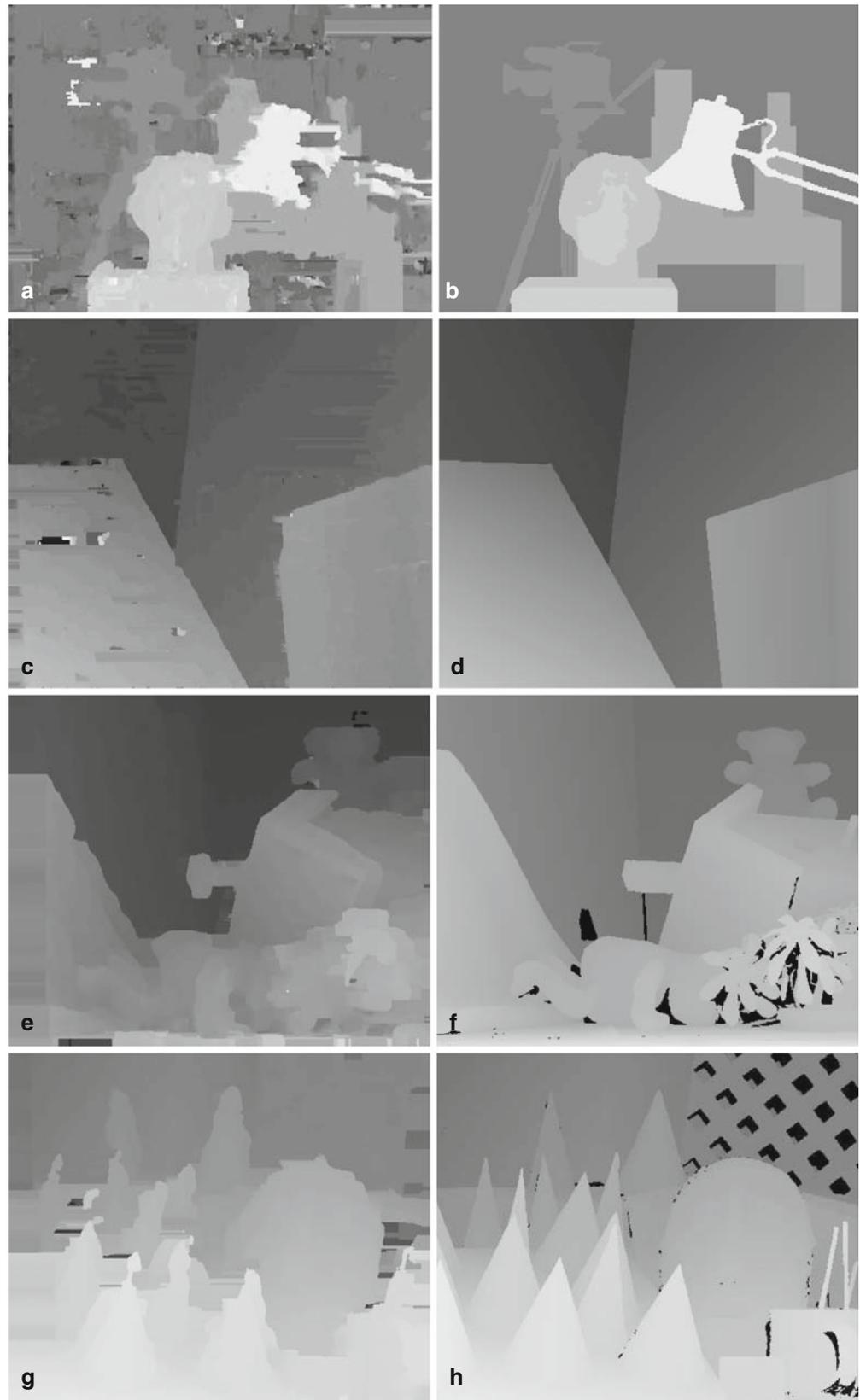
In general, the results of the algorithms available for comparison use subpixel resolution and a global approach to minimize the disparity. The original images can be obtained in gray-scale and color versions. We use the gray-scale images even if this represents a drawback with respect to the final result. The first column in Fig. 18 shows the four left images used as benchmark, as well as the computed disparity maps obtained with the infection algorithm. Because we tried to compute the best possible disparity map, we applied a 0% saving in order to be consistent with the benchmark. The results are visually comparable to other

algorithms that make similar assumptions: gray-scale image, window-based approach, and pixel resolution.¹³⁻¹⁵ In fact in the Middlebury database, the infection algorithm is officially ranked at the bottom of the chart where other algorithms with similar assumptions are classified. In order to improve the test results, we decide to enhance the quality of the input image with an interpolation approach.¹⁶ According to Table 1, the statistics represent the percentage of “bad” pixels over different image regions using the infection algorithm with the original and first interpolated images. The original images are the result of the infection algorithm with a saving of 0%, while the first interpolated images are the result of using the interpolation approach with a saving of 0%. The data used to evaluate the stereo algorithm in Table 1 were collected for all unoccluded image pixels (column $B_{\bar{O}}$), for all unoccluded pixels in the untextured regions (column $B_{\bar{T}}$), and finally for all unoccluded image pixels close to a disparity discontinuity (column $B_{\bar{D}}$). In this way, we obtained the new disparity images shown in Fig. 19 together with the ground truth. These results show that the same algorithm could be ameliorated if the resolution of the original images was improved with a first interpolation technique.⁷ We provide the results to illustrate the quality that can be achieved with the infection algorithm. However, as we have explained in the introduction, the infection algorithm was realized to explore the field of artificial life using the correspondence problem. Therefore, the final judgment should also be made from the standpoint of the ALife community. Note that the aim of the infection algorithm is to show the possibility of obtaining coherent disparities after a high number of guessed

Fig. 18. These four pairs of images were obtained from the Middlebury stereo matching web page. **a** Tsukuba left image. **b** Our result. **c** Venus left image. **d** Our result. **e** Teddy left image. **f** Our result. **g** Cones left image. **h** Our result



Fig. 19. The final computed disparity maps illustrate the quality of the infection algorithm. **a** Our improved image. **b** Ground truth. **c** Our improved image. **d** Ground truth. **e** Our improved image. **f** Ground truth. **g** Our improved image. **h** Ground truth



pixels. In the near future, we expect to use the evolutionary infection algorithm in the search of novel vantage viewpoints.

4 Conclusion

This article has shown that the problems of dense stereo matching and dense stereo disparity can be approached from an artificial life standpoint. We believe that the complexity of the problem reported in this research, and its solution, should be considered as a rich source of ideas in the artificial life community. A comparison with a standard test bed provides enough confidence that this kind of approach can be considered as state-of-the-art. The best algorithms use knowledge currently not used in our implementation. This point provides a clue for future research in which some hybrid approaches could be proposed by other researchers in the ALife community.

Acknowledgments This research was funded by CONACyT and INRIA through the LAFMI project 634-212 and Junta de Extremadura Government. The second author was supported by scholarship 0416442 from CONACyT.

References

1. Adami C (1998) Introduction to artificial life. Springer, TELOS
2. Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int J Comput Vision (IJCV)* 47(1–3): 7–42
3. Olague G, Fernández F, Pérez CB, et al (2004) The infection algorithm: an artificial epidemic approach to dense stereo matching. In: Yao X, et al (eds) *Parallel problem solving from nature, VIII. Lecture Notes in Computer Science 3242*, Birmingham, Springer, pp 622–632
4. Brown MZ, Burschka D, Hager GD (2003) Advances in computational stereo. *IEEE Trans Patt Anal Mach Intell* 25:993–1008
5. Fielding G, Kam M (2000) Weighted matchings for dense stereo correspondence. *Patt Recog* 33:1511–1524
6. Luo Q, Zhou J, Yu S, et al (2003) Stereo matching and occlusion detection with integrity and illusion sensitivity. *Patt Recog Lett* 24:1143–1149
7. Sun J, Zheng NN, Shum HY (2003) Stereo matching using belief propagation. *IEEE Trans Patt Anal Mach Intell* 25:787–800
8. Zitnick CL, Kanade T (2000) A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans Patt Anal Mach Intell* 22:675–684
9. Canny JF (1986) A computational approach to edge detection. *IEEE Trans Patt Anal Mach Intell* 8:679–698
10. Olague G, Fernández F, Pérez CB, et al (2006) The infection algorithm: an artificial epidemic approach for dense stereo matching. *Artificial Life* 12(4):593–615
11. Pérez CB, Olague G, Fernández F, et al (2005) An evolutionary infection algorithm for dense stereo correspondence. 7th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, *Lecture Notes in Computer Science 3449*, Lausanne, Springer, pp 294–303
12. Sipper M (1997) *Evolution of parallel cellular machines*. Springer, Heidelberg
13. Sun C (2001) Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *IEEE Computer Vision and Pattern Recognition 2001 Stereo Workshop*. *Int J Comput Vision*
14. Birchfield S, Tomasi C (1998) Depth discontinuities by pixel-to-pixel stereo. *IEEE International Conference on Computer Vision*, Mumbai, India, January 1998, pp 1073–1080
15. Shao J (2001) Combination of stereo, motion and rendering for 3D footage display. *IEEE Computer Vision and Pattern Recognition 2001 Stereo Workshop*. *Int J Comput Vision*
16. Legrand P, Levy-Vehel J (2003) Local regularity-based image denoising. *IEEE International Conference on Image Processing*, vol. 3, Barcelona, Spain, September 2003, pp 377–380