The Infection Algorithm: An Artificial Epidemic Approach for Dense Stereo Correspondence

Gustavo 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.
México
olague@cicese.mx

Francisco Fernández
Universidad de Extremadura
Computer Science Department
Centro Universitario de Merida
C/Sta Teresa de Jornet, 38
06800 Merida, Spain

Cynthia B. Pérez
EvoVisión Laboratory
CICESE Research Center
Km. 107 carretera Tijuana-Ensenada
22860, Ensenada, B.C.
México

Evelyne Lutton
INRIA Rocquencourt
Complex Team Domaine de
Voluceau, BP 105
78153 Le Chesnay Cedex
France
Evelyne.Lutton@inria.fr

Abstract We present a new bio-inspired approach applied to a problem of stereo image matching. This approach is based on an artificial epidemic process, which we call the *infection algorithm*. The problem at hand is a basic one in computer vision for 3D scene reconstruction. It has many complex aspects and is known as an extremely difficult one. The aim is to match the contents of two images in order to obtain 3D information that allows the generation of simulated projections from a viewpoint that is different from the ones of the initial photographs. This process is known as view synthesis. The algorithm we propose exploits the image contents in order to produce only the necessary 3D depth information, while saving computational time. It is based on a set of distributed rules, which propagate like an artificial epidemic over the images. Experiments on a pair of real images are presented, and realistic reprojected images have been generated.

Keywords
Image matching, epipolar geometry, stereo vision, artificial epidemics, infection algorithm

1 Introduction

The problem of matching the contents of a stereoscopic system is a classical (and extremely difficult) problem in computer vision. Elastic image matching has been shown recently to be NP-complete [12]. Stereo image matching has multiple practical applications, including robot navigation, object

* Author to whom correspondence should be addressed.

recognition, and more recently, realistic scene visualization or image-based rendering. Many approaches exist today that attempt to solve this problem. These approaches are now classified into local and global methods. Local methods include block matching, gradient-based optimization, and feature matching, while global approaches include dynamic programming, intrinsic curves, graph cuts, nonlinear diffusion, belief propagation, and correspondenceless methods [3].

The matching of two images can be formulated as a constrained optimization problem. It is however necessary to deal with a huge search space and with some aspects that are hard to model, such as the occlusion of parts in the scene, regions with a regular pattern, or even regions with similar textures. Classical techniques are limited and fail, due to the complexity and nature of the problem [7, 14, 24, 29]. This is the main reason why artificial life and evolutionary methods are considered now in this framework [18, 13].

Our work aims to improve the speed and the quality of stereo matching, while considering simultaneously local and global information [21]. The local information is obtained from the correlation or objective function used to match the contents of the two images, and from the constraints used to improve the matching process. The objective function and constraints are considered as local because they only depend on a local neighborhood. The global information is encapsulated within the algorithm that we will describe in Section 5.

Artificial life obviously does not try to reproduce natural life, but to create systems that generate characteristics, responses, or behaviors similar to the ones observed in natural systems [2]. One of the favorite topics of this domain is the study of emergence of complex behaviors from a set of elementary components acting according to simple rules. This work can be considered as a first attempt to study dense stereo matching from an artificial life standpoint.

This study is also based on the idea of persistence of vision. This natural phenomenon can be described as follows: the perceptual processes of the brain or the retina of the human eye retains an image for a split second. In other words, when the human eye is presented with a rapid succession of slightly different images, there is a brief period during which each image, after its disappearance, persists upon the retina, allowing that image to blend smoothly with the next image. Such an explanation indeed was the first to account for a sense of constancy of the light source. This idea was further developed to explain motion; it is, of course, a totally inadequate explanation of the illusion of motion in cinema. Instead, the eye-brain system has a combination of motion detectors, detail detectors, and pattern detectors, the outputs of all of which are combined to create the visual experience.

This work represents a first study that attempts to propose how a machine could achieve dense stereo matching using the principles of artificial life with the aim of providing a new and rich research avenue into how machine vision could be benefited by bio-inspired approaches.

This article is organized as follows. The next section describes the nature of the matching problem. Then, we introduce the principles of modeling and calibrating cameras. After that, we explain the concepts of stereo vision and epipolar geometry in order to derive a criterion, as well as a number of constraints, on dense stereo matching. Section 5 introduces the principles of the proposed algorithm, with the main goal of saving on the number of calculations while maintaining the quality of the virtual reprojected image. Section 6 explains the infection algorithm. Then, a set of images and graphs illustrates the behavior and performance of the infection algorithm. Section 8 concludes the article and proposes possible future research directions.

2 Statement of the Problem

Computational stereo, or the correspondence problem, refers to the problem of determining the three-dimensional structure of a scene from two or more images taken from different viewpoints. The fundamental assumption used by stereo algorithms is that a single three-dimensional physical point projects onto a unique pair of image points for two cameras. On the other hand, if it is possible to locate the image projected points on two observing cameras that correspond to the same physical point in space, then it is possible to calculate its precise 3D position. However, the problem becomes

(a) Left image. (b) Right image.

Figure 1. A stereo pair taken at the EvoVisión laboratory. Notice several classical problems: occlusion, textureless areas, sensor saturation, and optical constraints.

intractable for several reasons, so that there exists no closed-form solution to it. Stereo matching is an ill-posed problem with inherent ambiguities, due to ambiguous matches produced by occlusions, specularities, or lack of texture. Therefore, a variety of constraints and assumptions are commonly exploited to make the problem tractable. Figure 1 shows a classical pair of images illustrating the usual difficulties of computational stereo. The movement between the two images is typically a translation plus a small rotation. The wooden house and the calibration grid are clearly visible in both images. However, the wooden puppet only appears in the right image, while the file organizer behind the box with the children face only appears on the left image.

These are examples of visibility constraints: due to obstructions in the workplace, points of interest that lie within the field of view are not all projected on both cameras. Thus, the visibility of a feature of an object from a particular viewpoint depends on whether it is occluded, either by some part of the object itself (self-occlusion), or by other objects in the environment. This problem is due to the opaqueness of the objects. Additionally, if the feature lies outside the field of view of the sensor, it is not visible. Hence, matching is impossible for these cases. Moreover, matching large homogeneous (textureless) areas is also an ill-posed problem, since there is no way to decide which point corresponds to which other point over the entire area (no characteristic features, and hence ambiguity). This is clearly the case for the EvoVisíon house as well as for the calibration grid of the test stereo pair.

3 Background of Modeling and Calibrating Cameras

Before explaining our new algorithm we must first define some notation. According to the problem statement, we are working within the stereo-vision framework, which is represented by two cameras observing the same scene. Each camera is modeled from a geometric standpoint according to the pinhole camera model. This model is based on the fundamental assumption that the exposure center, the ground point and its corresponding image point all lie on a straight line.

Let u_{ij} and v_{ij} denote the photo coordinates of the image of point j in photograph i. For each pair of image coordinates $(u_{ij}, v_{ij})^t$ observed on each image the following relationship exists:

$$\begin{aligned} u_{ij} &= \frac{m^i_{11}X_j + m^i_{12}Y_j + m^i_{13}Z_j + m^i_{14}}{m^i_{31}X_j + m^i_{32}Y_j + m^i_{33}Z_j + m^i_{34}}, \\ v_{ij} &= \frac{m^i_{21}X_j + m^i_{22}Y_j + m^i_{23}Z_j + m^i_{24}}{m^i_{31}X_j + m^i_{32}Y_j + m^i_{33}Z_j + m^i_{34}}. \end{aligned} \tag{1}$$

This system of equations assumes that light rays travel in straight lines, that all rays entering a camera lens system pass through a single point, and that the lens system is distortionless or, as is usual in highly accurate measurement, that distortion has been canceled out after having been estimated. In this way, a point in the scene P_j, $j = 1, \ldots, n$, of homogeneous coordinates $(X_j, Y_j, Z_j, 1)^t$ is projected into points p_{ij} of image coordinates $(u_{ij}, v_{ij})^t$, through a projection matrix M_i, $i = 1, \ldots, k$, of size 3×4 corresponding to the ith image. In this case $k = 2$. Therefore, three-dimensional measurements can be obtained from several images. Each matrix M represents a mapping composed of a transformation $W \rightarrow C$ from the world coordinates W to the camera coordinates C given by

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} R_{\mathrm{WC}} & T_{\mathrm{WC}} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2}$$

where the rotation matrix R_{WC}, which is a function of three rotation parameters (α, β, γ) and the translation vector T_{WC}, also of three degrees of freedom, characterizes the camera's orientation and position with respect to the world coordinate frame. Under perspective projection, the transformation from the 3D-world coordinate system to the 2D-image coordinates is

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = K \begin{bmatrix} R_{\mathrm{WC}} & T_{\mathrm{WC}} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3}$$

where the matrix

$$K = \begin{pmatrix} -k_u f & 0 & u_0 & 0 \\ 0 & k_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

represents the intrinsic parameters of the camera, f is the focal length of the camera, (k_u, k_v) are the horizontal and vertical pixel sizes on the image plane, and (u_0, v_0) is the projection of the camera's center (principal point) on the image plane.

Calibration is the process of estimating the intrinsic and extrinsic parameters of a camera. It can be thought of as a two-stage process in which we first compute the matrix M and then compute the intrinsic and extrinsic parameters from M. Here we follow the approach proposed by Faugeras and Toscani [5, 6] to calibrate each camera in order to obtain the 10 intrinsic and extrinsic parameters.

4 Stereo Vision and Epipolar Geometry

Knowing the calibration for each camera, it is possible to compute the spatial relationship between the two. Stereo vision refers to the ability to infer spatial geometry from at least two images taken from different viewpoints. Stereo calibration consists in determining the transformation between two images. If the calibration is performed between the two camera coordinate systems using only the extrinsic parameters, then the transformation receives the name *essential matrix*. On the other hand, if the calibration is performed between the two image coordinate systems using the intrinsic and extrinsic parameters or no prior information on the stereo system, then the transformation receives the name *fundamental matrix*.

4.1 The Essential Matrix

The essential matrix can be considered as the calibrated form of the fundamental matrix. It encodes the epipolar geometry between two camera coordinate systems and can be represented as the product of two matrices: an antisymmetric matrix (rank 2) and an orthonormal matrix (rank 3). Hence, a point in the left camera coordinate system has a corresponding conjugate line in the right camera coordinate system of coordinates a', b', c' as follows:

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \mathbf{E} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{4}$$

4.2 The Fundamental Matrix

Notice that the above expression uses coordinates in the camera reference frame, but what we actually measure from images are pixel coordinates. Therefore, we use the intrinsic parameters in order to be able to make profitable use of the essential matrix. Let C and C' be the matrices of the intrinsic parameters of the left and right camera, respectively. Thus, it is possible to transform between the left image coordinate frame and the left camera coordinate frame as follows:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{C}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{5}$$

Similarly, for the right camera

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathbf{C'}^{-1} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}. \tag{6}$$

If $\mathbf{p}$ and $\mathbf{p}'$ are the points in the pixel coordinate system obtained by substituting Equations 5 and 6 into considering that $\mathbf{q}''^{t}\cdot\mathbf{l}' = 0$, we have

$$\mathbf{p}'^{t}(\mathbf{C}'^{-1})^{t}\mathbf{E}\mathbf{C}^{-1}\mathbf{p} = 0.$$

From the last equation we obtain the fundamental matrix, which is defined as follows:

$$\mathbf{F} = (\mathbf{C}'^{-1})^{t}\mathbf{E}\mathbf{C}^{-1}. \tag{7}$$

It describes the epipolar geometry. The equation

$$\mathbf{p}'^{t}\mathbf{F}\mathbf{p} = 0 \tag{8}$$

represents the epipolar line in the image coordinate system. Thus, as well as the essential matrix, the fundamental matrix can be computed from two previously calibrated cameras.

4.3 Correlation as an Objective Function

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 process of correlation can be thought as a search process in which the correlation gives the measure used to identify the corresponding pixels on both images. This process 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, $2W + 1$ the width (in pixels) of the correlation window; $\overline{I_l(x,y)}$ and $\overline{I_r(x,y)}$ be the mean values of the images in the windows centered on p_l and p_r; $R(p_l)$ be the search region in the right image associated with p_l; and $\phi(I_l, I_r)$ be a function of both image windows. The function ϕ is taken as the zero-mean normalized cross-correlation (ZNCC). It is used to match points in two images, as it is invariant to local linear radiometric changes. We have

$$\begin{aligned}
\phi(I_l, I_r) &= \frac{\sum_{i,j\in[-w,w]} AB}{\sqrt{\sum_{i,j\in[-w,w]} A^2 \sum_{i,j\in[-w,w]} B^2}}, \\
A &= I_l(x+i, y+j) - \overline{I_l(x,y)}, \\
B &= I_r(x'+i, y'+j) - \overline{I_r(x',y')}.
\end{aligned} \tag{9}$$

4.4 Constraints on Dense Stereo Matching

Stereo matching is a very difficult search process. In order to minimize false matches, some matching constraints must be set. The constraints we have considered in our algorithm are the following:

- *Epipolar geometry.* Given a feature point p_l in the left image, the corresponding feature point p_r must lie on the corresponding epipolar line. This constraint reduces the search space from two dimensions to one dimension. Unlike all other constraints, the epipolar constraint will never fail and can be applied reliably once the epipolar geometry is known (stereo-calibrated system).

- *Ordering*. If $p_l \leftrightarrow p_r$ and $p'_l \leftrightarrow p'_r$ and if p_l is to the left of p_r, then p'_l should also lie to the left of p'_r, and reversely. That is, the ordering of features is preserved.
- *Orientation*. We use as constraints the preservation not only of the gray value of the pixels but also of the orientation of the epipolar line on which the pixel lies.

5 Artificial Epidemics for Dense Stereo Matching

The motivation to use what we call the infection algorithm has two main origins. First, when we observe a scene, we do not observe everything in front of us. Instead, we focus our attention on some parts that keep our interest in the scene. Hence, we believe that many of the attention processes are handled by guessing, while we are developing our activities (which also may be the source of optical illusions). Second, the process of guessing information from characteristic points can be implemented via a matching propagation. This point also has some intuitive connections with disease spread in a population.

5.1 Artificial Epidemics

Invasion of a population by a disease is a natural phenomenon. This phenomenon can be found in all species populations, and has been a concern of human beings for centuries. The problem has been deeply studied during the last fifty years, with the aim of obtaining ideas about how a given disease will spread in a population, and also of deciding how to vaccinate the people in order to stop the propagation of the disease.

There have been several approaches to the study of disease propagation. Researchers agree that the first attempt to formulate a mathematical model is due to Lowell Reed and Wade Hapton in the 1920s [1], and their model was called the SIR (susceptible-infective-removed) model. In this model a population is divided into three classes according to their status in relation to the disease of interest: *susceptible*, meaning they are free of the disease but can catch it, *infective*, meaning they have the disease and can pass it to others, and *removed*, meaning they have been removed; for example, dead hosts are typically considered removed. Some probabilities per unit time are employed to calculate transitions from a given state to another one. Contact patterns between individuals in the population are also required for modeling the phenomenon.

The model has since been revised, and several researchers have studied exact solutions using what is called *epidemic models* [11]. Given the features of epidemic models, some authors have employed them to study different behaviors in computer networks and parallel machines. For instance, in [16] an epidemic-based protocol is applied for implementing computer models on parallel machines. Ganesh et al. [8] present an epidemic-based protocol for decentralized collection of information in a distributed network. Very recently, a new kind of models aimed at describing interaction through social, technological, and biological networks has been presented, and also employed for the study of the spread of disease. These models have been called *small-world networks*, and are intended to study how information traverses a network with certain properties [27]. The concept of small-world network has recently been applied to study the effects of epidemic algorithms [17]. But epidemic algorithms can also be seen from another point of view. Given that the model employs transition rules from the different states that each individual from the population may occupy during the epidemic process, other authors have studied and modeled disease spread in a population by means of cellular automata [15], which are another very well-known class of algorithms inspired by nature.

Cellular automata (CAs) are fundamental computational models of spatial phenomena, in which space is represented by a discrete lattice of cells. Each cell concurrently interacts with its neighborhood, which, in a traditional CA, is limited to the cell's nearest neighbors. CAs are considered as one of the best representatives of parallel algorithms inspired by nature [23].

The new algorithm we present in this article is called the *infection algorithm*. It possesses some properties borrowed from epidemic models: information is transmitted between individuals of the

population, based on some transition rules. Nevertheless, as we will see later, the number of different states that each individual of the population may occupy is larger than in an epidemic model. On the other hand, the pattern of contacts among individuals is restricted to the close neighbors, instead of establishing complex networks of contacts. The algorithm is also inspired by CAs, in that transition rules are employed to determine the next state of each individual of the population. But these transition rules use not only the states of the individual and its neighbors, but also some external information, such as correlation and constraints, which are computed from the problem we are addressing. With this in mind, the infection algorithm is described in Section 6, after the introduction of some fundamental concepts useful in the comprehension of epidemic CAs.

5.2 Artificial Life and Cellular Automata

CAs are a class of mathematical systems characterized by discreteness (in space, time, and state values), determinism, and local interaction. A CA consists of a finite-dimensional lattice of sites whose values are restricted to a finite (typically small) set of integers $Z_k = \{0,1,\ldots,k-1\}$. The value of each site at any time step is then determined as a function of the values of the neighboring sites at the previous time step. The general form of a one-dimensional CA, for example, is given by

$$\begin{aligned} x_i^{t+1} &= f(x_{i-r}^t,\ldots,x_i^t,\ldots,x_{i+r}^t), \\ f &: Z_k^{2r+1} \rightarrow Z_k, \end{aligned} \tag{10}$$

where x_i^t denotes the value of site i at time t, f represents the rule defining the automaton, and r is a non-negative integer specifying the radius of the rule. The simplest CAs are those with $r = 1$ and $k = 2$, designated by Wolfram [28] as "elementary."

In fact, the evolution of a CA is governed typically 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 rule.

CAs were introduced by von Neumann [26] in an attempt to model self-reproducing systems such as biological systems; they have been used as models of complex systems from biology to highly parallel computers. The canonical attribution is to Ulam and von Neumann [25, 26].[1] The dynamics of CAs are based on the following observations about physical systems:

- Information travels a finite distance in finite time.
- The laws of physics are independent of the position of the observer.

To this list, von Neumann added the simplifying assumptions of discrete time, space, a local discrete state space, and a regular grid topology. He used two dimensions, but the definition can be extended to any number of dimensions. Here we will use the two-dimensional case.

A universal computation property can be obtained in a cellular array called *life*, which was introduced in 1970 by J. Conway [4]. Life has been designed to simulate a population of interacting living organisms or cells. The population is supposed to "live" on a two-dimensional grid (potentially infinite). The cells change state, in parallel, by applying, each for itself, a transition function that depends only on the eight nearest neighbors of the cell on the grid. The global behavior that arises from such dynamics includes fixed points, limit cycles, and traveling configurations. The interesting feature of life is that it succeeds, despite its very simple *local* transition rules, in achieving a *global* behavior, bringing a nicely balanced compromise between the complexity of the dynamical evolutions and their stability.

1 At about the same time, but quite independently, Zuse [30] proposed structures, intended as digital models of mechanics, that are essentially CA.

5.2.1 Definitions of Cellular Automata

Given a finite set S and a dimension d, one can consider a d-dimensional lattice in which every point has a label from the set S. Formally, the lattice is the set $L = Z^d$ together with the *shift space*, the topological product S^L (i.e., the set of functions $L \rightarrow S$), where S is given the discrete topology, and the product is given the product topology. This space is compact by Tychonoff's theorem [10]. What is more, each of the lattice directions determines a natural shift map σ_i, which is a homeomorphism of the space.

Given the d-dimensional shift space, one can form a purely dynamical definition of cellular automata.

DEFINITION 1 (Cellular automata): *A cellular automaton is a continuous map $G : S^L \rightarrow S^L$ that commutes with $\sigma_i (1 \leq i \leq d)$.*

This definition, however, is not useful for computations. Therefore we consider an alternate characterization. Given a finite set S 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 CA 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 translates of v by elements in N.

DEFINITION 2 (Cellular automata): *A cellular automaton is determined by a quadruple $A = (S, d, N, f)$ where S is a finite set, d a positive integer, $N \subset Z^d$ a finite set, and $f : S^N \rightarrow S$ an arbitrary (local) function. The global function $G_f : S^L \rightarrow S^L$ is defined by $G_f(c)_v = f(c_v + N)$.*

A result of Hedlund [9] shows that all CAs arise in this fashion, so that the two definitions are equivalent. The proof hinges on the fact that every continuous function on a compact space is uniformly continuous. Notice that, as in classical examples, the local transition functions we consider are actually local, that is, each f_i depends only on some variables x_j. Here we will use a special neighborhood rather than familiar ones like the well-known von Neumann and Moore neighborhoods. Hence, according to the above definitions, we introduce now our notation in order to be able to introduce the infection algorithm.

5.3 Epidemic Automata Networks

Let I_l and I_r be considered each one as a lattice; thus, our CA needs to be able to select, to evaluate, and to propose automatically a pixel match between the two lattices. In order to achieve it, we will add two more lattices. The third is a lattice where the automaton registers the information that is being computed with the algorithm that we will detail in Section 6. This is actually the lattice of our epidemic CA. The fourth lattice is a virtual image, which represents a reprojected image from a novel viewpoint. Thus, three of the lattices represent images of size $N \times M$, where N is in the interval [1, 484] and M is in the interval [1, 768], where each cell has a pixel value $P = 0, 1, \ldots, 255$. The lattice of our epidemic cellular automaton is exactly of the same size, but each cell contains four different states and a wildcard that we will introduce below. Now, we introduce our automaton:

DEFINITION 3 (Epidemic cellular automata): *Our epidemic cellular automaton is formally introduced as a quadruple $E = (S, d, N, f)$ where S ($|S| = 5$) is a finite set composed of four states and the wildcard *, $d = 2$ a*

positive integer, $N \subset Z^d$ a finite set, and $f_i : S^N \rightarrow S$ an arbitrary set of (local) functions, where $i = 1, \ldots, 14$. The global function $G_f : S^L \rightarrow S^L$ is defined by $G_f(c)_v = f(c_v + N)$. Also, it is useful to mention that S is defined by the following sets:

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

Our epidemic cellular automaton has four states (see Figure 2). These states are described as follows:

- *Not exposed, or susceptible*, representing the cells that have not been infected by the virus (the pixels that remain in the initial state).
- *Exposed*, representing the cells that have been infected by the virus (the pixels that have been computed in order to find their matches).
- *Proposed* or *infected*, representing the cells that have acquired the virus with a probability of recovering from the disease (the pixels that have been guessed by the algorithm in order to decide later on a better match based on local information).
- *Automatically allocated* or *recovered*, representing the cells that cannot become infected by the virus. This state represents the cells that are immune to the disease (the pixels that have been confirmed by the algorithm in order to automatically allocate a pixel match).

In this way, pixels are labeled following a *susceptible-exposed-infected-recovered* (SEIR) model that leads to fast labeling. SEIR epidemics refer to diseases with incubation periods and latent infection.

In general, cellular automata uses a set of transformations called *rules*, which take a pattern of discrete values over a spatial lattice to another (future) pattern, with a local map that depends only on the values of sites in a local region of the lattice. The epidemic cellular automaton uses a special neighborhood in order to exploit the epipolar constraint (see Figure 3). This neighborhood can be divided in two parts: the interior neighborhood composed of the eight nearest cells, and the exterior neighborhood compose of 16 cells specially chosen so as to register the eight main directions that an epipolar line could follow. Hence, the neighborhood is composed of 25 cells including the central cell.

In order to describe the epidemic graphs that we have designed in our work, we formally introduce the definition of graph:

DEFINITION 4 (Graph): *A graph, which is denoted by $G = (X, \Gamma)$, is a pair consisting of a set X and a function Γ. It describes the relationships among a set of objects of any nature whatsoever, which are called its points (or its elements).*

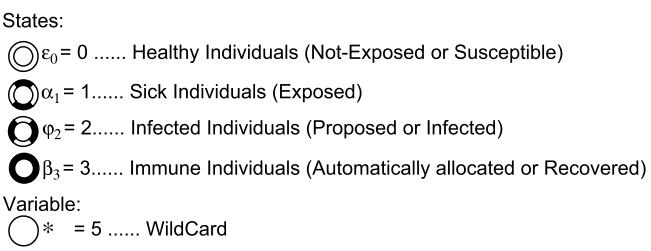


Figure 2. The epidemic cellular automaton uses four states and a wildcard.

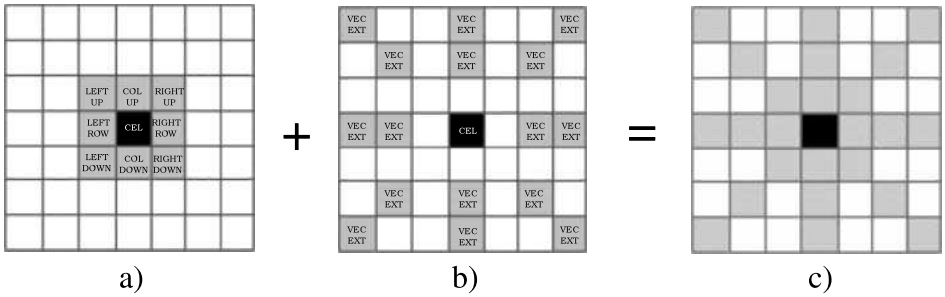


Figure 3. Neighborhood designed to use the epipolar constraint.

The relationships describing the behavior of epidemic cellular automata can be represented as a graph. Thus, we say that we have a graph whenever we have:

1. a set X,
2. a function Γ mapping X into X.

Whenever possible, the elements of a set X will be represented by points in the plane, and if X and y are two points such that $y \in \Gamma x$, they will be joined by a continuous line with an arrowhead pointing from x to y. Hence, an element of X is called a *point* or *vertex* of the graph, while the pair (x, y), with $y \in \Gamma x$, is called an *arc* of the graph. In the following, the set of the arcs of a graph will be designated by U, the arcs themselves being labeled u, v, or w (with an index if necessary). As can be seen, the set of arcs completely determines the function defining the graph, just as this function completely determines the set U; because of this, we may equally well use the forms (X, Γ) or (X, U) to describe the graph G.

A *subgraph* of a graph (X, Γ) is defined to be a graph of the form (A, Γ_A) where $A \subset X$ and in which the function Γ_A is defined by

$$\Gamma_A x = \Gamma_x \cap A.$$

It is worthy of mention that we have identified some subgraphs such that, if they are changed appropriately, the global behavior of the algorithm can be improved. The graph of Figure 4 provides

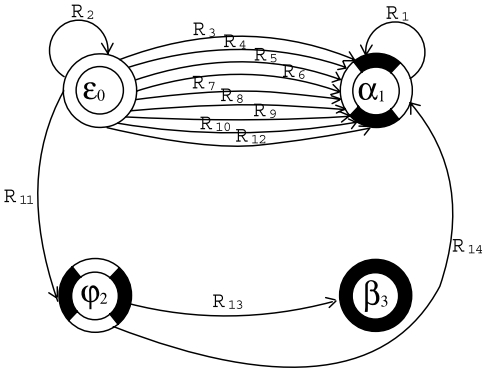


Figure 4. Epidemic graph used to obtain 47% computational savings.

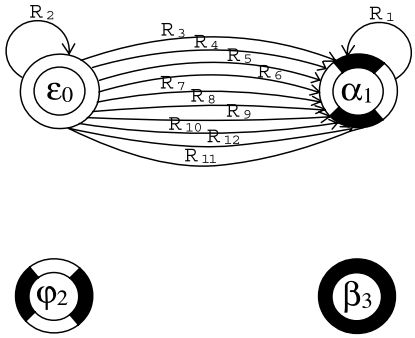


Figure 5. Epidemic graph used to obtain 0% computational savings.

47% computational savings; the graphs of Figures 5, 6, and 7 save around 0%, 78%, and 99%, respectively. The final results are given in Section 7.

6 The Infection Algorithm

The infection algorithm for searching the correspondences between real stereo images is based on the concept of a natural virus. The purpose is to find all existing corresponding points in stereo images while minimizing calculations and maintaining the quality of the reconstructed data.

The search process is based on transition rules, similarly to CAs. Our automaton can be seen as a distributed sequence of discrete time steps governed by a fixed set of rules. The rule entries depend on the state of the pixel's neighborhood. The neighborhood structure considered here is a 25-neighbor lattice including the central cell: it is positioned inside a 7×7 window centered on the pixel of interest, with 9 close neighbors and 16 external ones.

The initialization of the process is based on a set of seeds—or nuclei of infection—distributed over the whole image, which attack their neighboring cells. The infection nuclei are generated on the left image with the O&H and K&R corner detectors [20]. The infection evolves over the image according to a set of rules that changes the state of a pixel depending on the state of its neighborhood. Four states are defined as follows:

1. *Healthy individuals (not exposed)*. Nothing has been decided yet for the pixel.
2. *Sick individuals (exposed)*. The pixel has been computed using constraints of dense stereo matching.

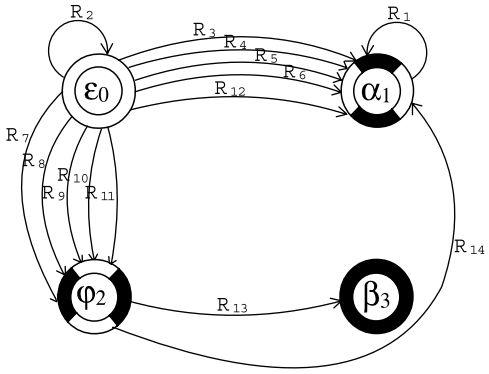


Figure 6. Epidemic graph used to obtain 78% computational savings.

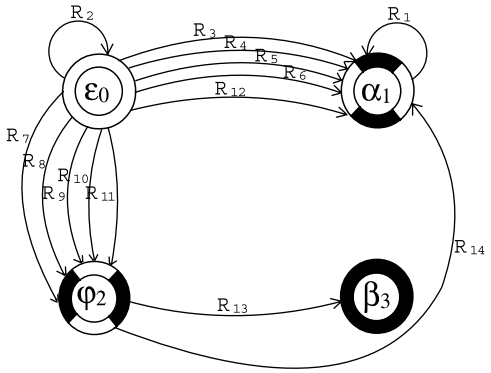
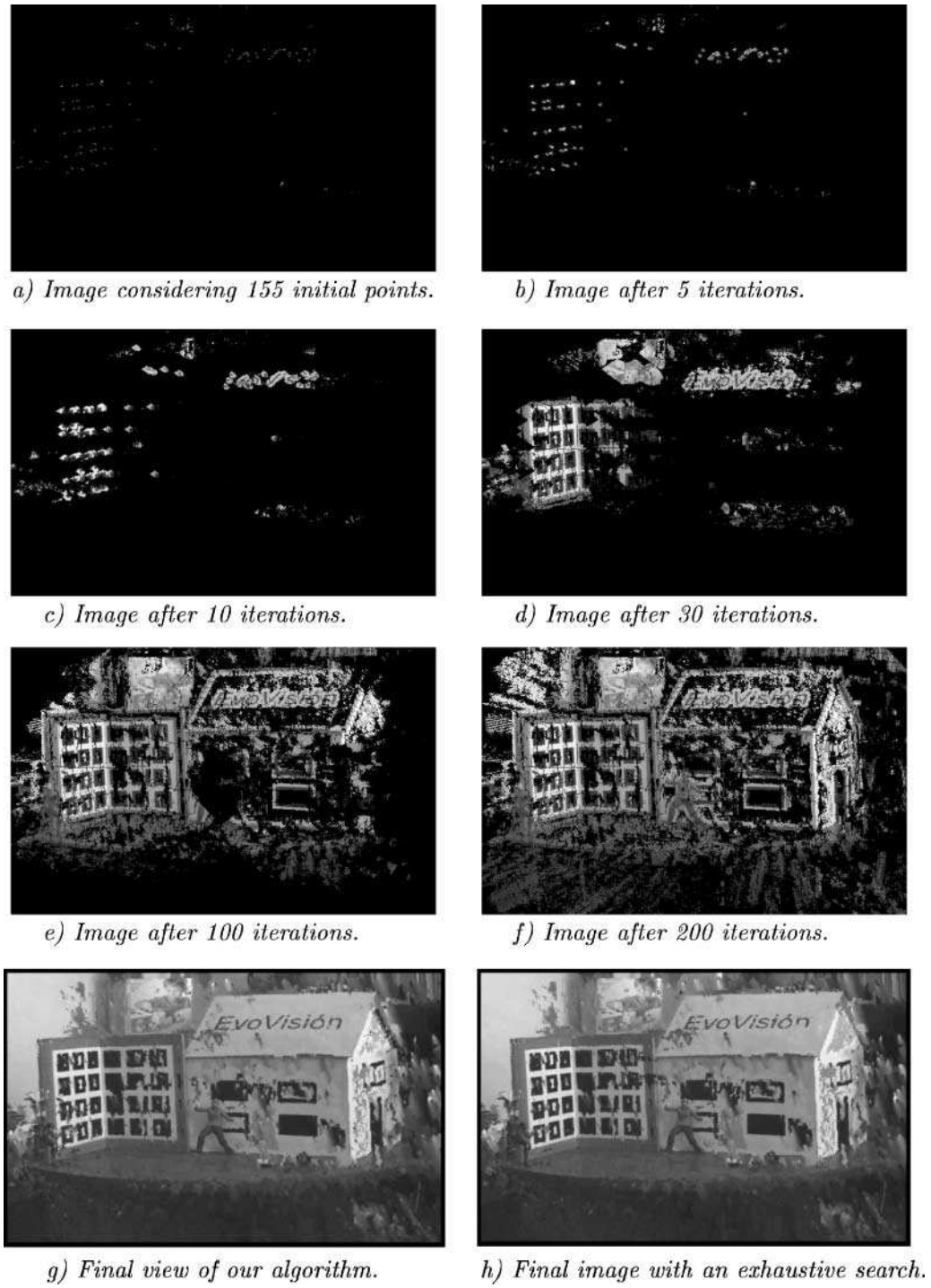


Figure 7. Epidemic graph used to obtain 99% computational savings.

3. *Infected individuals (proposed)*. The value of the pixel is guessed on the basis of its neighborhood state. Some conflicting information from various neighbors prevents fixing its status at the moment.
4. *Immune individuals (automatically allocated)*. All the information from the neighborhood is coherent, and the guessed value has been confirmed.

Table 1. Rules that save 47% of computational power.

Cell	No.	Left up	Col up	Right up	Left row	Right row	Left down	Col down	Right down	Vec exp	Vec prop	Vec ext	Vec aut	Action
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	1
0	8	5	5	5	5	5	5	5	5	3	5	5	3	1
0	9	5	0	5	1	0	1	2	5	5	5	5	5	1
0	10	5	0	5	0	1	5	2	1	5	5	5	5	1
0	11	5	5	5	5	5	5	5	5	3	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1
2	13	5	5	5	5	5	5	5	5	3	5	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1



a) Image considering 155 initial points. *b) Image after 5 iterations.*

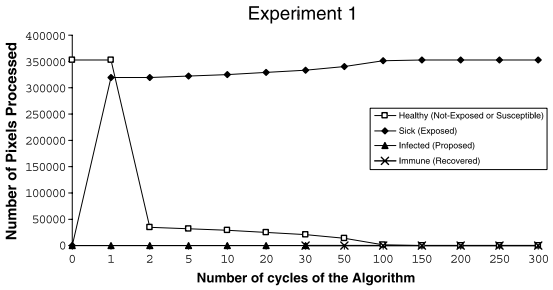
c) Image after 10 iterations. *d) Image after 30 iterations.*

e) Image after 100 iterations. *f) Image after 200 iterations.*

g) Final view of our algorithm. *h) Final image with an exhaustive search.*

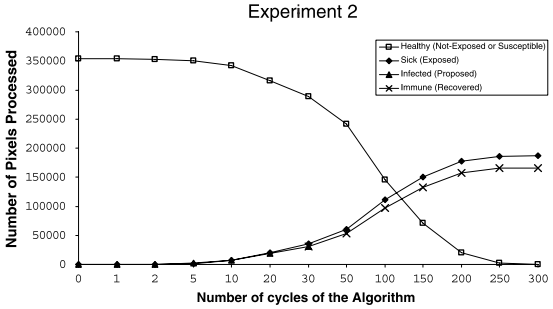
Figure 8. These eight images show the evolution of our algorithm within a synthetic view. The new image represents a new viewpoint between those of the two original images. The last two images show the final step of our algorithm. We apply a median filter to achieve the new synthetic image (g), as well as to obtain the image (h), which is the product of an exhaustive search. We compared images (g) and (h): the quality is slightly better in (g), while saving 47% of the calculations.

a) Final view with 0% savings.



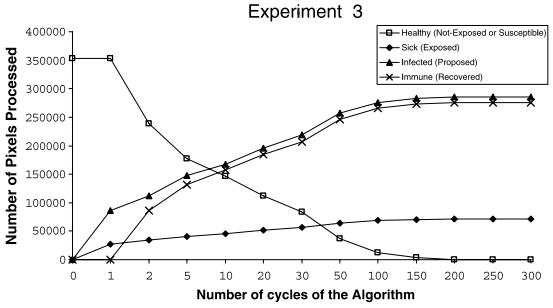
b) Performance at 0%.

c) Final view with 47% savings.



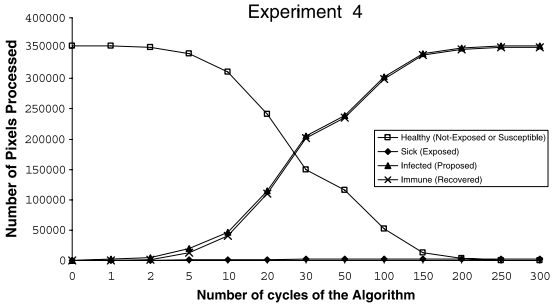
d) Performance at 47%.

e) Final view with 78% savings.



f) Performance at 78%.

g) Final view using mainly the seeds.



h) Performance at 99%.

Figure 9. These four images show the final results of our algorithm after modifying the rules in order to obtain different levels of savings in the number of computations. The quality of the virtual image is deteriorated when the level of automatically allocated pixels is increased. However, the quality of the final image is relatively good regardless the quality of the nuclei of infection.

Healthy individuals are the cells that have not been affected by the virus at time t (initial state). The sick individuals are the cells that have been affected by the virus, and represent pixels that have been calculated (exposed) in order to find the correspondence on the right image. The proposed, or infected, individuals represent cells or pixels such that we don't know yet if we need to calculate them or automatically allocate them a correspondence or matching. Finally, the immune individuals are in the most important state, because they represent the automatic allocation of the corresponding pixels in the right image without the need of calculating the matching. The assignment is made only through the consideration of the set of local rules, which produces then a global behavior.

The algorithm works in the following way, using the above information. First, we need to calculate the projection matrix and the fundamental matrix. We consider a calibrated system using a calibration grid, from which we calculate a set of correspondences with a high-accuracy corner detector [19]. The obtained information is used to calibrate rigorously the stereo system. It is important to mention that because the correspondence problem is focused on a calibrated system using real stereo images, we work now, for simplicity, with a scene that is static. The camera is moving, but not the scene. It is also important to mention that within the cells we create data structures composed of the following information: cell state, corresponding coordinates, and angle of the epipolar line. The whole virtual image is initialized to the healthy state (not exposed). The lattice is examined cell by cell in a number of cycles.

The aim is to find the maximum number of existing correspondences according to the employed rules. These rules are defined and coded in a file according to the nature of the problem. They

Table 2. Data generated by the infection algorithm using the rules that save 47%.

Cycle	Susceptible	Exposed	Infected	Recovered
0	353526	0	0	0
1	353471	452	161	0
2	352858	578	549	161
5	350545	1767	2255	1690
10	341881	6665	7250	5977
20	316448	20310	19465	18056
30	288861	34951	32375	31155
50	241116	60198	54279	53281
100	145904	110845	98433	97624
150	70456	150649	133660	133052
200	19533	177336	157304	156944
250	2112	186276	165156	165118
300	97	187354	166030	166021

Total time: 00:10:17, versus 00:17:18 for the exhaustive search.
Percentage of computational savings: (166021 × 100) ÷ 353626 = 46.9%.
Percentage of time savings: 41%.

determine the desired global behavior, even though their application only involves local information (state of the neighborhood). At each step, the algorithm scans the set of rules in order to make a decision based on local information.

This structure is flexible: the rules and neighborhood system directly tune the global behavior of the system. For example, it allows one to obtain various infection percentages (which are directly related to the saving of computational cost) and various quality levels of stereo matching results [22].

Each central cell is analyzed in order to precisely define its state, as well as the state of its neighborhood. The algorithm searches the set of rules in order to apply the rules that match the current local state (state of central cell, number of external neighbors in each state, and precise state of the close neighbors). An action is then activated that produces a path and sequence around the initial nucleus. 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. The correlation is the main measurement that is used to identify which pixel on the right image corresponds to the studied pixel of the left image (of course, improvements may be made by passing in both directions).

If the central cell is automatically allocated, then the angle of the epipolar line and the orientation constraints of the exposed neighbors are checked for consistency with the corresponding values of the central cell. The algorithm then saves the coordinates of the corresponding pixel, otherwise the algorithm calculates the pixel. On the other hand, if the central cell is in the state "proposed", the central cell is only marked as "infected." Finally, when the search is finished, the algorithm saves a

Table 3. Rules that save 0% of computational power.

Cell	No.	Left up	Col up	Right up	Left row	Right row	Left down	Col down	Right down	Vec exp	Vec prop	Vec ext	Vec aut	Action
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	1
0	8	5	5	5	5	5	5	5	5	3	5	5	3	1
0	9	5	0	5	1	0	1	2	5	5	5	5	5	1
0	10	5	0	5	0	1	5	2	1	5	5	5	5	1
0	11	5	5	5	5	5	5	5	5	1	5	5	5	1
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1
2	13	5	5	5	5	5	5	5	5	1	5	5	5	1
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

synthetic image, which represents the projection of the scene from a point of view between those for the two original images. This algorithm approaches then the problem of dense stereo matching. It is summarized as follows:

1. All pixels in the images are initiated to the state "not exposed."
2. Then, pixels of maximum interest are extracted. They are in the state "exposed" (nuclei of infection).
3. Transition rules are applied to every pixel in the images whose state is not "automatically allocated" or "exposed."
4. While there still exists pixels that are not automatically allocated or exposed, go to step 3.

7 Experiments

We have tested the infection algorithm on a stereo pair of real images; see Figure 1. The camera is a PULNiX TM-9701d with a C-mount lens (Fujinon, HF16A-2M1) of focal length $f = 16$ mm. The pixel size of the camera is 11.6×13.6 μm. The size of the image is 768×484 pixels. The infection algorithm was implemented under the Linux operating system on an Intel Pentium 4 at 2.0 GHz with 256 Mbit of RAM.

Table 4. Data generated by the infection algorithm using the rules that save 0%.

Cycle	Susceptible	Exposed	Infected	Recovered
0	0	0	0	0
1	353471	319278	0	0
2	34193	320004	0	0
5	32216	321777	0	0
10	29646	324310	0	0
20	24976	328940	0	0
30	20706	333170	0	0
50	13366	340430	0	0
100	2016	351580	0	0
150	0	353471	0	0
200	0	353471	0	0
250	0	353471	0	0
300	0	353471	0	0

Total time: 00:15:55, versus 00:17:18 for the exhaustive search.
Percentage of computational savings: 0%.
Percentage of time savings: 8%.

Our approach was to design first a simulation based on a set of 14 rules with the aim of obtaining around 50% computational savings. The idea was to obtain several global behaviors through a careful selection of these 14 rules. Table 1 presents a set of 14 rules that could achieve 47% computational savings. Experiments are presented in Figures 8 and 9. Figure 8 shows a few images that illustrate the evolution of the infection algorithm using a virtual view. The stereo pair used presents all the problems that prevent perfect correspondence. In general, the regions of the images with high texture are the regions where matching is easier to obtain. Here the final result is compared against an algorithm that uses exhaustive search. We see that the infection algorithm achieves a slightly better result while saving 47% of computational power. Table 2 illustrates that we could also achieve over 40% of time savings.

The epidemic CA could be modified to obtain 0% of computational savings; see Figure 5. Table 3 shows the set of 14 rules that we have used to achieve that result. Figure 9 shows a set of experiments where the local rules were changed in order to modify the behavior of the algorithm. It shows a set of images and graphs, which illustrates different levels of computational savings. In the case of 0% savings we still obtain a slight improvement in total time due to the difference in programming; see Table 4. The latter case represents a high percentage of automatically allocated pixels producing a good-quality image.

Table 5 provides a set of 14 rules that allows 99% computational savings. It confirms that the quality around the center of the virtual image is good enough to allow further visual processing: recognition, segmentation, and classification. Table 6 shows that we could achieve about 90% time savings.

Table 5. Rules that save 99% of computational power.

Cell	No.	Left up	Col up	Right up	Left row	Right row	Left down	Col down	Right down	Vec exp	Vec prop	Vec ext	Vec aut	Action
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	2
0	8	5	5	5	5	5	5	5	5	3	5	5	3	2
0	9	5	0	5	1	0	1	2	5	5	5	5	5	2
0	10	5	0	5	0	1	5	2	1	5	5	5	5	2
0	11	5	5	5	5	5	5	5	5	1	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	2
2	13	5	5	5	5	5	5	5	5	3	1	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

Table 6. Data generated by the infection algorithm using the rules that save 99%.

Cycle	Susceptible	Exposed	Infected	Recovered
0	353526	0	0	0
1	353471	2	1854	0
2	351615	447	4916	1402
5	340457	807	19672	12176
10	310964	1252	46619	41151
20	241621	1612	115120	110058
30	150014	1770	205343	201484
50	115794	1870	239190	235573
100	52416	2002	302086	298799
150	13033	2057	340955	338126
200	3223	2081	350324	347912
250	148	2081	353349	350987
300	0	2081	353461	351132

Total time: 00:02:16, versus 00:17:18 for the exhaustive search.
Percentage of computational savings: (351132 $\times$ 100) $\div$ 353626 = 99.2%.
Percentage of time savings: 87%.

In order to illustrate further the competence of the infection algorithm, a stereo pair of real images were acquired using a camera mounted on a robot manipulator. The robot, with six degrees of freedom, is an RX60B from Staubli, which achieves ± 0.02-mm repeatability. Three photographs were taken, called the left, right, and central images. The displacement among the photographs is described by a translation. The central view was taken approximately half way between of the left and right images. Three synthetic views were created using the infection algorithm, with savings of 0%, 47%, and 99% of calculations. The agreement between the synthetic views and the central image is evident. The scene contains a background with high texture, the calibration grid, and a person. Figure 10g shows the difference between the central image and the synthetic view with 0% savings. Figure 10h shows the robot manipulator and the camera in a hand-eye configuration.

In view of the continuing progress in computer technology and the usefulness of algorithms such as the one presented in this work, we could anticipate that our artificial life approach will be applied in real time.

8 Conclusion

We have derived a new bio-inspired algorithm and applied it, as a first trial, to dense stereo matching. It has been proved efficient for the aim of computational saving without losing the quality of the virtual image. In the future we intend to extend this model in order to identify occluded regions and

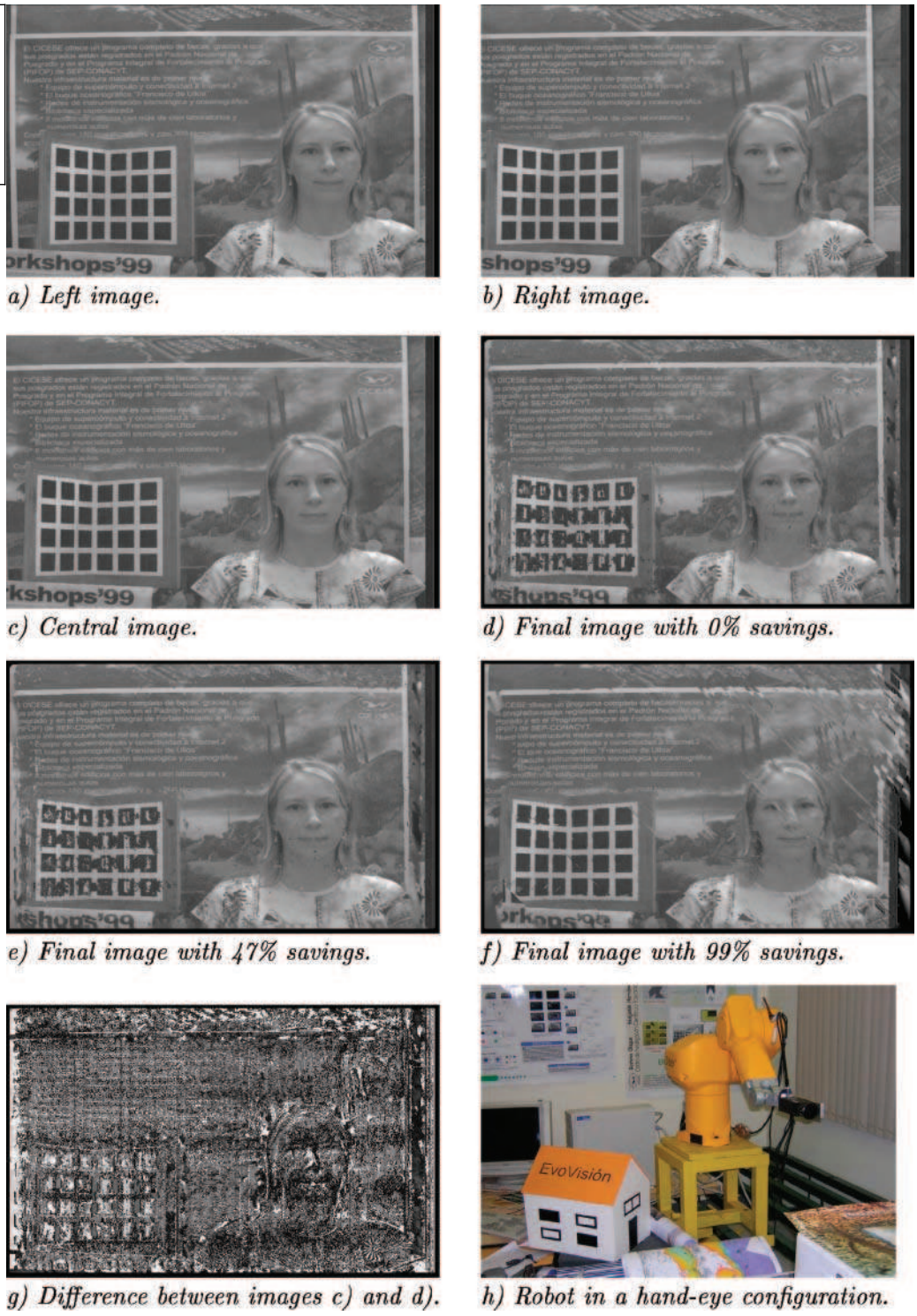


Figure 10. These images show the performance of the infection algorithm using a camera mounted on a robotic manipulator. The scene is composed of a background with high texture, the calibration grid, and a real person. Three images (left, right, and central) were acquired in order to test the agreement between the synthetic views and the central image.

vantage points for stereo vision. It seems that the infection algorithm may also be useful in other image applications, where local information needs to be propagated in a coherent way, such as monocular vision or "shape from…" problems, among others.

Acknowledgment

This research was funded by Ministerio de Ciencia y Tecnología, Spain, research project TIC2002-04498-C05-01, and by CONACyT and INRIA through the LAFMI project 634-212.

References

1. Abbey, H. (1952). An examination of the Reed-Frost theory of epidemics. *Human Biology*, *24*, 201–233.
2. Adami, C. (1998). *Introduction to artificial life*. Springer-Verlag, TELOS.
3. Brown, M. Z., Burschka, D., & Hager, G. D. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(8), 993–1008.
4. Conway, J. H., Berlekamp, E. R., & Guy, R. K. (1982). *Winning ways for your mathematical plays*. New York: Academic Press.
5. Faugeras, O. D., & Toscani, G. (1986). The calibration problem for stereo. In *Proceedings on Computer Vision and Pattern Recognition* (pp. 15–20). Miami Beach, FL: IEEE Computer Society.
6. Faugeras, O. D., & Toscani, G. (1987). Camera calibration for 3D computer vision. In *Proceedings of the International Workshop on Machine Vision and Machine Intelligence*. IEEE.
7. Fielding, G., & Kam, M. (2000). Weighted matchings for dense stereo correspondence. *Pattern Recognition*, *33*(9), 1511–1524.
8. Ganesh, A. J., Kermarrec, A. M., & Massoulié, L. (2001). Scamp: Peer-to-peer lightweight membership service for large-scale group communication. In J. Crowcroft & M. Hofman (Eds.), *Third International COST264 Workshop (NGC 2001)* (pp. 44–55). London: Springer-Verlag.
9. Hedlund, G. A. (1969). Transformations commuting with the shift. In J. Auslander & W. G. Gottschalk (Eds.), *Topological dynamics* (p. 259). New York: Benjamin 1968. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical Systems Theory*, *3*, 320.
10. Kelley, J. L. (1955). *General topology*. New York: Van Nostrand.
11. Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London, Series A*, *CXV*, 700–721.
12. Keysers, D., & Unger, W. (2003). Elastic image matching is NP-complete. *Pattern Recognition Letters*, *24*(1–3), 445–453.
13. Louchet, J. (2001). Using an individual evolution strategy for stereovision. *Genetic Programming and Evolvable Machines*, *2*(2), 101–109.
14. Luo, Q., Zhou, J., Yu, S., & Xiao, D. (2003). Stereo matching and occlusion detection with integrity and illusion sensitivity. *Pattern Recognition Letters*, *24*(9–10), 1143–1149.
15. Maniatty, W., Szymanski, B. K., & Caraco, T. (2001). *Parallel computing with generalized cellular automata*. Nova Science.
16. Maniatty, W., Szymanski, B. K., & Caraco, T. (1993). Epidemics modeling and simulation on a parallel machine. In *Proceedings of the International Conference on Applied Modeling and Simulation* (pp. 69–70). IEEE.
17. Moore, C., & Newman, M. E. J. (2000). Epidemics and percolation in small-world networks. *Physical Review E*, *61*(5), 5678–5682.
18. Olague, G. (2002). Automated photogrammetric network design using genetic algorithms. *Photogrammetric Engineering & Remote Sensing*, *68*(5), 423–431.
19. Olague, G., Hernández, G., & Dunn, E. (2003). Accurate L-corner measurement using USEF functions and evolutionary algorithms. In *Applications of evolutionary computing* (pp. 410–421). Springer-Verlag.
20. Olague, G., & Hernández, B. (2005). A new accurate and flexible model based multi-corner detector for measurement and recognition. *Pattern Recognition Letters*, *26*(1), 27–41.

21. Olague, G., Fernández, F., Pérez, C. B., & Lutton, E. (2004). The infection algorithm: An artificial epidemic approach to dense stereo matching. In X. Yao, E. Burke, J. A. Lozano, J. Smith, & J. J. Merelo-Guervós (Eds.), *Parallel problem solving from nature VIII* (pp. 622–632). Birmingham, UK: Springer-Verlag.

22. Pérez, C. B., Olague, G., Fernández, F., & Lutton, E. (2005). An evolutionary infection algorithm for dense stereo correspondence. In *7th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing* (pp. 294–303). Lausanne, Switzerland: Springer-Verlag.

23. Sipper, M. (1997). *Evolution of parallel cellular machines*. Springer-Verlag.

24. Sun, J., Zheng, N. N., & Shum, H. Y. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(7), 787–800.

25. Ulam, S. (1952). Random processes and transformations. In *Proceedings of the International Congress of Mathematician* Vol. 2, (pp. 264–275). Providences Amer. Math. Soc.

26. Von Neumann, J. (1966). Theory of self-reproducing automata. In A. W. Burks (Ed.). Champaign, IL: University of Illinois Press.

27. Watts, D. J. (1999). *Small worlds*. Princeton University Press.

28. Wolfram, S. (1986). *Theory and applications of cellular automata*. Singapore: World Scientific Press.

29. Zitnick, C. L., & Kanade, T. (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(7), 675–684.

30. Zuse, K. (1969). *Rechnender Raum*. Braunschweig: Vieweg. Translated as *Calculating space*, unpublished, Tech. Transl. AZT-70-164-GEMIT, MIT Project MAC (1970).