

An Evolutionary Infection Algorithm for Dense Stereo Correspondence

Cynthia B. Pérez¹, Gustavo Olague¹,
Francisco Fernandez², and Evelyne Lutton³

¹ 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
{cbperez, olague}@cicese.mx

² University of Extremadura, Computer Science Department,
Centro Universitario de Merida, C/Sta Teresa de Jornet, 38. 06800 Mérida, Spain
fcofdez@unex.es

³ INRIA Rocquencourt, Complex Team,
Domaine de Voluceau BP 105. 78153 Le Chesnay Cedex, France
Evelyne.Lutton@inria.fr

Abstract. This work presents an evolutionary approach to improve the infection algorithm to solve the problem of dense stereo matching. Dense stereo matching is used for 3D reconstruction in stereo vision in order to achieve fine texture detail about a scene. The algorithm presented in this paper incorporates two different epidemic automata applied to the correspondence of two images. These two epidemic automata provide two different behaviours which construct a different matching. Our aim is to provide with a new strategy inspired on evolutionary computation, which combines the behaviours of both automata into a single correspondence process. The new algorithm will decide which epidemic automata to use based on inheritance and mutation, as well as the attributes, texture and geometry, of the input images. Finally, we show experiments in a real stereo pair to show how the new algorithm works.

1 Introduction

Dense stereo matching is one of the main and most interesting problems to solve in stereo vision. It consists in determining which pair of pixels projected on at least two images, belongs to the same physical 3D point [1]. The correspondence problem has been one of the main subjects in computer vision and is clear that these matching tasks are to 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 [2]. The matching problem has been considered the hardest and most significant problem in computational stereo [3, 4]. The difficulty is related to the inherent ambiguities being produced during the image acquisition concerning the stereo pair; i.e., geometry, noise, lack of texture, occlusions, and saturation. Occlusion is the cause of complicated

problems in stereo matching, especially when there are narrow objects with large disparity and optical illusion in the scene [5, 6, 7]. Moreover, classical techniques are limited and fail in the general case, due to the complexity and nature of the problem. This is the main reason why we are considering a new approach based on artificial life and evolutionary methods that we called evolutionary infection algorithm. The infection algorithm presented in [8] 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. Finally, the algorithm provides the rendering of 3D information allowing the visualization of the same scene from novel viewpoints. In our past work, we had four different epidemic automata in order to observe and analyse the behaviour of the matching process. The better results we have obtained were those related to the case of 47% and 99% of effort savings in the correspondence process. The first case represents geometrically a good image with a moderate percentage of computational effort saving, see Fig. 7a. The second case represents a high percentage of automatically allocated pixels producing an excellent percentage of computational effort saving, with an acceptable image quality, see Fig. 7c. Our current work aims to improve the results based on a new algorithm that uses concepts from evolution; such as: inheritance and mutation. We want to combine the best of both epidemic automata in order to obtain a high computational effort saving with an excellent image quality.

This paper is organized as follows. The next section describes the nature of the correspondence problem. Section 2 introduces the new algorithm giving emphasis to the explanation on how the evolution was applied to the epidemic

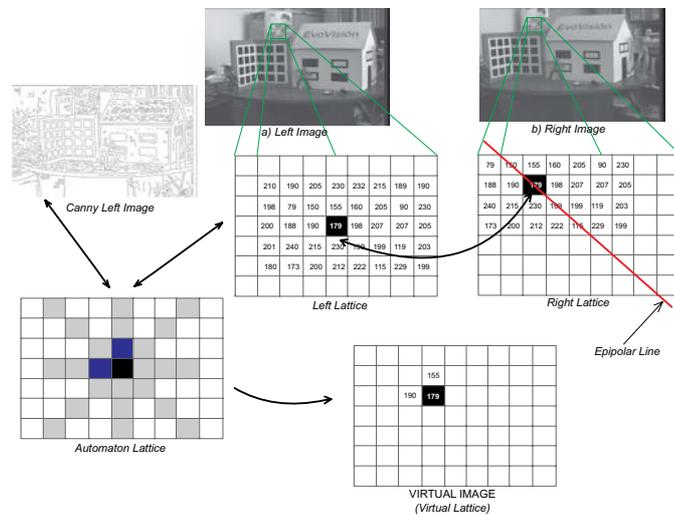


Fig. 1. Representation of the interactions among the five lattices used by the infection algorithm. A pixel in the left image is related to the right image, using several lattices such as: automaton lattice, canny image and virtual image lattice during the correspondence process

automata. Finally, the section 3 shows the results of the algorithm illustrating the behaviour, performance, and quality of the evolutionary infection algorithm.

1.1 Problem Statement

Computational stereo studies how to recover the three dimensional characteristics of a scene from multiple images taken from different viewpoints. 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. In this work the images are taken by a moving camera, with the hypothesis that a unique three-dimensional physical point is projected into a unique pair of image points, see Fig. 1. In order to solve the problem in a more efficient manner, several constraints and assumptions regarding occlusions, lack of texture, saturation, or field of view are exploited. The movement between the images is a translation with a small rotation along the x , y and z axes respectively: $T_x = 4.91 \text{ mm}$, $T_y = 114.17 \text{ mm}$, $T_z = 69.95 \text{ mm}$, $R_x = 0.84^\circ$, $R_y = 0.16^\circ$, $R_z = 0.55^\circ$. Fig. 1 shows also 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 automata in order to process the information that is being computed. The fourth lattice corresponds to the reprojected image, while the fifth lattice is used as a database in which we save information related to contours and texture.

2 The Evolutionary Infection Algorithm

The infection algorithm is based on the concept of 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 keeps our interest on the scene. As a result, it is not necessary to analyse each part of the scene in detail. The infection algorithm attempts to “guess” some parts of the scene through a process of propagation based on artificial epidemics with the purpose of saving computational time. The mathematical description of the infection algorithm is presented in [9]. This paper introduces the idea of evolution within the infection algorithm using the concepts of inheritance and mutation in order to achieve a balance between exploration and exploitation. As we could see in the paper, the idea of evolution is rather different from the traditional genetic algorithm. Concepts like an evolving population are not considered in the evolutionary infection algorithm. Instead, we incorporate aspects such as inheritance and mutation to develop a dynamical matching process. In order to introduce the new algorithm let us define some notations:

```

1. BEGIN
2. Corner detection of the left image with O&H or K&R detectors.
3. Calculate Projection Matrices MM1 and MM2 of the Left and Right image.
4. Calculate Fundamental Matrix (F) Left to Right.
5. Coding of rules matrices 47% and 99%.
6. Process the left image with the canny edge detector.
7. Instantiate epidemic automata to Healthy(Not-Explored) state.
8. Obtain the left image with canny edge detector.
9. While the number of Immune (Automatic) and Sick (Explored) states is
   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.     actionFam= critFam(stateCell,neighbors,pos,probFam)
17.     actionCont= critCont(stateCell,probContour)
18.     actionMut= critMut(stateCell, probMutation)
19.     IF( probFam > probContour && probFam>probMutation)
20.       action=actionFam
21.     IF(( probContour > probFam && probContour>probMutation)
22.       action=actionCont
23.     IF(( probMutation > probFam && probMutation>probContour)
24.       action=actionMut
25.     SWITCH(action)
26.     { 1: EXPLORED
27.       2: PROPOSED
28.       3: Analyse the inclination of the epipolar line
29.         IF the inclination = to the inclination of at least three
           neighbors
30.           { AUTOMATICALLY ALLOCATED }
31.         ELSE
32.           { EXPLORED } } } }
33. Virtual image obtained.
34. END

```

Fig. 2. Pseudo-code for the evolutionary infection algorithm

Definition 1. Epidemic cellular automata: Our epidemic cellular automata could be formally represented 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 $G_f(c_v) = f(c_v + N)$. 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,
- $S_f = \{\alpha_1, \beta_3\}$ is the set of final output states,
- $S_0 = \{\varepsilon_0\}$ is called the initial input state.

Each epidemic cellular automaton has 4 states that are defined as follows:

- *Explored*, representing the cells that have been infected by the virus (it refers to the pixels that have been computed in order to find their matches).
- *Not-explored*, representing the cells which have not been infected by the virus (it refers to the pixels which remain in the initial state).
- *Automatically allocated*, representing 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).
- *Proposed*, representing 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 the better match based on local information).

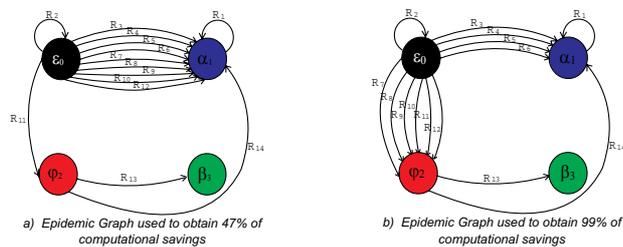


Fig. 3. Evolutionary epidemic graphs used in the infection algorithm

2.1 Infection Algorithm with an Evolutionary Approach

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% of effort saving while the other a 99% of effort saving. 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 behaviour 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 behaviour of the algorithm. In fact, the evolution of cellular automata 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 automata together with the values to which these states are to be mapped in one iteration of the algorithm.

The goal of the search process is to achieve a good balance between two rather different epidemic cellular automata in order to combine the benefit of each automaton. Hence, our algorithm not only finds a match within the stereo pair, but it provides an efficient and general process using geometric and texture information. It is efficient because the final image combine 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 image 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, we called A, provides a strategy which exploits the best solution. Here, best solution refers to areas where matching is easier to find. The 99% epidemic cellular automaton, we called B, provides a strategy which explores the search space when the matching is hard to achieve.

The pseudo-code for the evolutionary infection algorithm is depicted in figure 2. First, we calculate the projection matrices MM1 and MM2 related to the left and right image, through a process of calibration. Then, two sets of rules are defined A=46% and B=99%. 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 canny left image. Thus, we iterate the algorithm while the number of pixels with immune and sick states is different between time t and $t+1$. Each pixel is evaluated according to a decision that is made based on three criteria as follows:

1. The decision of using A or B is weighted considering the current evaluated pixels in the neighborhood. In this way inheritance is incorporated within the algorithm.
2. The decision is also made based on the current local information (texture and contour). Thus, environmental adaptation is contemplated as a driving force to the dynamical matching process.
3. A 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.

In this way, an action is activated producing 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. Thus during the search process, a sort of inheritance is used to decide which cellular automata to apply. Finally, our algorithm produces an image that is generated combining the two epidemic cellular automata.

2.2 Transitions of Our Epidemic Automata

The transitions of our epidemic automata are based on a set of rules. Each relationship between the four states is represented as a transition based on a local rule, which as a set is able to control the global behaviour of the algorithm. Each epidemic graph has 14 transition rules that we divide in 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 their 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 the close neighborhood. The complex structure rules consider not only the spatial set of relationships between the close neighborhood, but also those within the external neighborhood. The next section explains how each rule works according to the above classification. The basic rules correspond to rules 1 and 2. The initial structure is formed by the rules 3,4,5 and 6. Finally, the complex structure rules correspond to the rest of the rules. The basic and initial structure rules until this moment have not been changed. The rest of the rules are easily modified in order to produce different behaviours and a certain percentage of computational effort saving, see Fig. 3. The most important epidemic rules related to the case of 47% are explained next.

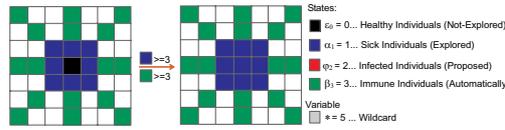


Fig. 4. Rule 7. The epidemic transition indicates that it is necessary to have at least three Sick (Explored) individuals on the closed neighborhood and three Immune (Automatically allocated) individuals in order to change the central cell

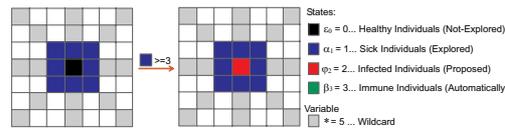


Fig. 5. Rule 11. This epidemic transition represents the quantity of Infected (Proposed) individuals in order to obtain the Immune (Automatically Allocated) individuals. In this case, if there are three Sick (Explored) individuals in the closed neighborhood, then the central cell changes to an Infected (Proposed) state

- **Rules 3,4,5 and 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 way that the information is distributed in several process. Thus, the propagation of the matching is realized in a higher number of directions from the central cell. We use these rules at the initialisation of the process only.
- **Rule 7 and 8.** These rules ensure the evaluation of the pixels in a region where exist Immune (Automatically Allocated) individuals, see Fig. 4. The figure of the rule 8 is similar to 7. The main purpose of these rules is to control the quantity of immune individuals within a set of rules.
- **Rule 11.** This rule generates the Infected (Proposed) individuals in order to obtain later a higher number of Immune (Automatically Allocated) individuals. If the central cell is on Healthy state (Not-Explored) and there are at least three Sick individuals (Explored) in the closed neighborhood, then, the central cell is Infected (Proposed).
- **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 closed neighborhood, the central cell is evaluated.
- **Rule 13.** This rule is one of the most important epidemic transition rules because it indicates the computational effort saving without any computation of individual pixels during the matching process. This is the reason why we called it immune or automatically allocated. Rule 13 can be summarized as follows: if the central cell is Infected (Proposed) and if there are at least three Sick (Explored) individuals in the closed neighborhood; then, we know automatically that the corresponding pixel in the right image is made

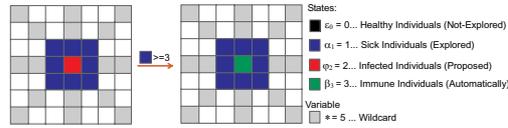


Fig. 6. Rule 13. This transition indicates when the Infected (Proposed) individuals will change to Immune (Automatically Allocated) individuals

without any computation. The number of Sick (Explored) individuals can be changed depending on the percentage of computational savings that we need.

Obviously, if we modified some of the complex structure rules from the case 47% we could obtain up to 99% of computational effort saving. The basic and initial structure rules do not change the coding, implying that the central cell remains without changes. The complex structure rules, except rule 14, are modified in two ways: the central cell state is modified according to the information within the neighborhood and the number of cells involved.

3 Experimental Results and Conclusions

We have tested the 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 2.0Ghz with 256Mb of RAM. We have used libraries programmed in C++, designed specially for computer vision, called VXL (Vision X Libraries). 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 epidemic automata 47% and 99%, in order to obtain a high computational effort saving together with an excellent image quality. We used knowledge based on geometry and texture in order to decide during the correspondence process, which epidemic automata is better to apply during the evolution of the algorithm.

Figure 7 shows a set of experiments where the epidemic cellular automata was changed in order to modify the behaviour of the algorithm and to obtain a better virtual image. Fig. 7a is the result of obtaining 47% of computational effort savings, while Fig. 7b is the result of obtaining 70% of computational effort savings and Fig. 7c shows the result to obtain 99% of computational effort savings. Fig. 7d presents the result that we obtain with the new algorithm. Clearly, the final image shows how the algorithm combines both epidemic cellular automata. However, the quality of the final image depends on the texture and contours of the stereo pair. We observe that the geometry is preserved with a nice texture reconstruction. We also observe that the new algorithm spends about the same time employed by the 70% epidemic cellular automaton with a slightly better texture result. Figure 8a shows the behaviour of the evolution-

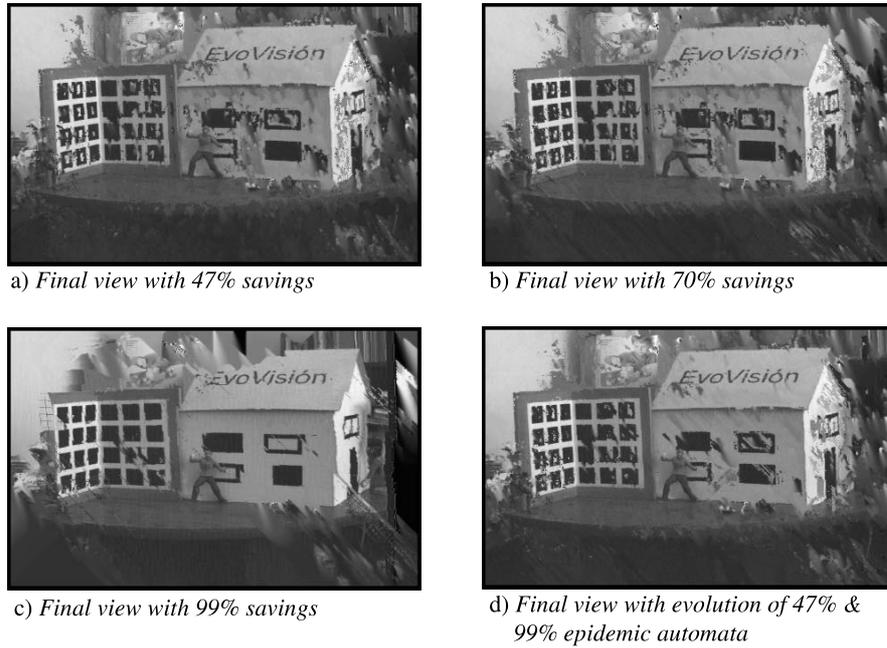


Fig. 7. Set of different experiments where epidemic cellular automata were change

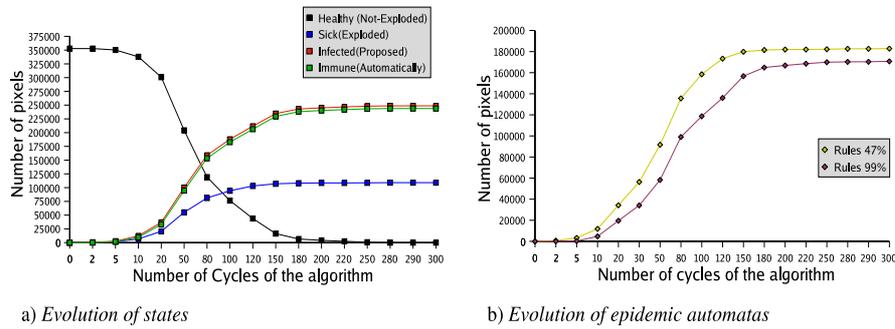


Fig. 8. Evolution of the epidemic cellular automata during the dense correspondence process

ary infection algorithm that corresponds to the final result of Fig. 7d. Figure 8b describes the behaviour of the two epidemic cellular automata during the correspondence process. In the near future, we expect to use the Evolutionary Infection Algorithm in the search of novel viewpoints.

Acknowledgments. This research was funded by Ministerio de Ciencia y Tecnología, SPAIN, research project number TIC2002-04498-C05-01, and by CONACyT and INRIA through the LAFMI project. First author supported by scholarship 171325 from CONACyT.

References

1. Lionel Moisan and Béranger Stival. A Probabilistic Criterion to Detect Rigid Point Matches Between Two Images and Estimate the Fundamental Matrix. *International Journal of Computer Vision*. Vol.57(3):201-218. 2004.
2. Myron Brown, Darius Burschka and Gregory Hager. Advances in Computational Stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol.25(8):993-1008. 2003.
3. Umesh Dhond and K. Aggarwal. Structure from Stereo. *IEEE Transactions on Systems and Man and Cybernetics*. Vol.19(6):1489-1509. 1989.
4. Barnard S, Fischler M."Computational stereo". *ACM, Computer Surveys*, Vol. 14(4):553-572. 1982.
5. Quiming Luo, Jingli Zhou, Shengsheng Yu, Degui Xiao. "Stereo Matching and occlusion detection with integrity and illusion sensitivity". *Pattern Recognition Letters*. 24(2003):1143-1149.
6. Stan Birchfield and Carlo Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo. *International Journal of Computer Vision*. Vol.35(3):269-293. 1999.
7. C.Lawrence Zitnick and Takeo Kanade. A Cooperative Algorithm for Stereo Matching and Occlusion Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol.22(7):675-684. July 2000.
8. Gustavo Olague, Francisco Fernández, Cynthia B. Pérez, and Evelyne Lutton. "The infection Algorithm: An Artificial Epidemic Approach to Dense Stereo Matching. Parallel Problem Solving from Nature VIII. X. Yao et al.(Eds.): LNCS 3242, pp.622-632, Springer-Verlag, Birmingham, UK, September 18-22, 2004.
9. Gustavo Olague, Francisco Fernández, Cynthia B. Pérez, and Evelyne Lutton. "The infection Algorithm: An Artificial Epidemic Approach for Dense Stereo Correspondence." Available as CICESE research report No.25260