SCIA '97

Proceedings of

The 10th Scandinavian Conference on Image Analysis Lappeenranta, Finland, June 9-11, 1997

Volume II

SCIA '97 The 10th Scandinavian Conference
on Image Analysis
Lappeenranta, Finland, June 9-11, 1997

Edge detection using a multiagent system

P. Ballet, V. Rodin, J Tisseau
Ecole Nationale d'Ingénieurs de Brest
Laboratoire d'Informatique Industrielle
Technopôle Brest-Iroise - Site de la Pointe du Diable
CP n° 15 - 29608 BREST Cedex, France
emails: {ballet,rodin,tisseau}@enib.fr

Abstract

In this paper, we introduce a multiagent system allowing edge detection in an image. This system is made up mainly reactive agents in interaction thanks to a shared memory. Each agent has a very simple behavior (perception-action), and has the capacity to adapt locally to what it considers its environment, that is to say the image. An agent can move and has its own position in its environment. The basic behavior for an agent consists of following the highest brightness gradient and inscribing its path, if estimating to be on an edge, in all the agents' shared memory. Its path thus corresponds to edges which are found in the image.

We think that this approach is original in its use of agents and may be used to implement parallel image processing

1 Introduction

For many years, contour detection is one of the most important problem in image processing and has originated many scientific studies. These studies are mostly based on convolution filters or non linear filters [2][6][7][8]. Another type of methods has recently been set for detecting features in images. Those methods are based on multiagent system [1][3].

In this article, we would like to detect the edges with the help of a multiagent system made up of reactive agents. The reactivity being very important, the agents' behavior is very simple (perception-action) : they have the capacity, nevertheless, to adapt locally to what they consider their environment, that is to say the image. An agent can move and has its own position in its environment. The basic behavior for an agent consists of following the highest brightness gradient and inscribing its path, if estimating to be on an edge, in all the agents' shared memory. Its path thus corresponds to edges which are found in the image. The notion of shared memory is very important because it allows the interaction among agents and the coordination of their actions [4][5][9]. The agents actually use already found edges for finding new ones or complete those already detected.

We have tested this system on images such as "indoor" and "biological" scenes (image of a muscle taken through a microscope), but as well on synthetic scenes allowing analysis of thus obtained results.

In the remaining part of the paper, we will first introduce actual organization of our multiagent system. We will then account into details for the decisional module of the agents specifying their behaviors, their actuators and their sensors.

2 Actual organization of our multiagent system

Our multiagent system for detecting edges is made up of three component parts (see figure 1) : the agents, the shared memory and the environment. An agent actually picks up a small part of the environment and also requires the shared memory. It reacts according to those informations and its present state. The exact behavior of an agent is detailed further on in the article. The shared memory contains the detected edges. It is, in fact, a table which has the same dimensions as the environment in which the agents inscribe the edges they have detected. To allow interaction among agents, the agents recover information as well, via a sensor, from that memory. The environment in which the agents move can be any grey level image

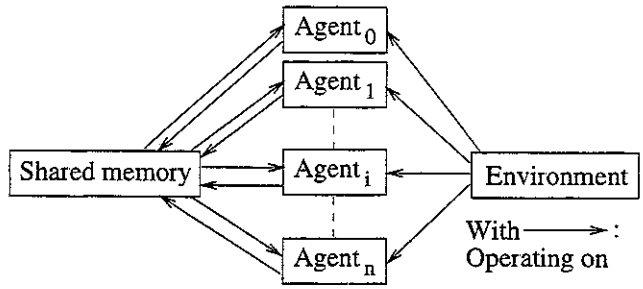


Figure 1: Actual organization of our multiagent system

A simulation involves one instance from the shared memory, one instance from the environment and several instances from agents. The instances from agents are initially placed at random in the environment (which is, in other words, the image)

3 Agent decisional module

The agent decisional module follows the perception-action model, for the sensors allow each agent to receive outer information. According to the data given by the sensors and its own inner state, the agent takes its decisions and applies them through its actuators (see figure 2). Hereafter, we will analyse the decisional module and specify the

Figure 2: Perception-action model of our agents.

agents' behavior as well as their sensors and actuators.

3.1 The agents' behavior

In our system, the agents can have several states allowing them to move on images for detecting edges according to the tactics quoted below. They are designed for detecting before anything else the straight edges, but they can also detect curves (circle, ellipse or any curve) However, the results are not as conclusive. The detection strategy is the following :

1- An agent is initially placed at random in the image and then acquires the status of "Nose". In other words it only "sniffs" around searching for an edge on which it could settle.

2- As soon as the agent considers, according to a threshold, that it is located on an edge (see figure 3), it acquires the status of "Settled".

3- It loses the status of "Settled" as soon as it considers not being on an edge anymore or that it may collide with an edge stored in the shared memory or if it does a 360-degree self rotation Then, the agent follows and records the edge It acquires the status of "Recorder".

4- As soon as the agent considers it is not located on an edge anymore or that it may collide with an edge stored in the shared memory, or if it reaches the start of its recorded edge, it acquires the status of "Marker". It writes its entire path to the shared memory, from the moment it turned back Please note that, in order to be noise resistant, the path is actually recorded only if it is long enough

5- After having written its path to the shared memory, it tries to mark out a new edge to go on its way (searcher state). If it finds one, it acquires back the status of "Recorder" If it does not, it initializes by relocating itself at random on the image and acquires back the status of "Nose"

In stage 5, for marking out a new edge, an agent follows the method quoted below :

a- It comes to a standstill (see figure 4-a).

b- It creates other sensors to be able to find other edges (see figure 4-b).

c- Should the occasion arise, it detects a new edge (see figure 4-c)

d- It settles in the prolongation of the latter so as to continue memorizing (see figure 4-d) That method

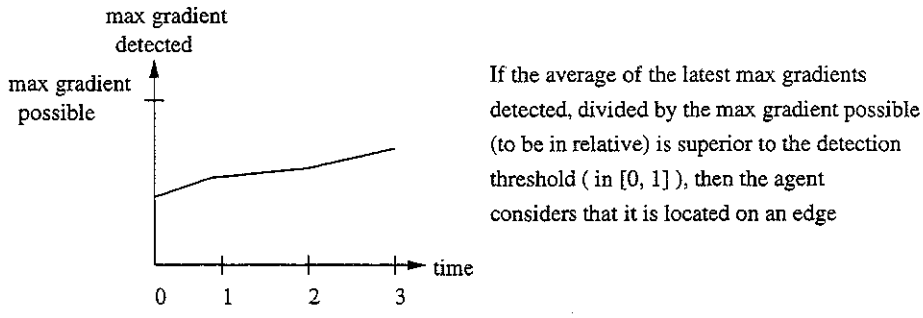


Figure 3: Agent's internal memory

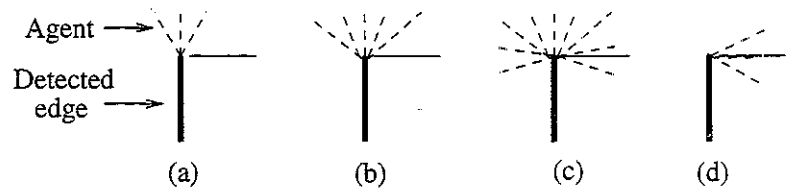


Figure 4: Agent's tactics for marking out a new edge

allows the agent to wholly memorize a polygon, and even memorize curved edges through successive markings and resettlements

The finite state machine showed below gives a more formal representation of an agent's behavior (see figure 5)

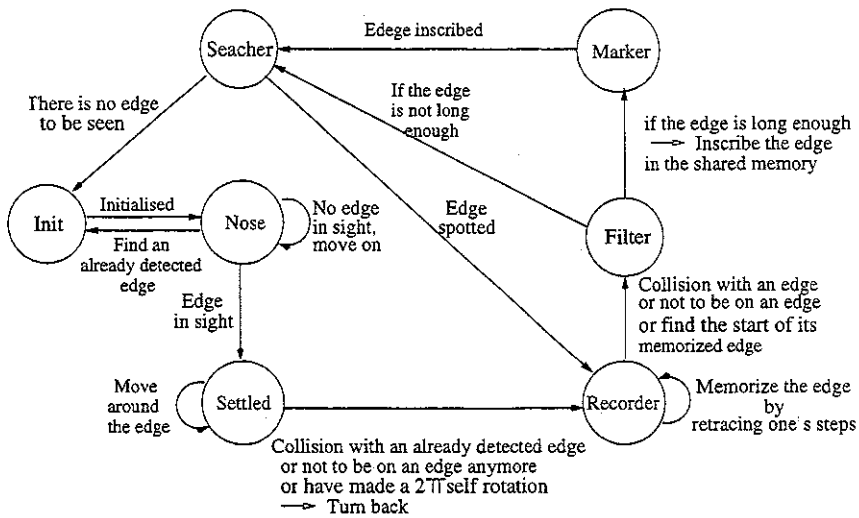


Figure 5: Finite state machine describing an agent's behaviour

3.2 Agents' sensors

The agents' sensors are divided into two groups. The sensor permitting to extract information from the shared memory is in the first group, which allows the agents to know the already detected edges The second group corresponds to the sensor on the image. Please note that a sensor may itself be made up of other sensors. The smallest sensor corresponds to the unit sensor, that is to say to the sensor that directly catches the brightness of an image pixel or of the shared memory The sensor in the shared memory is very simple insofar as it is made up of one single unit sensor. On the other hand, the sensor on image is more complex. It is made up, indeed, of three

or more bars which themselves are made up of a set of unit sensors The bars are located on the front part of the agent (motionwise) and each is inclined at an oblique angle with regard to the next Each bar adds up the values gathered by the unit sensors making it up The sensor on the image then estimates the differences between the values of two successive bars, retaining only the most important value which corresponds to the angular position of the most pronounced edge. The unit sensors are set in the form of bars so as to favor the detection of rectilinear edges Please note that the way it is set can be changed according to the application Figure 6 gives us more details about the sensors in use

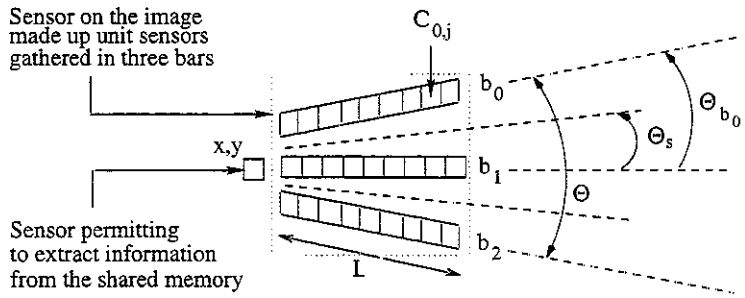


Figure 6: Agents' sensors

Where: $(x, y) \in I\!R^2$, position of the agent in the environment,
$\theta \in [0, 2\pi[$, distance between opposite bars,
$\theta_{b_i} \in [0, 2\pi[$, distance between two successive bars,
$L \in I\!N$, length of bars,
$i \in [0, 2] \in I\!N$, index of the bar,
$j \in [0, L-1] \in I\!N$, index of the unit sensors in the bar,
$C_{i,j} \in [0, C_{max}] \in I\!N$, value of unit sensor i, j

The sensor permitting to extract information from the shared memory is equal to 0 if no edge is found and is equal to 1 otherwise.

The sensor on the image is made up of unit sensors gathered in bars. Each of the unit sensors is equal to the intensity of the brightness found on the image, and each bar is equal to the sum of the intensities found in each sensor making it up :

$$b_i = \sum_{j=0}^{L-1} C_{i,j} \ , \ b_i \in N, \text{value of bar i}$$

The sensor on the image has two values. The first one corresponds to angle θ_s and the second one has the value of the maximum found gradient.

Where : $gradient_i = (b_{i+1} - b_i)$
$gradient_{max} = max(gradient_i)$
$\theta_s = (\theta_{b_{i+1}} + \theta_{b_i})/2$, for the maximum gradient.

3.3 Agents' actuators

An agent is made up of three actuators. Two of its actuators allow the agent to move around in its environment The first one allows it to turn around and the second one to move forward. The third actuator allows it to write the edges it has detected to the agents' shared memory.

We have applied that multiagent system on different types of images. The results are given hereafter.

4 Results

This last part sets out a few simulation results on different types of images ("indoor" and "biological" scenes). We will also submit results of a computer-generated image allowing to analyze the results In these simulations, the parameters have the following values :

$\theta = \pi/3$
$L = 10$
$i \in [0, 4]$
$threshold = 0\ 07$
image dimensions = 512X512 in 256 grey levels

4.1 Computer-generated image

Given the image in figure 7-a. That computer-generated image is made up of different types of straight or curved edge objects. On the whole, the main edges are indeed detected (even if they are curved). Some small objects

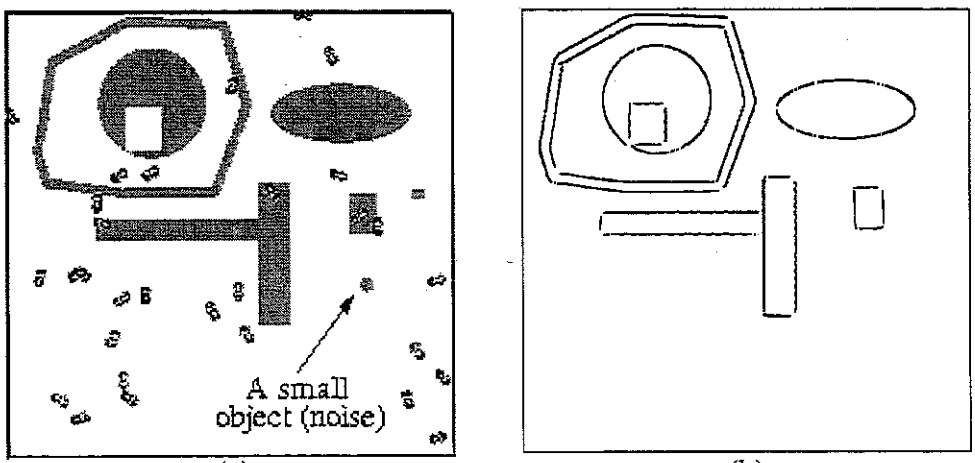


Figure 7: (a) Initial computer-generated image with a serie of agents
(b) Image of detected edges during a simulation.

(they could be noise, for instance) are not detected by our system (see figure 7-b). That due to the fact that we only take into account the edges which are long enough

4.2 Image of an office

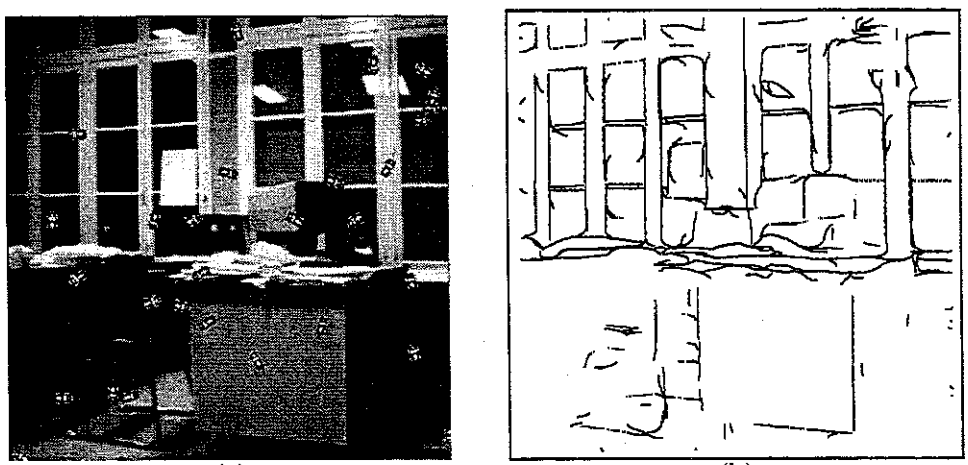
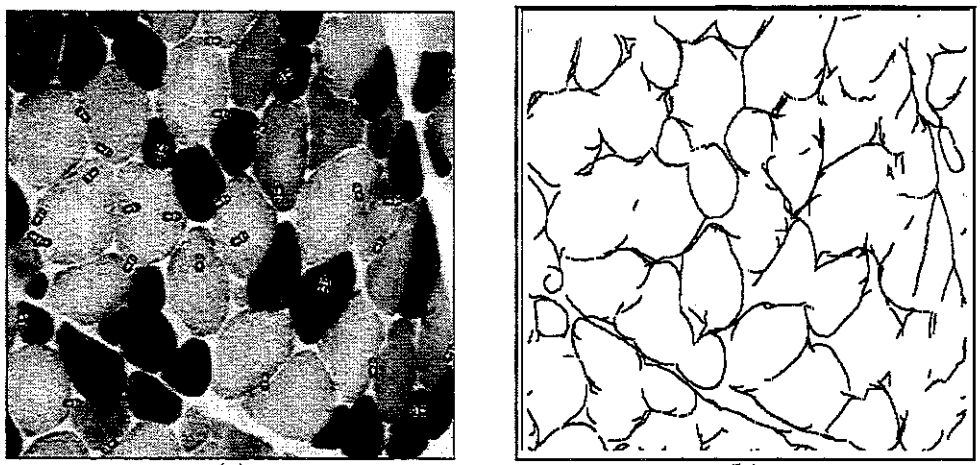


Figure 8: (a) Initial image with a serie of agents
(b) Image of detected edges during a simulation

Given the image in figure 8-a. As we can see figure 8-b, most of object boundaries have been detected on this image which is essentially made up of straight edges.

4.3 Image of a muscle taken through a microscope



(a) (b)

Figure 9: (a) Initial image with a serie of agents
(b) Image of detected edges during a simulation

Given the image in figure 9-a. This image is more complex than the previous image, for it is only made up of curved edges which are more difficult to detect with our system. Figure 9-b shows the detection results which are not as conclusive as in the case of the indoor scene.

5 Conclusion

We have introduced a multiagent system for edge detection. In our system, the agents are reactive and the interaction is done through a shared memory allowing storage of found edges. The results are promising though we have noted a few faults while detecting curved edges. Specialized agents could be designed for such edges.

Our multiagent system has been tested on a single-processor computer, and it has been note that the number of agents in a simulation neither affects the quality of the result nor CPU time necessary for segmentation of a given scene. This is true up to a maximum number of agents for, afterwards the performances decrease as the number of agents increases.

We think that this approach is original in its use of agents and may be used to implement parallel image processing by assigning, for instance, an agent to each processor.

References

[1] O. Boissier and Y. Demazeau. *"ASIC : An Architecture for Social and Individual Control and its application to computer vision"*. MAAMAW Conference, Odense, Danemark, August 2-4, 1994.

[2] J.F. Canny. *"A computational approach to edge detection"*. *IEEE Trans. on PAMI*, Vol. 8, No 6, pp 679-698, 1986.

[3] Y. Demazeau, O. Boissier and J.L. Koning. *"Using interaction protocols to control vision systems"*. IEEE International Conference on System, Man and Cybernetics, San Antonio, 1994.

[4] J.L. Deneubourg and S. Goss. *"Collective patterns and decision-making"*. Ecology, Ethology and Evolution, Vol. 1, pp. 295-311, 1989.

[5] J. Ferber. *"Multiagent systems" (in french)*. InterEdition, 1995.

[6] D. Marr and E.Hildreth. *"Theory of edge detection"*. Proc. Roy. Soc., London, B-207, pp 187-214, 1980.

[7] J.M.S Prewitt. *"Object Enhancement and Extraction"*. Picture Processing and Psychopictorics, Academic Press, London, pp 75-149, 1970.

[8] J. Shen and S. Castan. *"An optimal linear operator for step edge detection"*. CVGIP: Graphical models and image processing, Vol. 54, No 2, pp 112-133, 1992.

[9] L. Steels. *"Cooperation between distributed agents through self-organization"*. Elsevier/North-Holland, 1989.