European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2000
Barcelona, 11-14 September 2000
ECCOMAS

AUTOMATIC RULES GENERATION BY G.A. FOR EGGSHELL DEFECT CLASSIFICATION

Angela Ribeiro[*], Maria C. García-Alegre[*], Domingo Guinea[*], and Gabriel Cristobal[†]

[*]Instituto de Automática Industrial
Spanish Council for Scientific Reserch
Arganda del Rey, 28500 Madrid, Spain
e-mail: (angela, maria, domingo)@iai.csic.es, web page: http://www.iai.csic.es/users/gpa

[†]Instituto de Optica
Spanish Council for Scientific Reserch
28006 Madrid, Spain.
gabriel@optica.csic.es

Key words: Genetic Algorithms, Example-based Learning, Visual Inspection, Colour Processing, Industrial Application, Poultry Industry.

Abstract. *Present work is focused on the design and implementation of a classification algorithm for an artificial vision system devoted to the automatic rejection of defective eggs in the farms. The artificial vision system performance is measured in terms of two parameters: short computing time and accuracy in the grading process. An image splitting process is proposed to iteratively calculate the number of defective pixels from the whole image to ever-shorter image regions using evolutionary algorithms. The antecedents of the classification rules of the proposed algorithm are obtained by means of an examples-based learning method that derives the best antecedents. Genetic Algorithms are used as a learning process that discovers the best antecedent departing from two sets of binary images: "dirt_eggshells" images or positive examples and "clean_eggshells" images or negative examples. The classification results are displayed and the performance is compared with those obtained in former studies.*

1 INTRODUCTION

The early rejection of defective eggs is one of the major concerns in the poultry industry for both economical and sanitary reason. The classification stage is harder in Europe, wherein government regulations do not allow for a washing phase in the eggs production process. This generates a wider population of dirty eggs with multiple combination of defects as are: blood spots, dirt stains, cracks, and stains both white and yolk of the egg. The automation of the detection of such defects in the farms, will give rise to a reduction of contamination in the mechanical parts at the poultry packer industry[1,2].

Eggshells collecting and packing is always performed automatically either at farm or at the packers companies, but visual inspection is done manually in both working environments. Expert graders, which suffer from visual stress and tiredness, perform the inspection. Thus, the integration of a real time artificial vision system to improve quality control and productivity at the farm initial packers constitutes a fundamental issue.

Previous studies related to the poultry industry deal with a unique type of well-defined defect, for classification purposes. Some use colour images[3], but most studies have been done with monochromatic images where the classification is accomplished with artificial neuronal networks[4,5].

Present work deals with the automation of the visual inspection eggshell process, based upon Genetic Algorithms (G.A.) to perform a more accurate classification that takes into account not only the global amount of defective pixels[6,7], but also their spatial distribution. The quantification of the spatial distribution of the defective pixels will allow for the separation, not achieved yet, of two kind of eggshells: 1) Eggs presenting a small spot in a reduced area, that has to be considered as "dirt_eggshell" and 2) Eggs having equal or higher number of defective pixels spread all over the eggshell that have to be classified as "clean _eggshell".

First objective has been addressed to the development of a well-fitted algorithm for critic time, to enhance and detect any kind of inhomogeneous pattern on a regular eggshell background under controlled illumination conditions. As a result, a binary image is obtained, where defective pixels are in white and the rest of the image is black. The second phase deals with the design of a classification algorithm using a set of rules, where the antecedent is formed by the conjunction of seven thresholds and the consequent is the classification label (“clean_eggshell”). The thresholds explore the number of defective pixels of the binary image at different resolution levels. The levels are obtained from a recursive image fragmentation that departs from the whole image (level 1) and ends with the image split in 64 sub-regions with identical size (level 7). The proposed G.A. derives the seven thresholds from a learning process that discovers the more adequate antecedents from two input sets of binary images: “dirt_eggshell” images or positive samples and “clean_eggshell” images or negative samples.

2 THE VISION SYSTEM AND THE CLASSIFICATION ALGORITHM

2.1 The vision system

A SONY DXC-950P camera endowed with 3 CCD per pixel (RGB/Colour), has been used. The camera characteristics are: image device size 1/3 inch, 752x582 effective pixels (resolution), intensity depth 3x8 bits and speed 30 frames per second (fps). The camera selection is based on the low noise level of the three-video signals, Red, Green and Blue transmitted through separated channels. The camera video signal is digitised by means of a Matrox Meteor board and digital images are displayed and stored on a computer (Pentium 233Mhz) to be processed further on. In addition an interactive programming environment[8,9] has been developed in Visual Basic to integrate both the acquisition of the digital images and the filtering and pre-processing algorithms. This tool has proved to be very useful to test different algorithms under distinct illumination conditions. Diffuse halogen light has been selected and the eggshell reflected light is focused on the CCD camera in a top-bottom, zenith view, to get a more uniform illumination.

2.2 The image pre-processing

The high quality digitised colour image obtained has to be appropriately filtered to enhance the defects in such a way that they become outstanding for a real time quantification. The objective being the achievement of a binary image wherein white pixels (active pixels) represent the spots present in the original colour image and black pixels the background. The sequence of filtering performed on the colour image to obtain a binary image with enhanced defects, well suited for automatic defect quantification. The process is decomposed on five steps[7,] , Figure 1. A set of eggshell colour images with their corresponding binary results is displayed in Figure 2.

2.3 The classification algorithm

An efficient and low computational cost method is proposed, to achieve a classification that accounts for both total defective pixels and their spatial distribution on the eggshell. The process is iteratively accomplished in seven levels, which are related to the image area. The first level classifies the image considering the global amount of active pixels in the global eggshell binary image:

IF the amount of active pixel is lower than an "initial threshold" THEN the image is split in two identical subregions AND the image with the highest active pixels value is retained for classification purposes but with a different threshold.

The fragmentation process proceeds as far as either the highest number of pixels in a part and at each level remains below the level threshold or the seventh level is reached, Figure 3. The threshold at each level indicates the minimum number of active pixels required to classify the egg as "dirt_eggshell". Therefore, the requirement to label a binary image as

"clean_eggshell" is that the highest number of active pixels at each level remains below the corresponding level threshold.
To calculate the number of active pixel in the global image it is mandatory to scan the whole image and this entails a high computational cost. This is way to speed up the computational process, the image has been initially split in $64 = 2^5$ parts to calculate at once, the number of active pixel per part. The features vector is then composed of seven parameters *{v1, v2, v3, v4, v5, v6, v7}* that represent the highest number of active pixels per part at each particular resolution level. Thus, *v1* correspond to the overall active pixels of the eggshell image {Area1= Image_size} and *v7* to the highest active pixels of the seventh level part {Area7 = (1/64) * Area1}. The rest of the components are calculated by a recursive process merging the value of component at level *i* from the previously computed values at level *i+1*. This image fragmentation process reduces computing time as the image pixels are scanned only once and then only $63 = \{2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0\}$ additions are needed to get the features vector. With this fragmentation algorithm the computational cost is reduced to the values of former algorithm[7] that only considered the global number of active pixels, for grading with a unique threshold. By fulfilling the critic time requirement, the proposed algorithm presents the advantage of being able to capture the active pixels spatial distribution, a crucial point to distinguish "clean spotty eggs" from "dirt one spot eggs", Figure 2.

Once every binary image is characterising by a seven components vector, the classification process can be viewed, as a process where rules are sequentially applied while a classification label is not reached. Rules are formulated as follows:

1. IF $(v_1>=th1)$ AND $(v_2>=th2)$ AND $(v_3>=th3)$ AND $(v_4>=th4)$ AND $(v_5>=th5)$ AND $(v_6>=th_{6exp})$ AND $(v_7>=th_{7exp})$ THEN "clean_eggshell"
2. IF *true* THEN "dirt_eggshell"

3 THE GENETIC ALGORITHM

This section presents the G.A. that has been designed to obtain the seven thresholds needed in the classification process. The algorithm is an example-based learning[9] process that calculates the thresholds from two set of feature vectors derived from binary image associated to clean (positive examples) and dirty (negative example) eggshells.

3.1 Individual encoding

Framing the problem within the machine learning paradigm, each individual, in the proposed algorithm, represents a candidate descriptor of the "clean egg" class. In other words, seven thresholds that hold for every positive example without explaining any negative one, and are able to discriminate between the two input classes. So each individual depicts the antecedent of a classification rule of the type: ***if*** *individual*$_i$ ***then*** *"clean_eggshell”*

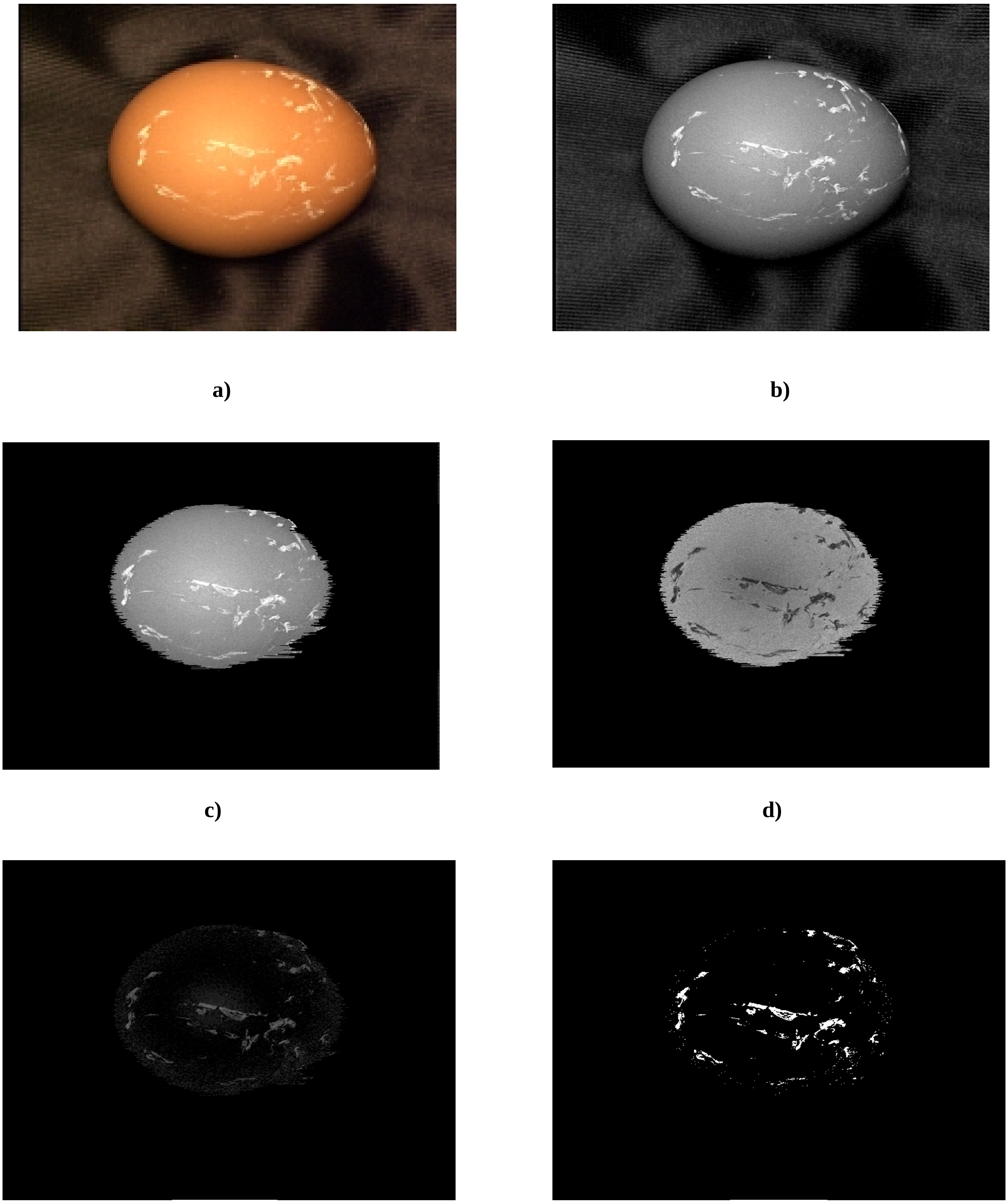

Figure 1: a) Original colour image; b) Green colour plane; c) Eggshell image; d) (RedPlane-BluePlane)/RedPlane+BluePlane); e) d)-d) mean value; and f) e)binary image with a segmentation threshold = 50 (grey scale: 0..255)

Figure 2: Examples of different types of defects: Original colour images and Binary images.

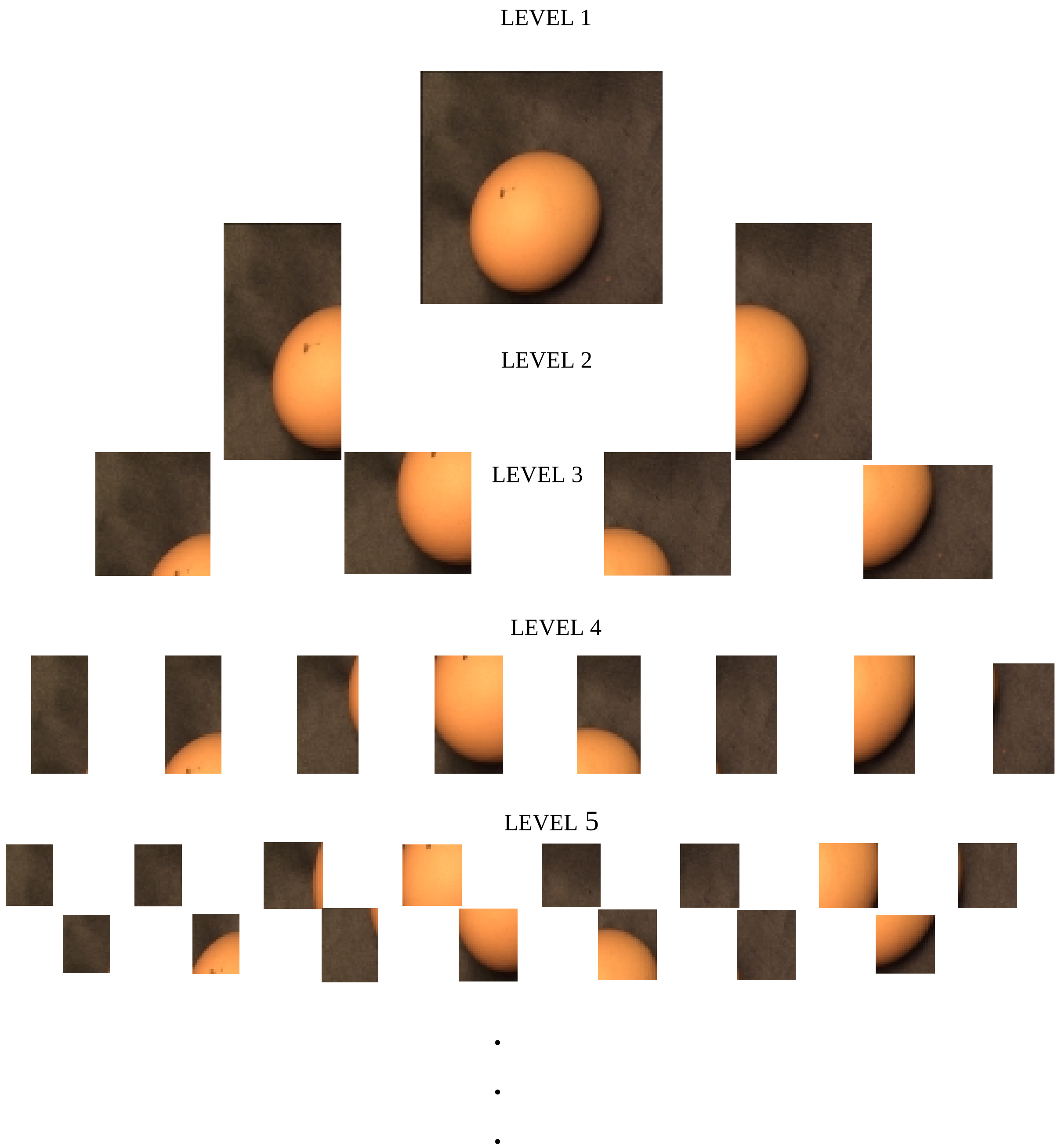

Figure 3: Image decomposition process

A fixed size string encodes an individual with seven genes representing the values that each threshold has in the rule, Figure 4. The number of bits associated to each gene is twelve, so that if only unsigned integer are considered, the codification range for each gene ranges between [0,4095].

Gene$_1$			**Gene$_2$**			**Gene$_3$**			**Gene$_4$**			**Gene$_5$**			**Gene$_6$**			**Gene$_7$**		
b_1	…	b_{12}	b_1	…	b_{12}	b_1	…	b_{12}	b_1	…	b_{12}	b_1	…	b_{12}	b_1	…	b_{12}	b_1	…	b_{12}

Figure 4: Individual encoding

3.2 Fitness function

The fitness function evaluates the quality of each individual. Thus, the fitness function would be associated to the individual cover, that is, the number of positive samples that can be explained by this individual from the whole set of positive samples, weighted by the number of negative samples explained also by this candidate

Four types of results[10] can describe the behaviour of a rule in the classification process:

1. ***True positive (tp)***: the rule classifies as positive an example that belongs to the set of positive examples.
2. ***False positive (fp)***: the rule classifies as positive an example that belongs to the set of negative examples.
3. ***True negative (tn)***: the rule classifies as negative an example that belongs to the set of negative examples.
4. ***False negative (fn)***: the rule classifies as positive an example that belongs to the set of positive examples.

By using the previous definitions it is possible to compute two indicators for each individual: 1) sensibility (S) and 2) specificity (E), that are expressed as:

$$S = tp/(tp+fn)$$

$$E = tn/(tn+fp)$$

Finally, the fitness function is defined as the product of these two indicators and the objective of the genetic algorithm is to maximise this product:

$$fitness\_funtion = S*E$$

4 EXPERIMENTAL RESULTS

The SUGAL[12] package (Sunderland Genetic Algorithm) has been used to develop the genetic algorithm. This package, written in C language, is specially suited to help in the design and experimentation with Genetic Algorithms and related techniques.

The test phase departs from a sample of 100 binary images eggs already classified in the farm by an expert grader. These images were randomly selected, thus containing similar rates of defects appearance than those found at the farm. One third of the former images were used in the training stage, but here the rate of conflictive eggs (miss-classified with former algorithm) was increased. First derived thresholds were employed to classify the two third of images not used in the training phase and miss-classified images were joined to the initial training set.

The generic G.A. used in the thresholds generation has the following characteristics:

1. The chromosomes are unsigned integer strings.
2. Each unsigned integer is codified with 12 bits.
3. The chromosomes are randomly initialised.
4. The entire population is replaced by its children in each generation.
5. Parents are randomly selected to produce children using the roulette method.
6. All children are produced by crossover of two parents; two children are produced simultaneously. One-point crossover is used.
7. Once children have been obtained, they may be subjected to mutation. The default mutation rate defined in SUGAL (1.0) has been used as well as the mutation per chromosome, then the average number of mutated genes per chromosome will be one. Uniform mutation has been selected, wherein a small amount is either added to or substrated from the current value of a gene and the step size varies uniformly in the range [-1,1].

The mean fitness evolution is shown in Figure 5, with a SUGAL run window at the bottom. The line with label 2 corresponds to the use of an elitism operator and that with label 1 is related to a run without the elitism operator.

With the derived thresholds vector, 80% of the eggs mis-classified with former algorithms were correctly classified and the corresponding a computer time is around 150ms.

5 CONCLUSIONS

The Genetic Algorithms are proposed to adjust a classification method as a learning example-based algorithm. The training phase is accomplished by means of two input set of examples and an adequate fitness function, to derive the antecedent of the classification rules.

This method improve the precision of former classification algorithms[7] with similar computational cost, by taking into account the spatial distribution of the defects in addition to their global area. This characteristic is fundamental to discriminate eggs with a unique concentrated spot ("dirt_eggshell") from those showing a random distribution of similar amount of defective pixels on the eggshell ("clean_egghell").

Genetic Algorithms allow the generation of the most appropriate classification threshold, based only on the two sets of examples pre-classified by experts farm graders. The way in which such thresholds are derived permit simple re-adjustments to allow for the modulation imposed by the market demands as well as changes in the government sanitary regulations.

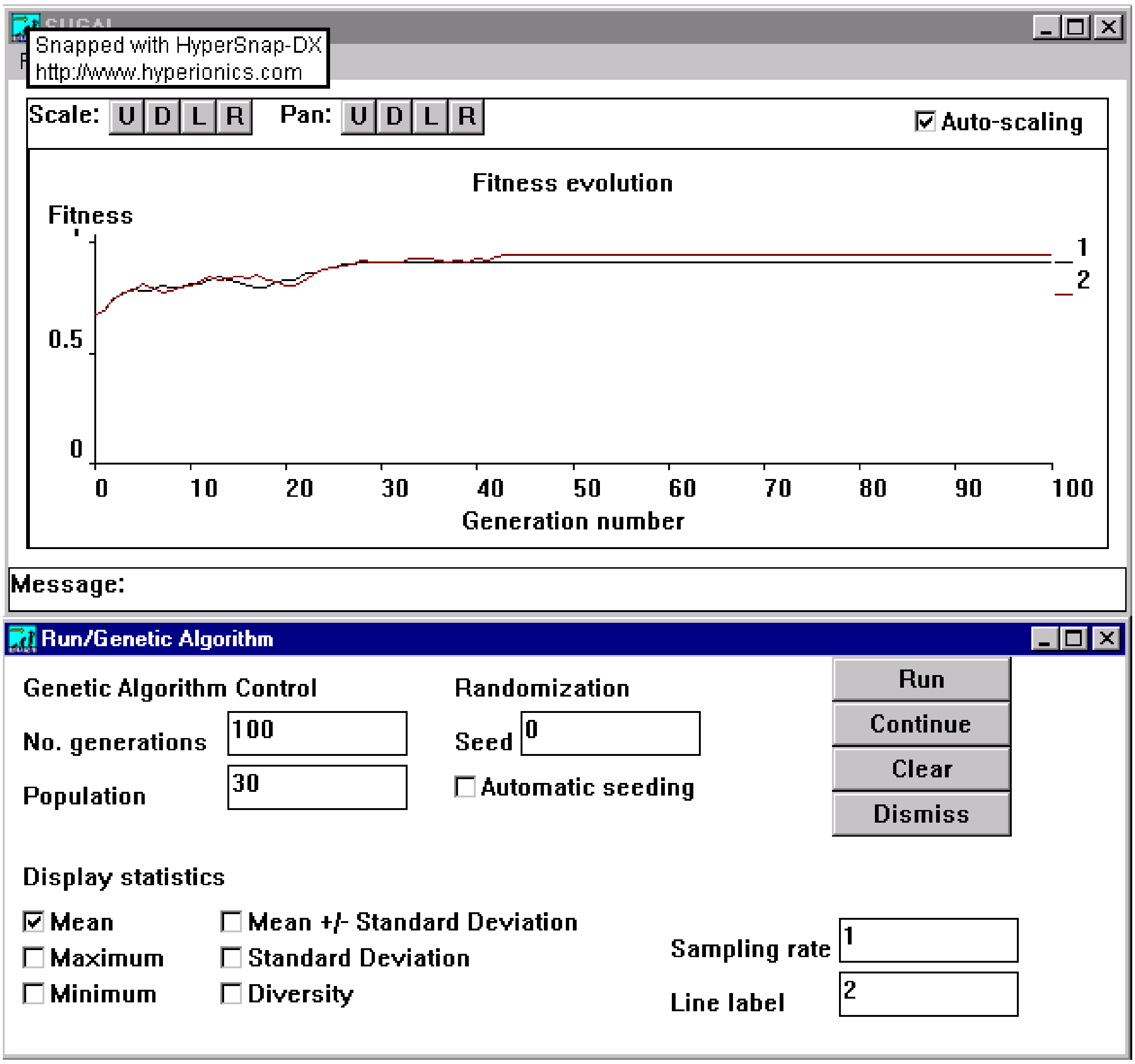

Figure 5: Mean fitness evolution in two cases: a) with Elitism, b) without Elitism.

6 ACKNOWLEDGMENTS

Present work was fully supported by Research Grants. CAM-06G-038-96: "Automation based on an artificial system at the poultry industry", CICYT 96-1392-C02-01: "Integrated active vision system", E.U. Thematic Network: "Networked industrial design and control applications using genetic algorithms and evolution strategies: INGENET", BRRT-CT97-5034. And the co-operation of several medium size companies: Jose Luis Ayuso AVICU-DAGU, Javier García GRANJA SOLEDAD, Pedro Ansoaín ZUCAMI, Felipe Fadón NEW DATA REAL TIME.

REFERENCES

[1] Moba Newsletters, *Omnia 330: the bold advance*. Barneveld, Holland. (1996).

[2] Diammond *8300 Report* Brochure, USA. (1995).

[3] V. C.Patel, R.W. McClendon, J.W.Goodrum, “Crack detection in eggs using computer vision and neural networks”, *A.I. Applications*, 8, 2, pp.21-31 (1994).

[4] V.C.Patel, R.W.McClendon, J.W.Goodrum, “Detection of cracks in eggs using color computer vision and artificial neural networks”, *ASAE Annual Intern. Meeting*, St.Joseph, (1995).

[5] V.C.Patel, W. McClendon, J.W. Goodrum, “Detection of blood spots and dirt stains in eggs using computer vision and neural network”, *Applied Eng. In Agriculture*, 12, 2, pp.253-258 (1996).

[6] M.C.García-Alegre, J.Enciso, A.Ribeiro, D.Guinea, “Towards an automatic visual inspection of eggshell defects”, *Proc. Intern.Workshop on Robotics and Automated Machinery for Bio-Productions*, pp.51-56, Gandia, Spain, (1997).

[7] M.C. García-Alegre, A. Ribeiro, D. Guinea, and G. Cristobal, “Eggshell defects detection based on color processing”, *SPIE 2000 Electronic Imaging Conf.*, CA, Jan. (2000).

[8] J.Enciso, D.Guinea, A.Ribeiro, M.C.García-Alegre, “Entorno interactivo de captura y análisis de imagen AVICOM: Detección de defectos en cáscara de huevo”, *Tech.Report TR-10/98*, Systems Dpt., IAI-CSIC, (1998).

[9] J.Enciso, R.Sucre, D.Guinea, M.C.García-Alegre, “Entorno Interactivo de Proceso de imagen real:FILEPIC”, *Tech.Report TR-01/98*, Systems Dpt., IAI-CSIC, (1998).

[10] T.M. Mitchell, *Machine Learning*, McGraw Hill (1997).

[11] M.V. Fidelis, H.S. Lopes, and A.A. Freitas, “Um algoritmo genético para descobrir regras de classificaçao em data mining”, *Encontro Nacional de Inteligencia Artificial (ENIA-99)*, 11-29. PUC-Rio(1999).