Image Segmentation using Isodata Clustering with Parameters Estimated by Evolutionary Approach: Application to Quality Control

M. Merzougui,
LABO MATSI, ESTO, B.P 473,
University of Mohammed I,
OUJDA, MOROCCO.

M. Nasri,
LABO MATSI, ESTO,B.P 473,
University of Mohammed I,
OUJDA, MOROCCO

B. Bouali,
LABO AGA, FSO,
University of Mohammed I,
OUJDA, MOROCCO

ABSTRACT

A segmentation method based on pixel classification by *Isodata* algorithm and evolution strategies is proposed in this paper. The *Isodata* algorithm is an unsupervised data classification algorithm. Its result depends strongly on two parameters: distance threshold for the union of clusters and threshold of typical deviation for the division of a cluster. A bad choice of these two parameters leads the algorithm to spiral out of control leaving the end only one class. To determine these parameters and improvements to this algorithm, evolution strategies are used. An evolutionary algorithm is adapted to estimate the two optimal thresholds to be used by the algorithm then *Isodata*.
To note that the other parameters are chosen empirically.
The application of this evolutionary method (*Evolutionary Isodata*: *EIsodata*) on synthetic and real images helps to validate this approach and show its interest in the problem of decision support in the quality control.

Keywords

Classification, Segmentation by pixel classification, *Isodata* algorithm, evolutionary strategies.

1. INTRODUCTION

The segmentation is an essential stage in image processing. There are many consistent methods available today for image segmentation, among these, there is the segmentation based on pixels classification as a function of their grey level values [1][2][3]. Every pixel in the image holds an inherent relationship with the pixels in its surrounding. The information at a particular pixel may be in relation with the information over the whole or part of the image. The mean or median value of the gray level of the pixel is selected.

The process of the proposed segmentation by pixel classification consists of three stages**:** [1][2][3].

- Acquisition of data for each pixel in order to form the attribute vector
- Estimation of the parameters of *Isodata* algorithm with the evolutionary strategies.
- Segmentation based on the acquired information

✓ **Acquisition stage**

For each pixel, two values are calculated, the mean value of the grey levels (*MGL*), and the difference between the *MGL* and the maximum value of the grey levels surrounding the particular pixel (*DMGL*). For this purpose, a square window centred at the particular pixel is used.

✓ **Estimate the parameters of *Isodata***

The *Isodata* algorithm suffers of adjustment of its parameters. This is a step that is always difficult as you want towards the expected solution. This algorithm converges in a finite number of iterations, but the resulting solution depends on the values of the parameters chosen. Indeed if the algorithm resets a second time with other parameters, it will converge to a solution completely different from the first or escape out of control leaving at the end one alone class.[4]

✓ **Segmentation stage**

Isodata and *evolutionary strategies* are used for this segmentation purpose. The algorithms use the information provided by the parameters *MGL* and the *DMGL*, associated with each and individual pixel in order to classify the pixel with respect to centers that evolve at each iteration.

In Section 2 image segmentation with *Isodata algorithm* is presented. Section 3 gives an introduction to evolution strategies approach. The proposed evolutionary *algorithm* is presented in section 4, along with the image evolutionary segmentation. In section 5, a validation of our approach is given, simulations and experimental results are obtained over synthetic and real images. Finally we give a conclusion.

2. ISODATA SEGMENTATION

2.1. Descriptive elements

Consider a set of *M* objects $\{O_1, O_2,..., O_M\}$ characterized by *N* attributes, grouped in a line vector form $V = (a_1\ a_2 \ldots a_N)$. Let $R_i = (a_{ij})_{1\le j\le N}$ be a line vector of $\boldsymbol{R}^N$ where a_{ij} is the value of the attribute a_j for the object O_i. Let *mat_obs* be a matrix of *M* lines (representing the objects O_i) and *N* columns (representing the attributes a_j):

$$mat\_obs = \left(a_{ij}\right)_{\substack{1\le i\le M \\ 1\le j\le N}} \qquad (1)$$

V is the *attribute vector*, R_i is the *observation associated* with O_i or the *realization* of the attribute vector *V* for this object, $\boldsymbol{R}^N$ is the *observations space* [1][2][3] and *mat_obs* is the *observation matrix associated* with *V*. The ith line of *mat_obs* is the observation R_i. R_i belongs to a class CL_s, s=1, …, *C*.

2.2. Isodata algorithm

The *Isodata* method is the method developed by Ball, Hall and others in the 1960s. The *Isodata* method is a method which added division of a cluster, and processing of fusion to the K-means method. The individual density of a cluster is controllable by performing division and fusion to the cluster generated from the K-means method. The individual in a cluster divides past [a detached building] and its cluster, and the distance between clusters unites them with past close. The parameter which set up division and fusion beforehand determines. The procedure of the *Isodata* method is shown as follows:

1. Parameters, such as the number of the last clusters, a convergence condition of rearrangement, judgment conditions of a minute cluster, branch condition of division and fusion, and end conditions, are determined.
2. The initial cluster center of gravity is selected.
3. Based on the convergence condition of rearrangement, an individual is rearranged in the way of the K-means method.
4. It considers with a minute cluster that it is below threshold with the number of individuals of a cluster, and excepts from future clustering.
5. When it is more than the threshold that exists within fixed limits which the number of clusters centers on the number of the last clusters, and has the minimum of the distance between the cluster center of gravity and is below threshold with the maximum of distribution in a cluster, clustering regards it as convergence and ends processing. When not converging, it progresses to the following step.
6. If the number of clusters exceeds the fixed range, when large, a cluster is divided, and when small, it will unite. It divides, if the number of times of a repetition is odd when there is the number of clusters within fixed limits, and if the number is even, it unites. If division and fusion finish, it will return to 3 and processing will be repeated.

- Division of a cluster: If it is more than threshold with distribution of a cluster, carry out the cluster along with the 1st principal component for 2 minutes, and search for the new cluster center of gravity. Distribution of a cluster is re-calculated, and division is continued until it becomes below threshold.
- Fusion of a cluster: If it is below threshold with the minimum of the distance between the cluster centers of gravity, unite the cluster pair and search for the new cluster center of gravity. The distance between the cluster center of gravity is re-calculated, and fusion is continued until the minimum becomes more than threshold.

Although the *Isodata* method can adjust the number of certain within the limits clusters, and the homogeneity of a cluster by division and fusion, global optimal nature cannot be guaranteed. Since the *Isodata* method has more parameters than the K-means method, adjustment of the parameter is still more difficult.[4][5]

2.3. Isodata segmentation

The objects that are processed by the *Isodata* algorithm are the pixels of the input image. The observation matrix in this case is formed by two columns which represent the attributes associated with each pixel of the image: the columns are associated with the *MGL* and the *DMGL*. The size of the square window used must have an odd length (3 * 3, 5 * 5 ...) [1][2][3][6][7].

In this process each pixel is attributed to a specific class. The resulting image is segmented into C different regions where each region corresponds to a class.

3. EVOLUTION STRATEGIES

Evolutionary strategies (*ES*) are particular methods for optimizing functions. These techniques are based on the evolution of a population of solutions which under the action of some precise rules optimize a given behaviour, which initially has been formulated by a given specified function called fitness function. [8][9][10][11]

An *ES* algorithm manipulates a population of constant size. This population is formed by candidate points called chromosomes. Each of the chromosomes represents the coding of a potential solution to the problem to be solved, it is formed by a set of elements called genes, and these are real.

At each iteration, called generation, is created a new population from its predecessor by applying the genetic operators: selection and mutation. The mutation operator perturbs with a Gaussian disturbance the chromosomes of the population in order to generate a new population permitting to further optimize the fitness function.

This procedure allows the algorithm to avoid the local optimums. The selection operator consists of constructing the population of the next generation. This generation is constituted by the pertinent individuals [8][9][10].

Figure 1 illustrates the different operations to be performed in a standard *ES* algorithm [8][9][2][3] :

```
Random generation of the initial population
Fitness evaluation of each chromosome
Repeat
    Select the parents
    Update the genes by mutation
    Select the next generation
    Fitness evaluation of each chromosome
Until Satisfying the stop criterion
```

Figure 1: Standard SE algorithm.

4. EVOLUTIONARY ALGORITHM PROPOSED

4.1. Proposed coding

The proposed algorithm consists of selecting among all of the possible partitions the optimal partition by minimizing a criterion. This yields the optimal parameters $(p_s)_{1\leq s\leq np}$. Thus, the real coding following is suggested:

$$chr = (p_s)_{1\leq s\leq np} = \left(p_1, p_2, p_3 \ldots p_{np}\right) \qquad (2)$$

The *chr* chromosome is a real line vector of dimension *np*. The genes $(p_s)_{1\leq s\leq np}$ are the components of the chromosome *chr*.

To avoid that the initial solutions be far away from the optimal solution, each chromosome *chr* of the initial population should satisfy the condition:

$$p_s \in [\min p , \max p] \qquad (3)$$

In the proposed algorithm, any chromosome with a gene that does not satisfy this constraint is eliminated. This gene, if any, is replaced by another one which complies with the constraint [2][3][6][7][12].

4.2. The proposed fitness function

Unlike the K-means algorithm that requires you to make the number of classes a priori, the *Isodata* algorithm itself determines the number of classes does not exceed a maximum previously selected and is among the parameters that must provide user. We are inspired by the behavior of the algorithm *Isodata* to choose selective function. Indeed, the repetition of this algorithm several times with parameters automatically generated by evolution strategies lead to the end of the optimal number of classes, this is the ultimate goal. To this end the test, well known, Xie and Beni is selected. The latter is based on a measure of compactness and separability of classes:
- The criterion of compactness, defined by:

$$comp = \frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\|R_i - g_s\|^2 \tag{4}$$

- And the separability criterion, defined by:

$$Sép = min_{s\neq s'}\|g_s - g_{s'}\|^2 \tag{5}$$

chr is a chromosome of the population formed by the parameters $(p_s)_{1\leq s\leq np}$, to calculate the selective value of *chr* we define the function F which reflects the selective behaviour to optimize [2][7]:

$$F(chr) = \frac{\frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\|R_i - g_s\|^2}{min_{s\neq s'}\|g_s - g_{s'}\|^2} \tag{6}$$

chr is optimal if F is minimal.

4.3. The proposed mutation operator

The performances of an algorithm based on evolutionary strategies are evaluated according to the mutation operator used [4]. One of the mutation operator forms proposed in the literature [2][10] is given by:

$$chr^* = chr + \sigma \times N(0,1) \tag{7}$$

where *chr** is the new chromosome obtained by a Gaussian perturbation of the old chromosome *chr*. *N(0,1)* is a Gaussian disturbance of mean value 0 and standard deviation value 1, σ is the strategic parameter. σ is high when the fitness value of *chr* is high. When the fitness value of *chr* is low, σ must take very low values in order to be not far away from the global optimum.
Of this approach, a new shape of the operator of the mutation is proposed. The fact to propose a new operator of the mutation is motivated by the interest to reach the global solution in a small computational time.

The mutation operator proposed in this work consists in generating, from the *chr*, the new chromosome *chr** formed by the parameters $(p^*_s)_{1\leq s\leq np}$, as:

$$p^*_s = p_s + f_m \times p_s \times N(0,1) \tag{8}$$

where f_m is a multiplicative constant factor taken to be randomly chosen between 0.5 and 1.

Generating children from parent chromosomes the technique of selection by ordering is adopted. The elitist technique is also used [2][3][11].

4.4.The EIsodata algorithm proposed

Figure 2 shows the different steps of the proposed algorithm *EIsodata* [2][3][6][7][12].

```
1.1. Fix:
- The size of the population maxpop.
- The maximum number of generations maxgen.
- The maximum number of classes C.
1.2. Generate randomly the population P:
        P = {chr1, .., chrk, ..., chrmaxpop}
1.3. Verify for each chr of P the constraint:
        $p_s \in [minp, maxp]$,
Repeat
2.1  Isodata for each chr of P.
2.2. Compute for each chr of P its fitness value F(chr).
2.3. Order the chromosomes chr in P from the best to the poor ( in an increasing order of F).
2.4. Choose the best chromosomes chr.
2.5. Generate randomly the constant $f_m$ ($f_m \in [0.5, 1]$).
2.6. Mutation of all the chromosomes chr of P except the first one (elitist technique):
        $p^*_s = p_s + f_m \times p_s \times N(0,1)$
2.7. Verify for each chr of P the constraint:
        $p_s \in [minp, maxp]$,
 (The population P obtained is the population of the next generation )
Until Nb_gen (generation number) > maxgen

3.1 To keep the optimal chr:  the first of the last P
3.2  Isodata for the optimal chr
```

Figure 2: The *EIsodata* algorithm proposed.

5. EXPERIMENTAL RESULTS AND EVALUATIONS

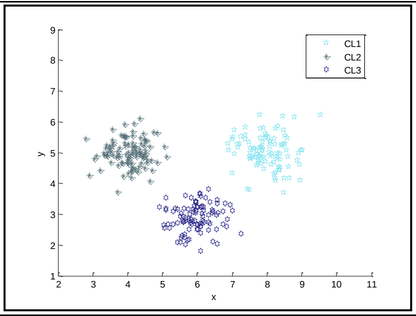
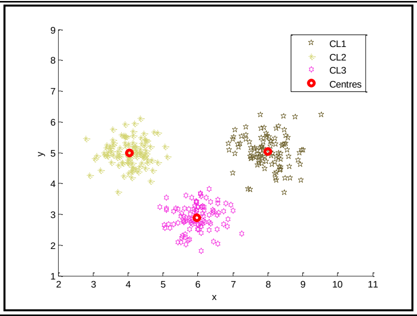
5.1. Introduction

In order to evaluate the performances of the proposed method, firstly, simulations point clouds are used. Then, the segmentation of a synthetic image and other real images are performed [1]. The segmentation is carried out by *Isodata* algorithm after that the optimal chr ($\mathbf{chr}_{op}$) has been obtained by the evolutionary algorithm proposed.
$\mathbf{chr}_{op}$= (**a**,**b**) where : **a** is the distance threshold for the union of clusters, **b** is threshold of typical deviation for the division of a cluster.

5.2. Simulation tests

Simulation tests are used in the observation space of dimension 2 (N = 2). These tests are different from each other by the type of distribution of classes in the space of observations. In each test, the classes are generated randomly by Gaussian distributions and each class contains 100 observations (Table 1).

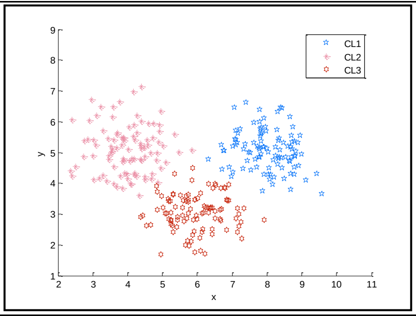
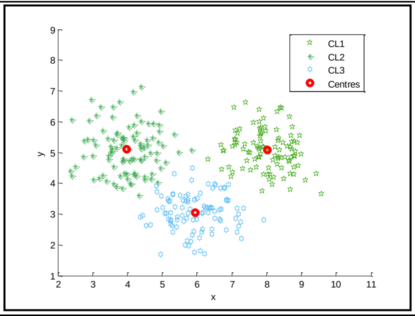
Table 1: Simulations tests

<table>
<tr>
<td>
</td>
<td>
</td>
<td>

$chr_{op} = (0.9501 \quad 0.7621)$

The error rate obtained is :

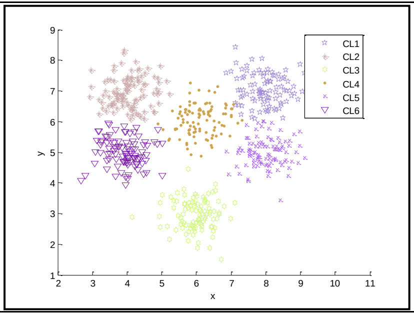
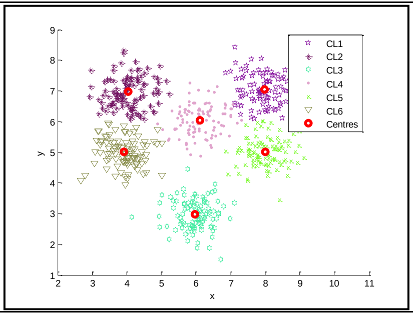
$\tau = 0\%$

</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>

$chr_{op} = (0.1114 \quad 0.9888)$

The error rate obtained is :

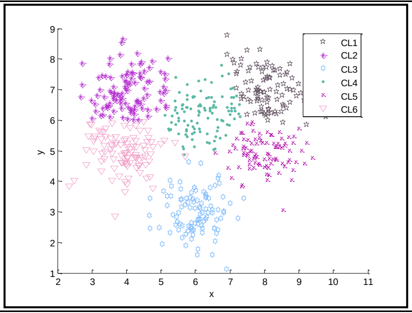
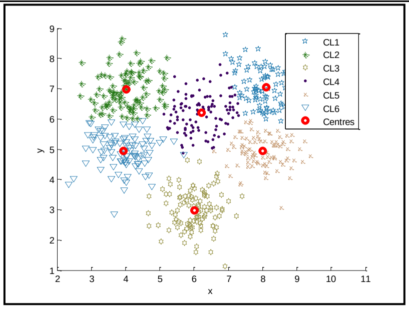
$\tau = \frac{2}{300} = 0{,}67\%$

</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>

$chr_{op} = \big(0.6135 \quad 0.9708\big)$

The error rate obtained is :

$\tau = \frac{6}{600} = 1\%$

</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>

$chr_{op} = (0.5430 \quad 0.8445)$

The error rate obtained is :

$\tau = \frac{14}{600} = 2{,}3\%$

</td>
</tr>
</table>

For several initialisations of *EISODATA* algorithm, the classes obtained in each test simulation are stable. The error obtained is low, this confirm the good performance of the proposed approach.

5.3. Synthetic image

A synthetic image is constructed that is named SYNTH1 of size 113 * 80 (figure 3).

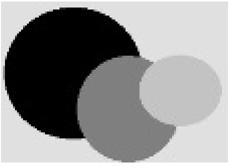


Figure 3: Synthesised image SYNTH1.

Table 2 shows the classes of SYNTH1 along with the grey level values and the number of pixels in each class.

Table 2: Information on SYNTH1

Class Region	Description	Level of gray	Number of pixels
1	Big disk	0	2593
2	Disk middle	125	1701
3	Small disk	190	1155
4	Bottom of picture	225	3591

a 3

Figure 4 shows the results of segmentation by the proposed algorithm:

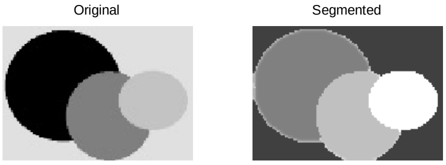


Figure 4 : Result of image segmentation
***chr*$_{op}$ = (0.9501 0.7621)**.

One notices that the number of classes gotten by the proposed evolutionary algorithm coincides precisely with the real number of the classes (4 classes). Synth1 image segmentation results are summarised in table 3.

Table 3: results of segmentation for Synth1 image

Class Region	Description	Level of gray	Number of pixels
1	Big disk	0.50	2486
2	Disk middle	0.75	1690
3	Small disk	1	1142
4	background	0.25	3360

The results show that the *proposed evolutionary algorithm* clearly detects the all objects of the image, background, big disk, small disk and disk middle. The number of pixels badly classified is estimated to 362.

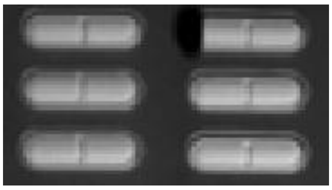
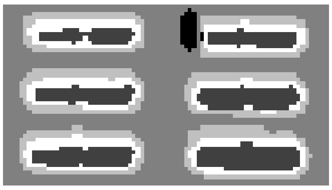
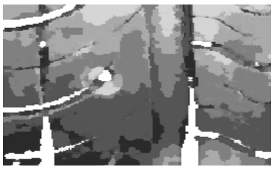
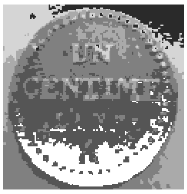
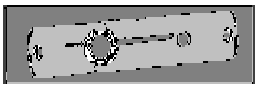
The error rate obtained is:

$$\tau = \frac{362}{9040} = 4\%$$

5.4. Real images

In this test phase, four images are taken. The objective is to detect if there is a default. For each image, the coordinates of each pixel are the average gray level and the difference between this average and the maximum gray level in a neighbourhood window size 3 * 3. The results of the segmentation and the information obtained are shown in the table 4:

Table 4: real image and results segmentation by EIsodata

Original image	Segmented image	The segmentation information
		Number of classes is 4 *chr*$_{op}$=(0.1210 0.2548)
		Number of classes is 6 *chr*$_{op}$ =(0.9501 0.7621)
		Number of classes is 6 *chr*$_{op}$ =(8.9130 4.4470)
		Number of classes is 4 *chr*$_{op}$ =(0.3705 0.3127)

The results obtained show that in each test, the default is correctly detected.

6. CONCLUSION

The use of the *Isodata* algorithm for the classification or for the image segmentation necessity inevitably to give several parameters. Thus, it is difficult by the experience to arrive to a good classification (or segmentation) with standard *Isodata* and often, this algorithm escapes all control and gives only one class at the end. The strategies of evolution are used to estimate a part of the parameters.

The proposed evolutionary algorithm is tested for the classification and the image segmentation. Although computation time became long by the proposed method, the gotten results are better.

This approach may be used for problems of decision support in quality control.

In a next research, all the parameters will be estimated by the strategies of evolution.

7. REFERENCES

[1] Cocquerez T.P. et Phillip S. 'Analyse d'images : Filtrage et segmentation'. Editions MASSON, Paris, 1995.

[2] Nasri M. 'Contribution à la classification de données par Approches Evolutionnistes : Simulation et Application aux images de textures'. Thèse de doctorat. Université Mohammed premier Oujda 2004.

[3] M. Nasri, M. EL Hitmy, H. Ouariachi and M.Barboucha. 'Optimization of a fuzzy classification by evolutionary strategies'. In proceedings of SPIE Conf., 6 th international conference on quality control by artificial vision. Repulished as an SME Technical Paper by The society of manufacturing Engineers (SME). Paper number MV03-233, IDTP 03 PUB 135, vol.5132, pp. 220-230 USA, 2003.

[4] Kohei Arai and XianQiang Bu, 'ISODATA clustering with parameter (threshold for merge and split) estimation based on GA: Genetic Algorithm' Reports of the Faculty of Science and Engineering, Saga University, vol. 36, No.1, pp.17-23, 2007.

[5] Nargess Memarsadeghi and al. 'A fast implementation of the isodata clustering algorithm', IJCGA, vol.17, No. 1, pp. 71-103, 2007.

[6] A. EL Allaoui, M. Merzougui, M. Nasri, M. EL Hitmy and H. Ouariachi. 'Optimization of Unsupervised Classification by Evolutionary Strategies'. IJCSNS International Journal of Computer Science and Network Security, ISSN: 1738-7906, vol. 10 No. 6 pp. 325-332 June, 2010.

[7] M. Merzougui, A. EL Allaoui, M. Nasri, M. EL Hitmy and H. Ouariachi. 'Unsupervised classification using evolutionary strategies approach and the Xie and Beni criterion'. IJAST International Journal of Advanced Science and Technology, ISSN: 2005-4238, vol. 19, pp 43-58 June, 2010.

[8] Presberger, T., Koch, M. 'Comparison of evolutionary strategies and genetic algorithms for optimization of a fuzzy controller'. Proc. of EUFIT'95, Aachen, Germany, august 1995.

[9] Sarkar, M. et *al.* 'A clustering algorithm using an evolutionary-based approach'. Pattern Recognition Letters, 1997.

[10] Presberger, T., Koch, M. 'Comparison of evolutionary strategies and genetic algorithms for optimization of a fuzzy controller'. Proc. of EUFIT'95, Aachen, Germany, august 1995.

[11] H.Ouariachi, 'Classification non –Supervisée de données par les réseaux de neurones et par une approche évolutionniste : application à la segmentation d'images'. Thèse de doctorat. Université Mohammed premier Oujda 2001.

[12] Renders, J.M., 'Algorithmes génétiques et Réseaux de Neurones'. Editions HERMES, 1995.