Unsupervised Evolutionary Segmentation Algorithm based on Texture Analysis

Cynthia B. Pérez, Gustavo Olague

CICESE, Research Center, División de Física Aplicada
Centro de Investigación Científica y de Educación Superior de Ensenada,
B.C., Km. 107 Carretera Tijuana-Ensenada, 22860, Ensenada, B.C., México.
cbperez@cicese.mx,olague@cicese.mx

Abstract. This work describes an evolutionary approach to texture segmentation, a long-standing and important problem in computer vision. The difficulty of the problem can be related to the fact that real world textures are complex to model and analyze. In this way, segmenting texture images is hard to achieve due to irregular regions found in textures. We present our *EvoSeg* algorithm, which uses knowledge derived from texture analysis to identify how many homogeneous regions exist in the scene without *a priori* information. *EvoSeg* uses texture features derived from the Gray Level Cooccurrence Matrix and optimizes a fitness measure, based on the minimum variance criteria, using a hierarchical GA. We present qualitative results by applying *EvoSeg* on synthetic and real world images and compare it with the state-of-the-art JSEG algorithm.

1 Introduction

Human vision is a complex process that is not yet completely understood, despite several decades of studying the problem from the standpoint of natural science and artificial intelligence. In computer vision, the complex physical process of identifying colors, shapes, textures and automatically grouping them into separate objects within a scene continues to be an open research avenue. Image segmentation denotes a process by which an input image is partitioned into regions that are homogeneous according to some group of characteristics, i.e., texture information. Formally, image segmentation could be defined as follows: Segmentation of I is a partition P of I into a set of M regions R_m, $m = 1, 2, ..., M$, such that:

$$\begin{array}{l} 1)\ \bigcup_{m=1}^{M} R_m = I \quad \text{with} \quad R_m \bigcap R_n = \emptyset, \quad m \neq n \\ 2)\ H(R_m) = true \quad \forall\ m \\ 3)\ H(R_m \bigcup R_n) = false \quad \forall\ R_m \ \text{and}\ R_n \ \text{adjacent} \end{array} \tag{1}$$

where I is the image and H is the predicate of homogeneity. Thus, each region in a segmented image needs to simultaneously satisfy the properties of homogeneity and connectivity [1]. A region is homogeneous if all of its pixels satisfy an homogeneity predicate defined over one or more pixel attributes such as intensity, texture or color. On the other hand, a region is said to be connected if a connected path exists between any two pixels within the region.

Because of the large diversity of segmentations methods, it is difficult to review each individual image segmentation technique proposed up to now. Segmentation methods usually can be classified as region or boundary based [2], histogram-based [3] or graph-based [4], to mention but a few. In the evolutionary computer vision community there are also a number of works dealing with image segmentation [1, 5–8]. The application of evolutionary techniques to image processing and computer vision have increased mainly due to the robustness of the approach [9]. In this paper we pose image segmentation as a combinatorial optimization problem. We present *EvoSeg* as an evolutionary image segmentation algorithm based on texture information extracted in a Gray Level Cooccurrence Matrix (GLCM). *EvoSeg* attempts to identify how many homogeneous regions exist in the scene without *a priori* information. We qualitatively compare our results with the JSEG algorithm [10].

The remainder of this paper is organized as follows. Section 2 describes the cooccurrence matrix and texture descriptors that are used for texture analysis. Section 3 introduces the *EvoSeg* algorithm giving emphasis to the explanation on how evolution was applied to the segmentation problem. Section 4 shows the results of the *EvoSeg* and JSEG algorithm, illustrating several comparative results. Finally, Section 5 gives some concluding remarks.

2 Texture Analysis

Texture analysis is a long-standing and important problem in computer vision. It comprises problems like texture classification, texture segmentation, texture synthesis and shape from texture. Obviously, a given application of texture analysis usually falls into more than one category; our work is focused on texture segmentation. The difficulty of the problem can be related to the fact that real world textures are complex to model and analyze. Historically, the most commonly used methods for describing texture information are statistical based approaches, which include first order, second order and higher order statistical methods. These methods analyze the distribution of a texture property for each pixel contained in the image. We are interested in second order statistical methods, which represent the joint probability density of the intensity values between two pixels separated by a given vector $\mathbf{V}$. This information is coded using the GLCM denoted by $M(i, j)$. The GLCM $M(i, j)$ describes the frequency of a gray value that appears in a specific spatial relationship with another gray value within a given window, denoted by $f(x, y)$.

Formally, the GLCM $M_{i,j}(\pi)$ defines a joint probability density function $f(i, j|\mathbf{V}, \pi)$ where i and j are the gray levels of two pixels separated by a vector $\mathbf{V}$, and $\pi = \{\mathbf{V}, R\}$ is the parameter set for $M_{i,j}(\pi)$. The GLCM identifies how often pixels that define a vector $\mathbf{V}\,(d, \theta)$, and it differs by a certain amount of intensity value $\Delta = i - j$ appearing in a region $\mathbf{R}$ of a given image $\mathbf{I}$.

The GLCM presents a problem when the number of different gray levels in region **R** increase, turning difficult to handle or use directly due to the dimensions of the GLCM. Fortunately, the information encoded in the GLCM can be expressed by a varied set of statistically relevant numerical descriptors. This reduces the dimensionality of the information that is extracted from the image using the GLCM. Extracting each descriptor from an image effectively maps the intensity values of each pixel to a new dimension. Descriptors extracted from $M(i,j)$ include the following: Entropy, Homogeneity, Local Homogeneity, Contrast, Moments, Inverse Moments, Uniformity, Maximum Probability, Correlation and Directivity [11, 12]. Such descriptors may be defined in the spatial domain, like those extracted from the GLCM, or can be extracted in other frequency domains.

3 EvoSeg Algorithm

The evolutionary segmentation algorithm *EvoSeg* attempts to identify how many homogeneous regions exist in the scene without *a priori* information. The algorithm uses a fitness function that evaluates possible segmentations based on region homogeneity and region distinctiveness. The *EvoSeg* algorithm carries out two general process, a statistical texture analysis process and a segmentation process immerse into a genetic algorithm. The complete flow-chart of the algorithm is shown in Fig. 1(a).

3.1 Statistical texture analysis process

The statistical texture analysis process is used as a way of obtaining representative compact data of the image texture through the GLCM and texture descriptors. In order to calculate the GLCM, we tested experimentally different windows sizes, directions and distances. The results showed only substantial differences when increasing the window size, it produces blurring descriptor images. We used the following parameters: the window size was set as 7 × 7 pixels, the direction as 0°, and the distance as 1 pixel. The GLCM information is used to calculate the texture descriptors for each pixel of the image; in that way, we obtain one matrix for each texture descriptor. Different texture descriptors were used one by one, or as a combination of them. However, the descriptor which gave better experimental results in our study was the second order moment, which is defined as follows:

$$Mom_k = \sum_i \sum_j (i-j)^2 \cdot M(i,j) \tag{2}$$

3.2 Segmentation process immersed into a genetic algorithm

Our approach evaluates the image segmentation using a genetic algorithm to decide which segmented image is the best from the set of possible solutions based on the following algorithm, see Fig. 1(a):

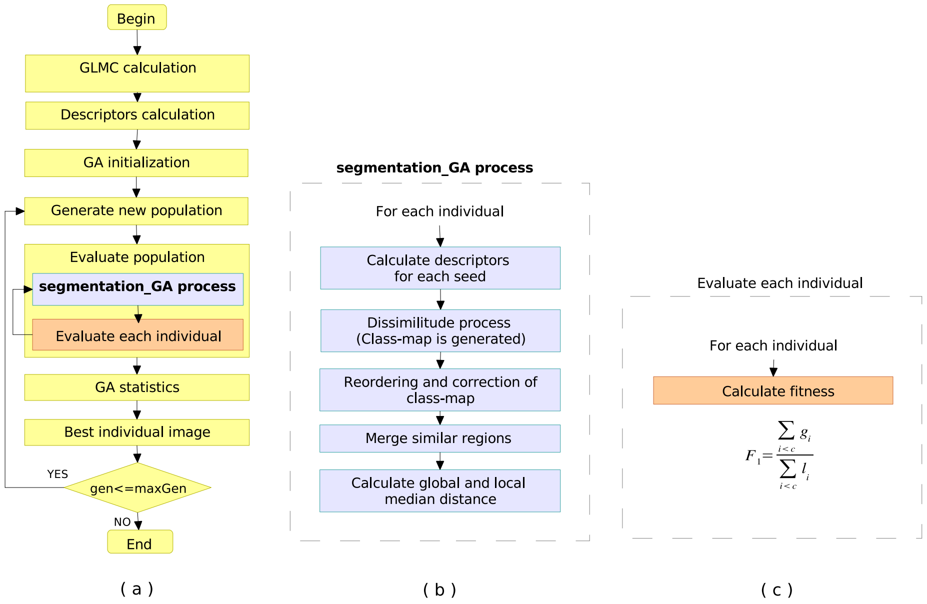


Fig. 1. *EvoSeg* Flow-chart. (a) Complete *EvoSeg* flow-chart. (b) General steps for segmentation process. (c) GA Fitness Function scheme.

1. **Initialize the genetic algorithm parameters**. The chromosome is coded with a hierarchical structure, it contains M elements representing the possible centroids of each image region denoted by C_i where $i = 1...M$ as shown in Fig. 2(a). The binary array, see Fig. 2(b), contains the control bits that indicates the final centroids to use in the segmentation process, see Fig. 2(c). We experimentally conclude that it is better to use the binary array because it allows a greater diversity of centroids in new individuals. Tournament was chosen as the selection method in the GA. The crossover is accomplished after the first generation of each individual with a probability of 90%. The mutation rate used was 10% with a population size of 50 and 30 generations.
2. **Segmentation process**. Once the algorithm has selected the possible regions, it is necessary to know how to classify the pixels. We use texture and spatial information in order to classify pixels of the possible regions. Suppose we have a set $P = \{x_1, x_2, ..., x_N\}$ of N pixels where N is the total amount of pixels. Let also have M disjoint subsets $R_1, R_2, ...R_M$ where each subset represents a region and has an associated centroid C_i. Moreover, we have a set $D = \{d_1, d_2, ..., d_N\}$ of N descriptor values. In this way, the segmentation process is as follows, see Fig. 1(b):
   (a) We calculate a descriptor value s_{C_i} for each initial centroid given by the mean of the descriptor values d_i within a 5×5 neighborhood around the centroid.

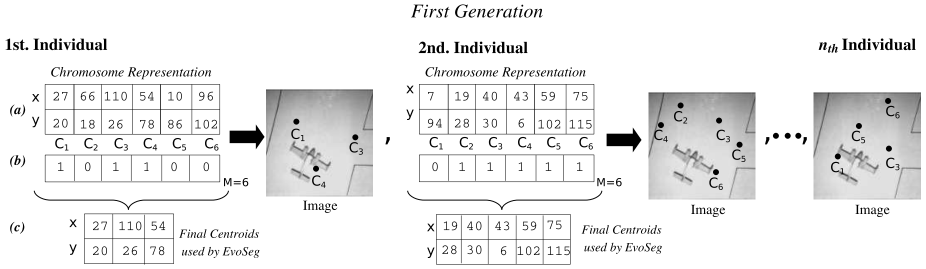


Fig. 2. Example of the chromosome representation in the first generation of the *EvoSeg* algorithm.

(b) We create an initial class map, that can be viewed as a special matrix indicating the region to which each pixel belongs. The class map is created using nearest neighbor classification. Two distances measures[1] are defined, Δ and δ. Δ is the distance in the descriptor space and δ is the distance within the image plane. In this way, a pixel x_j is assigned to a region R_i, if $\Delta(d_j, s_{C_i}) < \Delta(d_j, s_{C_l}) \;\; \forall \; l$ and if $\delta(x_j, C_i) < t$, where t is a threshold and $l = 1...M$ with $l \neq i$.

(c) The class map is rearranged and corrected with the purpose of improving the regions with a poor classification.

(d) Two regions R_i and R_j are merged if they satisfy a similarity criterion based on the median of each region. Then, during the segmentation process, before the fitness evaluation, the centroids are updated each time an element is added into a region, using the following expression:

$$centroid(x, y) = (\frac{\sum_{I \epsilon A} x_i}{\mid A \mid}, \frac{\sum_{I \epsilon A} y_i}{\mid A \mid}), \tag{3}$$

where x, y are the pixel coordinates in the image and $\mid A \mid$ is the number of elements in the region A. In this way, the genetic information of the chromosome is adjusted to facilitate the classification of image pixels.

3. **Fitness Evaluation.** The fitness function used by the genetic algorithm is based on local and global minimum distance measures between regions, see Fig. 1(c). The local distance is,

$$l = \sqrt{\sum_{i=1}^{c} \sum_{j=1}^{n_i} (d_j - m_i)^2}, \tag{4}$$

where d_j represents the descriptor values and m_i represents the median by region. The global distance is given by

$$g = \sqrt{\sum_{i=1}^{c} (m_i - m)^2}, \qquad \text{where} \qquad m = \frac{1}{N} \sum_{i=1}^{c} m_i \cdot n_i \tag{5}$$

[1] All distances measures are Euclidean.

m is the total median and n_i is the number of elements in the i^{th} region.

The proposed fitness function showed in Fig.1(c) indicates that the distances between the medians of different regions should be maximized in order to maintain the separability between regions. In addition, we have to minimize the distances between elements within a region because the elements nearest to a centroid should belong to the same region. This process is repeated while the maximum number of generations is reached.

4 Experimental Results

In this section, we present the experimental results obtained by EvoSeg and JSEG algorithms tested on seven images. The JSEG algorithm is a state-of-art segmentation algorithm that considers color and texture information in image and video, see [10]. The motivation to use this segmentation algorithm is due to its quality and simplicity, as well as its versatility on high level tasks.

Table 1 presents our experimental results, both algorithms produce good segmentations in images 1(a),(c),(d) and produce a misclassification of pixels in the image 1(g). The *EvoSeg* segmentation in Table 1(b) is better than JSEG, this is due to the fact that JSEG is sensitive to illumination changes. In this case, JSEG could not segment the image because the textures do not present high contrast. Otherwise, *EvoSeg* is robust to the lack of image contrast and almost achieves the complete segmentation of the texture D84. Table 1(e) shows the results for the baboon image, the segmentation done by the JSEG algorithm is superior due to the high contrast of the textures. On the other hand, *EvoSeg* presents problems segmenting the image because the contours of the descriptor image are not well defined. In this case, it would be interesting to add the JSEG color quantization method into the *EvoSeg* algorithm. The airplane shown in Table 1(f) is segmented by *EvoSeg* while JSEG was lost in the illumination. In general, the segmented images by *EvoSeg* are comparable with an algorithm that is state-of-art and it could be more robust by adding the color quantization method before generating the class map.

The images used in our experiments were obtained online at the USC-SIPI Image DataBase[2]. The original images in Table 1(a),(b),(c) and (d) are used by Yoshimura [7].

5 Conclusions

In this paper, we have presented *EvoSeg* as an unsupervised evolutionary segmentation algorithm. *EvoSeg* identifies a good image segmentation from multiple solutions using a genetic algorithm. Our segmentation process depends on the

[2] Signal and Image Processing Institute: http://sipi.usc.edu/database.

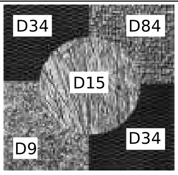
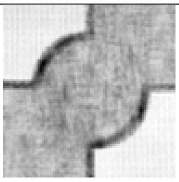
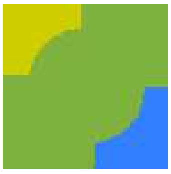
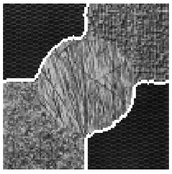
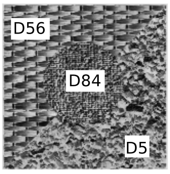
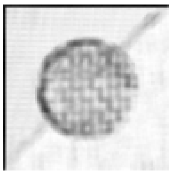
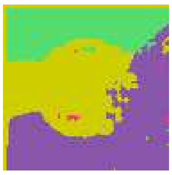
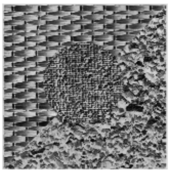
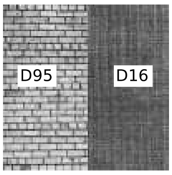
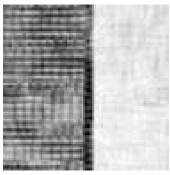
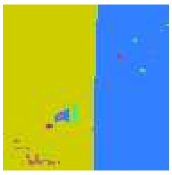
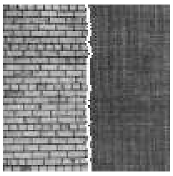
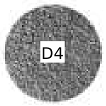
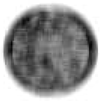
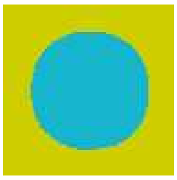
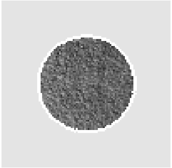
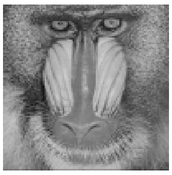
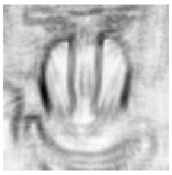
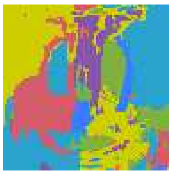
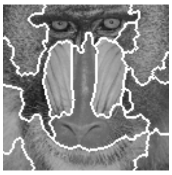
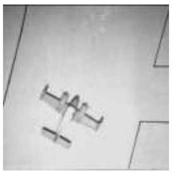
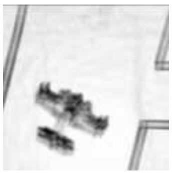
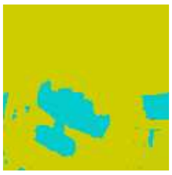
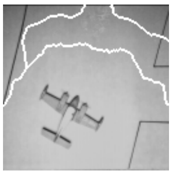
second moment descriptor extracted from the GLCM. If the descriptor image is well defined, it is easier for the algorithm to identify the region boundaries. The hierarchical GA allowed a better distribution of the regions during the segmentation process, and produces a variety of good solutions from which we select the best one. In further research, it would be interesting to analyze the problem as multiobjective and to obtain a pareto of possible segmentations. We compared our results with the JSEG algorithm and both segmentations were comparable.

Acknowledgments This research was funded by Ministerio de Ciencia y Tecnología, and by CONACyT and INRIA through the LAFMI project. First author supported by scholarship 0416442 from CONACyT.

References

1. Suchendra M.Bhandarkar and Hui Zhang. "Image Segmentation Using Evolutionary Computation". IEEE Transactions on Evolutionary Computation. Vol.3 (1). April 1999.
2. Timo Ojala and Matti Pietikäinen. "Unsupervised texture segmentation using feature distributions". Pattern Recognition. Vol.32. pp. 477-486. 1999.
3. Jan Puzicha, Thomas Hofmann and Joachim M. Buhmann. "Histogram Clustering for Unsupervised Segmentation and Image Retrieval". Pattern Recognition Letters. 1999.
4. Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.32 (8). 2000.
5. S. Cagnoni, A.B.Dobrzeniecki, R.Poli and J.C.Yanch."Genetic algorithm-based interactive segmentation of 3D medical images". Image and Vision Computing. Vol.17. pp.88-895. 1999.
6. Bir Bhanu, Sungkee Lee and John Ming. "Adaptive Image Segmentation Using a Genetic Algorithm". IEEE Transactions on Systems, Man, and Cybernetics. Vol.25 (12). December 1995.
7. Motohide Yoshimura and Shunichiro Oe. "Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas. Pattern Recognition. Vol.32. pp.2041-2054. 1999.
8. Leonardo Bocci, Lucia Ballerini and Signe Hässler. "A new evolutionary algorithm for image segmentation". Applications on Evolutionary Computation. LNCS 3449. pp. 264-273. 2005.
9. Olague, G., Cagononi, S., and Lutton, E. (eds). "Introduction to the Special Issue on Evolutionary Computer Vision and Image Understanding". Pattern Recognition Letters. Vol.27 (11). pp.1161-1163, 2006.
10. Yining Deng and B.S.Manjunath. "Unsupervised Segmentation of Color-Texture Regions in Images and Video". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.23 (8). pp. 800-810. August 2001.
11. J.R. Parker. "Algorithms for Image Processing and Computer Vision". John Wiley, New York. 1996.
12. R.M.Haralick, K. Shanmugam and I. Dinstein. "Textural features for image classification". IEEE Transactions on System, Man and Cibernetics. Vol.3 (6). pp.610-621. 1973.

Table 1. *EvoSeg* and JSEG results using different texture images.

	Original Image	Descriptor Image	*EvoSeg* Result	JSEG Result
(a)				
(b)				
(c)				
(d)				
(e)				
(f)				
(g)