Image Segmentation Based on Genetic Algorithms Combination

Vito Di Gesù and Giosuè Lo Bosco

DMA, Università di Palermo, Italy

Abstract. The paper describes a new image segmentation algorithm called *Combined Genetic segmentation* which is based on a genetic algorithm. Here, the segmentation is considered as a clustering of pixels and a similarity function based on spatial and intensity pixel features is used. The proposed methodology starts from the assumption that an image segmentation problem can be treated as a *Global Optimization Problem.* The results of the image segmentations algorithm has been compared with recent existing techniques. Several experiments, performed on real images, show good performances of our approach compared to other existing methods.

1 Introduction

The segmentation of images in *homogeneous* components plays a key role in each recognition system so that its accuracy may influence the performance of the overall recognition procedure. On the other hand, image segmentation depends on the context and it is subjective; the decision process is driven by the goal or the purpose of the specific task considered. The image segmentation problem can be formulated as a clustering based on region properties, in this case, one of the problems is to select more compact features space to allow a better separation between regions. An overview of clustering-based methods can be found in [1]. Shi and Malik [2] have considered a $2D$ image segmentation as a *Graph Partitioning Problem* (GPP) solved by a normalized cut criterion. The method finds approximated solutions by solving a generalized eigenvalue system. Here, we will consider the problem of extracting largest image regions that satisfy uniformity conditions in the *intensity/spatial* domains. The segmentation method here described incorporates and generalizes the approach to image segmentation by genetic algorithm described in [3]. The general design of the proposed segmentation procedure, named in the following *Combined Genetic Segmentation* (CGS), is sketched in Figure 1. The procedure includes two phases. In the first phase, the *Global Optimization Phase* (GOP) the segmentation algorithm *Unsupervised Tree Segmentation* (UTS) is applied. UTS is based on a genetic algorithm called *Genetic Segmentation Procedure* (GSP) that segments the image into a fixed number of regions. Here, the largest and uniform, but not necessarily connected, regions (*candidate segments*) are found. Candidate segments are represented by a graph SG that is described in section 5. In the second phase *Maximal Connected Components* (MCC) are computed from the SG. This phase is realized by the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 352–359, 2005.

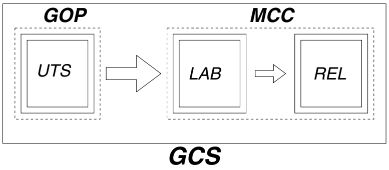


Fig. 1. Sketch of the two phases segmentation algorithm

procedure LAB that updates the labels of the nodes of the SG in order to reach a one to one correspondence between connected components and segments. After the LAB procedure, a generalized relaxation procedure (REL) acts on the SG nodes to embed small components into adjacent largest ones. Section 2 describes the similarity measure between pixels, section 3 gives details on the genetic optimization procedures GSP, section 4 describes the UTS procedure, section 5 describes the algorithm to compute the maximal connected component via the MCC, the generalized relaxation labeling (REL) and the LAB; in section 6 results of the algorithm on real data are shown; concluding remarks are given in section 7.

2 The Similarity Function

In this paper we use a similarity function between two pixels $\underline{x}, \underline{y} \in \mathbf{X}$, that is function of both intensity and spatial features:

$$\delta(\underline{x}, \underline{y}) = a \times d_g(\underline{x}, \underline{y}) + b \times d_E(\underline{x}, \underline{y}) \tag{1}$$

where, $a \geq 0$, $b \geq 0$, and $a + b = 1$. The two terms are defined as follows:

$$d_g(\underline{x}, \underline{y}) = \frac{|g_{\underline{x}} - g_{\underline{y}}|}{max(g_{\underline{x}}, g_{\underline{y}})}, \quad d_E(\underline{x}, \underline{y}) = \frac{1}{|C_r(\underline{x})|} \sum_{\underline{z} \in C_r(\underline{x}), z \neq x} \frac{|g_{\underline{z}} - g_{\underline{y}}|}{max(g_{\underline{z}}, g_{\underline{y}})} \tag{2}$$

where $C_r(\underline{x})$ is the neighborhood of the pixel $\underline{x}$ and r is the interaction radius. The assumption that pixels may interact at a given distance r is motivated mainly by physiological reasons and it has been also used in [2]; moreover, it allows us to reduce the algorithm complexity.

The component d_g is a metric and it takes into account the closeness of the pixels intensity, while the component d_E considers the attraction effect due to local pixels. Actually, it increases if the intensities of neighbors of $\underline{x}$ differ from the intensity of $\underline{y}$. Note that, d_E is not a metric in fact it is easy to show that $d_E(\underline{x}, \underline{y}) \neq d_E(\underline{y}, \underline{x})$ because in general $C_r(\underline{x}) \neq C_r(\underline{y})$. Therefore, δ is a distance functions only if $b = 0$.

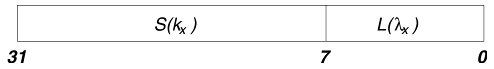


Fig. 2. Chromosome representation of a pixel

3 Genetic Segmentation Procedure

The formalization of the image segmentation problem by different perspective (e.g. as clustering, as graph partitioning) consists in solving it as a global optimization problem which is NP complete [2,4,5].

Genetic algorithms (*GA*'s) [6,7] are random algorithms that can often find a global optimal solution; however, local optima could be reached in consideration of their random nature. They have been already used to face clustering problems [8].

The *GSP* groups the input data in K segments, it consists in the assignment procedure of each pixel $\underline{x}$ to a given segment class chosen from a random population of assignments. *GSP* is the kernel of the *UTS* segmentation procedures described in section 4.

One of the most important step of a genetic algorithm is the data coding that may influence its efficiency (precision of the solutions, computational complexity). In our case, each pixel, (i_x, j_x, g_x), of $\mathbf{X}$, is coded by a binary string (the *chromosome*) α_{i_x,j_x} of 32 bits that represent position (i_x, j_x) in the 24 most significant bits and the pixel-label, λ_x, in the 8 least significant bits. The value λ_x identifies the cluster where x belongs. The genetic operators are applied to the entire binary content of a chromosome and this corresponds to some modification on the binary content part which represents a pixel position or a label. The resulting chromosome could identify a new label assignment for the same pixel (in the case only the least significant bits is modified) or a label assignment for a different pixel. This corresponds to a whole modification of the grouping. The functions used to code the pixel label and positions are:

$$L(\lambda_x) = (\frac{2^8 - 1}{K}) \times \lambda_x \qquad S(k_x) = (\frac{2^{24} - 1}{n \times m}) \times k_x \tag{3}$$

where $k_x = i_x \times m + j_x$ and K is the number of clusters. It follows that $\alpha_{i_x,j_x} \equiv \alpha_{k_x} \equiv b_{31}b_{30}...b_0$, with $b_{31}b_{30}....b_8 = S(k_x)$ and $b_7b_6....b_0 = L(\lambda_x)$ (See Figure 2). A generic chromosome α, represents a pixel in position $S^{-1}(\alpha)$, assigned to a cluster $\lambda = L^{-1}(\alpha)$.

The chosen coding is compact and easy to handle, because it represents the information in four bytes; moreover, it allows us to represent up to 256 segments per image and image-size up to 4096 × 4096. These values are adequate in most real applications. The accuracy of the approximated solution of the global optimization problem depends also on the choice of the *fitness function.* In our case, the optimization is related to the minimization of the internal variance of each image segment. The evolution of the *GA* will be driven by a *fitness function*, f, that is computed on the basis of the similarity function, δ, defined in the previous section. Each segment P_j is characterized by the mean value, mv_j, of the gray levels:

$$mv_j = \frac{\sum_{\alpha \in A_j} \mathbf{X}(S^{-1}(\alpha)))}{|P_j|} \tag{4}$$

where $A_j = \{\alpha | L^{-1}(\alpha) = j\}$. The fitness function is computed between a given chromosome $\alpha = (S(k), L(\lambda))$ and the corresponding segment P_λ as follows:

$$f(\alpha) = \delta(\mathbf{X}(S^{-1}(\alpha)), mv_{L^{-1}(\alpha)}) \tag{5}$$

To evolve the system the single point crossover and the bit mutation are used and their combination is denoted by Γ. Random labels in $\{0, 1, ..., K\}$ are assigned to the initial population of chromosomes. The population at the iteration t is denoted by $P(t) = \{\alpha_1(t), \alpha_2(t), ..., \alpha_N(t)\}$ where $N = n \times m$ is the size of the input image. The application of Γ generates the population $\Gamma(P(t)) = \{\beta_1(t), \beta_2(t), ..., \beta_N(t)\}$ where $\beta_r = \Gamma(\alpha_r)$. The *selection process* is performed by selecting for each $1 \leq r \leq N$ the best chromosome between α_r and β_r. Γ and the selection process are applied until the *halting condition* $|Var_{t-1} - Var_t| \leq \phi$ is satisfied, where $Var_t = \sum_k^K \sigma_t(k)$ is the total internal variance, $\sigma_t(k)$ is the variance of the cluster k at the iteration t and $\phi \geq 0$. The condition $|Var_{t-1} - Var_t| = 0$ is not usually reached and the value of ϕ is determined by the heuristics $\phi \approx \varepsilon \times Var_{t-1}$, with $\varepsilon \leq 0.1$. Finally, the halting condition takes also into account if a maximum of iterations T has been reached. From previous definitions it follows that the genetic algorithm GSP may be sketched as it follows:

```
**Procedure** $GSP((S_1, S_2, \ldots, S_K), T, p_c, p_m)$
   Set up a population of chromosome $P(0) = \{\alpha_1(0), \alpha_2(0), ....., \alpha_N(0)\}$
   by using $(S_1, S_2, \ldots, S_K)$;
   t=0;
   **repeat**
      Apply $\Gamma$ to current population $P(t)$;
      Build population $P(t+1)$ by selecting the *best chromosome*
      from $P(t)$ and $\Gamma(P(t))$;
      t = t + 1;
   **until** $(|Var_{t-1} - Var_t| \leq \phi) \bigvee (t > T)$;
   Set up $(S_1, S_2, \ldots, S_K)$ by using $P(t-1)$;
```

Note that the sets $(S_1, S_2, \ldots, S_K)$ represent an initial distribution of labels (segments) that could be user defined or totally randomized. The fitness function used in the GSP is that defined by Equation 5. Parameters a and b determine the influence of the pixel intensity and the local distribution of labels. Probabilities p_c and p_m establish the frequency of the *crossover* and *mutation* operators. Typical values used in the our experiments are $p_c = 0.7$ and $p_m = 0.01$. K indicates the initial maximum number of clusters and it may be reduced from the genetic process.

4 Unsupervised Tree Segmentation

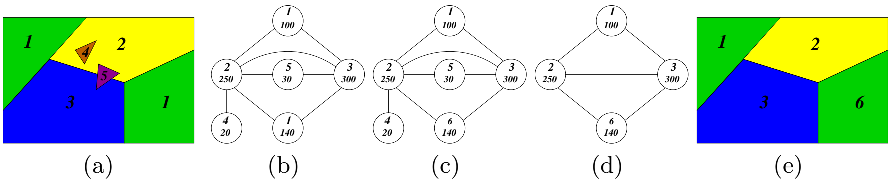
The *UTS* algorithm builds a balanced binary tree, where the apex represents the whole image, **X**, internal nodes are temporary segments, and leaves are final image segments. The UTS uses the genetic procedure *GSP* in the sense that on each internal node of the tree it is applied with $K = 2$. The binary split of each node continues until an internal *uniform condition* is reached or the tree has reached the maximum depth $log_2(K_{max})$ where K_{max} is the maximum number of clusters. The *UTS* algorithm can be sketched as it follows:

```
Procedure UTS($\mathbf{S}$,$K_{max}$,$d$)
   if $\neg uniform(\mathbf{S}) \bigwedge (d < log_2(K_{max}))$ then
      split randomly $\mathbf{S}$ in $(S', S'')$;
      $GSP((S', S''), T, pc, pm)$;
      $UTS(S'$,$d + 1)$;
      $UTS(S''$,$d + 1)$;
   else
      $K = K + 1$;
      $S_K = S$;
   end
```

The procedure *uniform*(**S**) returns a value that is true if the variance of the gray levels in the set **S** is greater than a given threshold. In most cases the threshold is set to $\overline{g} + \sqrt{\overline{g}}$; where $\overline{g}$ is the mean gray value of the pixels in A. This assumption is a good approximation in the case of Poisson's distribution. The computed segments $(S_1, S_2, \ldots, S_K)$ are represented by the leaf nodes on the tree.

5 Maximal Connected Components Computation

The *GOP* procedure assigns a label to each pixel of **X**. The resulting segmentation can be represented as a connected planar labelled graph, named the segmentation graph (SG). Nodes of SG correspond to connected homogeneous regions (pixels in the same segment); the pair (λ, w), assigned to each node, represents the segment-label (λ) and the number of pixels (w) in the corresponding



(a) (b) (c) (d) (e)

Fig. 3. (a) *GOP* segmentation; (b) corresponding *SG*; (c) *SG* after the application of *LAB* procedure; (d) *SG* after the application of *REL* procedure; (e) final segmentation.

connected region. Note that, segments constituted by disconnected regions corresponds to more than one node with the same λ, but in general different w. However, two adjacent nodes must have different λ, because they represent, by construction, adjacent regions that can't correspond to the same segment. The Maximal Connected Component (MCC) algorithm is performed in two phases:

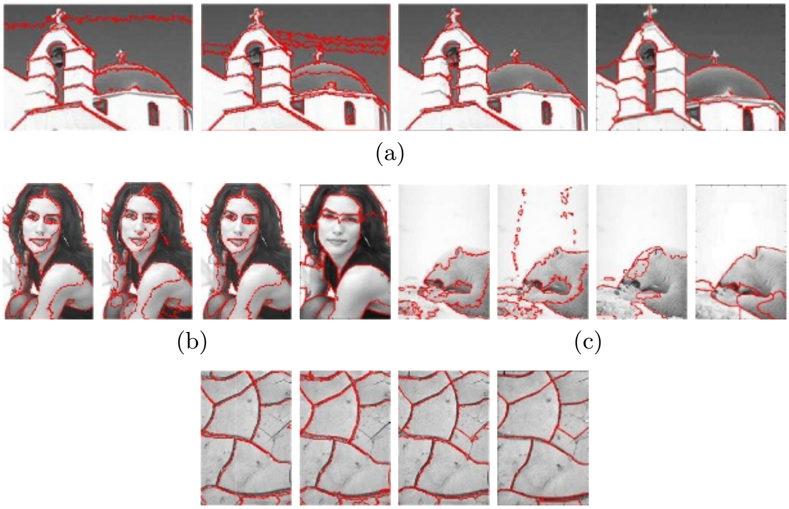
- LAB. During this phase, all λ's are recomputed to assign a different label to each node. After this phase each node of the SG corresponds to a maximal connected region of homogeneous pixels.
- REL. During this phase, nodes of SG, such that $w < \psi$, are embedded into adjacent greater nodes, in particular the small nodes are embedded in the neighbor which has the maximum cardinality greater than ψ all over the neighborhood of w. The threshold can be set to $\psi = \overline{w} - \sqrt{\overline{w}}$; where $\overline{w}$ is the mean value of the node weights. As a result small segments are eliminated.

For example, Figure 3a represents the result of a segmentation after UTS, Figure 3b represents the corresponding graph, SG. Figure 3c shows SG after the LAB procedure, finally Figure 3d shows SG after the application of REL. Figure 3e shows the final segmentation.

6 Experimental Results

The experiments has been performed on images belonging to Corel data set (http://elib.cs.berkeley.edu/photos/corel/) in order to compare the accuracy of CGS with other three segmentation algorithms : the first based on a C-means clustering (*C-means*), the second on a single link clustering (*Single-link*) and the last on a normalized cut minimization criterion (GPS). The first two algorithm has been implemented on the R.S.I.S. [9] image analysis system, the last one is the method by Shi and Malik [2]. In the following we provide segmentation examples of four images: church (192×128 pixels), face (128×192), bear (128×192) and fragments (128×192). The results of CGS, *C-means*, *Single-link*, and GPS segmentation techniques on the test images are reported in Figure 4.

The performance evaluation of segmentation methods is not an easy task because the expected results are subjective and they depend on the application. One evaluation procedure could be the comparison of the method against a robust and well experimented one, but this choice is not always feasible. Here, we evaluate the performances of all the algorithm by comparing machine and human segmentation. The comparison has been performed between the automatic segmentation and the segmentation deriving from the evaluation of an odd number (5) of persons. The human segmentation is obtained taking the maximum intersection of the regions selected by each person separately. This procedure has been suggested by the approach described and tested experimentally in [10]. In the following, Seg_k and S denote the *k-th* segment retrieved by humans and machine respectively, $|Seg_k|$ and $|S|$ denote the corresponding size, $\sharp\mathrm{agr}_k$ is the largest pixel intersection between Seg_k and S. A measure of agreement between human and automatic segmentation can been evaluated as follows:



(a)

(b)

(c)

(d)

Fig. 4. For each image the different segmentation results, from left to right : *CGS*; *C-means*; *Single-link*; *GPS*

Table 1. Segmentation Comparison

Method	η	T_{CPU}
		sec.
CGS	0.74	154
C-means	0.70	97
Single-link	0, 77	220
GPS	0, 73	300

$$\eta = \frac{1}{K} \sum_{k=1}^{K} \beta_k \times \frac{\sharp \mathrm{agr}_k}{max(|Seg_k|, |S|)}$$

where β_k is the percentage of pixels of the whole image included in the k-th segment. This measure error is related to the global consistence error introduced in [10]. Table 1 shows the average value of the agreement indicator (η) and the mean CPU time (T_{CPU}) for each segmentation method. All the segmentation algorithms have been implemented in C++ under windows system; the CPU time is referred to an INTEL PENTIUM III 1GHz. The results in the table show that all algorithms provide a comparable accuracy. *Single-link* has the best accuracy, while *CGS* provides a good compromise in time/accuracy. It must be pointed out that *GPS* [2] is more complete since it is also able to segment texture based images. The convergence of the *CGS* depends on the convergence of the components *GSP*. Their convergence has not been fully proved; nevertheless the *GA* converges in all our experiments.

7 Concluding Remarks

The image segmentation problem has been treated here as a complex global optimization problem that, because of its computational complexity, does not admit tractable exact solutions. Thus, we have proposed a segmentation algorithm based on a genetic algorithm to find an approximated solution of the problem. The comparison with three published algorithms shows its good performance. The agreement between natural and automatic grouping has been also done using more than one individual. Finally, the overall method seems to indicate a faster convergence to the correct solution.

References

1. A.K. Jain and P. Flynn, "Image Segmentation Using Clustering", *Advances In Image Understanding*: A Festschrift for Azriel Rosenfeld, K. Bowyer and N. Ahuja (eds.), pp. 65–83, IEEE Computer Society Press, 1996.
2. J. Shi, J. Malik, "Normalized Cuts and Image Segmentation", *IEEE transactions on PAMI*, Vol.22, N.8, pp.1–18, IEEE Computer Society Press, 2000.
3. G. Lo Bosco, "A genetic algorithm for image segmentation", *in Proc. of ICIAP 2001*, pp. 262–266, Palermo, Italy, IEEE computer society press, 2001.
4. M. Garey, D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman and Company, 1979.
5. R. Horst and P.M. Pardalos (eds.), Handbook of Global Optimization, Kluwer Academic Publishers, 1995.
6. J.H. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, MI, 1975.
7. D.E.Goldberg, "Genetic algorithms in search, optimization and machine leraring", Addison Wesley, 1989.
8. L.O. Hall, I.B. Ozyurt, J.C. Bezdek, "Clustering with a genetically optimized approach", *IEEE Transactions on Evolutionary Computation*, Vol.3, N.2, pp. 103–112, IEEE Computer Society Press, 1999.
9. R.S.I.S., EPRI TR-11838, WO-5144-03 & WO-8632-01, Palo Alto, October, 1999.
10. D. Martin, C. Fowlkes, D. Tal, J. Malik. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms", *in Proc. of ICCV01*, Vol.2, pp. 416–425, IEEE Computer Society press, 2001.