An Evolutionary Approach for Image Segmentation

Alessia Amelio amelio@icar.cnr.it
Institute for High Performance Computing and Networking (ICAR),
National Research Council of Italy (CNR), Via P. Bucci 41C, 87036 Rende, Italy

Clara Pizzuti pizzuti@icar.cnr.it
Institute for High Performance Computing and Networking (ICAR),
National Research Council of Italy (CNR), Via P. Bucci 41C, 87036 Rende, Italy

doi:10.1162/EVCO_a_00115

Abstract

The paper explores the use of evolutionary techniques in dealing with the image segmentation problem. An image is modeled as a weighted undirected graph, where nodes correspond to pixels, and edges connect similar pixels. A genetic algorithm that uses a fitness function based on an extension of the normalized cut criterion is proposed. The algorithm employs the locus-based representation of individuals, which allows for the partitioning of images without setting the number of segments beforehand. A new concept of nearest neighbor that takes into account not only the spatial location of a pixel, but also the affinity with the other pixels contained in the neighborhood, is also defined. Experimental results show that our approach is able to segment images in a number of regions that conform well to human visual perception. The visual perceptiveness is substantiated by objective evaluation methods based on uniformity of pixels inside a region, and comparison with ground-truth segmentations available for part of the used test images.

Keywords

Image segmentation, evolutionary computation, genetic algorithms, normalized cut.

1 Introduction

Image segmentation is a central problem in pattern recognition (Duda and Hart, 1973), and of prior importance for facing more complex higher level problems such as object recognition. It also constitutes an important phase for content analysis and image understanding. Image segmentation aims at partitioning an image into regions satisfying a homogeneity criterion that takes into account one or more pixel features, such as color, texture, intensity, and so on. Each region corresponds to a different object, and neighboring regions should be significantly dissimilar with respect to the homogeneity function.

The problem of generating a meaningful segmentation can be formulated as an optimization problem, where the objective to optimize formalizes the concept of the homogeneity criterion adopted. Because of the computational complexity of finding an exact solution to this problem, many different heuristics have been proposed

Manuscript received: January 18, 2013; revised: June 10, 2013 and September 30, 2013; accepted: October 23, 2013.

which guarantee a good local optimum and feasible computation times. Most of these methods can be classified in two main types (Zhang, 1997): boundary detection-based approaches (Sahoo et al., 1988; Helterbrand, 1996), and region clustering-based approaches (Chen et al., 1998; Pappas, 1992; Chen and Zhang, 2004). However, methods representing an image as a graph (Zahn, 1971; Urquhart, 1982; Wu and Leahy, 1993; Shi and Malik, 2000; Felzenszwalb and Huttenlocher, 2004) proved competitive both in terms of efficiency and segmentation quality.

This paper presents a graph-based algorithm, GeNCut (Genetic Normalized Cut), which deals with the image segmentation problem by using evolutionary computation (Holland, 1975). An image is represented as a weighted undirected graph, and a genetic algorithm (GA; Goldberg, 1989) optimizing a fitness function is run in order to find an optimal partitioning of the graph. This division corresponds to a segmentation of the image in real-world objects. The fitness function is an extension of the normalized cut concept of Shi and Malik (2000) that allows for a simultaneous k-way partitioning of an image, without the need to fix the number k of divisions beforehand, which is typical of many image segmentation approaches. In fact, because of the adopted representation of individuals, k is automatically determined by the optimal value of the objective function.

The introduction of a new mutation operator that considers both the spatial closeness of pixels and their similarity, biases the genetic approach to assign pixels to the foremost region, while avoiding the method to get stuck at local optima. An extensive and thorough experimentation on images coming from different domains, and an objective evaluation based on an internal criterion and two ground-truth assessment methods, show that GeNCut is competitive with the method of Shi and Malik (2000) in partitioning natural and human scenes in meaningful objects.

The paper is organized as follow. In the next section a description of the state of the art approaches to image segmentation is given, with particular emphasis to evolutionary-based methods. In Section 3, the problem of image segmentation is defined, together with its formalization as a graph partitioning problem, and a description of the adopted homogeneity measure. Section 4 introduces the concept of normalized cut and the fitness function used by GeNCut. Section 5 explains the genetic representation and operators employed. Section 6 describes the evaluation measures used to quantitatively and objectively compare our method with that of Shi and Malik (2000). Section 7 presents the experimental results. Finally, Section 8 summarizes the approach and outlines future work.

2 Related Work

The image segmentation problem has been intensively investigated with the use of several computational techniques, and many different methods have been proposed. A broad classification divides the existing methods in two main categories (Zhang, 1997): boundary detection-based approaches and region cluster-based approaches. The former approaches search for closed boundary contours by detecting pixels that sensibly change in intensity. Boundaries of objects are obtained by linking such pixels in contours. The main limitation of these approaches is that a threshold value must be set in order to produce a continuous contour (Sahoo et al., 1988; Helterbrand, 1996). Region cluster-based methods group similar close pixels into clusters. Many of these approaches use fuzzy C-means (Chen and Zhang, 2004) or the K-means method, such as Chen et al. (1998) and Pappas (1992). A drawback of these methods is that the number of clusters must be predetermined, which implies that a user should know in advance the number of relevant regions that should be detected in the image to be segmented. In order to overcome these limitations, methods based on representing an image as a graph have been introduced. One of the earliest graph-based methods dates back over 40 years and it is based on the minimum spanning tree (MST) of a graph (Zahn, 1971). Zahn's method gives a weight to edges on the base of differences between pixel intensities, and breaks edges having a weight greater than a fixed threshold. Improvements on the policy of edge breaking were proposed by Urquhart (1982). Wu and Leahy (1993) presented a method based on finding minimum cuts in a graph. The cut criterion adopted tries to minimize the similarity between pixels assigned to different regions, but it has the drawback of being biased to obtain small components. To avoid unnatural cuts of small groups of isolated nodes, Shi and Malik (2000) introduced a new measure of dissimilarity between two groups, called the normalized cut. More recently, Felzenszwalb and Huttenlocher (2004) defined a measure of evidence of the boundary between two regions by considering both the differences of intensity across the boundary and among neighboring pixels within a region. Arbeláez et al. (2011) reduced the problem of image segmentation to that of contour detection by transforming contours into a hierarchy of regions.

Because of the abundance in the literature of image segmentation techniques, an exhaustive description of the topic is not possible. A recent survey on graph-based approaches to image segmentation can be found in Peng et al. (2013). Since GeNCut is based on the evolutionary computation paradigm, next we review some of the evolutionary-based proposals.

In recent years, much effort has been put forth in the definition of effective evolutionary-based approaches for solving complex problems of computer vision (Cagnoni et al., 2008a, 2008b). Evolutionary techniques have been successfully applied to segment images in difficult application domains, such as the medical context (Maulik, 2009; Smith and Cagnoni, 2011), or for finding optimal parameter configurations of segmentation algorithms (Bhanu et al., 1995). Bhanu et al., for example, defined a framework to adapt the segmentation process when image characteristics change due to varying environmental conditions, and use genetic algorithms to search the hyperspace of segmentation parameters in order to maximize segmentation quality criteria. Genetic algorithms have also been used by Andrey and Tarroux (1998) to help in estimating parameters for a segmentation approach based on Markov random fields, or combined by Bhandarkar and Zhang (1999) with stochastic annealing algorithms to optimize cost functions for image segmentation. A survey on the application of genetic algorithms for image enhancement and segmentation can be found in Paulinas and Ušinskas (2007). In the following, some of the most recent proposals are described in more detail.

Many of the approaches use a representation of the image based either on the cluster centers or on the label of the cluster that a pixel is assigned to. A color image segmentation algorithm based on an evolutionary approach has been proposed by Halder and Pathak (2011). Each individual is a sequence of cluster centers and the cost function is the inverse of the sum of Euclidean distances of each point from its respective cluster centers. In order to determine the most appropriate number k of clusters, the algorithm is repeated for values of k from 2 to $k_{\max}$. The choice of the best k is done by computing a cluster validity index based on inter and intra distances between clusters, while the value of $k_{\max}$ must be given as input to the algorithm.

Jiao (2011) proposed an evolutionary image texture classification algorithm where the individuals are the cluster representatives, and the total distance between the pixels and the corresponding centroids is optimized. The distance between a pair of pixels is the value of the shortest path between them in the undirected weighted graph representing the image. In the same paper, the author defines a memetic image segmentation approach, where a genetic algorithm is applied on a set of regions previously extracted from a watershed segmentation in order to refine or merge partitions into clusters. In this case, each gene of a chromosome is the cluster label of the corresponding pixel. The association of the regions with the clusters is evolved by optimizing the total distance between the pixels and the corresponding centroids. In the evolutionary image texture classification algorithm, the number of clusters must be fixed a priori; whereas in the memetic image segmentation approach, an approximate initial number is obtained by using watershed segmentation, and then a final local search procedure merges regions to obtain the optimal number of clusters.

Merzougui et al. (2011) proposed an evolutionary-based image segmentation technique where the individuals are the components of the cluster centers and the fitness is the mean distance between the pixels and the centroids. In order to determine the optimal number of clusters, a criterion based on separability and compactness of clusters is first applied.

Lai and Chang (2009) presented a hierarchical structure of the chromosome, composed by control genes, representing a partitioning in regions, and parametric genes, containing the representative gray levels of each region. The goal is to optimize a fitness function that is the sum of the distances between the gray level of each pixel and the representative gray level of its region. The number of control genes, as stated by the authors, is a soft estimate of the upper bound of the number of regions.

Pérez and Olague (2007) proposed EvoSeg, a hierarchical GA algorithm that uses texture features to extract homogeneous regions in an image. The algorithm first performs a statistical texture analysis to obtain representative data of the image texture, then applies a GA to obtain an optimal segmented image. A chromosome consists of the set of possible centroids that an image can have; and a binary array determines the actual centroids used in the segmentation process. Each pixel is assigned to the region for which the distance in the descriptor space is minimal and the distance from the centroid is lower than a fixed threshold. Finally, two regions are merged if they satisfy a similarity criterion. An interactive EvoSeg extension was presented in Pérez et al. (2009) to allow the algorithm to use external information provided by a user.

DiGesú and Bosco (2005) introduced an image segmentation algorithm where each chromosome represents the position and the label of the region where the pixel is located. The fitness function is defined on the similarity value and the spatial proximity between a pixel (chromosome) and the mean gray value of its corresponding region.

A different evolutionary approach was proposed by Bocchi et al. (2005). The method aims to colonize a bidimensional world by a number of populations that represent the regions contained in an image.

Because of the representation adopted, one of the main problems of the described approaches is the determination of the number of regions. Although different criteria are suggested to fix this number beforehand, the GA often cannot change this number while executing. The method we propose in the following dynamically computes the number of regions that optimizes the fitness function.

3 Problem Definition

Let I be an image constituted by a set of pixels. The problem of segmenting the image I consists of subdividing I into a finite number of nonoverlapping and connected regions $R_1 \ldots R_k$ such that

$$I = \bigcup_{i=1}^{k} R_i, \quad R_i \cap R_j = \emptyset, i \neq j.$$

A homogeneity measure must be defined over pixels that takes into account characteristics such as intensity, color, and texture. Pixels belonging to the same region are similar on the basis of the homogeneity measure adopted, while adjacent regions are significantly dissimilar with respect to the same features.

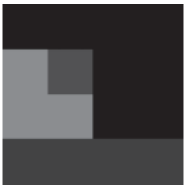
An image I can be represented as a weighted undirected graph $G = (V, E, w)$, where V is the set of n nodes of the graph G, E is the set of edges, and $w : E \rightarrow \mathcal{R}$ is a function that assigns a value to graph edges. Each node corresponds to a pixel in the image, and a graph edge (i, j) connects two pixels i and j, provided that these two pixels satisfy some suitably defined property that takes into account both pixel characteristics and spatial distance. The weight $w(i, j)$ associated with an edge (i, j) represents the likelihood that pixels i and j belong to the same image region and provides a similarity value between i and j. The higher the value of $w(i, j)$, the more likely it is that the two pixels are members of the same region. Let W be the adjacency weight matrix of the graph G. Thus W_{ij} contains the weight $w(i, j)$ if the nodes i and j are connected, and zero otherwise. Depending on the method adopted to compute the weights, any two pixels may or may not be connected.

In our approach, we employed the intervening contour method described in Leung and Malik (1998) and in Cour et al. (2005). In this framework, given a generic pixel, the magnitude of the orientation energy at that pixel is considered. If the maximum image edge magnitude along a straight line connecting the two pixels i and j in the image plan is large, then a deep change and, consequently, an intervening contour is present, indicating that the two pixels do not belong to the same segment. Hence, the weight $w(i, j)$ between these pixels will be low. On the other hand, if the image edge magnitude is sufficiently weak, which usually happens in a region that is unchanging in brightness, the affinity between the two pixels will be very high. More formally, the weight $w(i, j)$ between the two pixels i and j is computed as:

$$\mathrm{w(i, j)} = \begin{cases} e^{-\max_{x \in \mathrm{line}(i,j)} ||\mathrm{Edge}(x)||_2^2 / 2a^2} & \text{if } ||X(i) - X(j)||_2 < r, \quad i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $a = (\max_{y \in I} ||\mathrm{Edge}(y)||) \times \sigma$, Edge($x$) is the image edge strength at position x, I is the image plan, line(i, j) is a straight line between i and j, X(i) is the spatial location of pixel i, r is a distance threshold, and σ is the image edge variance. In order to compute the weight between pixels i and j, image edges across various scales are considered. The scale space is a set of images constructed by a progressive smoothing of the input image. It corresponds to gradually reducing the image resolution. The level of smoothing of the image is called image scale (Lindeberg, 1994). Small scales correspond to finer image details, while larger scales represent coarser image details. A multiscale approach is suitable to obtain accurate edges in image segmentation. Consequently, the input image is filtered by using a filter bank, where each filter is associated with a specific scale factor. After that, the magnitude of the orientation energy is recovered in correspondence with each image pixel location by considering the contribution of each image filter response (the contribution of the different scales) at that pixel. The obtained image edge magnitude map containing the contribution of multiple scales is adopted in the intervening contour framework for computing the affinity values among the pixels.

Figure 1(a) depicts a toy image I composed of 4 × 4 pixels and Figure 1(b) shows the matrix corresponding to I where each pixel is labeled with increasing numbers from 1 to 16. If we set the distance threshold $r = 2$, the spatial neighbors to consider for any pixel i are those for which the norm $||X(i) - X(j)||_2 < 2$. Thus, for example, if we use the Euclidean distance as the norm, the spatial Euclidean distance between pixel 1,



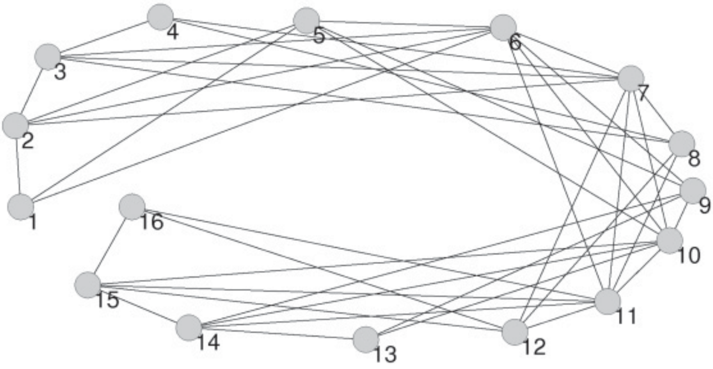
(a)

$$\left\| \begin{array}{cccc} \mathbf{1} & 2 & 3 & 4 \\ 5 & \mathbf{6} & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array} \right\| \qquad ||X(1) - X(6)||_2 = \sqrt{(1-2)^2 + (1-2)^2} = \sqrt{2}$$

(b)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	**0**	1	0	0	0.1	0.7	0	0	0	0	0	0	0	0	0	0
2	1	**0**	1	0	0.1	0.7	0.9	0	0	0	0	0	0	0	0	0
3	0	1	**0**	1	0	0.7	0.9	0.9	0	0	0	0	0	0	0	0
4	0	0	1	**0**	0	0	0.9	0.9	0	0	0	0	0	0	0	0
5	0.1	0.1	0	0	**0**	0.5	0	0	1	1	0	0	0	0	0	0
6	0.7	0.7	0.7	0	0.5	**0**	0.6	0	0.5	0.5	0.5	0	0	0	0	0
7	0	0.9	0.9	0.9	0	0.6	**0**	1	0	0.1	0.8	0.8	0	0	0	0
8	0	0	0.9	0.9	0	0	1	**0**	0	0	0.8	0.8	0	0	0	0
9	0	0	0	0	1	0.5	0	0	**0**	1	0	0	0.4	0.4	0	0
10	0	0	0	0	1	0.5	0.1	0	1	**0**	0.2	0	0.4	0.4	0.4	0
11	0	0	0	0	0	0.5	0.8	0.8	0	0.2	**0**	1	0	0.3	0.3	0.3
12	0	0	0	0	0	0	0.8	0.8	0	0	1	**0**	0	0	0.3	0.3
13	0	0	0	0	0	0	0	0	0.4	0.4	0	0	**0**	1	0	0
14	0	0	0	0	0	0	0	0	0.4	0.4	0.3	0	1	**0**	1	0
15	0	0	0	0	0	0	0	0	0	0.4	0.3	0.3	0	1	**0**	1
16	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0	0	1	**0**

(c)



(d)

Figure 1: (a) An image I composed of 4×4 pixels. (b) The matrix corresponding to I where each pixel is labeled with increasing numbers from 1 to 16, and computation of the spatial Euclidean distance between pixel 1 and pixel 6 at position (1,1) and (2,2), respectively. (c) Adjacency weight matrix of size 16×16, associated with I. Weights are computed by using Equation (1) with $r = 2$. (d) Graph representation corresponding to image I having 16 nodes.

at position (1,1), and pixel 6 at position (2,2), can be computed as $||X(1) - X(6)||_2 = \sqrt{(1-2)^2 + (1-2)^2} = \sqrt{2}$. Since this distance is lower than r, in the adjacency weight matrix W, $W(1,6) = W(6,1) > 0$. Figure 1(c) shows the adjacency weight matrix of size 16×16, associated with I, whose weights are computed by using Equation (1), and Figure 1(d) shows the corresponding graph representation of I.

4 Graph Partitioning for Image Segmentation

In recent years, many different criteria have been defined to partition a graph representing an image into nonoverlapping connected components. Shi and Malik (2000) introduced a dissimilarity measure, called the normalized cut, to divide a graph into two subgraphs, that was revealed successful for image segmentation. The concept of normalized cut is an extension of the notion of cut proposed by Wu and Leahy (1993) that avoids the problem, noted by the same authors, of generating small sets of isolated nodes. Given a partition of a graph $G = (V, E, w)$ in two disjoint sets of nodes A and B, the cut between A and B is defined as

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j). \tag{2}$$

Shi and Malik (2000) pointed out that the cut value diminishes when small sets of isolated nodes are generated. Thus, a disassociation measure that takes into account the total edge weight connecting two partitions, has been introduced. Let

$$\text{assoc}(A, V) = \sum_{i \in A, t \in V} w(i, t) \tag{3}$$

be the total connection weight from nodes in A to all the nodes in V, then the normalized cut is defined as

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}. \tag{4}$$

Shi and Malik formalize the problem of minimizing the normalized cut as a generalized eigenvalue problem and compute an optimal partition by using the eigenvector with the second smallest eigenvalue. A main limitation of this method is that the number k of desired partitions must be fixed beforehand. Two extensions of the approach to k-way partitioning are also proposed. The first method recursively partitions the groups obtained in the previous step by using the eigenvector with the second smallest eigenvalue. The second method exploits the top n eigenvectors, that is, those with the n smallest eigenvalues, and the K-means clustering algorithm. To perform simultaneous k-way partitioning, the objective function is modified as:

$$\text{Ncut}_k = \sum_{i=1}^{k} \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(A_i, V)}. \tag{5}$$

We now introduce an extension of the concept of normalized cut that is used as the criterion to partition a graph in a generic number k of regions. Note that the value of k in our approach must not be fixed in advance; instead it is determined by the optimal value of the objective function. Let $G = (V, E, w)$ be the graph representing an image, W its adjacency matrix, and $P = \{R_1, \ldots, R_k\}$ a partition of G in k clusters. For a generic cluster $R_i \in P$, let

$$c_r = \sum_{i \in R, j \notin R} W_{ij} \tag{6}$$

$$m_r = \sum_{i \in R, j \in R} W_{ij} \tag{7}$$

$$m = \sum_{i \in V, j \in V} W_{ij} \tag{8}$$

be respectively the sum of weights of edges on the boundary of R_i, the sum of weights of edges inside R, and the total graph weight sum. The weighted normalized cut, WNCut, measures, for each cluster in P, the fraction of the total edge weight to all the nodes in the graph, and summarizes the contribution of each of them as:

$$\text{WNCut} = \sum_{r=1}^{k} \frac{c_r}{m_r + c_r} + \frac{c_r}{(m - m_r) + c_r} \tag{9}$$

Because of the affinity measure w defined in the previous section, and the relationship between (cut) and (assoc) formalized in Shi and Malik (2000), more uniform regions can be obtained with low cut values between the subgraphs representing the regions and the rest of the graph. This implies that low values of WNCut should be preferred.

In order to more deeply explore the difference between the normalized cut and the weighted normalized cut function, we observe that

$$c_r = \text{cut}(A_i, V - A_i) \tag{10}$$

$$m_r + c_r = \text{assoc}(A_i, V) \tag{11}$$

$$(m - m_r) + c_r = \text{assoc}(V - A_i, V) \tag{12}$$

Consequently, Equation (9), which is the fitness function used by GeNCut to perform a simultaneous k-way partitioning, can be rewritten as:

$$\text{WNCut} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(A_i, V)} + \frac{\text{cut}(A_i, V - A_i)}{\text{assoc}(V - A_i, V)} \tag{13}$$

It is worth observing that the second term in Equation (13) is missing in Shi and Malik's Equation (5).

5 An Evolutionary Method to Segment Images

In this section, a description of the algorithm GeNCut is reported, along with the representation we used and the variation operators we adopted.

5.1 Genetic Representation

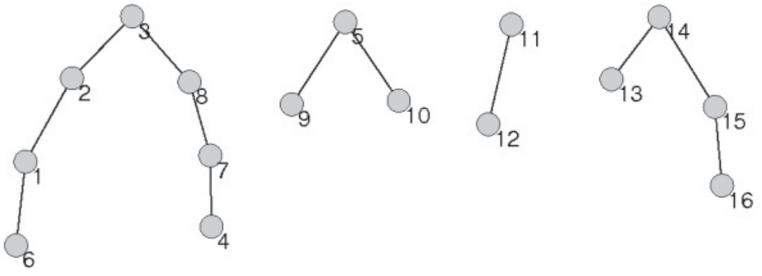
The genetic algorithm uses the locus-based adjacency representation proposed in Park and Song (1989), and adopted in Handle and Knowles (2007) for data clustering and in Pizzuti (2012) to detect communities in complex networks. In this graph-based representation, an individual of the population consists of n genes $g_1, \ldots, g_n$ and each gene can assume values in the range $\{1, \ldots, n\}$. Genes represent nodes of the graph $G = (V, E, w)$ modeling an image, and a value j assigned to the ith gene is interpreted as a link between the pixels i and j. This means that in the clustering solution, i and j will belong to the same region.

Figures 2–3 show the locus-based representation of two randomly generated individuals from the graph of Figure 1(d). The first individual J_1 represents a segmentation of the toy image I in Figure 1(a) into the four regions $\{1, 2, 3, 4, 6, 7, 8\}$, $\{5, 9, 10\}$, $\{11, 12\}$, and $\{13, 14, 15, 16\}$. The division corresponding to the second individual J_2 in Figure 3(a), instead, is composed of the five regions $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$, $\{13, 14, 15, 16\}$, $\{11, 12\}$, and $\{9, 10\}$.

The kind of crossover operator we adopted is uniform crossover. Given two parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 0 from the first parent, and the genes where the vector is a 1 from the second parent, and combines the genes to form the child. Figure 4 shows an example

pixel	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
neighbor	2	3	8	7	9	1	4	7	5	5	12	11	14	13	14	15

(a)



(b)

$$\left\| \begin{matrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{matrix} \right\|$$

(c)

Figure 2: (a) Locus-based representation of individual J_1. (b) Corresponding segmentation of the graph G of Figure 1(d) in four segments composed by the pixels $r_1 = \{1, 2, 3, 4, 6, 7, 8\}$, $r_2 = \{5, 9, 10\}$, $r_3 = \{11, 12\}$, and $r_4 = \{13, 14, 15, 16\}$. (c) Image partitioning.

of crossover between individuals J_1 and J_2 of Figures 2–3. A random mask is generated (Figure 4(a)) and the application of crossover operator produces the offspring reported in Figure 4(b). This new individual corresponds to the segmentation of the graph G of Figure 1(d) in four segments composed of the pixels $\{1, 2, 3, 4, 5, 6, 7, 8\}$, $\{9, 10\}$, $\{11, 12\}$, and $\{13, 14, 15, 16\}$.

The mutation operator, analogous to the initialization process, randomly assigns to a node i, chosen in a random way, one of its neighbors. Figure 5 shows an example of mutation where node 9 is chosen at random and its neighbor 5 is substituted by the other neighbor 6. This operation provokes a new segmentation in three components constituted by $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $\{11, 12\}$, and $\{13, 14, 15, 16\}$.

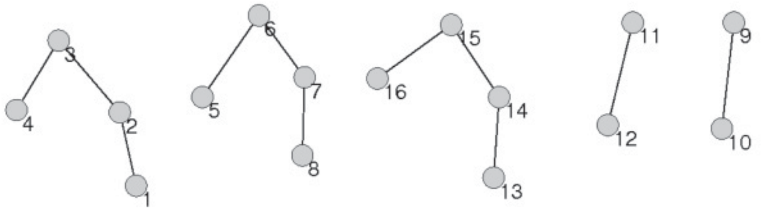
5.2 h-Neighborhood

The genetic operators have a requirement to determine the neighbors of each node. In our approach, we introduced the concept of neighbors of a node by taking into account not only the spatial closeness, but also the pixel affinity. In more detail, given a generic node i in the graph, let

$$w^h_{nn_i} = \{w^1, \ldots, w^h \mid w^1 \geq, \ldots, \geq w^h\} \tag{14}$$

pixel	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
neighbor	2	3	4	3	6	7	8	7	10	9	12	11	14	13	14	15

(a)



(b)

$$\left\Vert\begin{matrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_2 & r_2 \\ r_3 & r_3 & r_4 & r_4 \\ r_5 & r_5 & r_5 & r_5 \end{matrix}\right\Vert$$

(c)

Figure 3: (a) Locus-based representation of another individual J_2. (b) Corresponding segmentation of the graph G of Figure 1(d) in five segments composed of the pixels $r_1 = \{1, 2, 3, 4\}$, $r_2 = \{5, 6, 7, 8\}$, $r_3 = \{9, 10\}$, $r_4 = \{11, 12\}$, and $r_5 = \{13, 14, 15, 16\}$. (c) Image partitioning.

be the first h highest weights of row i in the weight adjacency matrix W. The h nearest neighbors of i, denoted as nn_i^h, are then defined as

$$nn_i^h = \left\{ j \mid w(i, j) \in w_{nn_i}^h \right\}. \tag{15}$$

nn_i^h is thus the set of those pixels whose spatial distance from i is less than the distance threshold r of Equation (1), and having maximum similarity with i. Note that the number of nearest neighbors of i can be greater than h, since if more neighbors of i have the same maximum similarity value, this value is counted only once. Figure 6 shows the computation of the first $h = 2$ nearest neighbors of each pixel. Consider, for example, pixel 3. As reported in the weight adjacency matrix of Figure 1(c), the neighbors and associated weights of node 3 are (2, 1), (4, 1), (6, 0.7), (7, 0.9), and (8, 0.9). The first two highest weights are 1 and 0.9 and occur twice, thus the first $h = 2$ nearest neighbors of node 3 are {2, 4, 7, 8}.

This definition of nearest neighbors guarantees the choice of the most similar neighbors during the initialization process, which results in a bias of the effects of the mutation operator toward the most similar neighbors; thus it contributes to improving the results of the method.

5.3 Algorithm

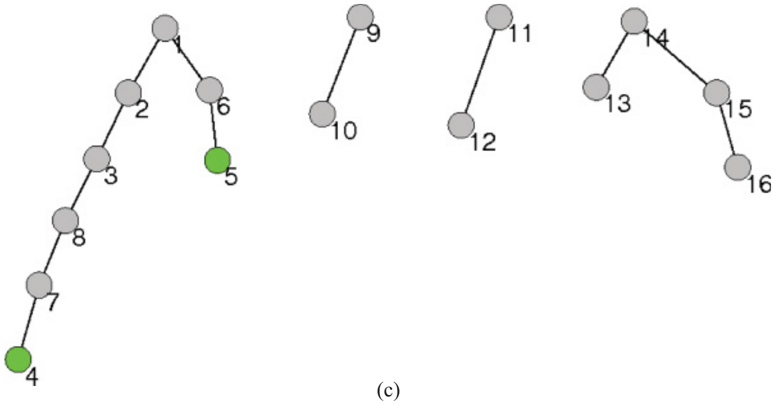
The segmentation procedure illustrated in Figure 7 is composed of three main phases: (1) a preprocessing step for enhancing the input image; (2) a segmentation phase to

mask	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0

(a)

pixel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
neighbor	2	3	8	7	6	1	4	7	10	9	12	11	14	13	14	15

(b)



(c)

$$\left\| \begin{matrix} r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{matrix} \right\|$$

(d)

Figure 4: Crossover between the two individuals J_1 and J_2 of Figures 2–3. (a) Random mask generated. (b) Offspring generated from J_1 and J_2. (c) Corresponding segmentation of the graph G of Figure 1(d) in four segments composed of the pixels $r_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $r_2 = \{9, 10\}$, $r_3 = \{11, 12\}$, and $r_4 = \{13, 14, 15, 16\}$. (d) Image partitioning.

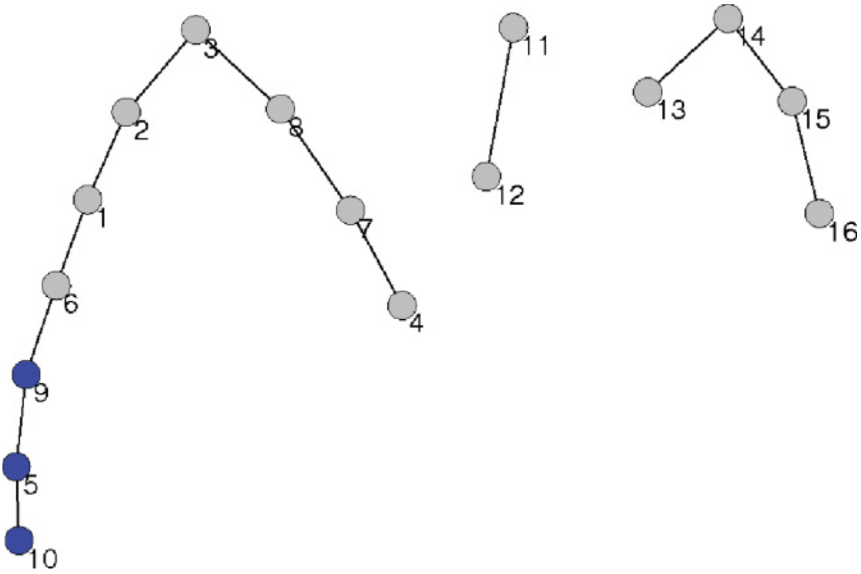
generate the segmentation map by running GeNCut; and (3) a final postprocessing activity for detecting the image boundaries from the segmentation map.

In particular, the preprocessing phase shown in Figure 7 consists of removing noise, through a Gaussian smoothing (step 1.1), or emphasizing fine details inside the images, through a spatial sharpening filtering (step 1.2). After that, a Sinc (Lanczos3) interpolation method is used to reduce the size of the test image (Duchon, 1979). The size reduction is such that it is able to preserve the main image object contours; thus, the main features of the original input image are preserved.

The preprocessed image is transformed in a gray scale image, the graph modeling the image is generated and given as input to GeNCut. As shown in Figure 7, the algorithm creates a random initial population (step 2.2.1), and, for a fixed number of

allele	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
gene	2	3	8	7	9	1	4	7	6	5	12	11	14	13	14	15

(a)



(b)

$$\begin{Vmatrix} r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_1 & r_1 \\ r_1 & r_1 & r_2 & r_2 \\ r_3 & r_3 & r_3 & r_3 \end{Vmatrix}$$

(c)

Figure 5: (a) Mutation on individual J_1 of Figure 2 where 9 is the randomly chosen node. Its neighbor 5 is substituted with 6. (b) The new generated individual corresponds to the segmentation of graph G of Figure 1(d) in three segments composed of $r_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $r_2 = \{11, 12\}$, and $r_3 = \{13, 14, 15, 16\}$. (c) Image partitioning.

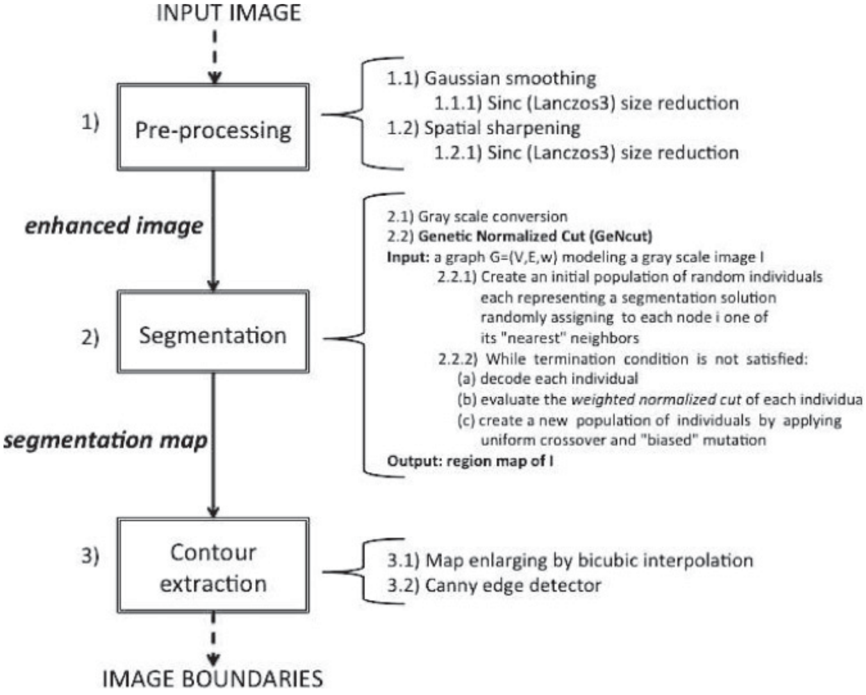
generations, decodes each individual to obtain the division of the graph in connected components (step 2.2.2(a)), evaluates the fitness of each individual (step 2.2.2(b)), applies crossover and mutation operators, and generates a new population (step 2.2.2(c)). The output of GeNCut is a segmentation map in which each pixel is assigned the label of the cluster it belongs to, representing the different regions discovered by the algorithm.

At this point, a contour extraction activity has to be performed. In particular, the segmentation map is enlarged to obtain the same size of the original input image, by

$1 \rightarrow \{(2; 1.0), (6; 0.7)\}$
$2 \rightarrow \{(1; 1.0), (3; 1.0), (7; 0.9)\}$
$3 \rightarrow \{(2; 1.0), (4; 1.0), (7; 0.9), (8; 0.9)\}$
$4 \rightarrow \{(3; 1.0), (7; 0.9), (8; 0.9)\}$
$5 \rightarrow \{(6; 0.5), (9; 1.0), (10; 1.0)\}$
$6 \rightarrow \{(1; 0.7), (2; 0.7), (3; 0.7), (7; 0.6)\}$
$7 \rightarrow \{(2; 0.9), (3; 0.9), (4; 0.9), (8; 1.0)\}$
$8 \rightarrow \{(3; 0.9), (4; 0.9), (7; 1.0)\}$
$9 \rightarrow \{(5; 1.0), (6; 0.5), (10; 1.0)\}$
$10 \rightarrow \{(5; 1.0), (6; 0.5), (9; 1.0)\}$
$11 \rightarrow \{(7; 0.8), (8; 0.8), (12; 1.0)\}$
$12 \rightarrow \{(7; 0.8), (8; 0.8), (11; 1.0)\}$
$13 \rightarrow \{(9; 0.4), (10; 0.4), (14; 1.0)\}$
$14 \rightarrow \{(9; 0.4), (10; 0.4), (13; 1.0), (15; 1.0)\}$
$15 \rightarrow \{(10; 0.4), (14; 1.0), (16; 1.0)\}$
$16 \rightarrow \{(11; 0.3), (12; 0.3), (15; 1)\}$

(a)

Figure 6: Nearest neighbors nn_i^h for $1 \leq i \leq 16$ and $h = 2$.



1.1) Gaussian smoothing
    1.1.1) Sinc (Lanczos3) size reduction
1.2) Spatial sharpening
    1.2.1) Sinc (Lanczos3) size reduction

```
2.1) Gray scale conversion
2.2) Genetic Normalized Cut (GeNcut)
Input: a graph G=(V,E,w) modeling a gray scale image I
        2.2.1) Create an initial population of random individuals
                each representing a segmentation solution
                randomly assigning to each node i one of
                its "nearest" neighbors
        2.2.2) While termination condition is not satisfied:
            (a) decode each individual
            (b) evaluate the weighted normalized cut of each individua
            (c) create a new population of individuals by applying
                uniform crossover and "biased" mutation
Output: region map of I
```

3.1) Map enlarging by bicubic interpolation
3.2) Canny edge detector

Figure 7: Main steps of the segmentation process and pseudocode of GeNCut.

adopting a bicubic interpolation (step 3.1), which computes a new pixel value inside the image as the weighted average of the 16 pixels closest to the specified input coordinates (Keys, 1981). This value is then associated with the output coordinates. Finally, the region boundaries are extracted from the segmentation map by using the Canny edge detector (step 3.2; Canny, 1986), implemented in MATLAB. The σ parameter, representing the standard deviation of the Canny edge detector, has been tuned in order to guarantee the extraction of regularly-shaped region boundaries.

5.4 Computational Complexity

The computation time of GeNCut depends, apart from the fitness function and operator computations, on the population size p, and the number of generations g it uses. At each generation, crossover needs $O(n)$ time, where n is the number of graph nodes, mutation $O(1)$ time, while fitness computation is composed of two parts: decoding of an individual, and WNCut (Equation (9)). Decoding can be efficiently performed in $O(n \log n)$ time (Cormen et al., 2007). To compute WNCut we need to consider, for each obtained segment, c_r (Equation (6)), m_r (Equation (7)), and m (Equation (8)), which is the sum of edge weight. Since we have to consider for each node all its neighbors, the time complexity of computing each of these formulas is $O(\hat{m})$, where $\hat{m}$ is the number of edges. Fitness computation can thus be realized in $O(n) + O(n \log n) + O(\hat{m})$ time. Therefore, the overall complexity of GeNCut is $O(gp \times (n \log n + \hat{m}))$.

6 Evaluation Methods

Evaluating the results of image segmentation algorithms, in order to determine whether one method is more accurate than another, is a critical aspect in image and computer vision research. This difficulty is mainly due to the lack of general criteria, independent of the application domain, that often influences the concept of good segmentation. In the last few years, many performance measures have been proposed to assess the quality of segmentation methods. In a recent survey, Zhang et al. (2008) classify segmentation evaluation methods in subjective and objective methods. In the former approaches, a human evaluator analyzes the segmented image visually. This implies that different evaluators can produce distinct estimates of the same image. Objective evaluators are divided into subcategories; in particular, the supervised and unsupervised methods are distinguished on whether a ground-truth reference image is available or not. A ground-truth image, also called the gold standard, is an image manually segmented by a domain expert. The unsupervised methods evaluate an image on the basis of some criteria that the image must satisfy. Thus, the image provides a quantitative measure that expresses the desired features.

For evaluating the performance of our algorithm, and comparing it with Shi and Malik's method, we employ three evaluation approaches, of which two are based on information theory. The first approach is unsupervised and measures the uniformity of pixel characteristics; the other two are supervised methods and evaluate the misclassified pixels versus the reference gold standard images. In the following, a brief description of the evaluation measures we adopted is reported.

6.1 Entropy-Based Evaluation Method

The unsupervised method we employ for evaluating the results of our segmentation algorithm was introduced by Zhang et al. (2004). It is based on the consideration that a good segmentation should maximize the uniformity of pixels inside the regions and minimize the uniformity across the regions. Consequently, a measure based on the concept of entropy, representing the disorder inside a region, could be a suitable evaluator: low entropy values should correspond to a good segmentation.

Let I be an image segmented in k regions $\{R_1, \ldots, R_k\}$, S_I the size of I, that is, the number of pixels, also referred to as the area of I, and $S_j = \mid R_j \mid$ the area of R_j. Let v be the feature used to assess uniformity inside a region, and ${V_j}^{(v)}$ the set of all possible values of feature v in region R_j. The entropy for R_j is then defined as:

$$H_v(R_j) = -\sum_{a \in {V_j}^{(v)}} \frac{L_j(a)}{S_j} \log \frac{L_j(a)}{S_j} \tag{16}$$

where a is a generic value of feature v in region j, and $L_j(a)$ is the number of pixels in region R_j having the value a for feature v.

The expected entropy across all regions, weighted proportionally to their areas, is the expected region entropy:

$$H_r(I) = \sum_{j=1}^{k} \frac{S_j}{S_I} H_v(R_j). \tag{17}$$

This measure is low when the segmentation generates very uniform regions in terms of luminance or other features. Consequently, it is not robust to over-segmentation, since entropy decreases when increasingly smaller regions are extracted. Therefore, a new measure, layout entropy, was introduced:

$$H_l(I) = -\sum_{j=1}^{k} \frac{S_j}{S_I} \log \frac{S_j}{S_I}. \tag{18}$$

The layout entropy represents the number of bits per pixel for specifying to which region a pixel belongs, given a fixed segmentation. It is high when a huge number of regions is present; thus, a combination of H_r and H_l is introduced to balance the effects of oversegmenting and undersegmenting when assessing the effectiveness of a segmentation method:

$$E = H_l(I) + H_r(I). \tag{19}$$

This final measure is not minimized when the image is over-segmented, because the layout entropy will be very high. On the other hand, a segmentation with very few regions will be penalized, because the expected region entropy will be high.

6.2 Ground-Truth Evaluation Methods

The other two methods we use to assess the effectiveness of our approach are ground-truth based. In this case, given a test image, a human-traced segmentation is available. Thus, each pixel is associated with a label representing the region the pixel belongs to.

6.2.1 Normalized Mutual Information

Normalized mutual information (NMI; Cover and Thomas, 1991) allows for the measurement of the similarity between a manual segmentation S_b of a reference image I and the segmentation S_a obtained by a method. It is used to evaluate the closeness between the two segmentations. The higher the similarity value, the better the quality of the segmentation obtained by the method under evaluation. The NMI considers the confusion matrix C, where rows correspond to the c_b regions of the gold standard segmentation, and columns to the c_a regions detected by the method. It is defined as follows:

$$\mathrm{NMI}(S_a, S_b) = -2\frac{\alpha}{(\beta + \gamma)} \tag{20}$$

$$\alpha = \sum_{i=1}^{c_a} \sum_{j=1}^{c_b} C_{i,j} \log\left(\frac{nC_{i,j}}{C_i C_j}\right) \tag{21}$$

$$\beta = \sum_{i=1}^{c_a} C_i \log\left(\frac{C_i}{n}\right) \tag{22}$$

$$\gamma = \sum_{j=1}^{c_b} C_j \log\left(\frac{C_j}{n}\right) \tag{23}$$

where n is the size of the test image in number of pixels, C_i is the sum of the elements of row i in C and C_j is the sum of the elements of column j in C.

6.2.2 Probabilistic Rand Index

Normalized mutual information allows for the comparison of two segmentations of the same image. However, as observed in Unnikrishnan et al. (2007), when multiple ground-truth segmentations are available for the same image, the comparison should be made against all the manually obtained segmentations. To this end, Unnikrishnan and Hebert (2005) and Unnikrishnan et al. (2007) introduced the Probabilistic Rand Index (PRI) as an extension of the concept of Rand Index (Rand, 1971), employed to assess clustering methods. Given a set $\{S_1, \ldots, S_T\}$ of ground-truth segmentations of an image I consisting of n pixels, and a test segmentation S, the Probabilistic Rand Index is defined as:

$$\mathrm{PRI}(S, \{S_1, \ldots, S_T\}) = \frac{1}{H} \sum_{i<j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})] \tag{24}$$

where c_{ij} denotes the event that pixels i and j have the same label, p_{ij} is its probability, and $H = n * (n - 1)/2$ is the total number of pixel pairs.

7 Experimental Results

In this section, we present the results of GeNCut on different kinds of images and compare the performance of our algorithm in partitioning these scenes in meaningful objects with the segmentation obtained by the algorithm of Shi and Malik (2000), referred to in the following as NCut.

The GeNCut algorithm was written in MATLAB 7.14 R2012a, using the Genetic Algorithms and Direct Search Toolbox 2. In order to set parameter values, a trial and error procedure was employed and then parameter values giving good results for the benchmark images were selected. Thus, we set the crossover rate to 0.9, the mutation rate to 0.2, the elite reproduction to 10% of the population size, and used the roulette selection function. The population size was 100, the number of generations 100. The value h of nearest neighbors to consider was fixed to either 1 or 2. As already pointed out, this does not mean that the number of neighbors is 1 or 2, but that the first (and second) most similar neighbors are taken into account for the initialization and mutation operators. The fitness function, however, is computed on the overall weight matrix. For all the datasets, the statistical significance of the results produced by GeNCut was checked by performing a t-test at the 5% significance level. The p values returned are very small, thus the significance level is very high, since the probability that a segmentation computed by GeNCut could be obtained by chance is very low.

The version of the NCut software that we used is written in MATLAB and it is available at http://www.cis.upenn.edu/~jshi/software/. The weight matrix of each image is computed in the same way for both methods, and, as already described in Section 3, it is based on the intervening contour framework by fixing $r = 10$, number of scales $= 3$, number of orientations $= 4$, and $\sigma = 0.1$.

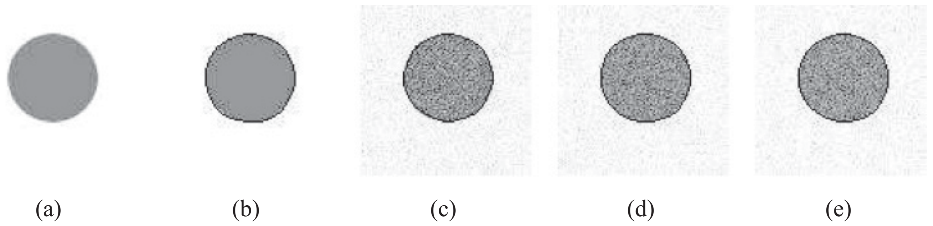


Figure 8: (a) A synthetic image and (b) its segmentation. Segmentation results on the synthetic image altered with increasing Gaussian noise having zero mean with increasing variance (c) 0.01; (d) 0.02; and (e) 0.03.

7.1 Evaluation on Synthetic Images

Before starting the evaluation of GeNCut by using the measures described in the previous section, we want to show the capability of the method in segmenting artificial images for which the ground-truth segmentation is obvious, and its sensitivity, when random noise is controlled, is added to an image. To this end we considered the artificial image reported in Figure 8(a), and three other variations of the same image obtained by adding Gaussian noise having zero mean with increasing variance 0.01, 0.02, and 0.03. The segmentation results are reported in Figures 8(b–e). From the figures, it is clear that GeNCut is able to correctly detect the image objects when noise is present.

7.2 Entropy-Based Evaluation

The entropy-based function described in Section 6.1, the layout entropy, can be used for the assessment of uniformity inside a region. To this end, analogous to Shi and Malik (2000), Leung and Malik (1998), and Cour et al. (2005), we consider changes in intensity as the feature to choose for discriminating image object contours. This means that a segmentation with regions having high uniformity in brightness will be preferred. Consequently, we chose the luminance as the feature in the entropy-based evaluation method: uniformity of regions in terms of luminance is evaluated, and segmentations with regions that are flat in brightness will exhibit low entropy values.

Since NCut needs a number k of clusters, in order to compare in terms of entropy, we executed the algorithm by setting the number k of segments to the same number of clusters found by GeNCut. Then, we evaluate the entropy value on the segmentation results of both NCut and GeNCut to perform the comparison between the two techniques.

For both methods and for each image, we also show the segmentation results of GeNCut and NCut by depicting the contours of the regions obtained by the two approaches. For a more clear visualization, we show three images. The first one is the original image. The second one reports the boundary lines of the segmentation obtained by GeNCut on the original color or grayscale image. The third image reports similar contour lines of the segmentation found by NCut. In order to formally validate the visual results, the unsupervised evaluation index E (Equation (19)), described in the previous section, is computed. In particular, we report the entropy-based evaluation function E for a set of 24 images for which the ground-truth segmentations are not known. The dataset includes satellite and medical images (Sections 7.2.1 and 7.2.2), pictures from nature, human faces, and images from archeology and painting (Section 7.2.3).

Table 1: Entropy evaluated for GeNCut and NCut on various kinds of natural and human images. *E* (Equation (19)) represents entropy, *nc* the number of segments found by GeNCut.

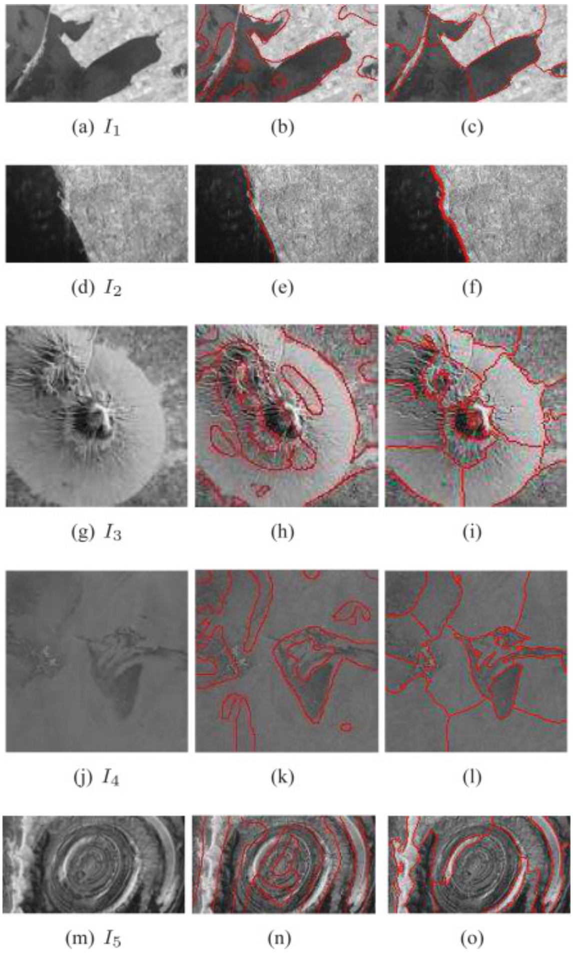
			GeNCut	NCut
		nc	E	E
Satellite	I_1	12	0.0737	0.3265
	I_2	2	0.7451	0.7544
	I_3	18	0.0331	0.1011
	I_4	15	0.0222	0.1627
	I_5	13	0.1484	0.2670
Medical	I_1	12	0.1430	0.5135
	I_2	12	0.1807	0.2708
	I_3	2	0.1044	0.2551
	I_4	15	0.0594	0.3365
	I_5	17	0.0536	0.0737
	I_6	13	0.0257	0.4008
	I_7	13	0.0453	0.3208
	I_8	13	0.0619	0.4137
	I_9	14	0.3280	0.3807
	I_{10}	14	0.2564	0.2636
Miscellaneous	I_1	16	0.2087	0.2861
	I_2	12	0.1177	0.2778
	I_3	16	0.0809	0.2526
	I_4	12	0.0685	0.2111
	I_5	13	0.0562	0.3697
	I_6	14	0.0529	0.2375
	I_7	28	0.2327	0.2836
	I_8	12	0.1691	0.2422
	I_9	18	0.1816	0.3054

The algorithms were executed 10 times on each test image and average entropy values are presented in Table 1. Standard deviation values of entropy are very low; thus, we did not report them.

7.2.1 Satellite Images

Figure 9 shows the segmentation results of GeNCut on five earth-observation satellite images, available at the website http://www.asi.it/. They are captured from COSMO-SkyMed, a constellation composed of four satellites equipped with synthetic aperture radar operating in the X-band, a project of the Italian Space Agency (ASI). In particular, the first column of Figure 9 shows the five images: the first is the Strait of Kerch, Nero Sea, the second image illustrates the Pompei area, Italy, the third depicts the Mount Egmont, New Zealand, the fourth and the last are, respectively, oil spills in the Gulf of Mexico and the geological structure of Richat, Mauritania.

The second column of Figure 9 shows the segmentation found by GeNCut for the original image appearing in the same row, while the third column shows the partitioning obtained by NCut. A first visual analysis highlights that by using our algorithm we are able to perform a segmentation that is more precise in capturing the real shape of the objects. NCut, in fact, with an equal number of segments, finds a partitioning that is



(a) I_1 (b) (c)

(d) I_2 (e) (f)

(g) I_3 (h) (i)

(j) I_4 (k) (l)

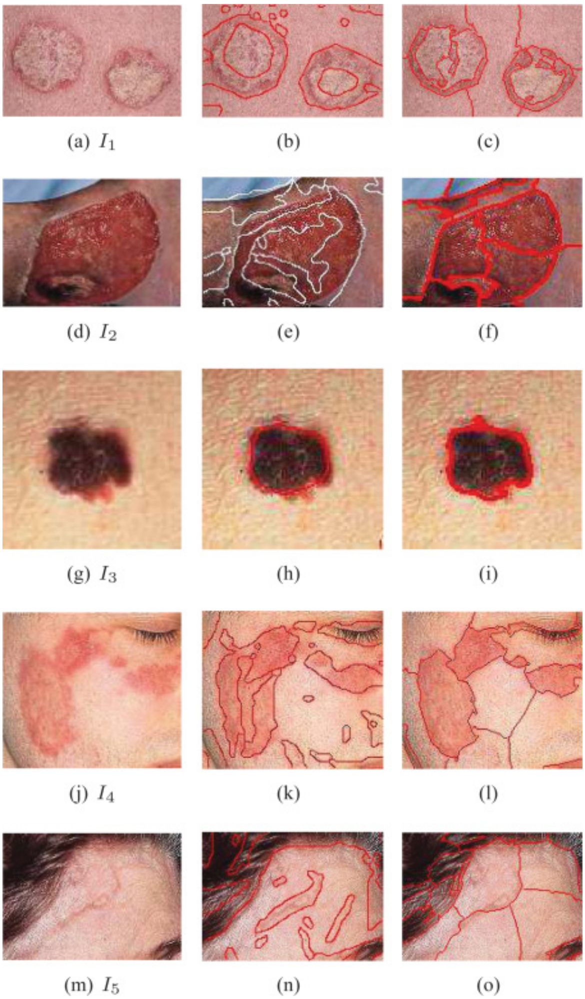
(m) I_5 (n) (o)

Figure 9: Segmentation on satellite images. From the first to the last image we have: Strait of Kerch, Nero Sea (image 1), Pompei area, Italy (image 2), Mount Egmont, New Zealand (image 3), oil spills in Gulf of Mexico (image 4) and geological structure of Richat, Mauritania (image 5). For each line, the original image together with the segmentation result of GeNCut and Ncut are presented.

not able to distinguish some inner parts of the original image. This can be observed especially in images with rounded objects inside, such as for the third and fifth images in the figure, but also in the fourth image, representing oil spills in the sea. In fact, some oil spills which are perfectly identified by our approach, are completely disregarded by NCut. The better visual quality is quantitatively confirmed by the entropy values computed for these satellite images and reported in Table 1. Lower entropy values found by using GeNCut prove that our technique is competitive with the NCut approach of Shi and Malik.

7.2.2 Medical Images

We also tested the effectiveness of our approach on 10 skin lesion images. Analysis of this kind of images is useful to pursue content-based image retrieval in dermatology



(a) I_1 (b) (c)

(d) I_2 (e) (f)

(g) I_3 (h) (i)

(j) I_4 (k) (l)

(m) I_5 (n) (o)

Figure 10: Segmentation on skin lesion medical images. From the first to the last image we have: psoriasis (image 1), cutaneous vasculitic ulcer (image 2), melanoma (image 3), lupus (image 4), and temporal arteritis (image 5). For each line, the original image together with the segmentation result of GeNCut and NCut are presented.

for skin cancer detection, but, more in general, to speed up the diagnosis process in dermatology. In particular, Figure 10 and Figure 11 show images extracted from the On-line Atlas of Dermatology and Rheumatology, University of Padua (Italy), available at http://www.archrheumatol.net/atlas/, and images from DermAtlas, Johns Hopkins University, available at http://dermatlas.med.jhmi.edu/derm/. Various kinds of pathologies are depicted in Figures 10–11: psoriasis (images 1 and 8), melanoma (images 3 and 10), cutaneous vasculitic ulcer (image 2), lupus (images 4 and 7), temporal arteritis (image 5), association between plaque morphea and erythema annulare centrifugum (image 6), lentigo solar (image 9).

Looking at the segmentation results of GeNCut in Figures 10–11, we can observe that the main features of the medical images are well detected, while the NCut

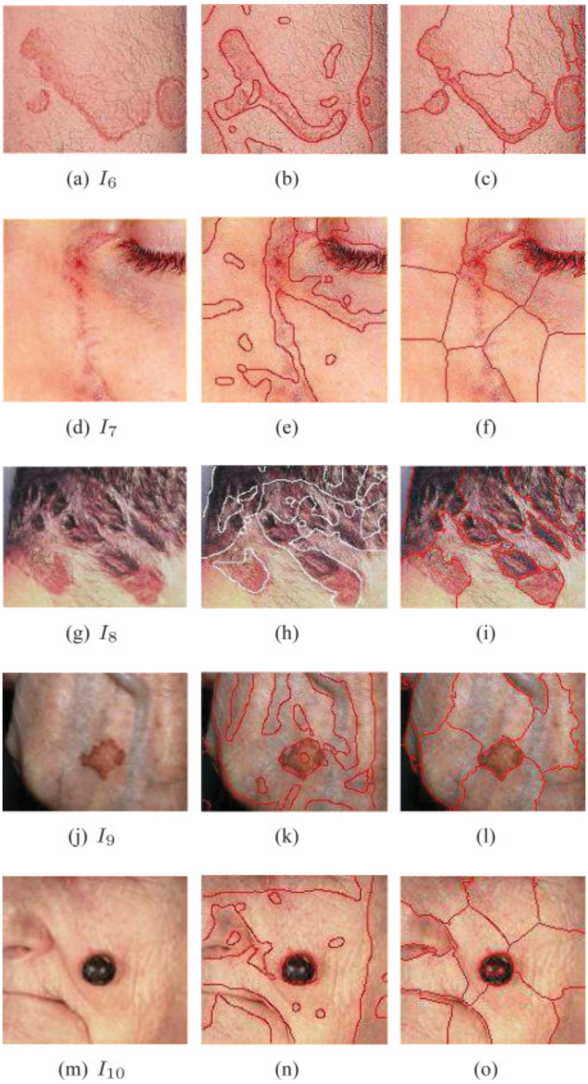


Figure 11: Segmentation on skin lesion medical images. From the first to the last image we have: an association between plaque morphea and erythema annulare centrifugum (image 6), lupus (image 7), psoriasis of the scalp (image 8), lentigo solar (image 9), melanoma (image 10). For each line, the original image together with the segmentation result of GeNCut and NCut are presented.

approach sometimes fails to capture the meaningful parts of the images. For example, in image 5, the temporal arteritis is correctly detected by GeNCut, while NCut wrongly traces boundaries in correspondence of the pathology. Another kind of problem is present in images 4, 5, 6, 7, and 10. In these cases, NCut, instead of capturing the details of the images, partitions the healthy part of the skin, like the cheek in image 4, or the brow in image 5. GeNCut, in contrast, correctly detects the illness boundaries for all the considered images. As already pointed out, NCut needs to know the number of segments to detect; thus, if it receives the true number of segments, it is able to find

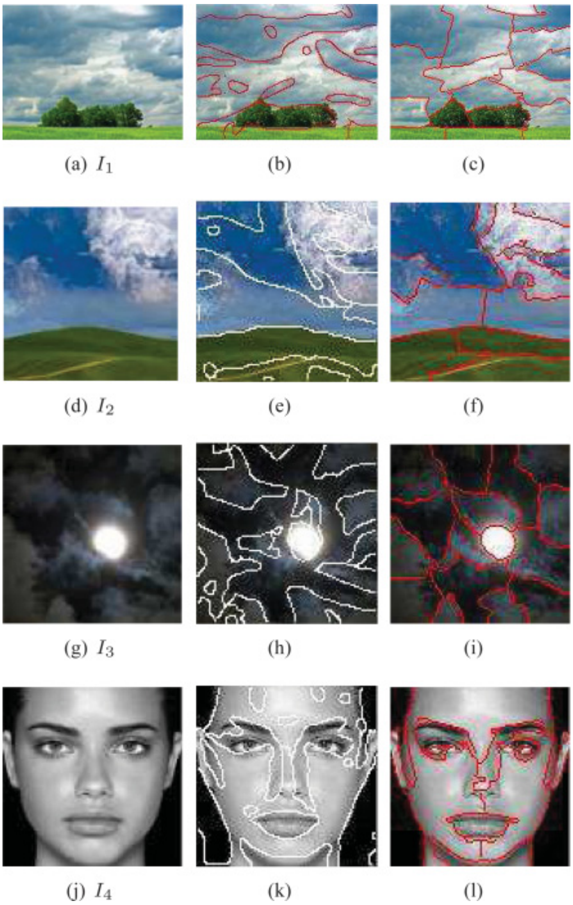


Figure 12: Segmentation of various kinds of natural and human images: pictures from nature, and human faces. From the first to the last image we have: landscapes (images 1–3), human face (image 4). For each line, the original image together with the segmentation result of GeNCut and Ncut are presented.

the correct partitioning. In fact, if we consider the image on the third row, NCut perfectly differentiates the melanoma from the skin if the number of segments is fixed to two. However, GeNCut is able to find the right partitioning and correctly discriminates between the melanoma and the skin, without requiring beforehand the number of segments. Note that, given an input image, it is hard to know a priori the true number of partitions.

Higher entropy values of NCut for the medical images in Table 1 confirm that our approach performs very well in comparison with the NCut approach.

7.2.3 Miscellaneous

In Figure 12 and Figure 13 we present the segmentation results on nine natural and human images of various kinds: landscapes and buildings (images 1, 2, 3, 8, 9), human faces (images 4 and 5), archaeological fragments, and painting (images 6 and 7, respectively). In particular, image 6 is a well-known fragment from murals discovered

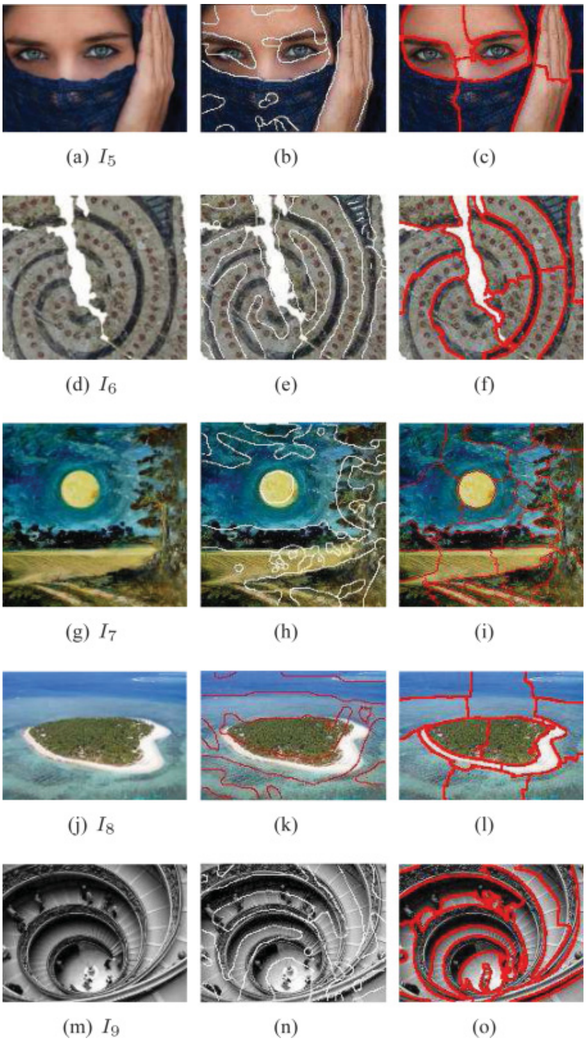


Figure 13: Segmentation on various kinds of natural and human images: pictures from nature, human faces, and images from archeology and painting. From the first to the last image we have: human face (image 5), archaeological fragment (image 6), painting (image 7), and landscape and building (images 8–9). For each line, the original image together with the segmentation result of GeNCut and Ncut are presented.

by professors Marinatos and Doumas in the Greek island of Thera and dated 1650 BC. Images of the fragments are freely available at http://www.therafoundation.org/. We point out that image segmentation of these fragments is a critical aspect in the automatic reconstruction of the murals and it is of fundamental importance for the archaeological research. Image 7 is an oil on board painting depicting a natural scene. Segment extraction from painting images is quite an interesting topic, and it is the basis of identifying an image of a painting as being an example of a given style of art. Finally, segmentation of human faces, as in images 4–5, is of prime importance for face recognition.

Image 3 represents a complex scenario, due to the presence of irregular shapes (clouds) around a spherical object (the moon). Although the halo makes it difficult to segment the moon, by using our algorithm we are able to perform a segmentation that adheres well to the human visual perception of the clouds. Although NCut also performs good segmentation, it is not able to capture some shape features.

Regarding images 1, 2, and 8, our algorithm is able to discover the main natural objects, while NCut often separates components that appear as single objects to a human, such as the sky in the second image, and the sea in image 8, despite the setting of the same number of segments, naturally extracted from our technique.

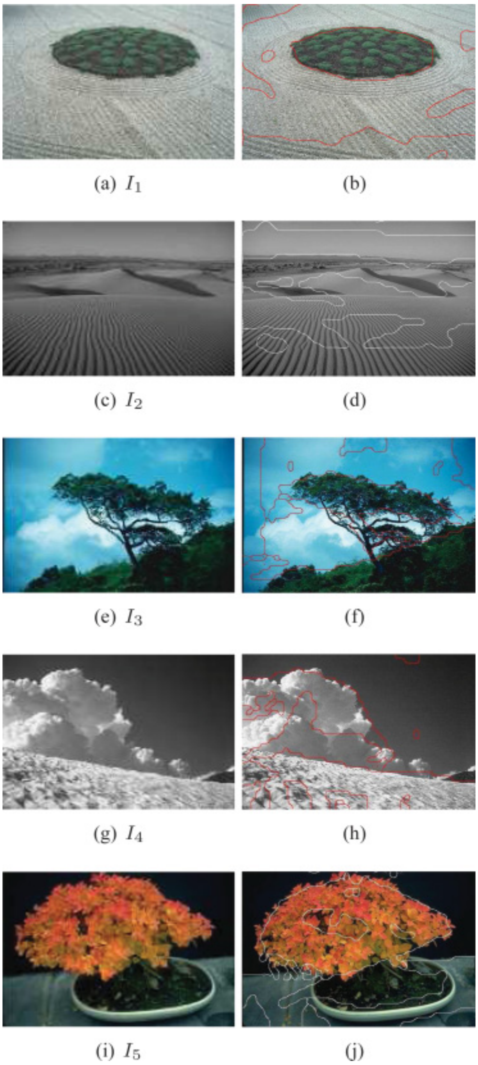
Looking at the segmentation results of GeNCut for the human faces, images 4 and 5, the two approaches are comparable. Although more details are discovered by NCut in correspondence to the eyes in image 4, it oversegments the face. On the other hand, GeNCut obtains a uniform and natural segmentation of the face that is also able to capture the shape of the nose, although it appears linked to the eyes, probably due to the similar gray intensities along the contours of the nose and the contours of the eyes.

Quantitative evaluation based on entropy for all the images is reported in Table 1. GeNCut performs quite well in the entropy domain, with lower entropy values with respect to the normalized cut approach.

7.3 Berkeley Images

The Berkeley image segmentation dataset (BSDS300; Martin et al., 2001) is a reference database for testing image segmentation algorithms. Images and gold-standard segmentations can be downloaded from http://www.eecs.berkeley.edu/Research/Projects/CS/vision/. In the Berkeley dataset, for each image, multiple human-traced segmentations are available. All the segmentations are considered to be equally reliable. We considered 10 images (see Figure 14 and Figure 15); for nine images, five different segmentations are provided, and for one image (the second image of Figure 14), six segmentations are available. For some images, different segmentations with the same number of regions are given. In order to compare GeNCut and NCut, given an image I, we first executed NCut by giving as input the same number of segments obtained by GeNCut, and then as many times as the number of different segmentations available for I, by giving as input the number k of ground-truth segments corresponding to the current human segmentation. This implies that the NCut method has been executed for the best input parameter value k. For each image and for each ground-truth segmentation, NCut was run 10 times. The average values of entropy E (Equation (19)), normalized mutual information (Equation (20)), and Probabilistic Rand Index (Equation (24)) were computed, together with the standard deviation, for the partitioning found by NCut, and compared with that obtained by GeNCut on the same image. All the results are reported in Table 2 and Table 3. Since GeNCut produces a single segmentation, the average value of entropy E in Table 2, and of PRI in Table 3, are the same for each of the segmentations of the image under consideration. Regarding the NMI value, instead, it is computed for each of the ground-truth segmentations associated with a given image.

In Table 2 for each image, the first row reports the number of segments obtained by GeNCut, the entropy value E, and the entropy value computed for the segmentation found by NCut when the number k of segments given as input is the same of that returned by GeNCut. Since for images I_8 and I_{10} the value k obtained by GeNCut coincides with one of the ground-truth segmentations, this row does not appear. The best values attained by either GeNCut or NCut are highlighted in bold. In particular, the entropy value of GeNCut is bold when its entropy value is lower than all entropy values computed for NCut. Note from the table that GeNCut obtains values of entropy



(a) I_1 (b)

(c) I_2 (d)

(e) I_3 (f)

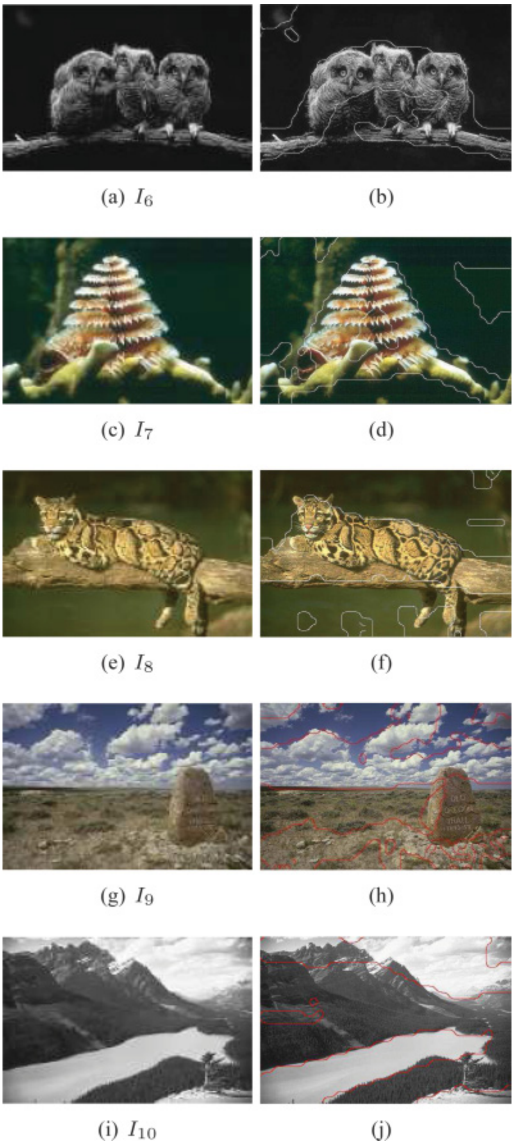
(g) I_4 (h)

(i) I_5 (j)

Figure 14: Segmentation of GeNCut on a subset of images from the Berkeley Image Segmentation dataset (BSDS300). For each line, the original image together with the segmentation result of GeNCut are presented.

lower than the entropies obtained by NCut in five out of the 10 images: in particular, for images I_1, I_2, I_6, I_7, and I_9, despite having been given to NCut the ground truth number of segments.

Looking at the NMI values on the same table, GeNCut overcomes the normalized cut approach for images I_2, I_3, and I_4, in all but one segmentation. In fact, NCut obtains higher NMI values only in segmentation 4 of image I_2, segmentation 3 of image I_3, and segmentation 2 of image I_4, notwithstanding that it has been given the true number of segments. For images I_6 and I_9, GeNCut returns better values in three out of five segmentations, and for image I_8, NMI values computed for GeNCut are always higher than the corresponding values computed for NCut. Thus, although we chose the best run conditions for NCut, by fixing the number of segments to the number of



(a) I_6

(b)

(c) I_7

(d)

(e) I_8

(f)

(g) I_9

(h)

(i) I_{10}

(j)

Figure 15: Segmentation of GeNCut on a subset of images from the Berkeley Image Segmentation dataset (BSDS300). For each line, the original image together with the segmentation result of GeNCut are presented.

partitions of the ground-truth segmentation that we are comparing to, in many cases the results of GeNCut are better than those obtained by NCut, or at least comparable with them.

Another important observation concerns the number of segments found by GeNCut on the test images. In some cases, such as in images I_8 and I_{10}, this number corresponds to the number of partitions of one of the ground-truth segmentations, or it is a value in the range of minimum and maximum number of segments. Note that the standard deviation values of GeNCut for both entropy and normalized mutual information are rather low, indicating that our approach is fairly stable.

Table 2: Normalized mutual information and entropy evaluated for GeNCut and NCut on a subset of the Berkeley Image Segmentation Dataset (BSDS300). *E* represents the entropy, NMI the normalized mutual information, *nc* the number of segments, ncGeNCut, the number of segments found by GeNCut.

		GeNCut			NCut		
		nc	*E*	NMI	*nc*	*E*	NMI
I_1	ncGeNCut	11	**0.0682** (0.0016)	—	11	0.2804 (0.0153)	—
	Seg1			0.3790 (0.0005)	24	0.1445 (0.0278)	**0.5905** (0.0004)
	Seg2			**0.4176** (0.0081)	4	0.5890 (0.4060)	0.4167 (0.0003)
	Seg3			0.3842 (0.0013)	23	0.1610 (0.0341)	**0.6050** (0.0030)
	Seg4			0.3455 (0.0006)	41	0.0898 (0.0124)	**0.5847** (0.0019)
	Seg5			0.3744 (0.0006)	17	0.2060 (0.0246)	**0.6217** (0.0003)
I_2	ncGeNCut	8	**0.0146** (0.0018)	—	8	0.5030 (0.0399)	—
	Seg1			**0.5778** (0.0140)	4	1.0427 (0.6662)	0.5570 (0.0458)
	Seg2			**0.5729** (0.0135)	5	0.9161 (0.2436)	0.4652 (0.0002)
	Seg3			**0.5546** (0.0110)	7	0.4426 (0.2226)	0.4749 (0.0002)
	Seg4			0.5457 (0.0094)	12	0.3435 (0.1866)	**0.5818** (0.0006)
	Seg5			**0.5541** (0.0108)	7	0.5519 (0.1803)	0.4970 (0.0002)
	Seg6			**0.5658** (0.0128)	9	0.3294 (0.1616)	0.5391 (0.0083)
I_3	ncGeNCut	13	0.2161 (0.0135)	—	13	0.2778 (0.0614)	—
	Seg1			**0.5092** (0.0260)	16	**0.2009** (0.0426)	0.4854 (0.0103)
	Seg2			**0.5043** (0.0262)	30	**0.1447** (0.0316)	0.4657 (0.0002)
	Seg3			0.3849 (0.0207)	10	0.3834 (0.0460)	**0.4371** (0.0001)
	Seg4			**0.5091** (0.0233)	26	**0.1461** (0.0363)	0.5020 (0.0015)
	Seg5			**0.4995** (0.0242)	16	0.2202 (0.0187)	0.4780 (0.0115)
I_4	ncGeNCut	16	0.5850 (0.4570)	—	16	**0.1966** (0.0002)	—
	Seg1			**0.6006** (0.0139)	6	0.6270 (0.2887)	0.4456 (0.0011)
	Seg2			0.5971 (0.0216)	12	**0.2914** (0.1389)	**0.6904** (0.0007)
	Seg3			**0.6042** (0.0196)	6	**0.4601** (0.2545)	0.4748 (0.0005)
	Seg4			**0.6023** (0.0141)	6	**0.5099** (0.2558)	0.4458 (0.0010)
	Seg5			**0.5978** (0.0140)	6	**0.4167** (0.2958)	0.4491 (0.0008)
I_5	ncGeNCut	13	0.4204 (0.1577)	—	13	**0.3169** (0.0213)	—
	Seg1			**0.6128** (0.0279)	6	0.4431 (0.1168)	0.5424 (0.0017)
	Seg2			0.4860 (0.0201)	8	0.5963 (0.3997)	**0.5875** (0.0009)
	Seg3			0.6135 (0.0277)	10	**0.2082** (0.1434)	**0.7068** (0.0002)
	Seg4			0.6118 (0.0282)	8	**0.3696** (0.3140)	**0.7399** (0.0011)
	Seg5			0.6303 (0.0255)	15	**0.1757** (0.0979)	**0.6774** (0.0029)
I_6	ncGeNCut	4	**0.0563** (0)	—	4	0.5396 (0.0734)	—
	Seg1			**0.4999** (0.0003)	7	0.3884 (0.2418)	0.4847 (0.0002)
	Seg2			**0.4983** (0.0009)	6	0.2778 (0.1107)	0.4948 (0.0003)
	Seg3			**0.4940** (0.0002)	6	0.2480 (0.0355)	0.4915 (0.0004)
	Seg4			0.4968 (0.0010)	6	0.4642 (0.2073)	**0.5033** (0.0002)
	Seg5			0.4923 (0.0023)	19	0.1579 (0.0094)	**0.5427** (0.0047)
I_7	ncGeNCut	9	**0.0276** (0)	—	9	0.3072 (0.0583)	—
	Seg1			0.4576 (0.0397)	6	0.5127 (0.0893)	**0.5168** (0.0002)
	Seg2			**0.4618** (0.0213)	4	0.6670 (0.0217)	0.3317 (0.0003)
	Seg3			0.4419 (0.0412)	10	0.3976 (0.1039)	**0.5936** (0.0009)
	Seg4			0.4749 (0.0382)	10	0.3693 (0.1359)	**0.4955** (0.0002)
	Seg5			**0.4370** (0.0293)	2	0.9410 (0.2241)	0.1248 (0.0020)

Table 2: Continued.

		GeNCut			NCut		
		nc	*E*	NMI	*nc*	*E*	NMI
	Seg1			**0.5643** (0.0134)	8	**0.4361** (0.0893)	0.4605 (0.0006)
	Seg2			**0.5546** (0.0134)	7	**0.3931** (0.1188)	0.4615 (0.0004)
I_8	Seg3	10	0.6235 (0.2301)	**0.5501** (0.0133)	8	**0.4065** (0.0774)	0.4452 (0.0008)
	Seg4			**0.5360** (0.0131)	7	**0.4363** (0.1109)	0.4565 (0.0004)
	Seg5			**0.5209** (0.0018)	10	**0.4034** (0.1175)	0.5189 (0.0084)
	ncGeNCut			—	18	0.2801 (0.0902)	—
	Seg1			**0.4639** (0.0122)	5	0.7625 (0.6048)	0.4075 (0.0018)
	Seg2			**0.4511** (0.0126)	3	1.8196 (0.0007)	0.2375 (0.0004)
I_9	Seg3	18	**0.0402** (0.0089)	**0.4618** (0.0116)	9	0.5654 (0.2546)	0.4399 (0.0002)
	Seg4			0.4773 (0.0034)	8	0.6866 (0.2405)	**0.5356** (0.0002)
	Seg5			0.6254 (0.0156)	28	0.1406 (0.0877)	**0.6346** (0.0121)
	Seg1			0.5220 (0.0187)	11	**0.2286** (0.1172)	**0.6065** (0.0001)
	Seg2			0.5641 (0.0059)	6	0.5659 (0.1871)	**0.6152** (0.0002)
I_{10}	Seg3	8	0.5605 (0.0670)	**0.5769** (0.0381)	3	1.1428 (0.6277)	0.5381 (0.0001)
	Seg4			0.5088 (0.0279)	11	**0.3838** (0.0809)	**0.6929** (0.0001)
	Seg5			0.5841 (0.0232)	8	**0.3451** (0.1485)	**0.6831** (0.0011)

Table 3 reports Probabilistic Rand Index values for both GeNCut and NCut. In particular, for each image *I*, we computed the PRI value of the segmentation returned by GeNCut, considered as test segmentation, against the set of ground-truth segmentations associated with *I* in the Berkeley dataset. With regard to NCut, PRI values were computed by considering as test segmentation that obtained by NCut for each of the executions performed, that is, one for the input parameter k fixed to the same number of segments obtained by GeNCut, and one for every segmentation available for the image under consideration. For example, if we consider image I_9, GeNCut found a segmentation of 18 segments; and the PRI value when this segmentation is considered as a test for the PRI Equation (24) is 0.7425. PRI values for NCut (when as test segmentation is used that obtained for input parameter k equal to 18, 5, 3, 9, 8, and 28) are 0.7071, 0.6682, 0.5025, 0.7273, 0.7267, and 0.6897, respectively, where the first $k = 18$ value is that obtained by GeNCut, and the others are those coming from the ground-truth segmentations. The table points out that the PRI value of GeNCut is higher for images I_2, I_3, I_4, I_6, I_8, and I_9. With regard to the others, NCut overcomes GeNCut on I_1, for four out of six segmentations on I_5, for two out of six segmentations on I_7, and all but one on I_{10}.

Finally, as already done for the images of the previous section, in Figure 14 and Figure 15, for each of the 10 images, we present the segmentation outputs of GeNCut by depicting the contours of the regions on the original image.

The visual perception of the segmentation results is quite positive. In addition, all the objects of a scene are identified and the main features extracted from the images by the segmentation process.

8 Conclusion

The paper presented a graph-based approach to image segmentation that employs GAs. A fitness function that extends the normalized cut criterion introduced in Shi and Malik (2000) is proposed, and a new concept of nearest neighbor that takes into

Table 3: Probabilistic Rand Index evaluated for GeNCut and NCut on a subset of the Berkeley Image Segmentation Dataset (BSDS300). PRI represents the Probabilistic Rand Index, *nc* the number of segments.

	GeNCut		NCut	
	nc	PRI	*nc*	PRI
			11	**0.7997** (0.0005)
			24	**0.7909** (0.0003)
			4	**0.6976** (0.0001)
I_1	11	0.6443 (0.0637)	23	**0.7936** (0.0008)
			41	**0.7897** (0.0005)
			17	**0.7991** (0.0001)
			8	0.6914 (0.0043)
			4	0.7580 (0.0004)
			5	0.7122 (0.0001)
I_2	8	**0.7822** (0.0016)	7	0.6974 (0.0001)
			12	0.7324 (0.0002)
			7	0.6975 (0.0001)
			9	0.7289 (0.0005)
			13	0.6592 (0.0001)
			16	0.6478 (0.0036)
			30	0.6310 (0.0001)
I_3	13	**0.7041** (0.0183)	10	0.6753 (0.0001)
			26	0.6340 (0.0001)
			16	0.6465 (0.0030)
			16	0.7460 (0.0004)
			6	0.7133 (0.0002)
			12	0.7770 (0.0008)
I_4	16	**0.8261** (0.0444)	6	0.7140 (0.0017)
			6	0.7135 (0.0006)
			6	0.7133 (0.0003)
			13	**0.8145** (0.0001)
			6	0.7708 (0.0002)
			8	**0.8867** (0.0003)
I_5	13	0.8088 (0.0198)	10	**0.8332** (0.0001)
			8	**0.8866** (0.0003)
			15	0.7978 (0.0003)
			4	0.7217 (0.0001)
			7	0.7181 (0.0001)
			6	0.7178 (0.0001)
I_6	4	**0.8137** (0.0215)	6	0.7178 (0.0001)
			6	0.7178 (0.0001)
			19	0.7540 (0.0001)
			9	0.7149 (0.0002)
			6	0.7278 (0.0002)
			4	0.7002 (0.0002)
I_7	9	0.7308 (0.0118)	10	**0.7359** (0.0002)
			10	**0.7361** (0.0001)
			2	0.6163 (0.0006)
			8	0.7704 (0.0001)
			7	0.7715 (0.0001)

Table 3: Continued.

	GeNCut		NCut	
	nc	PRI	*nc*	PRI
I_8	10	**0.8215** (0.0002)	8	0.7705 (0.0001)
			7	0.7715 (0.0001)
			10	0.7983 (0.0021)
			18	0.7071 (0.0002)
			5	0.6682 (0.0007)
			3	0.5025 (0.0004)
I_9	18	**0.7425** (0.0059)	9	0.7273 (0.0002)
			8	0.7267 (0.0001)
			28	0.6897 (0.0023)
			11	**0.7877** (0.0001)
			6	**0.8068** (0.0001)
I_{10}	8	0.7683 (0.0144)	3	0.6748 (0.0001)
			11	**0.7877** (0.0001)
			8	**0.8009** (0.0001)

account not only the spatial location of a pixel, but also the affinity with the other pixels contained in the neighborhood, is defined. The locus-based representation of individuals, together with the fitness function adopted, were shown to be particularly well suited to dealing with images modeled as graphs. In fact, as the experimental results showed, our approach is able to segment images in a number of regions that adhere well to human visual perception. The visual insight is corroborated by the objective evaluation measures reported.

It is known that evolutionary approaches are computing-time demanding; thus, the segmentation of large size images could require high runtimes. In fact, because of the adopted representation, the length of each individual of the population corresponds to the number of pixels of the image; thus, both space and time requirements could be high even for medium-size images. A preprocessing phase that reduces the original image size, as employed in our approach, allows for a noticeable reduction of execution time and further allows work with large images. As an example, for a 321 × 481 image, the execution of GeNCut on a MacBook Pro four-core, having an Intel Core i7, population size = 100, and number of generations = 100, takes about 3.5 min, including preprocessing and postprocessing. However, since the code is written in MATLAB, by exploiting the four cores, and running GeNCut with the parallel computing toolbox, the computation time is reduced to about 1.5 min. Thus, a parallel implementation could make the approach competitive, in terms of computing time, in real-life domains, such as medical imaging.

Future work aims at integrating the method inside a content-based image retrieval system. The identification of uniform image regions as a preprocessing step in image retrieval, in fact, has the advantage of improving the interpretation and, consequently, the retrieval of images and scenes, by extracting features at region level, and, possibly, their semantic content. Furthermore, an extension of the segmentation algorithm for detecting regions inside multispectral satellite data could be an interesting aspect, by considering that, in these kinds of images, each pixel is an array of spectral values, and not just a single gray value.

Acknowledgments

This work was partially supported by the project *MERIT: MEdical Research in Italy*, funded by MIUR.

References

Andrey, P., and Tarroux, P. (1998). Unsupervised segmentation of Markov random field modeled texture images using selectionist relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):252–262.

Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

Bhandarkar, S. M., and Zhang, H. (1999). Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 3(1):1–21.

Bhanu, B., Lee, S., and Ming, J. (1995). Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(12):1543–1567.

Bocchi, L., Ballerini, L., and Hässler, S. (2005). A new evolutionary algorithm for image segmentation. In *Applications of Evolutionary Computing, EvoIASP Workshop. Lecture notes in computer science*, Vol. 3449 (pp. 264–273). Berlin: Springer-Verlag.

Cagnoni, S., Lutton, E., and Olague, G. (2008a). Editorial introduction to the special issue on evolutionary computer vision. *Evolutionary Computation*, 16(4):437–438.

Cagnoni, S., Lutton, E., and Olague, G. (2008b). *Genetic and evolutionary computation for image processing and analysis*. New York: Hindawi Publishing Corporation.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Chen, C. W., Luo, J., and Parker, K. J. (1998). Image segmentation via adaptive *k*-means clustering and knowledge-based morphological operations with biomedical applications. *IEEE Transactions on Image Processing*, 7(12):1673–1683.

Chen, S., and Zhang, K. (2004). Robust image segmentation using FCM with spatial constraints based on a new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics B*, 34(4):1907–1916.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2007). *Introduction to algorithms*, 2nd ed. Cambridge, MA: MIT Press.

Cour, T., Bénézit, F., and Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, pp. 1124–1131.

Cover, T. M., and Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

DiGesú, V., and Bosco, G. (2005). Image segmentation based on genetic algorithms combination. In *Proceedings of Image Analysis and Processing, ICIAP 2005*, pp. 352–359.

Duchon, C. E. (1979). Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022.

Duda, R. O., and Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Felzenszwalb, P. F., and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Halder, A., and Pathak, N. (2011). An evolutionary dynamic clustering based colour image segmentation. *International Journal of Image Processing*, 4(6):549–556.

Handle, J., and Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76.

Helterbrand, J. D. (1996). One pixel-wide closed boundary identification. *IEEE Transactions on Image Processing*, 5(5):780–783.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: *University of Michigan Press*.

Jiao, L. (2011). Evolutionary-based image segmentation methods. In P. G. H. (Ed.), *Image segmentation* (Chap. 10, pp. 180–224). Rijeka, Croatia: Intech.

Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160.

Lai, C.-C., and Chang, C.-Y. (2009). A hierarchical evolutionary algorithm for automatic medical image segmentation. *Systems with Applications*, 36(1):248–259.

Leung, T. K., and Malik, J. (1998). Contour continuity in region based image segmentation. In *Proceedings of the 5th European Conference on Computer Vision, ECCV '98*, pp. 544–559.

Lindeberg, T. (1994). *Scale-space theory in computer vision*. Norwell, MA: Kluwer Academic.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference on Computer Vision*, Vol. 2, pp. 416–423.

Maulik, U. (2009). Medical image segmentation using genetic algorithms. *IEEE Transactions on Information Technology in Biomedicine*, 13(2):166–173.

Merzougui, M., Allaoui, A. E., Nasri, M., Hitmy, M. E., and Ouariachi, H. (2011). Evolutionary image segmentation by pixel classification and the evolutionary Xie and Beni criterion; Application to quality control. *International Journal of Computational Intelligence and Information Security*, 2(8):4–13.

Pappas, T. N. (1992). An adaptive clustering algorithms for image segmentation. *IEEE Transactions on Signal Processing*, 40(4):901–914.

Park, Y., and Song, M. (1989). A genetic algorithm for clustering problems. In *Proceedings of 3rd Annual Conference on Genetic Algorithms*, pp. 2–9.

Paulinas, M., and Ušinskas, A. (2007). A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and Control*, 36(3):278–284.

Peng, B., Zhang, L., and Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020–1038.

Pérez, C. B., and Olague, G. (2007). Unsupervised evolutionary segmentation algorithm based on texture analysis. In *Applications of Evolutionary Computating, EvoIASP Workshop. Lecture notes in computer science*, Vol. 4448 (pp. 407–414). Berlin: Springer-Verlag.

Pérez, C. B., Olague, G., Lutton, E., and Fernándezt, F. (2009). Texture image segmentation using an interactive evolutionary approach. In *Evolutionary image analysis and signal processing: Studies in computational intelligence*, Vol. 13 (pp. 3–19). Berlin: Springer-Verlag.

Pizzuti, C. (2012). A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation*, 16(3):418–430.

Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Sahoo, P. K., Soltani, S., and Wong, A. K. C. (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233–260.

Shi, J., and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Smith, S. L., and Cagnoni, S. (2011). *Genetic and evolutionary computation: Medical applications*. New York: Wiley.

Unnikrishnan, R., and Hebert, M. (2005). Similarity measures. In *IEEE Workshop on Computer Vision Applications*, pp. 416–423.

Unnikrishnan, R., Pantofaru, C., and Hebert, M. (2007). Towards objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944.

Urquhart, R. (1982). Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3):173–187.

Wu, Z., and Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and applications to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113.

Zahn, C. T. (1971). Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86.

Zhang, H., Fritts, J. E., and Goldman, S. A. (2004). An entropy-based objective evaluation method for image segmentation. In *Proceedings of the SPIE,* M. M. Yeung, R. Lienhart, and C.-S. Li (Eds.), *Storage and Retrieval Methods and Applications for Multimedia*, Vol. 5307, pp. 38–49. SPIE.

Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280.

Zhang, Y. J. (1997). Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18(10):963–974.