Journal of Electronic Imaging

JElectronicImaging.org

Genetic cuts for image segmentation

Shiueng-Bien Yang

SPIE.

IS&T imaging.org

Journal of Electronic Imaging 23(5), 053024 (Sep/Oct 2014)

Genetic cuts for image segmentation

Shiueng-Bien Yang*
Wenzao Ursuline University of Languages, Department of Digital Content Application and Management, 900 Mintsu 1st Road Kaohsing 80424, Taiwan

Abstract. The normalized cut (Ncut) method is a popular method for segmenting images and videos. The Ncut method segments an image into two disjoint regions, each segmented by the same method. After the Ncut method has been recursively applied to an image, its final segmented image is obtained. The main drawback of the Ncut method is that a user cannot easily determine the stop criteria because users have no idea about the number of regions in an image. This work proposes the genetic cut (Gcut) algorithm to resolve this shortcoming. Users do need not to specify thresholds in the Gcut algorithm, which automatically segments an image into the proper number of regions. Also, the neighbor-merging (NM) algorithm is proposed for preprocessing the images and improves the performance of the Gcut algorithm. Thus, the proposed Gcut method combines the NM and Gcut algorithms. Furthermore, a heuristic method is proposed to identify a good segment for the Gcut method. In all experiments, the proposed Gcut method outperforms traditional Ncut methods. [DOI: 10.1117/1.JEI.23.5.053024]

Keywords: genetic algorithm; normalized cut; genetic cut.

Paper 14027 received Jan. 19, 2014; revised manuscript received Sep. 13, 2014; accepted for publication Oct. 1, 2014; published online Oct. 27, 2014.

1 Introduction

The purpose of image segmentation is to divide the image into many regions such that each region is nearly homogenous and any two neighboring regions are not. The existing image segmentation algorithms can be classified into three categories; feature-space-based clustering, spatial segmentation, and graph-based methods. In the feature-space-based clustering methods, the image features are selected and calculated based on the color and texture.[1,2] Although the clustering methods are efficient for finding the salient region, the spatial structure and the detail edge information of an image are not preserved. The spatial segmentation methods are regarded as region based when they is based on region entities.[3] However, the spatial segmentation method may undesirably produce a very large number of small but quasihomogeneous regions. Thus, the merging algorithm is applied to these regions in Refs. 4 and 5. The graph-based methods can be regarded as image perceptual grouping and organization methods based on the spatial information. In the graph-based methods, the visual group is based on several key factors such as similarity, proximity, and continuation.[6–9] Felzenszwalb and Huttenlocher propose an alternative graph-based approach that has been applied to generate superpixels.[10] It performs an agglomerative clustering of pixels as nodes on a graph such that each superpixel is the minimum spanning tree of the constituent pixels. In Ref. 11, Moore et al. propose a method to generate superpixels that conform to a grid by finding optimal paths, or seams, that split the image into smaller vertical or horizontal regions. In Ref. 12, Veksler et al. used a global optimization approach. Superpixels are obtained by stitching together overlapping image patches such that each pixel belongs to only one of the overlapping regions. The methods[10–15] can be formed as a graph, where the weight of each edge connecting two pixels and two regions represents the likelihood that it belongs to the same segment. The graph is partitioned into many components that minimize some of the cost function.

Shi and Malik[16] proposed a general image segmentation method based on the normalized cut (Ncut). The Ncut method can robustly generate balanced clusters and is superior to other spectral graph-based methods, such as the average cut and average association.[16] Although the Ncut method has been applied in video summarization, scene detection,[17] and cluster-based image retrieval,[18] the image segmentation based on Ncut requires high computational complexity. In the Ncut method, the image is partitioned into two regions at a time, and then these two regions are continuously partitioned again, respectively. Thus, the user must set the stop criteria (or thresholds) before the Ncut method is recursively applied to the image. However, it is hard for the user to determine the stop criteria for the Ncut method. The reason is that the user has no idea about the number of regions contained in the image. Thus, the user usually sets the Ncut value as the threshold when the Ncut method is recursively applied to the image.[19] However, how to set the Ncut value is still a problem for the user. In Ref. 20, Merzougui et al. proposed an evolutionary-based image segmentation technique where the fitness is the mean distance between the pixels and the centroids. In Ref. 21, the biased Ncut is a modification of the normalized cut, which was proposed for constrained color-texture-based image segmentation.

In the last years, much effort has been devoted to defining the effective evolutionary-based approaches for solving complex problems related to computer vision. In particular, evolutionary techniques have been successfully applied to the image segmentation problem. A survey on the application of genetic algorithms for image enhancement and

*Address all correspondence to: Shiueng-Bien Yang, E-mail: 98010@mail.wzu.edu.tw

segmentation can be found in Ref. 22. In Ref. 23, a new graph-based algorithm, called the GeNCut (genetic NCut), was proposed to solve the image segmentation problem by using an evolutionary approach. In particular, a genetic algorithm optimizing a fitness function is executed in order to achieve good segmentation of the image. The fitness function is an extension of the Ncut concept of Shi and Malik;[16] it allows for a simultaneous k-way partitioning of the image without the need to fix the number k of divisions beforehand, i.e., the value of k is automatically determined by the GeNCut. In Ref. 24, the C-GeNCut (color GeNCut) extends the GeNCut to segment color-texture images in a number of regions that cater well to human visual perception. The C-GeNCut considers not only brightness, but also the color and texture for image segmentation. However, both the GeNcut and C-GeNCut use the same evolutionary technique and weighted normalized cut (WNCut) to segment images.

The contributions of our proposed Gcut method are described as follows.

(1) The Gcut method, such as the GeNCut, is proposed to segment the image without prior knowledge about the images. However, the GeNCut, such as the Ncut method, is designed to segment the image based on the pixels in the image; more time is required to segment an image with the genetic approach when the image size is large. Figure 1 shows the design of the Gcut method based on both the neighbor-merging (NM) and Gcut algorithms. The NM algorithm had been used to improve the performance of the genetic algorithms;[25,26] thus, the extended NM algorithm is also proposed to enhance the performance of the Gcut algorithm. The main goal of the NM algorithm is to merge similar and neighboring pixels (or blocks) into small regions in the image; each small region is regarded as a component that cannot be divided into the Gcut algorithm. Thus, the Gcut algorithm uses these components instead of pixels (or blocks) to enhance the efficiency of segmentation because the number of components is less than the number of pixels (or blocks) in an image. Also, the difference between the features in a component is larger and more robust than that between the pixels.

(2) A new Gcut criterion is proposed to measure the segmentation result in the Gcut algorithm. The Gcut criterion emphasizes the segmentation result, such that the intra-region similarity is high and the inter-region similarity is low. The intra-region similarity denotes the similarity between components in the same region, and the inter-region similarity denotes the similarity of the components in the adjacent regions. In contrast to the GeNCut method, the WNCut criterion considers the differences between the pixels in different regions throughout the whole image, and ignores emphasizing the differences between the pixels in the neighboring regions. Differences in the neighboring regions constitute one very important human visual perception for image segmentation. Figure 2 shows an example to illustrate the differences between the Gcut and GeNCut. In Fig. 2(a), the image can be broadly classified into white, black, and gray regions. The GeNCut segments the image into five regions, as shown in Fig. 2(b), because these small regions, a, b, c and d, are belong to the same gray regions according to the WNCut criterion. The benefit of the Gcut criterion is to emphasize the differences between the

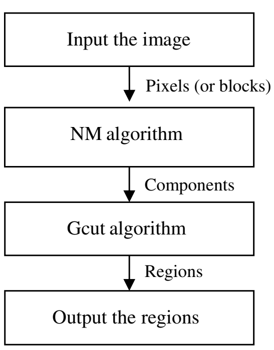


Fig. 1 The flow chart of our proposed method.

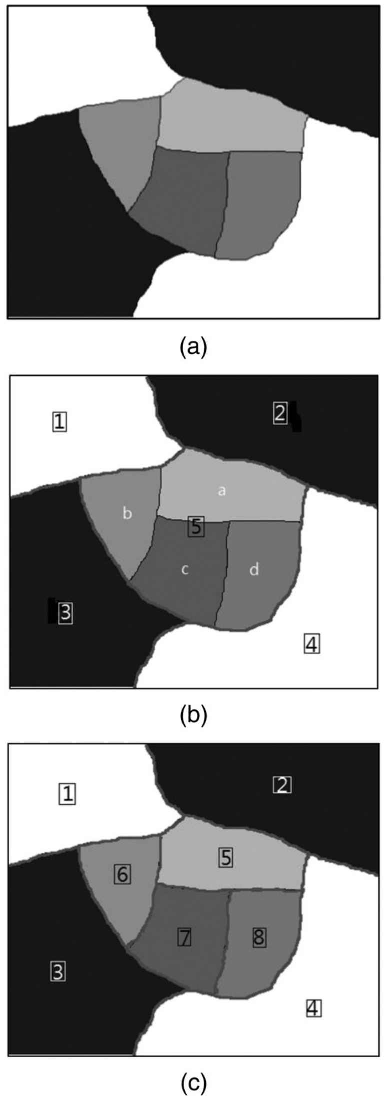


Fig. 2 The segmentation results of the Gcut and GeNCut methods. (a) Original image. (b) Five regions are obtained by the GeNCut method. (c) Eight regions are obtained by the Gcut method.

neighboring regions in the image, as shown in Fig. 2(c).

(3) Although the Gcut algorithm is able to automatically find the proper number of regions in the image, an additional parameter is provided to control the scope of the number of regions generated by the Gcut algorithm. If the parameter is large, the Gcut algorithm tends to generate fewer regions in the image. Otherwise, the Gcut algorithm tends to generate more regions. In Refs. 25 and 26, the heuristic method was used to achieve a good clustering result in the genetic algorithm. Thus, an extended heuristic method for finding a good segmentation is also proposed to determine the optimal number of segmentation regions in an image.

This paper is organized as follows. The Ncut and WNCut criterions are described in Sec. 2. Section 3 shows the design of the Gcut criterion. In Sec. 4, we demonstrate the Gcut algorithm. Section 5 shows the heuristic method used to find a good segmentation. The experiments are given in Sec. 6. Finally, Sec. 7 concludes this paper.

2 Normalized Cut and Weighted Normalized Cut Criterions

Let $G(V,E,W)$ be a graph, which can be partitioned into two disjoint sets, A, B, $A\cup B=V$, $A\cap B=\phi$. V is the set of pixels, and E is the set of edges connecting the pixels in the graph. Each pixel is represented as a node in $G(V,E,W)$. A pair of any two nodes is connected by an edge that is weighted by the matrix, W, which is used to measure their dissimilarity. The degree of dissimilarity between the two sets, namely $\mathrm{cut}(A,B)$, can be computed as the total weight of the removed edges, and is defined as

$$\mathrm{cut}(A,B)=\sum_{u\in A,v\in B}w(u,v), \tag{1}$$

where both u and v are two pixels that represent the nodes, and $w(u,v)$ is a weight between the two nodes, u and v, and is defined as

$$w(u,v)=e^{-\frac{\|u-v\|^2}{c}}. \tag{2}$$

The value of c is a constant in $w(u,v)$, and $\|\ \|$ denotes the Euclidean distance.

Much research[1–4] has been generated to find the minimum cut. However, the minimum cut criterion favors grouping small sets of isolated nodes in the graph. In Ref. 4, the modified graph partition criteria, Ncut, are defined as

$$\mathrm{Ncut}(A,B)=\frac{\mathrm{cut}(A,B)}{\mathrm{cut}(A,V)}+\frac{\mathrm{cut}(A,B)}{\mathrm{cut}(B,V)}. \tag{3}$$

After the graph is broken into two pieces, the Ncut criteria can recursively partition the two pieces. The recursion stops once the Ncut value exceeds a certain limit.

The WNCut criterion is applied to both the GeNcut and C-GeNCut methods. The WNCut is designed as follows. Let $G=(V,E,W)$ be the graph representing an image, W is its adjacency matrix, and $P=\{R_1,R_2,\ldots,R_k\}$ is a partition of G in k clusters. For a generic cluster $R\in P$, let

$$c_r=\sum_{u\in R,v\notin R}w(u,v), \tag{4}$$

$$m_r=\sum_{u\in R,v\in R}w(u,v), \tag{5}$$

$$m=\sum_{u\in V,v\in V}w(u,v), \tag{6}$$

be, respectively, the sum of the weights of the edges on the boundary of R, the sum of weights of edges inside R, and the total graph weight sum. The WNCut for each cluster $R\in P$ measures the fraction of total edge weight connections to all the nodes in the graph

$$\mathrm{WNCut}=\sum_{r=1}^{k}\frac{c_r}{m_r+c_r}+\frac{c_r}{(m-m_r)+c_r}. \tag{7}$$

This implies that low values of WNCut are preferred.

3 Design of Genetic Cut Criterion

Figure 1 displays the flow chart of the Gcut method. In the Gcut method, the image is first partitioned into components by the NM algorithm; these components are the input to the Gcut algorithm to generate the segmentation regions. The design of Gcut criterion is described as follows. Let there be q segmentation regions, $r_1,r_2,\ldots,r_q$, generated in the image, and let u_k be a component that belongs to the region, r_i. Then the $\mathrm{intra\_cut}(u_k,r_i)$ denotes the similarity between the components in the same region, and is defined as follows:

$$\mathrm{intra\_cut}(u_k,r_i)=\frac{\sum_{v_l\in r_i}w(u_k,v_l)}{|r_i|}, \tag{8}$$

where v_l is the component in r_i and $|r_i|$ denotes the number of components contained in r_i. Also, the $\mathrm{inter\_cut}(u_k,r_i)$ denotes the similarity of the components in adjacent regions, and is defined as follows:

$$\mathrm{inter\_cut}(u_k,r_i)=\frac{\sum_{\substack{r_j\subset \mathrm{NR}(r_i)\\ v_l\in r_j}}w(u_k,v_l)}{\sum_{r_j\subset \mathrm{NR}(r_i)}|r_j|}, \tag{9}$$

where $\mathrm{NR}(r_i)$ denotes the set of regions which are the neighbors of r_i. Figure 3 shows an example to illustrate the neighbors of a region. For example, 93 components are produced by the NM algorithm in Fig. 3(a), then the Gcut algorithm further merges these 93 components to generate 17 segmentation regions, as shown in Fig. 3(b). In Fig. 3(b), the region r_1 is surrounded by four neighboring regions: r_2, r_3, r_4 and r_5, which form the set of the neighbors of region r_1, $\mathrm{NR}(r_1)=\{r_2,r_3,r_4,r_5\}$. Figure 3(b) also shows the region r_1 containing five components: u_1, u_2, u_3, u_4, and u_5.

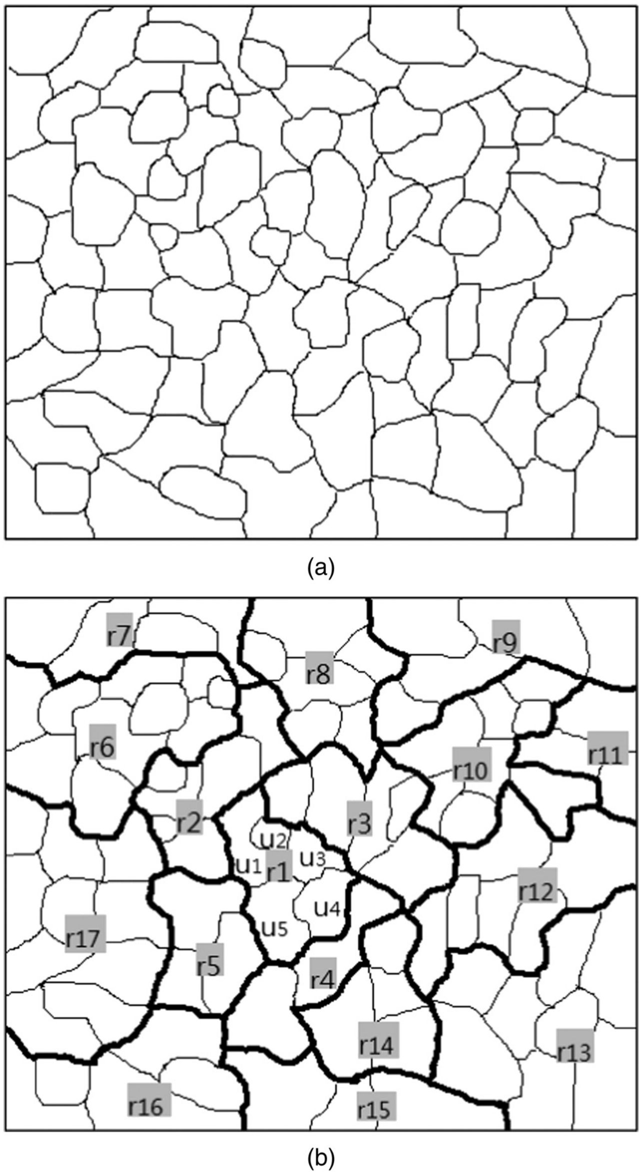


Fig. 3 An example to illustrate the neighbors of a region. (a) An example of 93 components in the image. (b) An example of 17 regions in the image.

Therefore, the Gcut criterion is defined as follows:

$$\mathrm{Gcut}(r_1, r_2, \ldots, r_q) = \sum_{\substack{i=1 \\ u_k \in r_i}}^{q} \alpha \frac{\mathrm{inter\_cut}(u_k, r_i)}{\mathrm{intra\_cut}(u_k, r_i)}. \tag{10}$$

In Eq. (10), the Gcut criterion emphasizes that the $\mathrm{intra\_cut}(u_k, r_i)$ is as large as possible and the $\mathrm{inter\_cut}(u_k, r_i)$ is as small as possible. Therefore, a low value of Gcut criterion is preferred. In Eq. (10), if the parameter α is a large value ($\alpha > 1$), the Gcut criterion is dominated by the value of the $\mathrm{inter\_cut}(u_k, r_i)$. Therefore, minimizing the Gcut criterion minimizes the value of the $\mathrm{inter\_cut}(u_k, r_i)$. Minimizing the $\mathrm{inter\_cut}(u_k, r_i)$ indicates that the component u_k in r_i must differ greatly from the components in the surrounding regions of $r_i[\mathrm{NR}(r_i)]$. A larger region containing more different components increases the difference with the component u_k. Fewer regions will make each region larger in the image. Then, the image tends to be segmented into lesser regions. Otherwise, the Gcut criterion is dominated by the value of the $\mathrm{intra\_cut}(u_k, r_i)$ while the value of α is smaller than 1. Thus, minimizing the Gcut criterion is to maximize the value of the $\mathrm{intra\_cut}(u_k, r_i)$. Maximizing the $\mathrm{intra\_cut}(u_k, r_i)$ indicates that the component u_k and the other components that belong to the same region r_i are similar. A small region containing fewer components will help to minimize the difference between the components in the region. More regions will make each region smaller in the image. Then the image tends to be segmented into more regions. Notably, the WNCut criterion defined in Eq. (7), like the Gcut criterion defined in Eq. (10), can also be multiplied by the parameter α, and then the GeNCut method is able to generate fewer or more regions in the image. In Eq. (7), the WNCut criterion can also be regarded as a case when the parameter α is set to 1. In Sec. 5, we discuss how the variance of α will produce segmentation results.

4 Design of the Genetic Cut Method

The Gcut method consists of the NM and the Gcut algorithms. The NM algorithm is described as follows. The goal of the NM algorithm is to merge the similar pixels to generate the components in the image. However, a large image has a large number of pixels, and the NM algorithm should take more time to generate the components in a large image. In our method, to reduce the computing time of the NM algorithm, the image should be divided into blocks of size 4×4 (pixels) before the image is segmented. In the experiment, the efficiency of the NM algorithm is discussed when both the pixels and blocks are applied to the NM algorithm, respectively.

Let the number of blocks in the image be n. Each block can be regarded as a node and each node is connected to its (upper, down, right, and left) neighbors in a graph. In the NM algorithm, each block is merged with its neighbors when they are similar, and into small regions (components) in the image. Each small region can be regarded as a component which cannot be divided any further. These components generated by the NM algorithm are the input for the Gcut algorithm. Thus, the Gcut algorithm does not process the pixels and blocks, but rather on these components in the image. The goal of the Gcut algorithm is to further merge these components into many segmentation regions. The NM algorithm is described as follows:

Algorithm: NM

Input: The set of blocks, $U = \{B_1, B_2, \ldots, B_n\}$, in image I. The value of m.

Output: The set of components, $V = \{v_1, v_2, \ldots, v_m\}$, in image I. $m \ll n$

Step 1. These n blocks, $B_1, B_2, \ldots, B_n$, are regarded as n nodes, $v_1, v_2, \ldots, v_n$, in the graph. Each node is connected to its neighboring nodes in the graph. Set $V = \{v_1, v_2, \ldots, v_m\}$, $v_i = B_i$, for $1 \le i \le n$. Set $\mathrm{Num} = n$.

Step 2. For each node, v_i, $1 \le i \le \mathrm{Num}$, do the following.

Step 2.1. Let $H(v_i)$ denoted the set of neighboring nodes of v_i. Calculate the minimal distance, d_i, between the node, v_i, and its neighboring nodes in $\mathrm{NR}(v_i)$. Then,

$$d_i = \min_{v_k \in \mathrm{NR}(v_i)} \|v_i - v_k\|. \tag{11}$$

Step 3. Find the minimal value $\hat{d}$ of the d_i, for $1 \le i \le \mathrm{Num}$. Then,

$$\hat{d} = \min_{1 \le i \le \mathrm{Num}} d_i, \tag{12}$$

$$j = \arg \min_{1 \le i \le \mathrm{Num}} d_i. \tag{13}$$

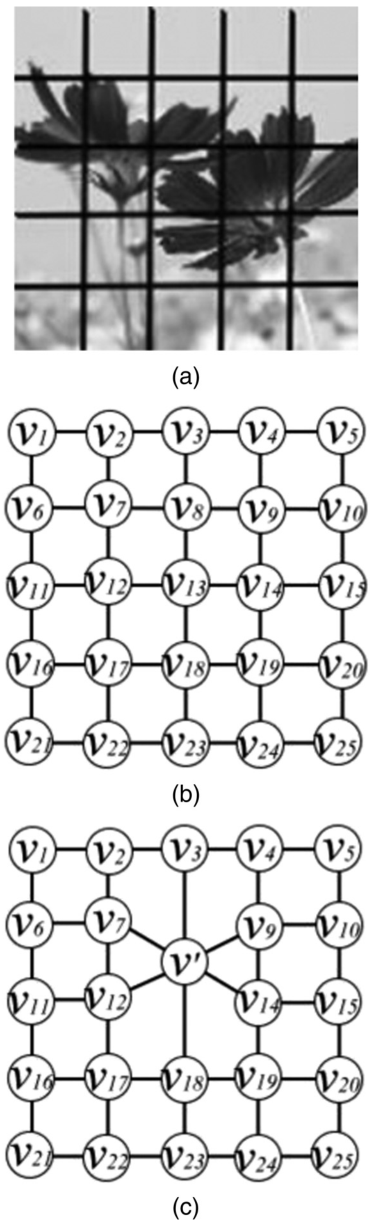


Fig. 4 An example to illustrate the NM algorithm. (a) The image is divided into the 4×4 blocks. (b) The nodes in the graph. (c) The nodes, v_8 and v_{13}, are merged into the node, v'.

Step 4. Let the distance between v_j and its neighboring nodes v_q be $\hat{d}$. Merge these two nodes, v_j and v_q, to generate the new node v'. Then, these two nodes, v_j and v_q, are deleted in V, and the new node v' is added to V. Set $\mathrm{NR} = (v') = \mathrm{NR}(v_j) \cup \mathrm{NR}(v_q)$ and $\mathrm{Num} = \mathrm{Num} - 1$.

Step 5. Go to Step 2 until the value of Num is equal to m.

Step 6. Each node in V is regarded as a component. These m components in V are the output.

End.

Figure 4 presents an example to illustrate the NM algorithm. Figure 4(a) displays the blocks of an image. Figure 4(b) shows the corresponding nodes of the blocks in the graph. In the NM algorithm, the node can be merged with its neighboring nodes if they are close enough. For example, in Fig. 4(c), the nodes v_8 and v_{13} are merged into node v'. Then, the nodes, v_3, v_7, v_9, v_{12}, v_{14} and v_{18}, are regarded as the neighboring nodes of v'. The merging process in the NM algorithm is continued until the desired number of components is obtained.

The Gcut algorithm is described as follows. Let there be m components, $v_1, v_2, \ldots, v_m$, obtained from the NM algorithm. The main goal of the Gcut algorithm is to further merge these components into the regions in the image. The Gcut algorithm is designed based on the genetic algorithm with the flow chart shown in Fig. 5.

4.1 *Initialization Step*

In the initialization step, a population of P strings is randomly generated. Then the length of each string is set to m. P strings are generated such that the 1s in the strings are uniformly distributed within $[1, m]$. Each string represents a subset of $\{v_1, v_2, \ldots, v_m\}$. If v_i is in this subset; then the i'th position of the string will be 1; otherwise, it will be 0. Each v_i in the subset is a seed to generate a region in the image. That is, each string denotes a segmentation result of the image.

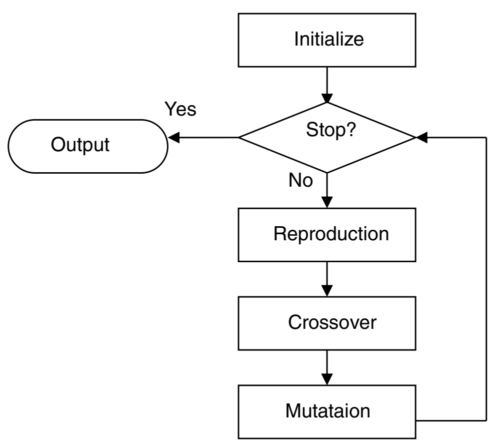


Fig. 5 The flow chart of the genetic algorithm.

4.2 *Reproduction Phase*

The main issue of the reproduction phase is to design the fitness function for each string. The string with a higher fitness indicates that it represents a better segmentation result. The design of the fitness function for the string R contains two stages, generating the segmentation result and calculating the fitness function. First, we describe how to generate the segmentation result for the string R. Let $R = (b_1, b_2, \ldots, b_m)$ be a bit string in the population. Each bit b_i represents the corresponding component v_i. Then, the string R includes a subset of components, Q, which is defined as

$$Q = \{v_i | b_i = 1, 1 \le i \le q\}. \tag{14}$$

In Q, q components, v_i for $1 \le i \le q$, are used as the seeds to generate q regions, and r_i for $1 \le i \le q$. Initially, each region, r_i, contains only one component, v_i. That is, $r_i = \{v_i\}$, $C(r_i) = v_i$, $S(r_i) = 1$. Then, each region r_i in Q is calculated the maximal similarity, $D(r_i)$, between the region, r_i, and its neighbors. Then,

$$D(r_i) = \max_{r_j \in \mathrm{NR}(r_i)} w[C(r_i), C(r_j)], \tag{15}$$

and

$$A(r_i) = \arg \max_{r_j \in \mathrm{NR}(r_i)} w[C(r_i), C(r_j)], \tag{16}$$

for $1 \le i \le n$. The maximum of $D(v_i)$, for $1 \le i \le n$, is thus calculated as

$$k = \arg \max_{1 \le i \le q} D(r_i). \tag{17}$$

The region, $r_{A(r_k)}$, is then merged with the region, r_k. The region, r_k, is updated as the new region, r'_k. Thus,

$$r'_k = r_k \cup \{v_{A(r_k)}\}, \tag{18}$$

$$C(r'_k) = \frac{C(r_k)S(r_k) + C(r_{A(r_k)})S(r_{A(r_k)})}{S(r_k) + S(r_{A(r_k)})}, \tag{19}$$

$$S(r'_k) = S(r_k) + S(r_{A(r_k)}), \tag{20}$$

$$\mathrm{NR}(r'_k) = \mathrm{NR}(r_k) \cup \mathrm{NR}(v_{A(r_k)}). \tag{21}$$

This merging process is continued until all of the components are merged to these q regions. Therefore, q regions are regarded as the segmentation result of the string R.

After the segmentation result of the string is produced, the second stage of the reproduction phase is continued to calculate the fitness of the strings. The computing complexity of the fitness function depends on the Gcut criterion. The design issue of the fitness function emphasizes that the Gcut criteria defined as in Sec. 3 are as small as possible. Let R be a string that separates the image into q regions, $r_1, r_2, \ldots, r_q$. Then the fitness function for the string R, $\mathrm{Fit}(R)$, is defined as

$$\mathrm{Fit}(R) = \frac{1}{\mathrm{Gcut}(r_1, r_2, \ldots, r_q)}. \tag{22}$$

In the fitness function, the Gcut criterion is possibly minimized. That is, the fitness of a string is possibly maximized. After the fitness of each string in the population is calculated, the reproduction operator is implemented using a roulette wheel with slots sized according to the fitness.

4.3 *Crossover Phase*

If the crossover operator is applied to a selected pair of strings I and J, then two random numbers e and f in $[1, m]$ are generated to decide which pieces of the strings are to be interchanged. After the crossover phase, two new strings, I' and J', replace the strings, I and J, in the population. The significance of the crossover phase is that it exchanges seeds between the different strings to yield the various segmentations.

4.4 *Mutation Phase*

During the mutation phase, the bits of the strings in the population are chosen from $[1, m]$ with a certain probability. Each chosen bit is then changed from 0 to 1 or from 1 to 0. That is, if one bit is chosen, then a selected cluster is discarded or produced in a string. After the mutation phase, the new string I' can be obtained and will replace the original string I.

The user may specify the number of generations over which to run the Gcut algorithm before obtaining the string with the best fitness. Suppose that the string $\hat{I}$ with the best fitness generates $\hat{n}$ regions. Then these $\hat{n}$ regions are the final segmentation result of the image.

The time complexity of the Gcut algorithm is analyzed as follows. The Gcut algorithm consists of an initialization step and iterations with three phases in each generation. Let N denote the size of the population and m denote the total number of components obtained by the NM algorithm. In the Gcut algorithm, it takes $\mathbf{O}(m^2)$ time for each component to find the nearest region. The time complexity of the Gcut algorithm is dominated by the calculation of the fitness function. It takes $\mathbf{O}(Nm^2)$ time in the worst case. Suppose the Gcut algorithm is asked to run G generations, the time complexity of the whole design of the Gcut algorithm is $\mathbf{O}(GNm^2)$.

5 Heuristic Method to Find a Good Segmentation

In the Gcut algorithm, the parameter α is used to control the segmentation result of the image. How to determine the parameter α is an important problem for the users. In this section, a heuristic method for finding a good segmentation is proposed, then the users do not need to determine the parameter α in Eq. (10). Assume q regions $\{r_1, r_2, \ldots, r_q\}$ are generated by the Gcut algorithm with the parameter α. We define the segmentation measure, $D(\alpha)$, as follows:

$$D(a) = \frac{\sum_{i=1}^{q} \mathrm{Variance}(r_i)}{q}, \tag{23}$$

where $\mathrm{Variance}(r_i)$ denotes the variance of the region, r_i, which is defined as follows:

$$\text{Variance}(r_i) = \frac{\sum_{v_l \in r_i} \|v_l - \text{Mean}(r_i)\|}{|r_i|}, \tag{24}$$

where $\text{Mean}(r_i)$ denotes the mean of the blocks contained in the region, r_i, which is defined as follows:

$$\text{Mean}(r_i) = \frac{\sum_{v_l \in r_i} v_l}{|r_i|}. \tag{25}$$

In general, if the value of $D(\alpha)$ is small, the image tends to be segmented into a large number of regions, and each region has a small size and low variance. Otherwise, if the value of $D(\alpha)$ is large, the image tends to be segmented into a small number of regions, and each region has a large size and high variance.

In the heuristic method, a good segmentation result is determined by using the Gcut algorithm with the parameter α varying within a range $[\alpha_1, \alpha_2]$. The values of α's are chosen from $[\alpha_1, \alpha_2]$ by some kind of binary search. The heuristic method is described in the following.

Step 1: Initially, let variables α_S and α_L indicate, respectively, the smallest value and the largest value within the given range, that is, $\alpha_S = \alpha_1$ and $\alpha_L = \alpha_2$. Use the Gcut algorithm with the parameters α_S and α_L to segment the image, respectively, and output $D(\alpha_S)$ and $D(\alpha_L)$.

Step 2: Do while $\alpha_L - \alpha_S > \lambda$

Begin

Step 2.1: Let $\alpha_M = [(\alpha_S + \alpha_L)/2]$. Use the Gcut algorithm with the parameter α_M to segment the image.

Step 2.2: Calculate the ratios $D(\alpha_M)/D(\alpha_S)$ and $D(\alpha_L)/D(\alpha_M)$.

Step 2.3: Among all the subranges within the whole range $[\alpha_1, \alpha_2]$, find the subrange $[\alpha_a, \alpha_b]$ that has the largest ratio of $D(\alpha_b)/D(\alpha_a)$. Set $\alpha_S = \alpha_a$ and $\alpha_L = \alpha_b$.

End

Step 3: Find the subrange $[\alpha_a, \alpha_b]$ that has the largest ratio, $|\alpha_b - \alpha_a|/|\alpha_2 - \alpha_1|$, and satisfies $\text{D}(\alpha_a = D(\alpha_b)$. Output the segmentation result obtained by the Gcut algorithm with a parameter α selected within the range $[\alpha_a, \alpha_b]$.

In Step 3, the subrange $[\alpha_a, \alpha_b]$ that has the largest ratio, $|\alpha_b - \alpha_a|/|\alpha_2 - \alpha_1|$, is called the "stable range." In the Gcut method, the segmentation result in the stable range is defined as arising more in line with the human visual sense. Thus, the segmentation result obtained by the Gcut algorithm, with the parameter α selected within the stable range $[\alpha_a, \alpha_b]$, is considered as the output. The segmentation results in a stable range; it will be compared with the results obtained from the other methods in the experiment.

Because the scale is different for both cases, the value of α is larger than 1 or less than 1, so these two cases must be processed separately, i.e., the Gcut algorithm segments the image into fewer regions by using the values of α in the range $[\alpha_1 = 1, \alpha_2]$ $(\alpha_2 > \alpha_1)$. The value of α_2 denotes the maximal value by which the inter_cut can be multiplied in Eq. (10). Also, the Gcut algorithm segments the image into more regions using the values of α in the range $[\alpha_3, \alpha_4 = 1]$ $(0 < \alpha_3 < 1)$. To fairly compare these two cases, we set $\alpha_3 = (1/\alpha_2)$; then the value of α_2 is also the maximal value by which the intra_cut can be multiplied in Eq. (10). Therefore, the segmentation result obtained by the Gcut algorithm with the parameter α selected within the stable range that has the largest ratio in the above two cases is considered as the output.

6 Experiments

In the experiments, the segmentation measure (SM) is used to measure the segmentation results of the image. Let that image consists of q regions. The SM is such that in Eq. (23), it is defined as

$$\text{SM} = \frac{\sum_{i=1}^{q} \text{Variance}(r_i)}{q}, \tag{26}$$

where $\text{Variance}(r_i)$ is defined that in as Eq. (24). From Eq. (26), a good segmentation result is one for which the SM of image I is as small as possible.

The Gcut method is based on the NM and Gcut algorithms. To verify the performance of the NM algorithm, the Gcut method with and without using the NM algorithm is tested separately. Table 1 lists the computing time of both the NM and Gcut algorithms when the Gcut method segments the artificial grayscale image $(256 \times 256$ pixels) shown in Fig. 6(a). The computing times of both the merging pixels and merging blocks in the NM algorithm are also separately presented in Table 1. In the Gcut algorithm, the population size is 100, the crossover rate is 80%, the mutation rate is 5%, and the parameter $\alpha = 1$ in Eq. (10). It is run for 100 generations and the best solution is retained.

Table 1 The performance of both the NM and Gcut algorithms.

		Merging pixels ($256 \times 256 = 65536$ pixels)		Merging blocks ($64 \times 64 = 4096$ blocks)	
		$M = 10$ components	$M = 20$ components	$M = 10$ components	$M = 20$ components
Exp. (1)	NM	16.81 s	16.13 s	14.51 s	13.61 s
	Gcut	2.35 s	2.38 s	2.35 s	2.38 s
	Total	19.16 s	18.51 s	16.86 s	15.99 s
Exp. (2)	Gcut	29.32 s		23.42 s	

In Exp. (1) of Table 1, the NM algorithm generates m components in the image, and the Gcut algorithm merges these m components to generate the segmentation regions. The "Merging Pixels" indicate that all of the pixels of an image are directly used as the input to the NM algorithm, each pixel is regarded as a node in the graph, and then the merging process of nodes in the NM algorithm is continued until the desired number of components is obtained. Also, the "Merging Blocks" denote that the image should be divided into blocks (4×4 pixels); these blocks can then be merged in the NM algorithm. We observe that the computing time of merging blocks is lower than that of merging pixels in the NM algorithm because the total number of blocks is less than the number of pixels in the image. Figures 6(b) and 6(c) show these 10 ($m = 10$) and 20 ($m = 20$) components in the image, respectively. Then, the Gcut algorithm merges these components, as shown in Figs. 6(b) and 6(c), respectively, and generates the same five segmentation regions that are shown in Fig. 6(d). Thus, we can conclude that the value of m in the NM algorithm does not affect the final segmentation result, because the Gcut algorithm can automatically find the proper number of segmentation regions from these m components. Also, we observe that the computing time of the Gcut algorithm shows a significant reduction in Exp. (1) of Table 1. The reason is that the Gcut algorithm uses fewer components instead of using pixels (or blocks) to segment the image.

In Exp. (2) of Table 1, the Gcut method only uses the Gcut algorithm to segment the image shown in Fig. 6(a), and then the Gcut algorithm also segments the image into five regions, as shown in Fig. 6(d). In this case, each pixel (or block) is regarded as a component; then all of the components are used as the input to the Gcut algorithm. From Exp. (2) of Table 1, we observe that the Gcut method spends more time segmenting the image because the number of pixels in the image is large. However, the total time in Exp. (1) is still less than that in Exp. (2). Therefore, we conclude that the NM algorithm can effectively improve the efficiency of the Gcut algorithm. Furthermore, although the computing time for merging blocks is less than that for merging pixels, the pixel position of the boundary is preferably accomplished by merging pixels.

Figures 7–9 show the segmentation results of the natural images with the Gcut method. The NM algorithm generates 100 components for each natural image, and then these 100 components are considered as the input to the Gcut algorithm for segmentation. Figure 7(a) shows the original image. Figure 7(c) shows the corresponding values of $D(\alpha)$ when the parameter α is in the range: $[(1/5)(=0.2),\ 1]$ and $\lambda = 0.1$, and Fig. 7(d) shows the corresponding values of $D(\alpha)$ when the parameter α is in the ranges $[1, 5]$ and $\lambda = 0.5$. The parentheses in both Figs. 7(c) and 7(d) denote [Sequence, number of regions, $D(\alpha)$]. The field "Sequence" denotes the sequence of the experiment when the heuristic method conducts a binary search on the values of parameter α. The field "Number of regions" denotes the number of regions that are generated by the Gcut algorithm with the given value of α. From Fig. 7(c), the segmentation result is stable when the parameter α is within the subranges $[0.3, 0.4]$, $[0.6, 0.7]$, and $[0.8, 0.9]$, that have the same largest ratio, $1/8$ ($|0.4 - 0.3|/|1 - 0.2| = |0.7 - 0.6|/|1 - 0.2| = |0.9 - 0.8|/|1 - 0.2| = 1/8$). From Fig. 7(d), the

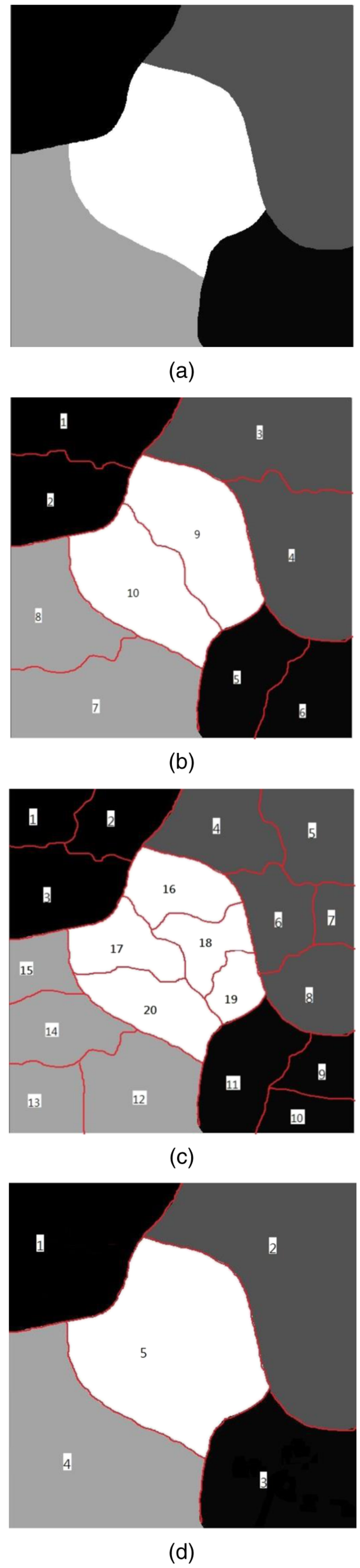


Fig. 6 The artificial image is segmented by the Gcut method. (a) Original image. (b) Ten components (merging blocks, $m = 10$). (c) Twenty components (merging blocks, $m = 20$). (d) Five segmentation regions ($\alpha = 1$).

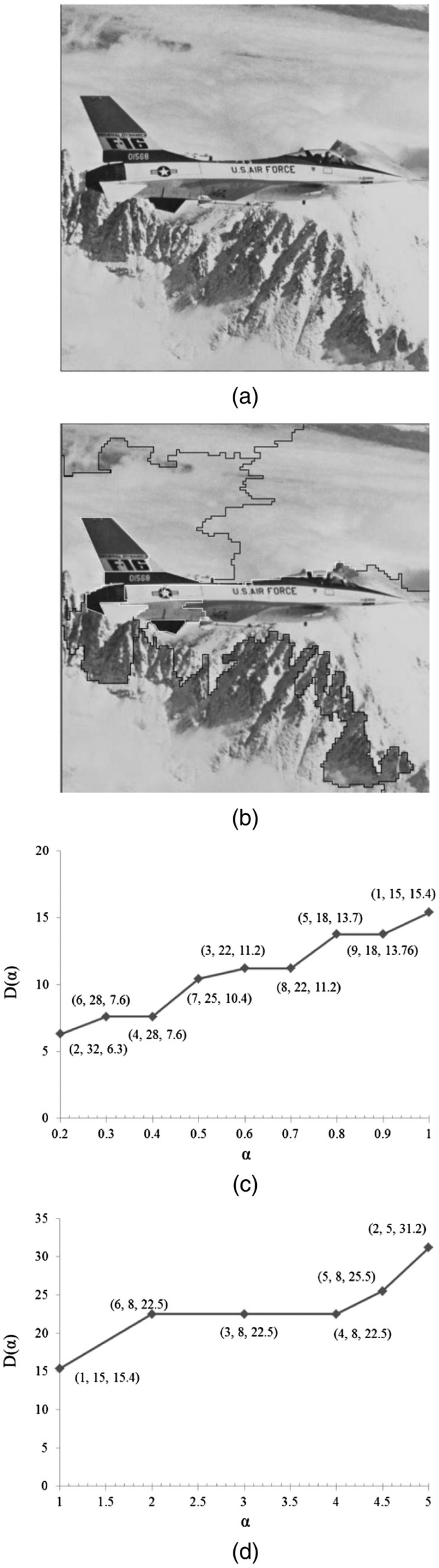


Fig. 7 The image "Jet" is segmented by the Gcut method. (a) The original image. (b) The segmentation result (8 regions). (c) The values of $D(\alpha)$ when the α is in the range [0.2, 1]. (d) The values of $D(\alpha)$ when the α is in the range [1, 5].

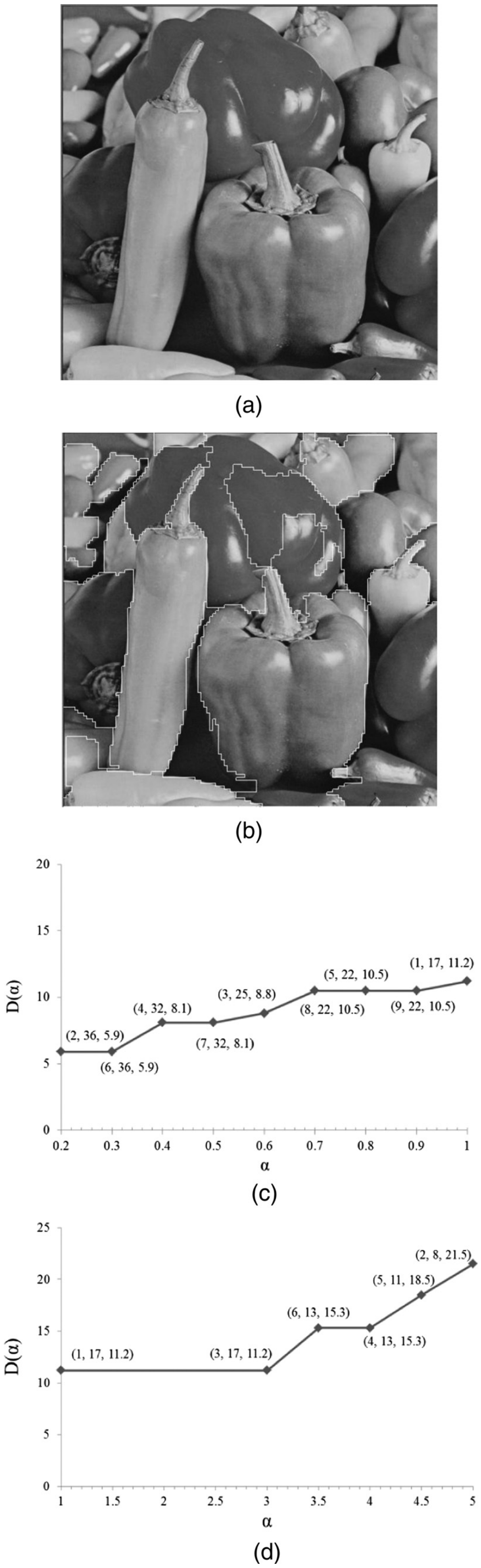


Fig. 8 The image "Pepper" is segmented by the Gcut method. (a) The original image. (b) The segmentation result (17 regions). (c) The values of $D(\alpha)$ when the α is in the range [0.2, 1]. (d) The values of $D(\alpha)$ when the α is in the range [1, 5].

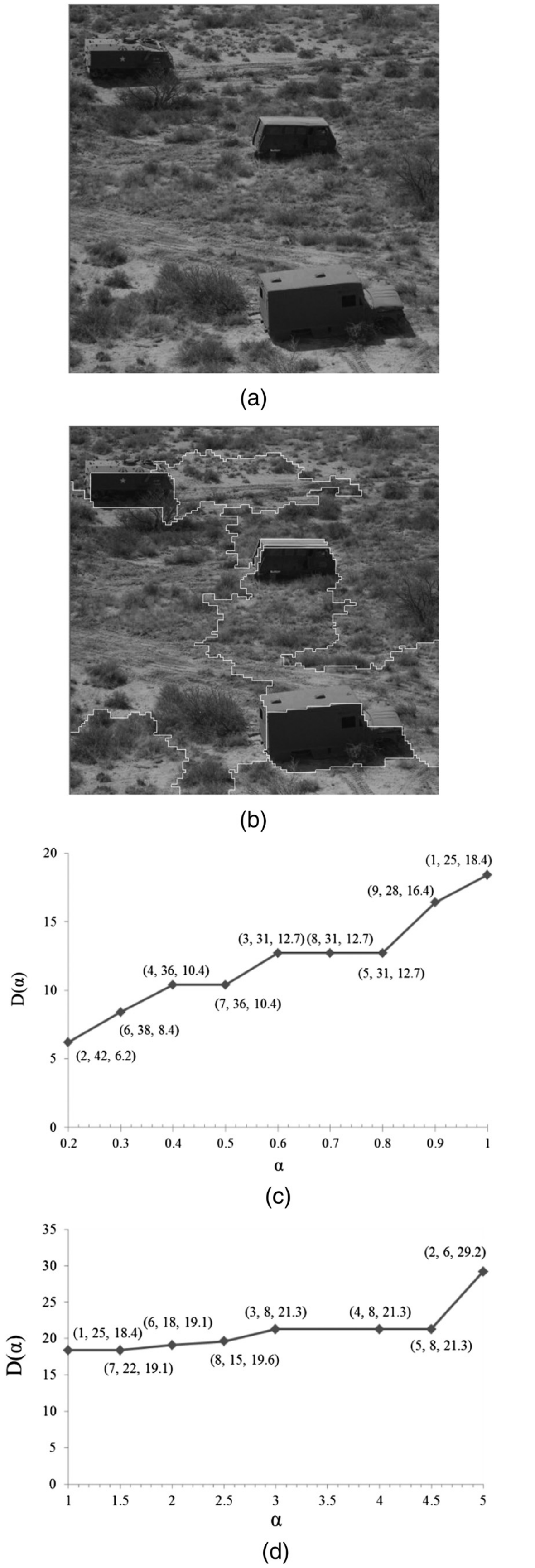
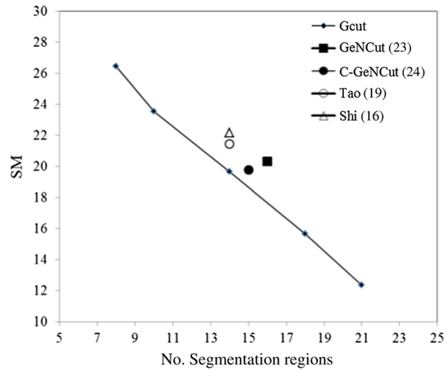


Fig. 9 The image "Trucks" is segmented by the Gcut method. (a) The original image. (b) The segmentation result (8 regions). (c) The values of $D(\alpha)$ when the α is in the range [0.2, 1]. (d) The values of $D(\alpha)$ when the α is in the range [1, 5].

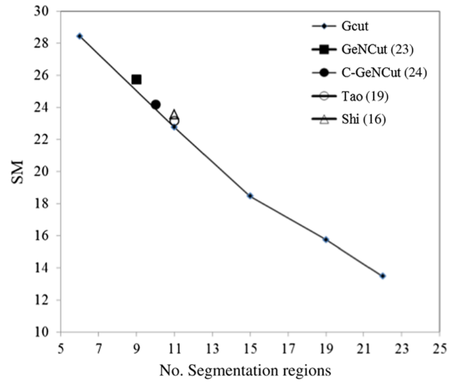
segmentation result is stable when the parameter α is within the subranges [2, 4], that have the largest ratio: $1/2$ $(|4-2|/|5-1| = 1/2)$. Because $1/2$ is larger than $1/8$, the segmentation result obtained by the Gcut algorithm with the parameter α selected within the range [2, 4] is the output. Therefore, the original image in Fig. 7(a) is segmented into eight regions (SM $= 22.5$) shown in Fig. 7(b). In Figs. 8(c) and 8(d), the segmentation result is stable when the parameter α is within the subrange [1, 3]. The image in Fig. 8(a) is segmented into 17 regions (SM $= 11.2$), which are shown in Fig. 8(b). Also, Fig. 9(a) is segmented into eight regions (SM $= 21.3$), which are shown in Fig. 9(b). In Figs. 9(c) and 9(d), the segmentation result is stable when the parameter α is within the subrange, [3, 4.5].

In the experiments, we provide some results using images from the Internet. The images are obtained from Google's Image Search, which is a keyword-based image retrieval system. Four image classes are obtained by Google's Image Search with four queries, "Car," "House," "Plane," and "Tiger." Each image class consists of 100 real images. Thus, a total of 400 images is used in our experiments.

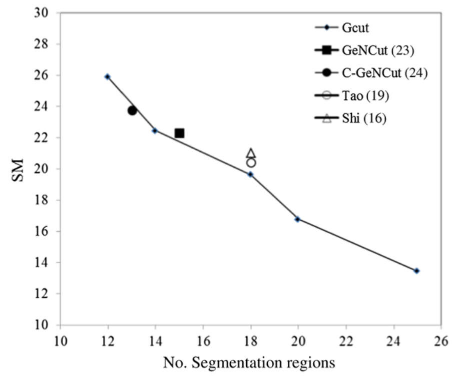
The Gcut method is compared with four methods based on the Ncut, Shi,[16] Tao,[19] GeNCut,[23] and C-GeNCut.[24] To fairly compare the Gcut method with the methods proposed in Refs. 16 and 19, the Gcut method is first used to segment the image; then both methods proposed in Refs. 16 and 19 generate the same number of regions in the image. The Gcut method is also compared with both methods, GeNCut[23] and C-GeNCut.[24] Our proposed Gcut method and both GeNCut and C-GeNCut methods have two common characteristics. First, the designs of these three methods are based on the genetic algorithms. Next, these three methods have the ability to identify the number of segmentation regions in the image. When the numbers of segmentation regions generated by the three methods differ, the SM value is not acceptable for comparison among these three methods. Thus, different segmentation results obtained by the Gcut algorithm using the different parameter α within the subranges, $[1/5, 1]$ and [1, 5], are compared to those obtained by both the GeNCut and C-GeNCut methods. Figure 10(a) shows an example of a "Car" image, and Fig. 10(b) shows the segmentation results of the image shown in Fig. 10(a) by the Gcut and other methods. From Fig. 10(b), the SM value obtained by the Gcut method is lower than that obtained by the other methods when they have the same number of segmentation regions in Fig. 10(a). Figure 10(d) also shows that the Gcut method has a lower SM value than the other methods when they have the same number of segmentation regions in the image shown in Fig. 10(c). Figures 10(f) and 10(h) show the examples that illustrate that the C-GeNCut has a lower SM value than the Gcut method when they have the same number of segmentation regions. The reason is that both the Gcut and GeNCut methods only use the gray-level information to segment the images, while the C-GeNCut method considers not only brightness, but also color and texture for image segmentation. Figure 11 shows the comparisons of the segmentation results of the 400 images using the Gcut and other methods. In Fig. 11, the statistics denote the number of images with the smallest SM values obtained by the method, assuming that all of the methods generate the same number of segmentation results in the images. From Fig. 11, the Gcut method outperforms



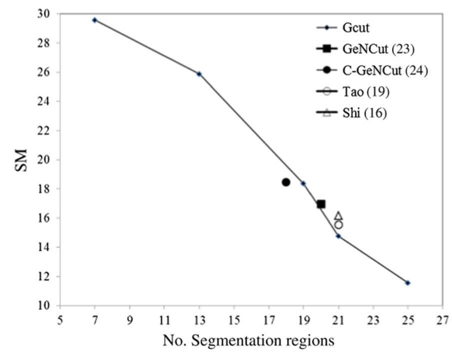
(a) (b)



(c) (d)



(e) (f)



(g) (h)

Fig. 10 Comparison of performance between the Gcut and other methods. (a) Class 1 (query="Car") (b) The segmentation results of our method and other methods. (c) Class 2 (query= "House"). (d) The segmentation results of our method and other methods. (e) Class 3 (query= "Plane"). (f) The segmentation results of our method and other methods. (g) Class 4 (query= "Tiger"). (h) The segmentation results of our method and other methods.

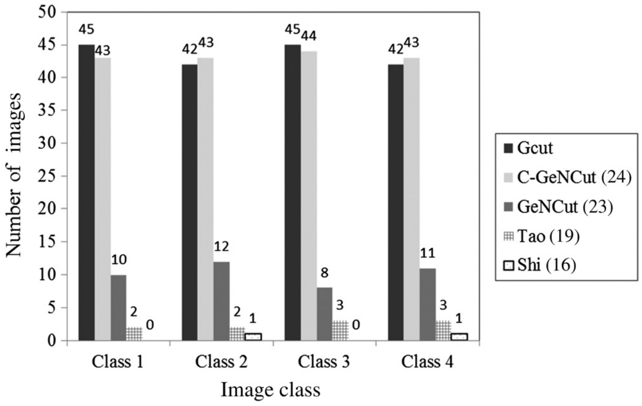


Fig. 11 The number of images with the smallest SM values obtained by the different methods.

the GeNCut,[23] Shi,[16] and Tao[19] methods in four classes of images because the Gcut criterion in the Gcut algorithm makes inter-region similarity as small as possible and intra-region similarity as large as possible. In particular, the Gcut criterion emphasizes the differences between the neighboring regions in the segmentation results. Also, the Gcut and the C-GeNCut methods have similar image segmentation quality in the four classes of images in Fig. 11. Notably, the Gcut method only considers the grayscale information for image segmentation, while the C-GeNCut method uses color features to segment the images. However, the above has proven that the Gcut method outperforms the GeNCut method, in which the proposed genetic approach and the WNCut criterion are also applied to design the C-GeNCut.

7 Conclusions

This paper applies the novel Gcut method for image segmentation. The Gcut method consists of the NM and Gcut algorithms. The NM algorithm is proposed to merge similar and neighboring pixels (or blocks) into small components in the image, and then the Gcut algorithm uses these components instead of pixels (or blocks) to segment the images. Furthermore, the Gcut criterion is proposed to replace the Ncut criterion for image segmentation in the Gcut algorithm. The Gcut criterion seeks to make the inter-region similarity as small as possible and the intra-region similarity as large as possible. The advantage of the Gcut algorithm based on the Gcut criterion is that it can automatically segment an image into the proper number of regions. Furthermore, the proposed heuristic method ensures a good segmentation when the Gcut algorithm is applied to segment an image. The Gcut method outperforms the variance of Ncut methods in the experiments.

References

1. J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, “A new method for gray-level picture thresholding using the entropy of the histogram,” *Comput. Vis. Graph. Image Process.* **29**, 273–285 (1985).
2. S. K. Pal, R. A. King, and A. A. Hashim, “Automatic grey level thresholding through index of fuzziness and entropy,” *Pattern Recognit. Lett.* **1**(3), 141–146 (1983).
3. K. Chen, D. Wang, and X. Liu, “Weight adaptation and oscillatory correlation for image segmentation,” *IEEE Trans. Neural Networks* **11**(5), 1106–1123 (2000).
4. S. Makrogiannis, G. Economou, and S. Fotopoulos, “A region dissimilarity relation that combines feature-space and spatial information for color image segmentation,” *IEEE Trans. Syst. Man Cybern. B, Cybern.* **35**(1), 44–53 (2005).
5. K. Haris et al., “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Trans. Image Process.* **7**(12), 1684–1699 (1998).
6. G. Kanizsa, *Grammatica del Vedere. Saggi su percezione e gestalt*, Il Mulino, Bologna, Italy (1980).
7. B. Caselles, B. Coll, and J.-M. Miorel, “A Kanisza programme,” *Progr. Nonlinear Differential Equations Appl.* **25**(2), 35–55 (1996).
8. M. Wertheimer, “Laws of organization in perceptual forms (partial translation),” in *Sourcebook of Gestalt Psychology*, W. B. Ellis, Ed., pp. 71–88, Harcourt Brace Jovanovich, Orlando, Florida (1938).
9. O. J. Morris, J. Lee, and A. G. Constantinides, “Graph theory for image analysis: an approach based on the shortest spanning tree,” *IEE Proc. F Commun., Radar Signal Process.* **133**(2), 146–152 (1986).
10. P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.* **59**(2), 167–181 (2004).
11. A. Moore et al., “Superpixel lattices,” in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE Computer Society, Anchorage, AK (2008).
12. O. Veksler, Y. Boykov, and P. Mehrani, “Superpixels and supervoxels in an energy optimization framework,” in *Proc. European Conf. on Computer Vision*, pp. 211–224, Springer-Verlag, Berlin, Heidelberg (2010).
13. S. X. Yu and J. Shi, “Multiclass spectral clustering,” in *Proc. IEEE Int. Conf. on Computer Vision*, pp. 313–319, IEEE Computer Society, Pittsburgh, Pennsylvania (2003).
14. S. Wang and J. M. Siskind, “Image segmentation with ratio cut,” *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(6), 675–690 (2003).
15. Z.-Y. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: theory and its application to image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(11), 1101–1113 (1993).
16. J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
17. C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, “Video summarization and scene detection by graph modeling,” *IEEE Trans. Circuits Syst. Video Technol.* **15**(2), 296–305 (2005).
18. Y. Chen, J. Z. Wang, and R. Krovetz, “CLUE: cluster-based retrieval of images by unsupervised learning,” *IEEE Trans. Image Process.* **14**(8), 1187–1201 (2005).
19. W. Tao, H. Jin, and Y. Zhang, “Color image segmentation based on mean shift and normalized cuts,” *IEEE Trans. Syst. Man Cybern.—Part B* **37**(5), 1382–1389 (2007).
20. M. Merzougui et al., “Evolutionary image segmentation by pixel classification and the evolutionary Xie and Beni criterion: application to quality control,” *Int. J. Comput. Intell. Inf. Secur.* **2**(8), 4–13 (2011).
21. S. Maji, N. K. Vishnoi, and J. Malik, “Biased normalized cuts,” in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 2057–2064, IEEE Computer Society, Providence, Rhode Island (2011).
22. M. Paulinas and A. Uinskas, “A survey of genetic algorithms applications for image enhancement and segmentation,” *Inf. Technol. Control* **36**(3), 278–284 (2007).
23. A. Amelio and C. Pizzuti, “An evolutionary and graph-based method for image segmentation,” *Lec. Notes Comput. Sci.* **7491**, 143–152 (2012).
24. A. Amelio and C. Pizzuti, “A genetic algorithm for color image segmentation,” *Lec. Notes Comput. Sci.* **7835**, 314–323 (2013).
25. L. Y. Tseng and S. B. Yang, “A genetic approach to the automatic clustering problem,” *Pattern Recognit.* **34**, 415–424 (2001).
26. L. Y. Tseng and S. B. Yang, “A genetic clustering algorithm for data with non-spherical-shape clusters,” *Pattern Recognit.* **33**, 1251–1259 (2000).

Shiueng-Bien Yang received his BS and PhD degrees from the Department of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan, in 1993 and 1999. Currently, he is a professor of the Department of Digital Content Application and Management, Wenzao Ursuline University of Languages, Kaohsiung, Taiwan. His research interests include pattern recognition, speech coding, image processing, and neural network.