

Improving the Quality of Color Image Segmentation Using Genetic Algorithm^{*}

Aniceto C. Andrade Jr.,
Zenilton K.G. Patrocínio Jr., and Silvio Jamil F. Guimarães

Audio-Visual Information Proc. Lab. (VIPLAB)
Computer Science Department – ICEI – PUC Minas
anicetojunior@gmail.com, {zenilton,sjamil}@pucminas.br

Abstract. Color image segmentation is the process of grouping regions according to some criterium. In this work, we cope with this problem using a graph-based approach based on removal of minimum spanning tree edges, however the tuning of parameters is a difficult task. To better identify the set of parameters which optimizes the error producing good segmentations, we propose the use of genetic algorithm in order to establish the best set of parameters. According to test experiments, our proposed method presents better results when compared to other approaches from the literature.

Keywords: Color image segmentation, genetic algorithm.

1 Introduction

Color image segmentation aims to group image regions using some criterium, however the choice of a strategy for grouping is a difficult task, and it is dependent on the application domain. One approach to cope with this problem is to model an image as a grid graph whose vertices correspond to the pixels and the edges connect the nearest neighbor pixels and their labels represent a dissimilarity measure computed from the connected pixels [4, 13] followed by the computation of a minimum spanning tree (MST). The first appearance of this tree in pattern recognition dates back to the seminal work of Zahn [15]. Lately, its use for image segmentation was introduced by Morris *et al.* [11] in 1986 and it was popularized in 2004 by Felzenszwalb and Huttenlocher [4]. Considering that, for a given image, one can tune the parameters of the well-known method [4] for obtaining a reasonable segmentation of this image. In [8], it was presented a framework to transform the non-hierarchical method proposed by [4] into its hierarchical version. In [7] two methods were proposed for color image segmentation, these methods also are MST-based and are called *Hierarchical Euclidean Minimum Spanning Tree (HEMST)* and *Maximum Standard Deviation Reduction Clustering Algorithm (MSDR)*. In fact, image segmentation methods may

^{*} The authors are grateful to PUC Minas – Pontifícia Universidade Católica de Minas Gerais, CNPq, CAPES and FAPEMIG for the financial support of this work.

be considered as a clustering problem that are usually handled by MST and K-means approaches (see [3, 5, 10, 12, 13, 14] for more informations).

In computer vision and pattern recognition, Fuzzy C-means (FCM) has been extensively used to improve the compactness of segmented regions, but its implementations always suffer from initialization problems related to the specification of the cluster number and with the selection of initial cluster centroids [12]. In [14], a method based on ant colony optimization – named *Ant System Algorithm* (AS) – was proposed to overcome the sensitiveness of FCM to the initialization conditions. However, the solutions obtained by AS are not very compact in the feature space, so to improve that the *Ant Colony-Fuzzy C-means Hybrid Algorithm* (AFHA) was introduced by the same authors in [14], which incorporates the FCM algorithm to the AS in order to improve the compactness of the clustering results in the feature space. Moreover, the *Improved Ant Colony-Fuzzy C-means Hybrid Algorithm* (IAFHA) is also presented in [14] in order to reduce AFHZ computational complexity using a sub-sampling method. Although the efficiency of has been increased, it still suffers from high computational complexity. The *Histogram Thresholding-Fuzzy C-means Hybrid Algorithm* (HTFCM), presented in [12], uses a histogram thresholding method for setting the initial conditions for the FCM algorithm. This approach does not require high computational complexity when compared to the ant-based proposals.

In [13], a simple method was proposed for simplifying color image using a minimum spanning tree approach for improving fire pixel classification. Even if the results for this application were quite good, the tuning of parameters is a real challenge. To deal with that, we consider genetic algorithms to look for the best choice of parameters. Genetic algorithms (GA) are mathematical algorithms inspired by natural evolution and genetic recombination. This technique provides a search engine which is based on adaptive *Darwinian principle of natural selection* [6, 9]. According to [6], the basic principle of this theory is related to features transmission through generations of individual, and the individuals which are better adapted to the environment have the higher possibility to survive and reproduce, and consequently, perpetuating their features. In GA, a chromosome is a data structure that can represent a solution. Thus, chromosomes are subjects to an evolutionary process which involves evaluation, selection, sexual recombination (*crossover*) and mutation. A final solution is obtained after several evolution cycles of a population in which only the best individuals are preserved [6]. In order to identify the best individual, fitness function according to a specific metric is needed.

In [17], it was presented an analysis and comparison of some metrics using a methodology for evaluation of image segmentation methods. Figure 1 illustrates that methodology with its three main components: (i) analytical method; (ii) goodness method; and (iii) discrepancy method. Analytical method acts directly on the segmentation algorithms, and not over the segmented image, defining its principles, requirements, utility, complexity, and other features of the algorithm. A goodness method assesses the segmentation result without using any reference image, e.g., it could be made indirectly through goodness metrics according to

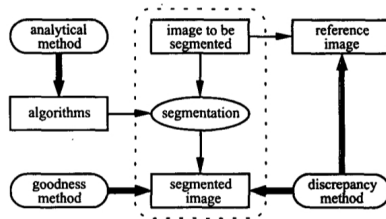


Fig. 1. The methodology for evaluation image segmentation methods proposed by [17]

the human intuition. It is also important to notice that a discrepancy method is used to quantify the difference between the segmented image and a reference one.

In this paper, we present a method for improving the quality of image segmentation using a genetic algorithm. Our proposal is based on the *Hierarchical Clustering Method* (HCM) presented in [13] and, therefore, we named it *Genetic Improvement of Hierarchical Clustering Method* (GHCM). The main contribution of this work is a graph-based approach for image segmentation, which uses a genetic algorithm to optimize the quality assessment according to some goodness method [17].

This paper is organized as follows. Section 2 presents our approach for image segmentation. Experimental results are presented in Section 3. And, finally in section 4, some considerations and future works are presented.

2 Proposed Approach

In this section, we present our proposal – GHCM – for improving the quality of image segmentation according to some metric using a genetic algorithm. Figure 2 shows the main steps of GHCM.

Our proposal uses the algorithm HCM presented in [13] (see Algorithm 1) for simplifying color images in order to decrease the number of colors. This method needs six parameters [13]: (i) original image; (ii) number of colors, which represents the number of representative colors that will be identified; (iii) outlier threshold, which represents the smaller permitted connected component size in terms of color frequency; (iv) a color space, which is the basis for graph creation values; (v) a distance measure, which is used to define the weight of graph edges; and (vi) number of nearest points considered during the graph creation. The computation time is directly related to the number of colors and, of course, to the adjacency relation of the graph, which will influence the graph size. In the adjacency relation, we will consider only the nK nearest points to each color c_1 in the color space. The strategy for the clustering process is presented in Algorithm 2 in which the best possible edge is identified to be eliminated and, consequently, its removal will divide a cluster into two others.

In order to apply a GA to a given problem, it is necessary to define the genotype required by the problem, i.e. the chromosome representation. In other

Algorithm 1. HCM – Hierarchical clustering method

input : An image – f ; Number of colors – nC ; Outlier threshold – T_n ;
 Color space – cS ; Distance measure – dM ; Nearest points number – nK .
output: Color set of segmented image.
 Color graph $G^\delta \leftarrow \text{createGraph}(f, cS, dM, nK)$;
 $\text{MST}_{G^\delta} \leftarrow \text{computeMinimumSpanningTree}(G^\delta)$;
for $i \leftarrow 1$ **to** nC **do**
 $e \leftarrow \text{getBestViableEdge}(\text{MST}_{G^\delta}, T_n)$;
 if ($e = \text{null}$) **then** **exit**;
 $C_1 \leftarrow \text{findConnectedComponentWithEdge}(e)$;
 $(C_2, C_3) \leftarrow \text{removeEdge}(e)$;
 $\text{MST}_{G^\delta} \leftarrow (\text{MST}_{G^\delta} - C_1) \cup C_2 \cup C_3$;
end for
return MST_{G^δ} ;

words, a decision must be made on how the parameters of the problem will be mapped into a finite string of symbols (genes), encoding a possible solution in the problem space. In this work, a chromosome was coded by a decimal string which is divided in five parts that are responsible for coding the parameters of the algorithm presented in [13], except for the original image.

At initialization, our method generates an initial population – see *step 1* in Figure 2. Each one of those individuals represents a specific combination of parameters that could be used with Algorithm 1. The fitness function evaluation plays a very important role in guiding the GA to obtain the best solutions within a large search space. Good fitness functions help the GA to explore the search space more effectively and efficiently. On the other hand, inappropriate fitness functions can easily weaken the GA search ability and result in getting stuck into a local optimum solution. In *step 2*, the fitness of each individual is measured by applying the Algorithm 1 to an image (or a set of images) – *step*

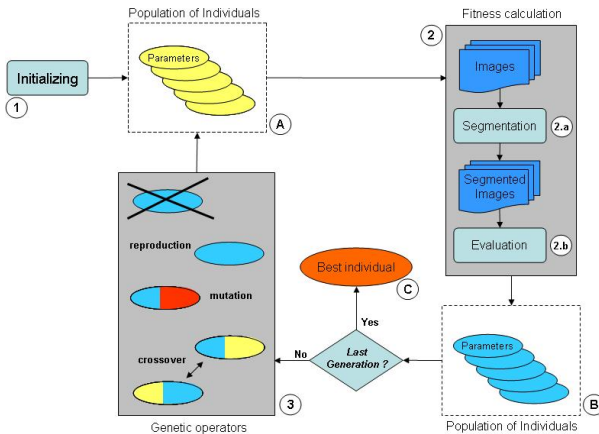


Fig. 2. Outline of the method of automatic segmentation

Algorithm 2. getBestViableEdge – Identifies the best edge to remove

```

input : Minimum spanning tree –  $MST_{Gs}$ ; Outlier threshold –  $T_n$ .
output: Best edge to remove –  $best$ .
 $best \leftarrow null$ ;
 $freqTotal \leftarrow sumOfFrequencies(MST_{Gs})$ ;
while ( $best = null$ ) do
     $w_e \leftarrow 0$ ;
    foreach edge  $e \in MST_{Gs}$  do
        | if ( $weight(e) \geq w_e$ ) then  $w_e \leftarrow weight(e)$ ;
    end foreach
    if ( $w_e = 0$ ) then return null;
    ( $C_1, C_2$ )  $\leftarrow componentsConnectedByEdge(e)$ ;
     $freqSum1 \leftarrow sumOfFrequencies(C_1)$ ;
     $freqSum2 \leftarrow sumOfFrequencies(C_2)$ ;
     $freq \leftarrow min(freqSum1, freqSum2)$ ;
    if ( $freq \geq (freqTotal \times T_n)$ ) then  $best \leftarrow e$  else  $weight(e) \leftarrow 0$ ;
end while
return  $best$ ;

```

2a – and, then, the goodness of the segmented image(s) is(are) computed – *step 2b*, according to an unsupervised evaluation method (also known as empirical goodness methods) [16]. More details about specific evaluation method used in this work will be discussed later (in Section 3).

If the maximum number of generations is not reached, a set of genetic operators (selection, recombination, and mutation) is applied in order to produce a new generation of individuals. Then the process starts again. In Figure 2, the individuals of the population **A** have not been evaluated, while the fitness calculation has already be done for all individuals of the population **B**. Finally, when the number of generations reaches a limit, the best individual is found (see **C** in Figure 2). After that, it could be used to segment the image(s) in order to obtain an improved final result(s) according to a specified metric (the one used in fitness calculation).

3 Experimental Analysis

In this section, we discuss some implementation issues. And we also present quantitative and qualitative assessments to show the effectiveness of our method when compared to AS [14], AFHA [14], IAFHA [14], and HTFCM [12].

Our method was implemented in C# using a genetic algorithm library named GALib¹. In order to compare our method to others, we used the same images considered in [12]: House (256 x 256), Football (256 x 256), Smarties (256 x 256), Capsicum (256 x 256), Gantry Crane (400 x 264), Beach (321 x 481) and Girl (321 x 481).

¹ <http://www.codeproject.com/Articles/54151/GALib>

Table 1. Genotype description: each gene represents a parameter for Algorithm 1

Gene <i>Locus</i>	Phenotypic feature (Coded parameter)	Alleles (Possible values for parameters)
1	Color space (<i>cS</i>)	RGB (1); YC_bC_r (2); $L^*u^*v^*$ (3); YIQ (4); $L^*a^*b^*$ (5)
2	Distance measure (<i>dM</i>)	Euclidean (1); Manhattan (2); Max (3)
3	Nearest points (<i>nK</i>)	Values in [5, 10]
4	Number of colors (<i>nC'</i>)	Values in [5, 15]
5	Outlier threshold (<i>T_n</i>)	Values in [1, 5]

With respect to the our population, each individual is represented by a decimal string. Table 1 illustrates the genotype structure of each individual. Other parameters used in GA are: (i) population size: 30; (ii) the maximum number of generations: 10, 15, or 20; (iii) mutation probability: 0.10; (iv) crossover probability: 0.90; and (v) percentage of elitism: 6%. Those values were obtained after preliminary empirical tests. We have also adopt one-point crossover and fitness proportional selection.

Two different metrics were used as unsupervised evaluation to assess the goodness of segmented image(s) during fitness calculation: (i) V_{PC} ; and (ii) $Q(I)$ – both were used in [12, 14]. These metrics allow us to assess the compactness and the homogeneity of pixel clusters based on a pixel representation. Let \mathbf{x}_p be a vectorial representation of the pixel p in a color space, e.g., $\mathbf{x}_p = (r_p, g_p, b_p)$, where r_p , g_p , and b_p are the values of the red, green, and blue components of the pixel p in RGB color space. The first metric is Bezdek partition coefficient – V_{PC} [1] – and it measures the compactness of the generated clusters. A smaller V_{PC} value indicates a fuzzier result, thus the larger the V_{PC} value, the better the clustering result. This metric ranges from 0 (less compact) to 1 (more compact), and it is defined as follows:

$$V_{PC} = \frac{\sum_{i=1}^N \sum_{j=1}^M u_{ji}^2}{N}, \quad (1)$$

where N is the total pixel number in image, M is number of clusters, and u_{ji} is the membership degree of i th pixel to j th cluster whose value is calculated by:

$$u_{ji} = \frac{1}{\sum_{k=1}^M (d_{ji}/d_{ki})^{2/(m-1)}}, 1 \leq j \leq M, 1 \leq i \leq N \quad (2)$$

where d_{ji} is the (euclidean) distance between i th pixel and j th cluster centroid, and m is the exponential weight of membership degree (that was set to 2). Notice that if $d_{ji} = 0$, then $u_{ji} = 1$ and other membership degrees of this pixel are set to 0.

The second metric – $Q(I)$ [2] – is used to penalize a segmentation that forms too many regions and whose regions are non-homogeneous. This metric produces higher values for over-segmented images with non-homogeneous regions and smaller values otherwise, and is defined by:

$$Q(I) = \frac{1}{(1000 \times N)} \sqrt{R} \sum_{j=1}^R \left[\frac{e_j^2}{1 + \log N_j} + \left(\frac{S(N_j)}{N_j} \right)^2 \right], \quad (3)$$

where I is the image and N is the total pixel number in I . The segmentation can be described as an assignment of pixels in image I into R regions. Let C_j denote the set of pixels in region j , $N_j = |C_j|$ denote the number of pixels in C_j . We also define \mathbf{c}_j as the region centroid, $\mathbf{c}_j = (\sum_{p \in C_j} \mathbf{x}_p) / N_j$. The squared error of region j is defined as $e_j^2 = \sum_{p \in C_j} (\mathbf{x}_p - \mathbf{c}_j)^2$. Finally, $S(a)$ denotes the number of regions in image I that have an area of exactly a pixels [2].

3.1 Experimental Results Using V_{PC}

The number of color groups found by the five different methods is presented in Table 2(a) – for GHCM the best results found after 4 runs are shown. One can observe that GHCM produces the smaller number of groups using V_{PC} after 15 iterations, except for one image (these results correspond to the best individuals whose V_{PC} values are shown in Table 2(c)). Tables 2(b), 2(c) and 2(d) show the V_{PC} values obtained by GHCM after 10, 15, and 20 iterations, respectively, along with the values obtained by the other methods. One can easily see that GHCM presents a similar result when compared to HTFCM [12] after only 10 iterations, and, for 15 and 20 iterations, GHCM outperforms all tested methods. Therefore, all the remaining tests were made with a maximum number of iterations (generations) equals to 15.

3.2 Experimental Results Using $Q(I)$

In this experiment, the aim is to verify the behavior of GHCM in different runs using an empirical goodness method to estimate the segmentation quality with some human characterization about the properties of “ideal” segmentation without any prior knowledge of correct segmentation [16]. Therefore, metric $Q(I)$

Table 2. Results obtained using V_{PC} (Bold values represent the best results)

(a) Number of regions after 15 iterations

Image	AS	AFHA	IAFHA	HTFCM	GHCM
House	10	10	10	07	08
Football	10	10	10	07	05
Capsicum	15	15	17	08	05
Smarties	07	07	07	06	05
Beach	15	15	14	10	06
Gantry Crane	10	10	09	08	05
Girl	14	14	15	09	06

(b) Values of V_{PC} after 10 iterations

Image	AS	AFHA	IAFHA	HTFCM	GHCM
House	0.742	0.736	0.729	0.804	0.758
Football	0.628	0.633	0.634	0.679	0.673
Capsicum	0.474	0.464	0.449	0.593	0.654
Smarties	0.771	0.784	0.782	0.793	0.827
Beach	0.593	0.587	0.603	0.689	0.679
Gantry Crane	0.648	0.631	0.649	0.691	0.772
Girl	0.654	0.629	0.596	0.668	0.706

(c) Values of V_{PC} after 15 iterations

Image	AS	AFHA	IAFHA	HTFCM	GHCM
House	0.742	0.736	0.729	0.804	0.758
Football	0.628	0.633	0.634	0.679	0.699
Capsicum	0.474	0.464	0.449	0.593	0.659
Smarties	0.771	0.784	0.782	0.793	0.828
Beach	0.593	0.587	0.603	0.689	0.696
Gantry Crane	0.648	0.631	0.649	0.691	0.772
Girl	0.654	0.629	0.596	0.668	0.733

(d) Values of V_{PC} after 20 iterations

Image	AS	AFHA	IAFHA	HTFCM	GHCM
House	0.742	0.736	0.729	0.804	0.758
Football	0.628	0.633	0.634	0.679	0.699
Capsicum	0.474	0.464	0.449	0.593	0.659
Smarties	0.771	0.784	0.782	0.793	0.828
Beach	0.593	0.587	0.603	0.689	0.696
Gantry Crane	0.648	0.631	0.649	0.691	0.800
Girl	0.654	0.629	0.596	0.668	0.767

Table 3. Results obtained using $Q(I)$ after 15 iterations ($\times 10^3$)

Image	IAFHA	HTFCM	GHCM			
			Run 1	Run 2	Run 3	Run 4
House	0.4004	0.1237	0.0012	0.0017	0.0059	0.0022
Football	0.8403	0.1895	0.0039	0.0032	0.0029	0.0029
Capsicum	0.3806	0.3606	0.0065	0.3513	0.0193	0.0131
Smarties	0.1456	0.1442	0.0017	0.0022	0.0015	0.0017
Beach	0.1513	0.1495	0.0065	0.0074	0.0064	0.0042
Gantry Crane	0.3166	0.1717	0.0063	0.0021	0.0068	0.0089
Girl	0.2987	0.1864	0.0099	0.0095	0.0087	0.0090

was used for fitness calculation. Table 3 presents the results after 15 iterations. One can see that even for different runs, GHCM is always better than both – IAFHA [14] and HTFCM [12].

3.3 Qualitative Analysis

In our experiments, the color model $L^*u^*v^*$ was chosen in 39.28% for the best individual, followed by the model YIQ in 28.58% of best individuals, YC_bC_r in 25.00% and the last one is the RGB model in 7.14% of the cases. With respect to distance measure, Max was choosen in 42.86%, followed by Manhattan (32.14%) and Euclidean (25.00%). Table 4 shows some examples of the best individuals found after 15 iterations in 4 distinct runs using metric $Q(I)$ for fitness calculation.

Finally, Fig. 3 illustrates the segmented images obtained with the best individuals found by GHCM after 15 iterations using V_{pc} and $Q(I)$ for fitness calculation. It is important to notice that, even though these metrics are very

Table 4. Examples of best individuals found after 15 iteration in 4 runs using $Q(I)$

Image	Feature	Run 1	Run 2	Run 3	Run 4
House	Color space	YIQ	YC_bC_r	$L^*u^*v^*$	$L^*u^*v^*$
	Distance measure	Euclidean	Manhattan	Euclidean	Manhattan
	Nearest points	10	07	07	08
	Number of colors	05	07	08	07
	T_n	01	01	01	03
Football	Color space	YIQ	YIQ	YIQ	YIQ
	Distance measure	Manhattan	Euclidean	Manhattan	Manhattan
	Nearest points	09	05	06	06
	Number of colors	07	05	05	05
	T_n	02	01	01	02
Capsicum	Color space	YC_bC_r	YIQ	YC_bC_r	YIQ
	Distance measure	Maxim	Maxim	Maxim	Maxim
	Nearest points	09	05	06	06
	Number of colors	07	05	05	05
	T_n	01	01	01	01
Beach	Color space	$L^*u^*v^*$	$L^*u^*v^*$	$L^*u^*v^*$	YIQ
	Distance measure	Maxim	Euclidean	Maxim	Manhattan
	Nearest points	09	06	07	05
	Number of colors	06	06	05	05
	T_n	05	02	05	05

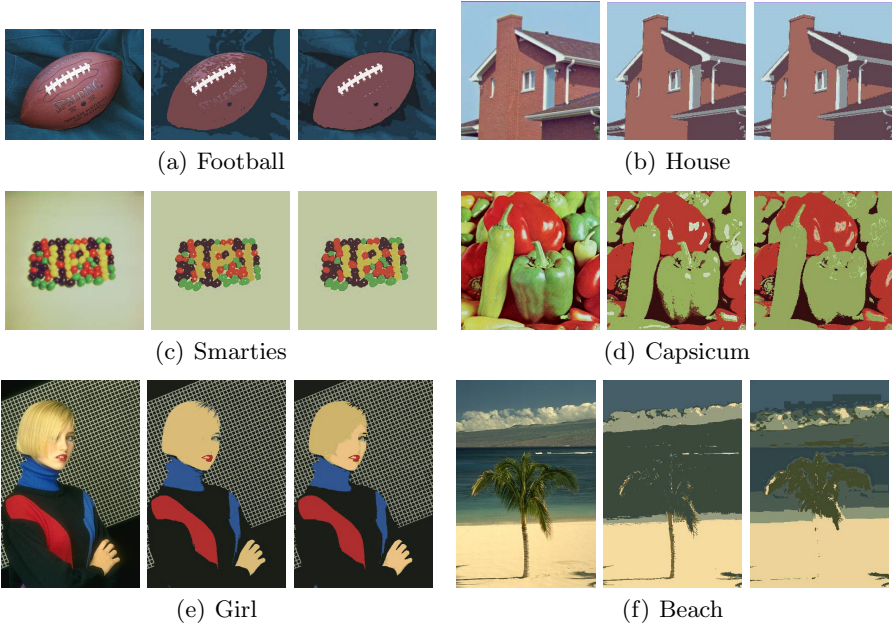


Fig. 3. Results for GHCM: original image (left), segmented image using V_{PC} (center); and segmented image using $Q(I)$ (right)

common in literature about unsupervised evaluation of image segmentation methods, they may produce some results (i.e., segmented images) which are not visually good/acceptable (specially for Fig. 3(d) and 3(f)). Despite of this, GHCM remains valid as a method adaptive selection of parameters since one can change the evaluation metric for a more suitable for a given task (or set of images).

4 Conclusion

In this paper, we present a method for improving the quality of segmentation results using a genetic algorithm to select the most suitable parameters in order to apply a graph-based approach for image segmentation. The main contribution of this work is a framework to optimize the quality assessment of the segmented image(s) according to some goodness method that could be selected according to a specific task (or image set).

Experimental results show that GHCM outperforms all other tested methods from the literature. But, in the future, more extensive experiments will be conducted to evaluate GHCM performance over different datasets. Another future line of work is to analyze the impact of the use of different metrics as empirical goodness methods.

References

1. Bezdek, J.C.: Cluster validity with fuzzy sets. *Journal of Cybernetics* 3(3), 58–73 (1973), <http://www.tandfonline.com/doi/abs/10.1080/01969727308546047>
2. Borsotti, M., Campadelli, P., Schettini, R.: Quantitative evaluation of color image segmentation results. *Pattern Recogn. Lett.* 19(8), 741–747 (1998)
3. Çiğla, C., Alatan, A.: Efficient graph-based image segmentation via speeded-up turbo pixels. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 3013–3016 (September 2010)
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision* 59, 167–181 (2004), <http://dl.acm.org/citation.cfm?id=981793.981796>
5. Garcia, A., Vachier, C.: Simplification of color images using semi-flat morphological operators and statistical metrics. In: Proceedings of the 16th IEEE International Conference on Image Processing, ICIP 2009, pp. 469–472. IEEE Press, Piscataway (2009), <http://dl.acm.org/citation.cfm?id=1818719.1818865>
6. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
7. Grygorash, O., Zhou, Y., Jorgensen, Z.: Minimum Spanning Tree Based Clustering Algorithms. In: 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2006, pp. 73–81 (November 2006)
8. Guimarães, S.J.F., Cousty, J., Kenmochi, Y., Najman, L.: A hierarchical image segmentation algorithm based on an observation scale. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) *SSPR&SPR 2012*. LNCS, vol. 7626, pp. 116–125. Springer, Heidelberg (2012)
9. Holland, J.H.: *Adaptation in natural and artificial systems*. MIT Press, Cambridge (1992)
10. Mikolov, T.: Color Reduction Using K-Means Clustering. In: 11th Central European Seminar on Computer Graphics (April 2007)
11. Morris, O., Lee, M.J., Constantinides, A.: Graph theory for image analysis: an approach based on the shortest spanning tree. *IEE Proceedings F Communications, Radar and Signal Processing* 133(2), 146–152 (1986)
12. Siang Tan, K., Mat Isa, N.A.: Color image segmentation using histogram thresholding - Fuzzy C-means hybrid approach. *Pattern Recogn.* 44(1), 1–15 (2011), <http://dx.doi.org/10.1016/j.patcog.2010.07.013>
13. Souza, K.J.F., Guimarães, S.J.F., Patrocínio Jr., Z., Araújo, A.D.A., Cousty, J.: A Simple Hierarchical Clustering Method for Improving Flame Pixel Classification. In: 2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 110–117 (November 2011)
14. Yu, Z., Au, O.C., Zou, R., Yu, W., Tian, J.: An adaptive unsupervised approach toward pixel clustering and color image segmentation. *Pattern Recogn.* 43(5), 1889–1906 (2010), <http://dx.doi.org/10.1016/j.patcog.2009.11.015>
15. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.* 20, 68–86 (1971)
16. Zhang, H., Fritts, J.E., Goldman, S.A.: Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding* 110(2), 260–280 (2008), <http://www.sciencedirect.com/science/article/pii/S1077314207001294>
17. Zhang, Y.J.: A review of recent evaluation methods for image segmentation. In: Sixth International, Symposium on Signal Processing and its Applications, vol. 1, pp. 148–151 (2001)