A Genetic Algorithm for Color Image Segmentation

Alessia Amelio and Clara Pizzuti

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

Abstract. A genetic algorithm for color image segmentation is proposed. The method represents an image as a weighted undirected graph, where nodes correspond to pixels, and edges connect similar pixels. Similarity between two pixels is computed by taking into account not only brightness, but also color and texture content. Experiments on images from the Berkeley Image Segmentation Dataset show that the method is able to partition natural and human scenes in a number of regions consistent with human visual perception. A quantitative evaluation of the method compared with other approaches shows that the genetic algorithm can be very competitive in partitioning color images.

1 Introduction

Image segmentation is an important problem in pattern recognition that aims at dividing an image into a number of regions having high homogeneity inside the same region, while adjacent regions are significantly dissimilar with respect to some adopted homogeneity measure. Many approaches have been proposed to segment monochrome and color images [7, 25, 27]. However, as observed by Cheng et al. [7], color image segmentation techniques are considered more appealing since they can provide more information than grey level images, and human eye is able to better detect objects when color is present. Most of these proposals, as outlined in [7], extend gray level image segmentation methods, such as histogram thresholding, boundary detection, region based approaches, with color representations. Furthermore, it has been recognized that no general algorithm exists for all monochrome and color images. Thus, techniques specialized for particular application domains have been presented [3–6, 12–14].

Among the several techniques proposed for image segmentation, methods representing an image as a graph [9, 20, 23, 24, 26], in which nodes correspond to pixels, and an edge between two pixels exists if they are similar, on the base of a suitably defined similarity criterion, revealed competitive both in terms of efficiency and segmentation quality [9]. In particular, Shi and Malik [20] introduced the concept of *normalized cut* that allows the partitioning of a graph in groups of nodes such that the homogeneity inside each region is maximized, while minimizing the dissimilarity between regions. More recently, Maji et al. [16] proposed the *biased normalized cut*, a modification of the normalized cut to incorporate priors which can be used for constrained color-texture based image segmentation.

In this paper a genetic algorithm for color image segmentation that adopts the representation of an image as a graph is proposed. The algorithm, named *C-GeNCut* (*Color*

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2013, LNCS 7835, pp. 314–323, 2013.

Genetic Normalized Cut), extends the approach proposed in [1] to take into account color, brightness, and texture, by computing the affinity between two pixels with the *Intervening Contour* cue [8, 10, 11, 15], that uses the *multispectral Pb detector* as defined by Arbelaez et al. in [2]. Experiments on ten color images from the Berkeley Image Segmentation Dataset (BSDS300) [17] show that the method is able to partition natural and human scenes in meaningful objects. A quantitative evaluation based on the well known concept of *Probabilistic Rand Index*, defined by Unnikrishnan et al. [21, 22], shows that the inclusion of color and texture improves the segmentation accuracy with respect to the algorithm of Amelio and Pizzuti [1], that considers only the gray-level information, as well as with the segmentations obtained by the algorithm of Maji et al. [16] for color images.

The paper is organized as follows. In the next section the problem of image segmentation is defined, together with its formalization as a graph partitioning problem, and a description of the homogeneity measure adopted. Section 3 describes fitness function, the genetic representation, and operators employed. Section 4 describes the evaluation measure used. Section 5 presents the experimental results. Finally, section 6 summarizes the approach.

2 Graph-Based Segmentation

An image R can be represented as a weighted undirected graph $G = (V, E, w)$, where V is the set of the nodes, E is the set of edges in the graph, 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 an edge (i, j) connects two pixels i and j, provided that these two pixels satisfy some property suitably defined that takes into account both pixel characteristics and spatial distance. The weight $w(i, j)$ associated with a graph 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 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, zero otherwise. Depending on the method adopted to compute the weights, any two pixels may or may not be connected.

2.1 Affinity Computation

In order to compute the weights, differently from [1], in this approach we employed the *Intervening Contour* cue [8, 10, 11, 15] that uses the multispectral *Pb* detector as defined in [2]. In this framework, given a generic pixel, the value of the multiscale *Pb* detector at that pixel is considered. If the maximum value 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 don't belong to the same segment. Hence, the weight $w(i, j)$ between these pixels will be low. On the other hand, if the value of the multiscale *Pb* detector is sufficiently weak, this usually happens in a region that is flat based on brightness, color and texture, the affinity between the two

pixels will be very high. More formally, the weight $w(i,j)$ between the pixels i and j is computed as:

$$w(i,j) = \begin{cases} e^{-max_{p \in line(i,j)}\{mPb(p)\}/\rho} & \text{if } ||X(i) - X(j)||_2 < r, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where *line(i,j)* is a straight line between i and j, *X(i)* is the spatial location of the pixel i, *r* is a distance threshold and ρ is a constant.

Multiscale *Pb* detector is based on the *Pb* detector function $Pb(x, y, \theta)$. Given an image pixel at position (x, y), it represents the posterior probability of a boundary with orientation θ at that pixel. This measure is obtained by evaluating the difference in local image brightness, color and texture channels.

Specifically, input image is transformed into four distinct channels. The first three channels are those of the CIE Lab colorspace: brightness, color a and color b. Color a represents the position of the color between red/magenta and green, while color b indicates the position of the color between yellow and blue. The last channel is related to the image texture content and it assigns to each pixel a texton id. Associations between pixels and texton ids come from another previous filtering stage. In particular, the input image is converted to grayscale and processed by a set of 17 Gaussian derivative and center-surround filters. Consequently, each pixel is represented by a 17-dimensional vector of responses, composed of one value from each filter. After that, these vectors are clustered by using $k - Means$: the cluster centers identify the image textons and each pixel is associated with the id in $[1, k]$ of the closest cluster center. Experiments provided 32 as a sufficient value for k. Finally, the texton channel is built, where each pixel of the original image is substituted by its corresponding texton id.

For each image channel, an oriented gradient signal $G(x, y, \theta)$ is computed at position (x, y), by placing a circular disc centered at location (x, y) and splitting it into two half-discs g and h by a diameter at angle θ. For each half-disc, an histogram of the intensity values of the pixels covered by it, is built. The gradient magnitude G at location (x, y) is defined by the χ^2 distance between the two half-disc histograms g and h.

$$\chi^2(g,h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)}$$

Furthermore, gradients at three scales $[\sigma/2, \sigma, 2\sigma]$ are considered for each channel, in order to detect fine and coarse image features.

The *Pb* detector processes the channels separately and then combines the oriented gradient signals obtained from the different channels at multiple scales into a single multiscale oriented signal:

$$mPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} G_{i,\sigma(i,s)}(x, y, \theta)$$

where s represents the scales index, i the feature channel index (brightness, color a, color b and texture) and $G_{i,\sigma(i,s)}(x, y, \theta)$ the oriented gradient signal at (x, y) in channel i where the radius of the disc is $\sigma(i, s)$ and the angle is θ. The parameters $\alpha_{i,s}$ weight the contribution of each gradient signal. The angle θ defining the orientation, takes eight

different values in the interval $[0, \pi)$. The final value of the multispectral *Pb* detector is the maximum response over the eight orientations:

$$mPb(x, y) = max_\theta\{mPb(x, y, \theta)\}$$

3 Algorithm

In this section we briefly describe the genetic operators and fitness function proposed in [1] and adopted also for *C-GeNCut*. The representation of individuals is based on the locus-based adjacency representation proposed in [18]. In this graph-based representation an individual of the population consists of N genes $g_1, \ldots, g_N$ and each gene can assume allele values j in the range $\{1, \ldots, N\}$. Genes and alleles represent nodes of the graph $G = (V, E, w)$ modelling an image, and a value j assigned to the ith gene is interpreted as a link between the pixels i and j. The initialization process assigns to each node i one of its neighbors j, and the kind of crossover operator adopted is uniform crossover. The mutation operator randomly assigns to each node i one of its neighbors. For both initialization and mutation, an important aspect to consider is the determination of the neighbors of each node. The concept of neighbors of a node introduced in [1] takes into account not only the spatial closeness but also the pixel affinity. More in details, given a generic node i in the graph, let $w^h_{max} = \{w^1, \ldots, w^h \mid w^1 \geq, \ldots, \geq w^h\}$ be the first h highest weights of row i in the weight adjacency matrix W.

The h nearest neighbors of i, denoted as nn^h_i, are then defined as $nn^h_i = \{j \mid w(i, j) \in w^h_{max}\}$. nn^h_i is thus the set of those pixels that are no more than r pixels apart from i, and that have maximum similarity with i. It is worth to note that, even if h is fixed to 1, the number of nearest neighbors of i could be sufficiently large if many of its spatial neighbors have the same maximum weight w^h_{max}. This definition of nearest neighbors guaranties to choose the most similar neighbors during the initialization process, and to bias the effects of the mutation operator towards the most similar neighbors, thus it contributes to improve the results of the method.

The fitness function is an extension of the concept of normalized cut of Shi and Malik [20]. 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 $R \in P$, let

$$c_r = \textstyle\sum_{i \in R, j \notin R} W_{ij} \qquad m_r = \textstyle\sum_{i \in R, j \in R} W_{ij} \qquad m = \textstyle\sum_{i \in V, j \in V} W_{ij}$$

be respectively the sum of weights of edges on the boundary of R, 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 total edge weight connections to all the nodes in the graph

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

Because of the affinity measure w defined in the previous section, 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 $WNcut$ are preferred.

4 Evaluation Measure: Probabilistic Rand Index

In the Berkeley dataset, for each image, multiple human-traced segmentations for color images are available. All the segmentations are considered equally reliable. As observed in [22], 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 et al. [21, 22] introduced the *Probabilistic Rand Index* as an extension of the concept of *Rand Index* [19], 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 :

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

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. The PRI value varies between 0 and 1. When its value is 0 it means that S and $\{S_1, \ldots, S_T\}$ are completely dissimilar.

5 Experimental Results

In this section we present the results of *C-GeNCut* on ten images from the Berkeley Image Segmentation Dataset (BSDS300) [17] and compare the performances of our algorithm in partitioning natural and human scenes in meaningful objects with the segmentations obtained by the algorithm of Maji et al. [16] (*Biased NCut*) for color images, in the following referred as *C-NCut*, and by the algorithm of Amelio and Pizzuti [1] (*GeNCut*), that takes into account only grayscale information, on the same images.

The version of the *Biased NCut* software is written in MATLAB and it is available at http://ttic.uchicago.edu/ smaji/projects/biasedNcuts/. However we eliminated the interactive mode from the available algorithm specifically for performing comparisons with our technique.

The *C-GeNCut* algorithm has been 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 has been employed and then the parameter values giving good results for the benchmark images have been selected. Thus we set crossover rate to 0.9, mutation rate to 0.2, elite reproduction 10% of the population size, roulette selection function. The population size was 100, the number of generations 60. The value h of nearest neighbors to consider has been 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 data sets, the statistical significance of the results produced by *C-GeNCut* has been 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 *C-GeNCut* could be obtained by chance is very low.

The weight matrix of each image is computed in the same way for both *C-NCut* and *C-GeNCut* methods, and, as already described in section 2, it is based on the Intervening

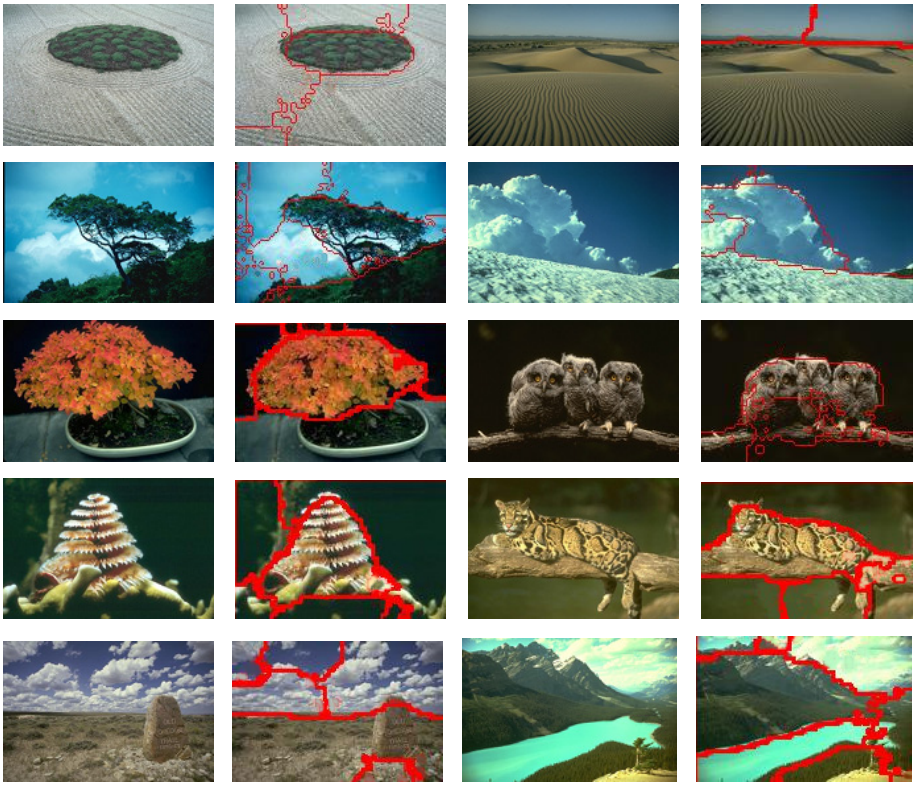

Fig. 1. Segmentation of *C-GeNCut* on ten images of the Berkeley Image Segmentation dataset (BSDS300). For each image, the original version together with the segmentation result of *C-GeNCut* are presented.

Contour framework that uses the multiscale *Pb* detector by fixing $r = 5$ and $\rho = 0.1$. About the *Pb* detector, the parameter σ, which defines the scales, is set to 5 pixels for the brightness channel, while for color and texture channels σ is set to 10 pixels. The parameters $\alpha_{i,s}$ are fixed to $0.01, 0.01, 0.02, 0.02, 0.02, 0.03, 0.02, 0.02, 0.02, 0.01, 0.01$ and 0.01. The weight matrix of each image is computed for *GeNCut* as in [1], by fixing $r = 10$, number of scales 3, number of orientations 4 and $\sigma = 0.1$. The value h of nearest neighbors to consider has been fixed to either 1 or 2.

In order to compare *C-GeNCut* and *C-NCut*, given an image I, we executed *C-NCut* as many times as the different number of segmentations available for I, by giving as input the distinct values k of ground-truth segments corresponding to the color human-segmentations. This implies that *C-NCut* has been executed for the best input parameter value k. For each image and for each distinct number of segments from the ground-truth segmentations, *C-NCut* has been run 10 times. The average values of the Probabilistic Rand Index (PRI) [21] have been computed, together with the standard deviation, for the partitioning found by *C-NCut* and *GeNCut*, and compared with that obtained

Table 1. Probabilistic Rand Index evaluated for *C-NCut*, *C-GeNCut* and *GeNCut* on a subset of the Berkeley Image Segmentation Dataset (BSDS300). PRI represents the Probabilistic Rand Index, nc the number of segments.

	GeNCut		***C-GeNCut***		***C-NCut***	
	nc	PRI	nc	PRI	nc	PRI
I_1	11	0.6443 (0.0637)	5	0.7526 (0.0263)	5	0.7299 (0.0003)
					24	**0.7858** (0.0008)
					4	0.7333 (0.0001)
					23	**0.7862** (0.0003)
					41	**0.7858** (0.0002)
					17	**0.7856** (0.0006)
I_2	8	0.7035 (0.0081)	3	**0.7613** (0.0063)	5	0.6728 (0)
					3	0.7068 (0.0001)
					8	0.6794 (0.0078)
					7	0.6799 (0.0001)
					9	0.6774 (0.0001)
I_3	13	0.7041 (0.0183)	11	**0.7052** (0.0183)	11	0.6545 (0.0001)
					16	0.6446 (0.0001)
					30	0.6202 (0.0007)
					10	0.6651 (0.0017)
					26	0.6235 (0.0009)
I_4	16	0.7889 (0.0368)	5	**0.8339** (0.0213)	7	0.7277 (0.0003)
					3	0.8200 (0.0001)
					5	0.8260 (0.0001)
					8	0.7082 (0.0144)
					6	0.8047 (0.0001)
I_5	13	0.8088 (0.0198)	7	0.8235 (0.0065)	7	0.8168 (0.0001)
					6	**0.8338** (0.0001)
					10	0.7977 (0.0001)
					8	0.8191 (0.0060)
					15	0.7861 (0.0001)
I_6	4	0.8036 (0.0186)	6	**0.8288** (0.0379)	13	0.7565 (0.0001)
					6	0.7455 (0.0003)
					19	0.7405 (0.0001)
					23	0.7405 (0.0013)
I_7	9	0.7308 (0.0118)	6	**0.7820** (0.0101)	6	0.7512 (0.0018)
					4	0.7345 (0.0001)
					10	0.7101 (0.0062)
					2	0.5763 (0)
I_8	10	0.8215 (0.0002)	5	**0.8361** (0.0163)	5	0.7827 (0.0001)
					8	0.8217 (0.0001)
					7	0.8122 (0.0001)
					10	0.8109 (0.0012)
I_9	18	0.7425 (0.0059)	6	**0.7653** (0.0076)	6	0.7399 (0.0001)
					5	0.7352 (0.0001)
					3	0.7114 (0.0001)
					9	0.7212 (0.0001)
					8	0.7041 (0.0001)
					28	0.6825 (0.0009)
I_{10}	8	0.7797 (0.0375)	8	0.8361 (0.0075)	39	0.8339 (0.0005)
					10	**0.8557** (0.0001)
					8	**0.8446** (0.0001)
					15	**0.8471** (0.0001)
					26	**0.8405** (0.0001)
					24	**0.8447** (0.0001)

by *C-GeNCut* on the same images. Since *C-GeNCut* and *GeNCut* generate a single segmentation, the average value of PRI is the same for each of the values of ground-truth segments of the image under consideration.

Table 1 reports the PRI for *C-GeNCut*, *C-NCut* and *GeNCut*. In particular, for each image I we computed the PRI value of the segmentation returned by *C-GeNCut* and by *GeNCut*, considered as test segmentation, against the set of ground-truth segmentations associated with I in the Berkeley dataset. As regards *C-NCut*, the PRI values have been computed by considering as test segmentation that obtained by *C-NCut* for each of the executions performed, i.e. one for the input parameter k fixed to the number of segments obtained by *C-GeNCut*, and one for every distinct value of ground-truth segments available for the image under consideration.

For example, if we consider image I_1, for which five human segmentations are available, *C-GeNCut* found a segmentation of 5 segments with PRI value equal to 0.7526, while *GeNCut* obtained 11 regions and PRI value 0.6443. The PRI values for *C-NCut* are 0.7299, 0.7858, 0.7333, 0.7862, 0.7858, 0.7856 when as test segmentations are used those obtained for input parameter k equal to 5, which is the number of segments returned by *C-GeNCut*, and 24, 4, 23, 41, and 17, respectively, that correspond to the distinct number of segments from the ground-truth segmentations. Note that, for images I_2, I_4, I_6, I_7, and I_{10} *C-GeNCut* found a number of segments equal to one of the ground-truth segmentations.

The table points out that the PRI value of *C-GeNCut* is always higher than the corresponding PRI value of *GeNCut*. Furthermore, the PRI value of *C-GeNCut* is the highest for seven out of the ten images, i.e. for I_2, I_3, I_4, I_6, I_7,I_8, and I_9, also with respect to *C-NCut*. As regards the other three images, *C-NCut* overcomes *C-GeNCut* on I_1 for 4 out of 6 segmentation values, on I_5 for 1 out of 5 segmentation values, and on I_{10} for 5 out of 6 segmentation values. *C-GeNCut* thus improves the results of the genetic approach when color and texture information are included, and it is competitive with respect to *C-NCut*, that is specialized for color images.

Finally, in Figure 1, for each of the ten images, we present the segmentation outputs of *C-GeNCut* by depicting the contours of the regions on the original image. The visual perception of the segmentation results is quite positive: the main objects of a scene are identified and the most meaningful features extracted from the images by the segmentation process.

6 Conclusions

The paper presented a graph-based approach to image segmentation that employs genetic algorithms. The method extends the method proposed in [1], by considering not only brightness but also color and texture for image segmentation. The method revealed particularly apt to deal with color-texture images modeled as graphs. In fact, as experimental results showed, the genetic approach is able to segment color-texture images in a number of regions that well adhere to the human visual perception, and it is competitive with state-of-the-art methods for color image segmentation.

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

References

1. Amelio, A., Pizzuti, C.: An Evolutionary and Graph-Based Method for Image Segmentation. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 143–152. Springer, Heidelberg (2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Anal. Mach. Intell. 33(5), 898–916 (2011)
3. Ballerini, L., Bocchi, L., Johansson, C.B.: Image Segmentation by a Genetic Fuzzy c-Means Algorithm Using Color and Spatial Information. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 260–269. Springer, Heidelberg (2004)
4. Bevilacqua, V., Mastronardi, G., Piazzolla, A.: An Evolutionary Method for Model-Based Automatic Segmentation of Lower Abdomen CT Images for Radiotherapy Planning. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 320–327. Springer, Heidelberg (2010)
5. Chaabane, S.B., Sayadi, M., Fnaiech, F., Brassart, E.: Dempster-shafer evidence theory for image segmentation: Application in cells images. International Journal of Information and Communication Engineering 5(2), 126–132 (2009)
6. Chen, C.W., Luo, J., Parker, K.J.: Image segmentation via adaptive k-means clustering and knowledge-based morphological operations with biomedical applications. IEEE Transactions on Image Processing 7(12), 1673–1683 (1998)
7. Cheng, H., Jiang, X., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition 34, 2259–2281 (2001)
8. Cour, T., Bénézit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)), pp. 1124–1131 (2005)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
10. Fowlkes, C., Malik, J.: How much does globalization help segmentation. Tech. rep. (2004)
11. Fowlkes, C., Martin, D., Malik, J.: Learning Affinity Functions for Image Segmentation: Combining Patch-based and Gradient-based Approaches. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2003)
12. Ghosh, P., Mitchell, M.: Segmentation of medical images using a genetic algorithm. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 1171–1178. ACM (2006)
13. Harrabi, R., Braiek, E.B.: Color image segmentation using multi-level thresholding approach and data fusion techniques: application in the breast cancer cells images. Eurasip Journal of Image and Video Processing 11 (2012)
14. Kim, S.-M., Kim, W.: An Algorithm for Segmenting Gaseous Objects on Images. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 322–328. Springer, Heidelberg (2004)
15. Leung, T., Malik, J.: Contour Continuity in Region Based Image Segmentation. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 544–559. Springer, Heidelberg (1998)
16. Maji, S., Vishnoi, N.K., Malik, J.: Biased normalized cuts. In: CVPR, pp. 2057–2064. IEEE (2011)

17. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)
18. Park, Y., Song, M.: A genetic algorithm for clustering problems. In: Proc. of 3rd Annual Conference on Genetic Algorithms, pp. 2–9 (1989)
19. Rand, W.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)
20. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
21. Unnikrishnan, R., Hebert, M.: Measures of similarity. In: Seventh IEEE Workshops on Application of Computer Vision, WACV/MOTIONS 2005, vol. 1 (2005)
22. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Towards objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 929–944 (2007)
23. Urquhart, R.: Graph theoretical clustering based on limited neighborhood sets. Pattern Recognition 15(3), 173–187 (1982)
24. Wu, Z., Leahy, R.: 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 (1993)
25. Xu, Y., Olman, V., Uberbacher, E.C.: A segmentation algorithm for noisy images: Design and evaluation. Pattern Recognition Letters 19, 1213–1224 (1998)
26. Zahn, C.T.: Graph theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers 20(1), 68–86 (1971)
27. Zhang, Y.: Evaluation and comparison of different segmentation algorithm. Pattern Recognition Letters 18, 963–974 (1997)