Image Segmentation Using Automatic Seeded Region Growing and Instance-based Learning

Octavio Gómez, Jesús A. González, and Eduardo F. Morales

National Institute of Astrophisics, Optics and Electronics
Computer Science Department,
Luis Enrique Erro Num 1, Puebla, México
{gomezo,jagonzalez,emorales}@ccc.inaoep.mx
http://ccc.inaoep.mx

Abstract. Segmentation through seeded region growing is widely used because it is fast, robust and free of tuning parameters. However, the seeded region growing algorithm requires an automatic seed generator, and has problems to label unconnected pixels (unconnected pixel problem). This paper introduces a new automatic seeded region growing algorithm called ASRG-IB1 that performs the segmentation of color (RGB) and multispectral images. The seeds are automatically generated via histogram analysis; the histogram of each band is analyzed to obtain intervals of representative pixel values. An image pixel is considered seed if its gray values for each band fall in some representative interval. After that, our new seeded region growing algorithm is applied to segment the image. This algorithm uses instance-based learning as distance criteria. Finally, according to the user needs, the regions are merged using ownership tables. The algorithm was tested on several leukemia medical images showing good results.

Key words: Image Segmentation, Seeded Region Growing, Instance-based learning, Color image, Multispectral image

1 Introduction

The image segmentation process consists in grouping parts of an image into units that are homogeneous with respect to one or more characteristics [2]. Image segmentation also can be viewed as a process of pixel classification in the sense that all pixels that belong to the same region are assigned the same label [6]. Automatic image segmentation is a fundamental step in many image processing applications such automatic object recognition, because it allows to separate areas of interest of an image and, consequently, reduce the processing effort.

To perform image segmentation there exist five main approaches: thresholding techniques [12], boundary-based methods [9], region-based methods [11] clustering-based techniques [8] and hybrid techniques [4]. A good review of this approaches can be found in [3]. Despite the numerous segmentation algorithms that have been proposed in the literature, image segmentation is still subject of research, and is not possible to state that the segmentation problem has been solved because of the diversity of applications [17].

Seeded region growing (SRG) is a hybrid method proposed by R. Adams and L. Bischof [1]. This method starts with a set of n initial seeds $A_1, A_2, \ldots, A_n$, and, at each steep, it grows the seeds A_i by merging a pixel x into its nearest neighboring seed region A_i. This algorithm is fast, robust, and free of tuning parameters [6], nevertheless, the algorithm does not generate seeds automatically, and also has problems to label unconnected pixels [6] (the unconnected pixel problem). To deal with the first problem, F. Shih and S. Cheng [14] proposed an automatic seeded region growing algorithm for color image segmentation. The algorithm transforms the input RGB image into a YC_bC_r color space, and selects the initial seeds considering a $3X3$ neighborhood and the standard deviation of the Y, C_b and C_r components. Afterwards, the seeds are grown to segment the image. Finally, region merging is used to merge similar or small regions. In [6] tree methods to automatically generate seeds are proposed. The first one partitions the image into a set of rectangular regions with fixed size and selects the centers of these rectangular regions as the seeds. The second method finds the edges of the image and obtains the initial seeds from the centroid of the color edges. Finally, the third method extends the second method to deal with noise applying an image smoothing filter. A. Tremeau and N. Borel [16] present a color segmentation algorithm that combines region growing and region merging. The algorithm starts with the region growing process taking into account color similarity and spatial proximity, afterwards, the resulting regions are merged on the basis of a criterion that only takes into account color similarity.

This paper introduces a new automatic seeded region growing algorithm called ASRG-IB1 that performs the segmentation of color (RGB) and multispectral images. First, homogeneous seeds are automatically obtained via histogram analysis. The histogram of each band is analyzed to obtain a set of representative pixel values, and the seeds are generated with all the image pixels with representative gray values (section 4.1). Second, a modified seeded region growing algorithm is applied to perform the segmentation. This algorithm makes use of instance-based learning as similarity criteria. Finally, according to user needs, the regions are merged using ownership tables.

This paper is organized as follows. Section 2 gives an overview of the original seeded region growing algorithm and Section 3 gives an overview of instance based learning. In Section 4 our proposed algorithm is described. In Section 5 the experimental results are presented and in Section 6 we present the main conclusions of this work.

2 Seeded Region Growing

To begin, the seeded region growing algorithm needs n seeds $A_1, A_2, \ldots, A_n$. The decision of what is a feature of interest is embedded in the choice of seeds [1]. Let T be the set of all unallocated (non labeled) pixels that border at least one A_i region after m iterations:

$$T = \left\{ x \notin \bigcup_{i=1}^{n} A_i | N(x) \cap \bigcup_{i=1}^{n} A_i \neq \oslash \right\}$$

where $N(x)$ is the second-order neighborhood (8-neighbors) of pixel x. If we have that $N(x)$ intersects only one labeled region A_i, then, we define the label $i(x) \in \{1, 2, \ldots, n\}$ to be an index such that:

$$N(x) \cap A_{i(x)} \neq \oslash$$

If we have that $N(x)$ meets two or more regions A_i then we define $\delta(x, A_i)$ to be a measure of how different is x from the region A_i that $N(x)$ intersects:

$$\delta(x, A_i) = |g(x) - mean_{y \in A_{i(x)}}[g(y)]|$$

were $g(x)$ is the gray value of the pixel x. The value of $i(x)$ will be the value of i such that $N(x)$ meets A_i and $\delta(x)$ is minimized:

$$i(x) = \{i | N(x) \cap A_i \neq \oslash \wedge \delta(x) is\, the\, minimum\}$$

3 Instance-based Learning

3.1 Learning Task and Framework

Instance-based learning algorithms are derived from the nearest neighbor pattern classifier. This kind of algorithms store and uses only selected instances to generate classification predictions by means of a distance function. The learning task of these algorithms is supervised learning from examples.

Each instance is represented by a set of attribute-value pairs, and all instances are described by the same set of n attributes. This set of n attributes defines an n-dimensional instance space. One of the attributes must be the category attribute and the other attributes are predictor attributes.

The primary output of an Instance-based learning algorithm is a function that maps instances to categories called concept description; this concept description includes a set of stored instances and, possibly, information about the classifiers past performance. The set of stored instances can be modified after each training instance is processed. All Instance-based learning algorithms are described by the following three characteristics:

1. *Similarity function:* computes the similarity between a training instance i and the instances stored in the concept description. The similarities are numerical-valued.
2. *Classification function:* This function receives the results of the similarity function and the performance records stored in the concept description. It yields to a classification for the training instance i.

3. *Concept description updater:* Keeps the records of classification performance and decides the instances to be included in the concept description. It yields to a modified concept description.

Similarity and classification functions determine how the instances stored in the concept description are used to predict the category of the training instance i.

3.2 IB-1 Algorithm

IB-1 is the simplest Instance-based learning algorithm. The distance function that it uses is:

$$Distance(x, y) = \sqrt{\sum_{i=1}^{n} f(x_i - y_i)^2}$$

where x is a test instance, y is a trainig instance and x_i is the value of the $i - th$ attribute of instance x. The instances are described by n features. The IB-1 algorithm is presented in Table 1.

Table 1. IB-1 Algorithm (CD = Concept Description)

$CD \leftarrow \oslash$
For each $x \in$ Training set **do**
1. **For each** $y \in CD$ **do**
$Dist[y] \leftarrow Distance(x, y)$
2. $Mdist \leftarrow$ the $y \in CD$ with minimum $Dist[y]$
3. $class(x) = Mdist$
4. $CD \leftarrow CD \cup x$

To label an instance, the IB-1 algorithm computes the distance between the test instance and the instances stored in the concept decription, and stores the nearest instance. The class of the test instance will be the class of the nearest instance.

4 SRG-IB1 Segmentation Algorithm

4.1 Automatic Seed Generation

An overwiew of the automatic seed generation algorithm is shown in Fig. 1.

The first step divides the histogram in subintervals. Let $h_b(p)$ be the histogram function, this function receives a gray value p ($0 \leq p \leq 255$) and returns the number of pixels of band b with gray value equal to p. To divide the histogram we must find the *cut points*. All the gray values p that satisfy the next two conditions will be taken as cut points:

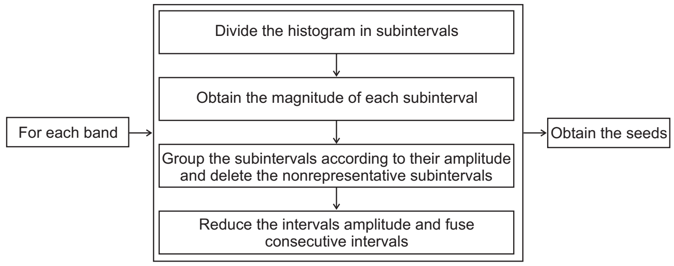

Fig. 1. Overwiew of the automatic seed generation algorithm

1. $h_b(p-1) \geq h_b(p)$
2. $h_b(p+1) > h_b(p)$

The cut points indicate the end and the beginning of each subinterval. Table 2 shows the subintervals S_j obtained from a given histogram function $h_b(p)$ with q cut points, where C_i is a cut point $(1 \leq i \leq q)$, S_j is a subinterval $(1 \leq j \leq m)$ and m is the number of resultant subintervals.

Table 2. Subintervals S_j obtained from a given histogram function $h_b(p)$ with q cut points

$S_1 = [0, C_1]$
$S_2 = [C_1 + 1, C_2]$
$S_3 = [C_2 + 1, C_3]$
$\ldots$
$S_m = [C_q, 255]$

The second step obtains the amplitude of each subinterval. For a given subinterval $S_j = [S_{j,1}, S_{j,2}]$ the amplitude is given by:

$$amp(S_j) = S_{j,2} - S_{j,1} + 1$$

The third steep groups the subintervals according to their amplitude to delete the non representative subintervals. For all subintervals S_j with amplitude $amp(S_j) = \alpha$, the most representative subinterval is the one with the largest amplitude:

$$mrs(\alpha) = \arg \max_{\forall S | amp(S) = \alpha} amp(S)$$

A subinterval S_j is nonrepresentative if:

$$amp(S_j) \leq \frac{1}{2} mrs(\alpha)$$

The fourth step reduces the representative intervals amplitude. For a given representative subinterval $S_j = [S_{j,1}, S_{j,2}]$ of band b, the most representative gray value is:

$$mrg(S_j) = \arg \max_{\forall S_{j,1} \leq \beta \leq S_{j,2}} h_b(\beta)$$

A gray value γ of a representative subinterval S_j of band b is representative if:

$$h_b(\gamma) > \frac{1}{2} mrg(S_j)$$

All the nonrepresentative gray values must be removed from the interval, producing a reduced interval.

Depending of the application, the consecutive resultant reduced intervals can be merged. For example, the reduced intervals [12-18], [19-25] produces the new merged interval [12-25]. Interval merging lower the quantity of homogeneous seeds, and must be avoided if the application needs the highest separation between seeds (i.e. the user needs the maximum level of homogeneity in the regions).

The final step is to generate the seeds. A pixel x is considered as a seed if its gray values on each band fall inside a representative interval of the same band. If the gray values of two seed pixels fall inside the same representative intervals, the pixels will be labeled with the same region ID. The output of the seed generator is a set with n seeds $A_1, A_2, \ldots, A_n$.

4.2 Region Growing and Instance-based Learning

The region growing algorithm is shown in Fig 2. The automatically generated seeds are used to construct the classifier using the region ID as the class of the pixel. Before the region growing step, the sets of pixels to label P and unallocated (non labeled) pixels Q must be defined. All the seeds must be grouped according to their region ID (region sets R). The region growing step obtains the pixels that must be labeled (set P) and updates the set Q. We use the IB1 classifier to label the regions. Because all the pixels are considered without concerning what regions they meet, pixels that in the original seeded region growing algorithm can not be reached by the region to which they belong (unconnected pixel problem) are labeled. After labeling, the IB1 classifier must be updated to consider the newly labeled instances. The algorithm stops when set Q is empty.

4.3 Ownership Tables and Region Merging

In many real world applications the user may need the segmentation of an image over different levels of abstraction, for example, in remote sensing, several regions can form a concept (a region with a specific semantic for the user), and these concepts can be merged to form a higher level concept.

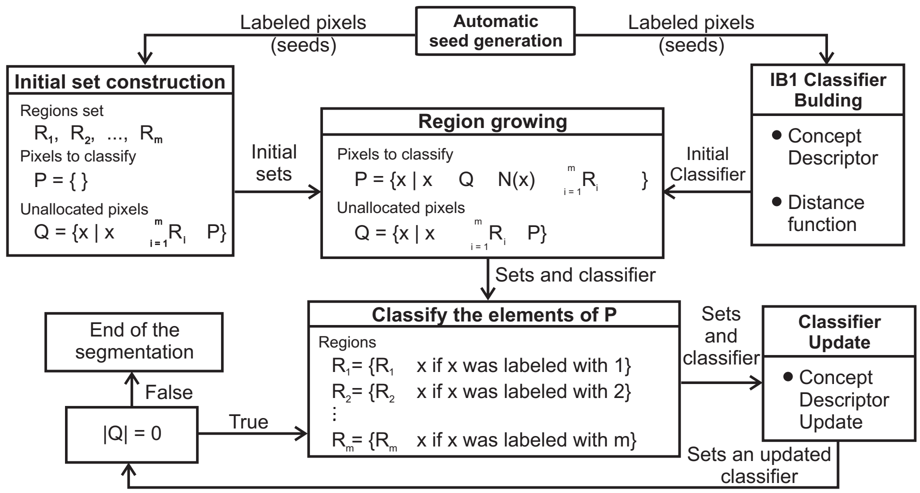

Fig. 2. Region growing algorithm.

At this point, the algorithm has obtained the homogeneous regions of the image, these regions represent a segmentation at the lowest level of abstraction. To complete the task it is necessary to merge the regions according to the user needs.

Ownership tables allow the user to merge regions according to his needs. The user manually selects the regions that must be merged and those regions ID's are stored in a table. An ownership table indicates which regions must be merged to form the concept that the user wants, so, the concept must be completely defined by its ownership table, and distinct concepts can not have the same table. The elements of an ownership table can be of two kinds, ambiguous and unambiguous. The unambiguous elements are regions that only belong to one ownership table and ambiguous elements can belong to two or more tables.

Fig. 3(a) represents a white cell blood with cytoplasm in the bottom. Fig. 3(b) shows the result of the proposed algorithm SRG-IB1. Finally, Fig. 3(c) shows the result after the user-guided region merging trough ownership tables. An example of an ownership table is shown in Fig. 4.

5 Experimental Results

This section shows the results of the proposed algorithm on RGB leukemia medical images. Leukemia is a cancer of the blood characterized by an abnormal proliferation of white blood cells (leukocytes). Experiments were made over thirty distinct images, with the objective of segmenting white blood cells of the image to study their characteristics and determine if a given patient has leukemia.

There is not a generally accepted methodology (in the field of computer vision) which elucidates on how to evaluate segmentation algorithms [10], [15].

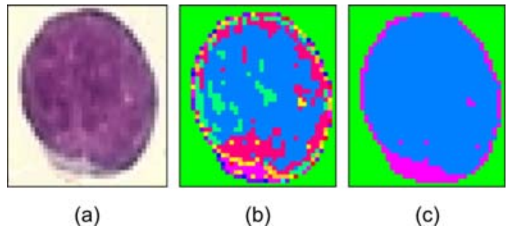

Fig. 3. (a) RGB image of a white blood cells with cytoplasm. (b) Image segmented with SRG-IB1. (c) Image segmented after region merging

White corpuscle		Cytoplasm	
	Region 1		Region 4
	Region 2		Region 5
	Region 3		Region 6
			Region 7
			Region 8
			Region 9
			Region 10

Fig. 4. Ownership table for Figure 2(c)

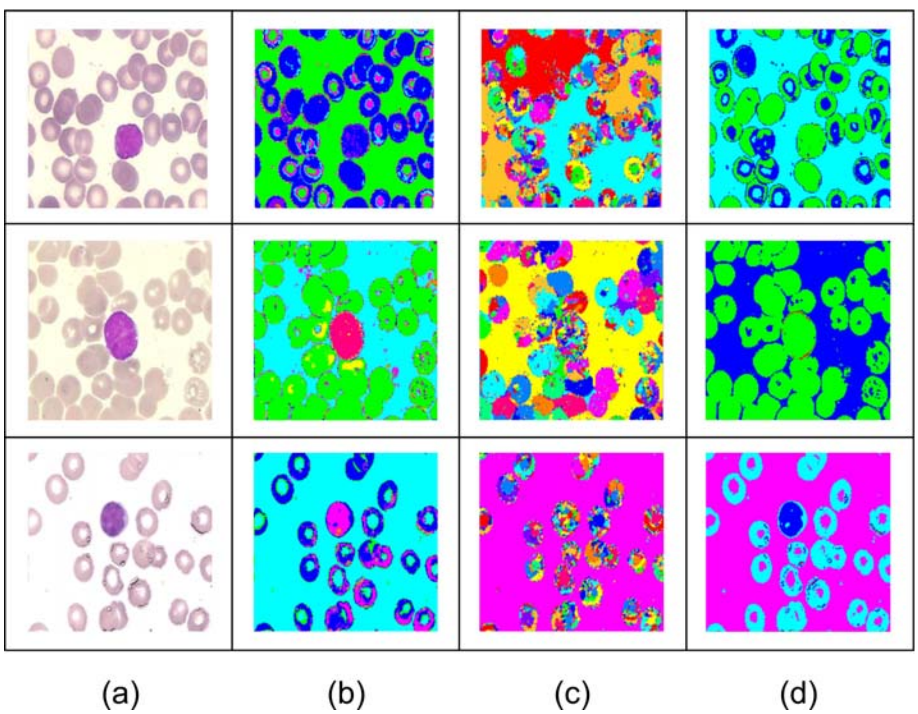

Fig. 5. (a) Original RGB image. (b) Image segmented with SRG-IB1. (c) Image segmented with region growing. (d) Image segmented with auto threshold

Comparing different segmentation algorithms with each other is difficult mainly because they differ in the properties they try to satisfy. Segmentation quality assessment requires one manually generated segmentation (for reference) plus computer-generated segmentations corresponding to different image segmentation algorithms or algorithm parameter settings [10]. In this domain, is difficult to find or generate manual segmentations so, the most common method for segmentation quality evaluation is a visual inspection made by domain experts.

For this comparison we used the HALCON [7] implementations of the region growing algorithm, and the implementation of the auto threshold algorithm. Auto threshold segments images using multiple thresholding. First, the relative histogram of the gray values are determined, then, relevant minima are extracted from the histogram, which are used successively as parameters for a thresholding operation. The thresholds used are 0, 255, and all minima extracted from the histogram (after the histogram has been smoothed). For each gray value interval one region is generated. The number of regions is the number of minima + 1.

Results are shown in Fig. 5. It can be observed that the proposed algorithm (b) improves the original instance based algorithm (c) that oversegments the image (a). The segmentations results of the proposed algorithm are highly competitive with respect to auto-threshold, even more, the proposed algorithm can find more homogeneus regions and allows the user to define a concept hierarchy by means of ownership tables.

6 Conclusions

We presented a new automatic seeded region growing algorithm that makes use of instance based learning as its distance criteria. This algorithm preserves all the advantages of the original SRG algorithm; furthermore, we presented a novel method for automatic seed generation via histogram analysis, and a region growing scheme that eliminates the unconnected pixel problem when considering all pxeles to label like a single set. Instance based learning is the most suited machine learning algorithm for this task because at each growing step the algorithm is updated, opossed to other algorithms that construct an explicit representation of the training data, and the representation is not updated during the classification step. Finally, ownership tables allow adjusting the segmentation result to the user needs, and make possible the definition of levels of abstraction to represent a concept hierarchy.

Acknowledges The first author acknowledges to CONACYT the support provided through the grant for Master's studies number 201804. The first author also acknowledges to Erika Danaé López Espinoza for his valuable comments.

References

1. Adams, R., Bischof, L.: Seeded Region Growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16. IEEE Computer Society, Los Alamitos California (1994) 641–647

2. Ballard, D.H., Brown, C.M.: Computer Vision. 1st edition, Prentice-Hall, Boston Massachusetts (1982)
3. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition, Vol. 34. Elsevier, Amsterdam (2001) 2259–2281
4. Cheng, H.D., Jiang, X.H., Wang, J.: Color image segmentation based on homogram thresholding and region merging. Pattern Recognition, Vol. 35. Elsevier, Amsterdam (2002) 373–393
5. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. Machine Learning: Proceedings of the Twelfth International Conference, Vol. 12. Morgan Kaufmann, San Francisco (1995) 194–202
6. Fan, J., Zeng, G., Body, M., Hacid, M.: Seeded region growing: and extensive and comparative study. Pattern Recognition, Vol. 26. Elsevier, Amsterdam (2005) 1139–1156
7. MVTec Software GmbH. Halcon: Machine vison software for business applications. MVTec Software GmbH, Munchen Germany (2007)
8. Jeon, B., Jung, Y., Sang, K.: Image segmentation by unsupervised sparse clustering. Pattern Recognition Letters, Vol. 27. Elsevier, Amsterdam (2005) 1139–1156
9. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Proceedings 1st International Conference on Computer Vision. International Journal of Computer Vision, Vol. 1. Springer-Verlag, Netherlands (1988) 321–331
10. Paglieroni, D.W.: Design considerations for image segmentation quality assessment measures. Pattern Recognition, Vol. 37. Elsevier, Amsterdam (2004) 1607–1617
11. Pichel, J.C., Singh D.E., Rivera, F.F.: Image segmentation based on merging suboptimal segmentations. Pattern Recognition Letters, Vol. 27. Elsevier, Amsterdam (2006) 1105–1116
12. Quiao, Y., Hu, Q., Qian, G., Luo, S., Nowinski W.L.: Thresholding based on variance and intensity contrast. Pattern Recognition, Vol. 40. Elsevier, Amsterdam (2007) 596–698
13. Quinlan, J.R.: Induction of Decision Trees. Machine Learning, Vol. 1. Springer-Verlag, Netherlands (1986) 81–106
14. Shih, F.Y., Cheng, S.: Automatic seeded region growing for color image segmentation. Image and Vision Computing, Vol. 23. Elsevier, Amsterdam (2005) 877–886
15. Siebert, A.: Dynamic Region Growing. Vision Interface 97. Massachusetts Institute of Technology, Cambridge Massachusetts (1997)
16. Tremeau, A., Borel, N.: A region growing and merging algorithm to color image segmentation. Pattern Recognition, Vol. 30. Elsevier, Amsterdam (1997) 1191–1203
17. Zouagui, T., Benoit-Cattin, H., Odet, C.: Image segmentation functional model. Pattern Recognition, Vol. 37. Elsevier, Amsterdam (2004) 1785–1795