Correspondence

Seeded Region Growing

Rolf Adams and Leanne Bischof

Abstract— We present here a new algorithm for segmentation of intensity images which is robust, rapid, and free of tuning parameters. The method, however, requires the input of a number of seeds, either individual pixels or regions, which will control the formation of regions into which the image will be segmented. In this correspondence, we present the algorithm, discuss briefly its properties, and suggest two ways in which it can be employed, namely, by using manual seed selection or by automated procedures.

Index Terms— Image segmentation, region growing, semiinteractive image processing, watershed.

I. INTRODUCTION

In this correspondence, we deal with the segmentation of intensity images in which the individual objects or regions in the image are characterized by connected pixels of similar value. Thus, the method presented may not be applicable to highly textured images or to range images. It may be applied to images affected by lighting variation but only after suitable preprocessing (i.e., conventional background removal techniques).

For the segmentation of intensity images, there are four main approaches [3], [8], namely, threshold techniques, boundary-based methods, region-based methods, and hybrid techniques which combine boundary and region criteria. Threshold techniques [13] are based on the postulate that all pixels whose value (gray level, color value, or other) lie within a certain range belong to one class. Such methods neglect all of the spatial information of the image and do not cope well with noise or blurring at boundaries.

Boundary-based methods [6] use the postulate that the pixel values change rapidly at the boundary between two regions. The basic method here is to apply a gradient operator such as the Sobel or Roberts filter [2]. High values of this filter provide candidates for region boundaries, which must then be modified to produce closed curves representing the boundaries between regions. Converting the edge pixel candidates to boundaries of the regions of interest is a difficult task.

The complement of the boundary-based approach is to work with the regions [16]. Region-based methods rely on the postulate that neighboring pixels within the one region have similar value. This leads to the class of algorithms known as region growing of which the "split and merge" technique [9] is probably the best known. The general procedure is to compare one pixel to its neighbor(s). If a criterion of homogeneity is satisfied, the pixel is said to belong to the same class as one or more of its neighbors. The choice of the homogeneity criterion is critical for even moderate success [4], [5], [12], and in all instances the results are upset by noise.

The fourth type are the hybrid techniques which combine boundary and region criteria. This class includes morphological watershed segmentation [10] and variable-order surface fitting [3]. The watershed method is generally applied to the gradient of the image. This gradient image can be viewed as a topography with boundaries between regions as ridges. Segmentation is equivalent to flooding the topography from the seed points [15] with region boundaries being erected to keep water from different seed points from meeting. Unlike the boundary-based methods above, the watershed is guaranteed to produce closed boundaries even if the transitions between regions are of variable strength or sharpness. The technique encounters difficulties with images in which regions are both noisy and have blurred or indistinct boundaries. The variable-order surface fitting method [3] starts with a coarse segmentation of the image into several surface-curvature-sign primitives (e.g., pit, peak, ridge, etc.) which are then refined by an iterative region growing method based on variable-order surface fitting. Because the technique was developed for machine vision applications where the image content may vary considerably, the segmentation is entirely data driven with no scope to involve higher level knowledge. The method is also computationally very expensive.

We present here a new method known as "seeded region growing" (SRG) which is based on the conventional region growing postulate of similarity of pixels within regions, but whose mechanism is closer to that of the watershed. Instead of tuning homogeneity parameters as in conventional region growing, SRG is controlled by choosing a (usually small) number of pixels, known as seeds. This form of control and the corresponding result is readily conceptualized, which allows relatively unskilled users to be able to achieve good segmentations on their first attempt. The result is a robust and easy-to-use routine where higher level knowledge of the image composition can be readily incorporated into the technique through the choice of seeds.

The SRG algorithm is presented in Section II. In Section III, we examine its properties and compare them to those of watershedding. Possible approaches to manual and automatic seed selection are discussed in Section IV and V, respectively.

II. THE ALGORITHM

Seeded region growing performs a segmentation of an image with respect to a set of points, known as seeds. We start with a number of seeds which have been grouped into n sets, say, $A_1, A_2, \cdots, A_n$. Sometimes, individual sets will consist of single points. It is in the choice of seeds that the decision of what is a feature of interest and what is irrelevant or noise is embedded. Given the seeds, SRG then finds a tessellation of the image into regions with the property that each connected component of a region meets (nonempty intersection with) exactly one of the A_i and, subject to this constraint, the regions are chosen to be as homogeneous as possible. We present here a description of the method as applied to gray-scale images. The extension to color or multispectral images is straightforward (after choice of suitable metric in color or multispectral space). The method can be implemented on any shape grid (or graph) in any number of dimensions.

The process evolves inductively from the seeds, namely, the initial state of the sets $A_1, A_2, \cdots, A_n$. Each step of the algorithm involves
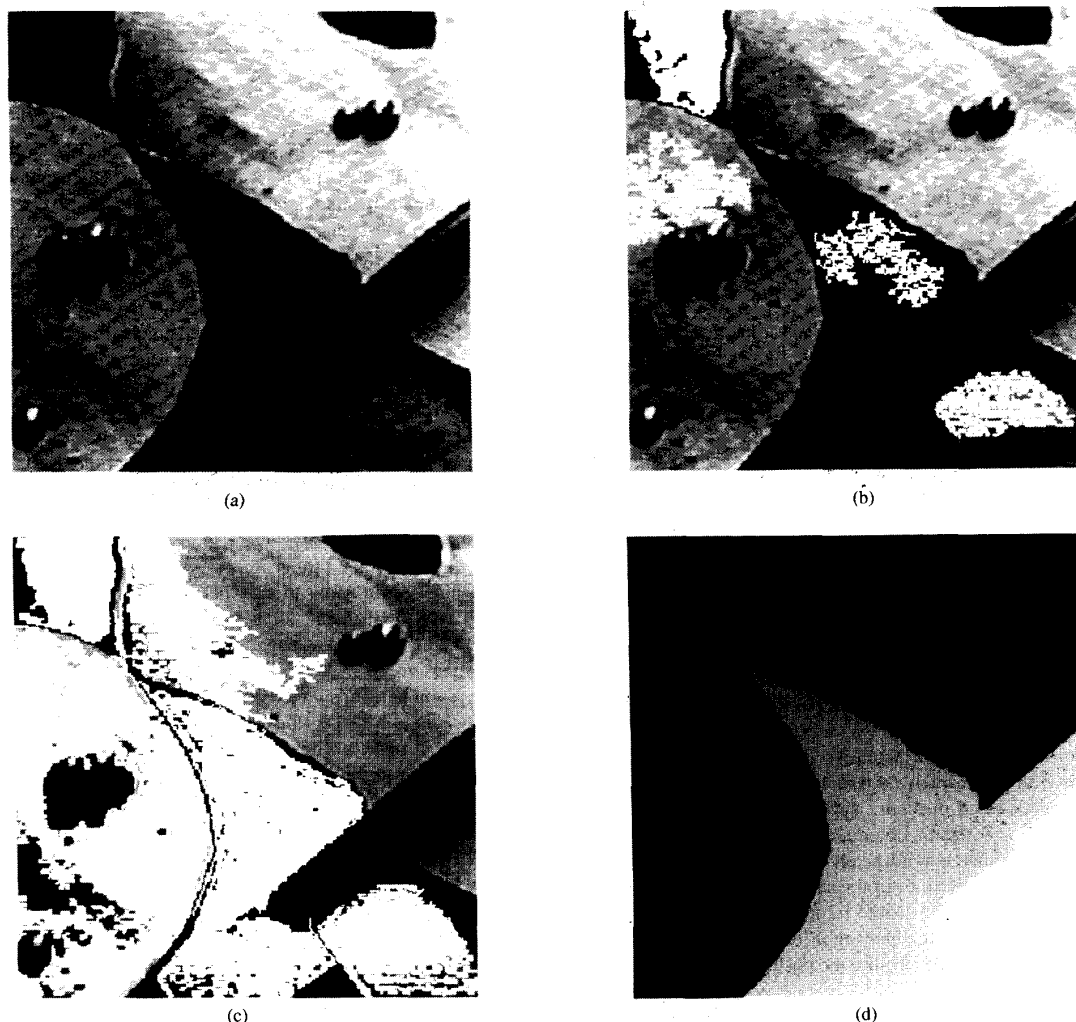
(a)

(b)

(c)

(d)

Fig. 1.   (a) Gray-scale image with 4 seeds marked (white diamonds). (b) The 4 regions after 10% of pixels have been allocated by SRG. (c) The 4 regions after 50% of pixels have been allocated. (d) The final 4 regions. Notice how noise and unwanted features become engulfed and forced to join one of the desired regions.

the addition of one pixel to one of the above sets. We now consider the state of the sets A_i after m steps. Let T be the set of all as-yet unallocated pixels which border at least one of the regions

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

where $N(x)$ is the set of immediate neighbors of the pixel x. In the examples to be presented in this correspondence, we will use a rectangular grid with immediate neighbors being those which are 8-connected to the pixel x. If, for $x \in T$ we have that $N(x)$ meets just one of the A_i, then we define $i(x) \in \{1, 2, \cdots, n\}$ to be that index such that $N(x) \cap A_{i(x)} \neq \emptyset$ and define $\delta(x)$ to be a measure of how different x is from the region it adjoins. The simplest definition for $\delta(x)$ is

$$\delta(x) = |g(x) - \text{mean}_{y \in A_{i(x)}} [g(y)]| \tag{1}$$

where $g(x)$ is the gray value of the image point x. If $N(x)$ meets two or more of the A_i, we take $i(x)$ to be a value of i such that $N(x)$ meets A_i and $\delta(x)$ is minimized. Alternatively, in this circumstance,

we may wish to classify x as a boundary pixel and append it to the set B of already-found boundary pixels. Flagging such boundary pixels is useful for display purposes or for use with a semiinteractive corrective procedure as introduced in Section IV. We then take a $z \in T$ such that

$$\delta(z) = \min_{x \in T} \{\delta(x)\} \tag{2}$$

and append z to $A_{i(z)}$.

This completes step $m + 1$. The process is repeated until all pixels have been allocated. The process commences with each A_i being just one of the seed sets. The definitions (1) and (2) ensure that the final segmentation is into regions as homogeneous as possible given the connectivity constraint.

In programming SRG, we make use of a data structure which we will term a sequentially sorted list (SSL). This is nothing new, although it has not often been used in image processing applications. An SSL is just a linked list of objects, in this case pixel addresses, which are ordered according to some attribute. When considering a new pixel, for example, at the beginning of each step of SRG, we take that one at the beginning of the list. When adding a pixel
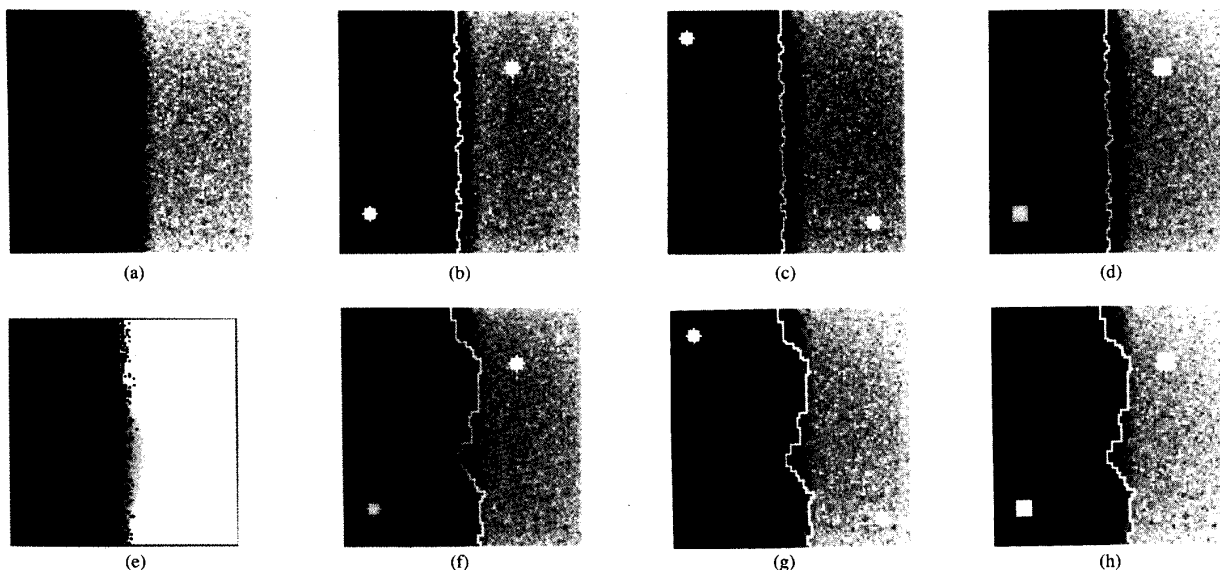
Fig. 2. (a) Artificial image with noise $\sigma = 2.0$. (b), (c) SRG segmentation using differently positioned single point seeds (shown as circles). (d) SRG segmentation using square seed areas (7×7 boxes). (e) ML classification. (f), (g), (h) Watershed segmentation using the same seeds as (b), (c), (d).

to the list, we must place it according to its value of the ordering attribute. In our case, the SSL stores the data of T which is ordered according to δ.

The algorithm for implementing SRG (boundary flagging case) is as follows:

> Label seed points according their initial grouping.
> Put neighbors of seed points (the initial T) in the SSL.
> *While the SSL is not empty:*
>> Remove first point y from SSL.
>> Test the neighbors of this point:
>>> If all neighbors of y which are already labeled
>>> (other than with the boundary label) have
>>> the same label—
>>>> Set y to this label.
>>>> Update running mean of corresponding region.
>>>> Add neighbors of y which are neither already
>>>> set nor already in the SSL to the SSL according
>>>> to their value of δ. (See note below).
>>> Otherwise
>>>> Flag y with the boundary label.

Note that previous entries in the SSL are not updated to reflect their differences from the new region mean. This leads to negligible difference in the results, but greatly enhanced speed.

This stepwise description shows that, in executing the algorithm, each pixel is visited just once, although at each visit we also view each of the neighbors. Hence, it makes for a very rapid program.

III. PROPERTIES OF SRG

Fig. 1 gives a visual demonstration of the region growing mechanism. It also shows the effect of SRG in the presence of unwanted artifacts (e.g., the small round objects in the image). Provided none of the seeds falls within an unwanted feature, such features become subsumed by surrounding regions. An appreciation of the speed of the algorithm can be gained from the fact that the user-wait time to segment this 256×256 8-bit image was ~ 4 s on a DECstation 5000/200.

The natural question which arises from a demonstration of SRG is: how critical is the seed choice to a good segmentation? If achieving a good segmentation were as dependent on having chosen a particular "correct" set of seeds as the choice of the homogeneity parameter in standard region growing [4], [5], [12], we would have made no progress. Luckily this is not the case. If the regions are relatively noiseless, all that is necessary, for a good segmentation is that each seed pixel have a gray value which is typical of its region. However, if the regions are noisy, single point seeds may fall on an atypical pixel (i.e., an outlier statistically). A poor starting estimate of that region's mean would result and the segmentation may be incorrect. To prevent this, it is recommended that small seed areas be used (instead of single pixels) when segmenting noisy images. Each seed area should be sufficiently large to ensure that a stable estimate of its region's mean is obtained.

The assumption inherent in the choice of the sample region mean for definition of δ in (1) is that the noise in each region is of equal variance. If this assumption is not reasonable, then the appropriate choice of δ would be

$$\delta(x) = \left| \frac{g(x) - \text{mean}_{y \in A_{i(x)}} [g(y)]}{\text{SD}_{y \in A_{i(x)}} [g(y)]} \right|$$

where SD refers to the sample standard deviation of the region. However, this implementation of seeded region growing is much more computationally expensive and should be avoided, if possible. Where the standard deviation is a known function of the mean (e.g., for Poisson counts in SPECT imagery), a suitable variance-stabilizing transformation of the image should first be applied.

A feeling for the stability of the SRG algorithm can be obtained by looking at its effect on the artificial two-dimensional image given by

$$g(x, y) = \begin{cases} -10 & x \in [-50, -10) \\ x & x \in [-10, 10) \\ 10 & x \in [10, 50] \end{cases}$$
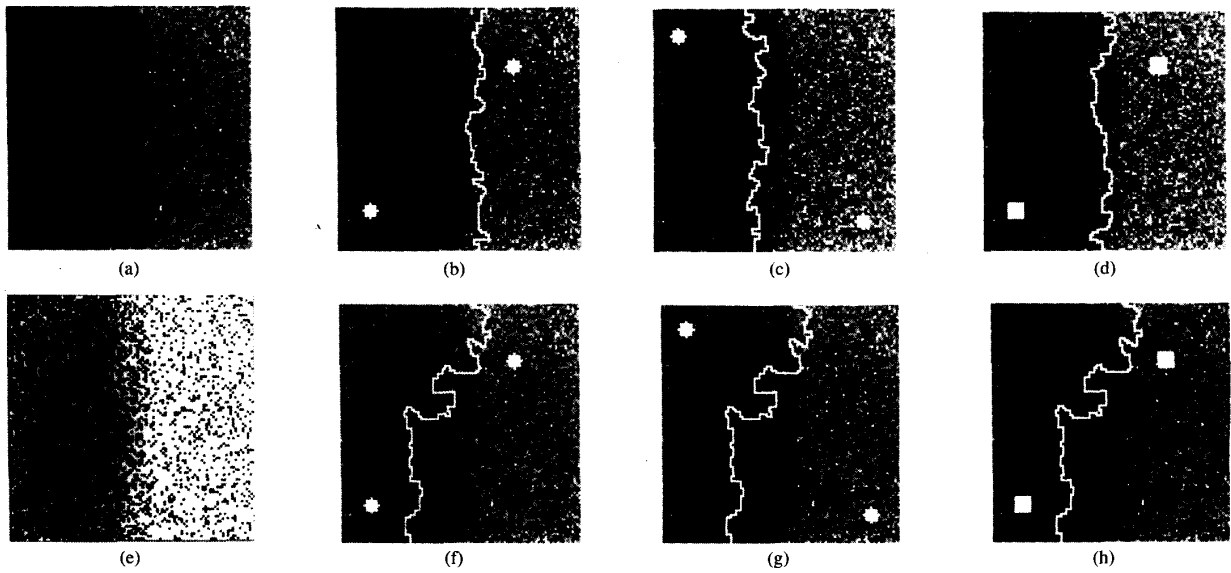
for $y = 0$ to 100.

Fig. 3.   Artificial image with noise $\sigma = 10.0$. (b), (c) SRG segmentation using differently positioned single point seeds (shown as circles). (d) SRG segmentation using square seed areas (7 × 7 boxes). (e) ML classification. (f), (g), (h) Watershed segmentation using the same seeds as (b), (c), (d). Note that the single seed SRG boundaries in (b) and (c) are biased to either side but the seed area boundary in (d) is in the center of the transition zone.
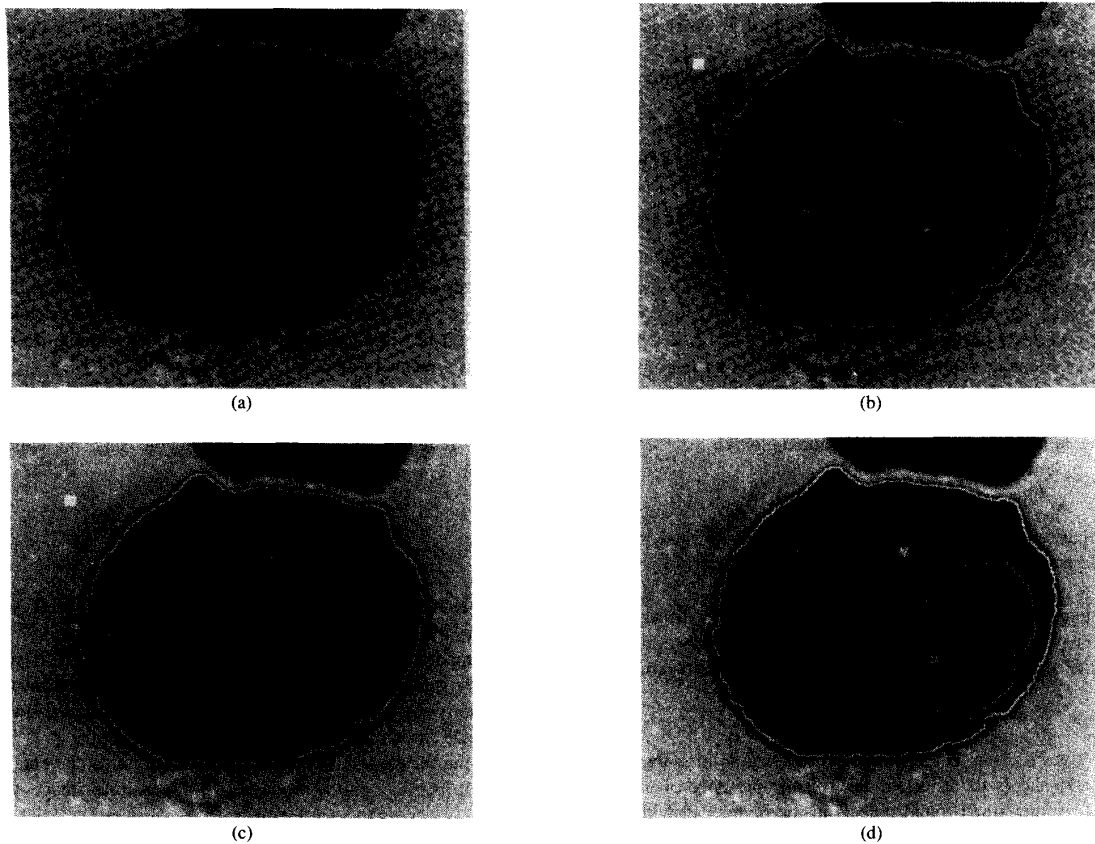

Fig. 4.   (a) Micrograph of cell stained to differentiate cytoplasm and nucleus. (b) Result of segmentation by SRG if all 4 seeds (for background, cytoplasm and 2 nuclei) are manually selected. (c) Result of segmentation by SRG if only 3 seeds are selected. (d) Taking the first incomplete segmentation in (c), the specified region is then broken down using a masked seeded region growing with respect to the 2 additional seeds shown. The resulting corrected segmentation is nearly identical to the original segmentation in (b) (i.e., the two boundaries differ in position by 1 pixel or less).
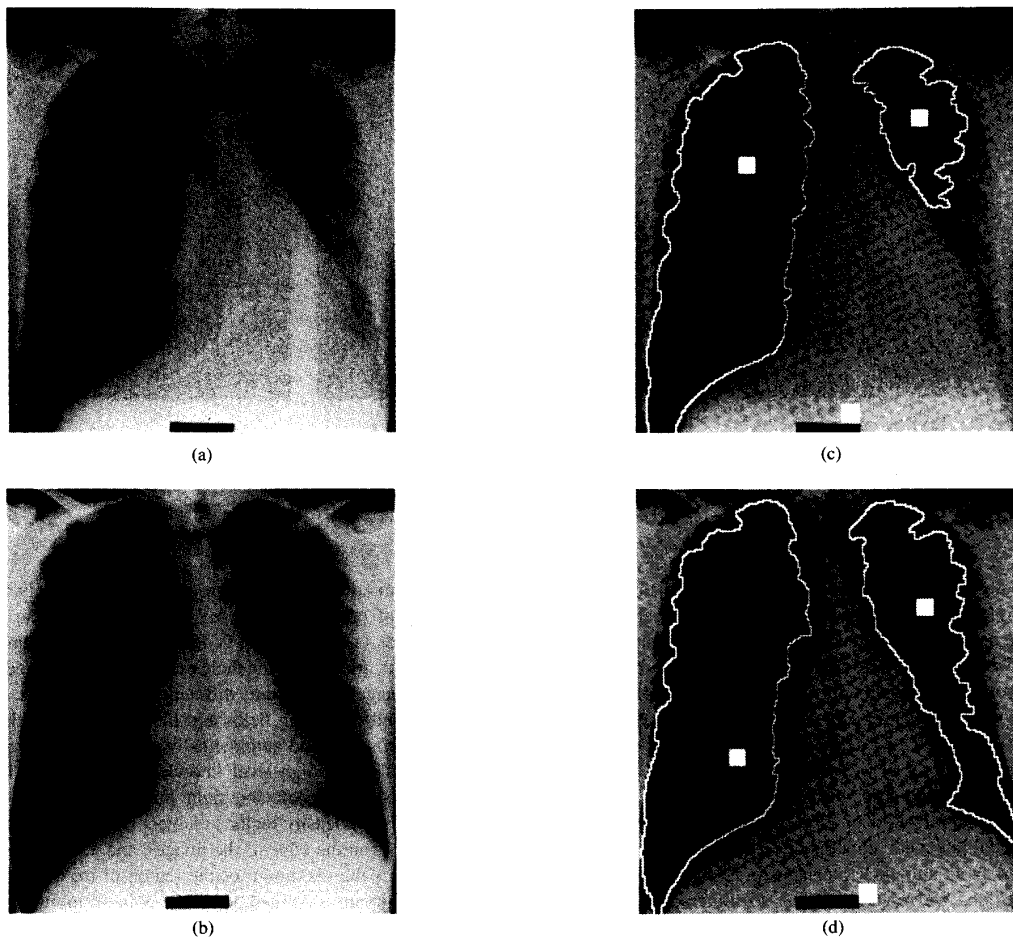
(a)

(c)

(b)

(d)

Fig. 5. (a) X-ray image of human chest in which the left and right lung fields are to be segmented. (b) A second X-ray image of human chest with very different left lung field. (c) Result of segmentation of (a) by SRG using automatically derived seed areas (10×10 boxes). These seeds, for the left and right lung fields and the region in between them, are found using a converging squares algorithm. (d) Result of segmentation of (b).

In this example, we have two regions with a difference in gray levels of 20 and separated by a broad transition region. We will assume that the seeds for the segmentation of this image will only be placed within the region they represent and not within the transition region. A priori, the most appropriate boundary between the two regions is a vertical line at $x = 0$. To test the stability of SRG with respect the presence of noise, we have added Gaussian noise with zero mean and standard deviation of $\sigma = 2.0$ and $\sigma = 10.0$ (Figs. 2(a) and 3(a), respectively). Since the assumption of equal variance noise is reasonable for these images, the implementation of SRG will use the definition of δ in (1).

To examine the effect of seed position and seed size variations, images (b), (c), and (d) of both figures show the SRG segmentations resulting from single point seeds in two different configurations, and small area seeds (7×7 areas), respectively. These results can be compared to the morphological watershed segmentation using the same seeds in images (f), (g), and (h), respectively. (Note that the watershed operation is actually applied to the gradient of the image, obtained using a Sobel filter.) For comparison, a nonspatial maximum likelihood (ML) classification (assuming Gaussian errors) [7] is included as image (e) using the known class statistics (e.g.,

$\mu_1 = -10.0, \sigma_1 = 2.0$ for region 1, and $\mu_2 = +10.0, \sigma_2 = 2.0$ for region 2 in Fig. 2). ML class allocation is shown (class 1 as black, class 2 as white) rather than the boundary between the two classes.

From Figs. 2 and 3, we see that SRG gives superior results to the watershed segmentation and ML classification for both levels of image noise. They will give similar results only when there is no noise present. As for the influence of seed size on SRG, single pixel seeds provide a reasonable estimate of their region means for lower levels of noise (i.e., $\sigma = 2.0$) and so accurate segmentation will result. However, for higher levels of noise (i.e., $\sigma = 10.0$), the larger area seeds are required. In both noise cases, provided the seed pixels or areas give good estimates of their region means, their positions are not important.

Of course, the test images used here are a lot simpler than a true image. It is hoped that these simple examples will still provide insight into the claimed robustness of seeded region growing and to the motivation behind the chosen algorithm.

IV. SEMIINTERACTIVE IMAGE SEGMENTATION

One fundamental problem of the segmentation task is that there is rarely one unambiguous separation of a given scene. Rather, the

(a)   (b)   (c)
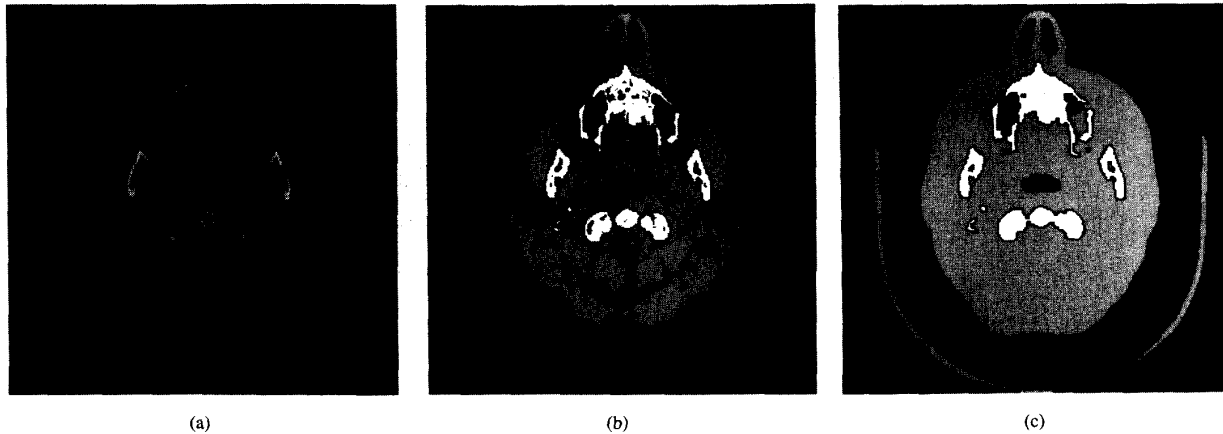
Fig. 6. (a) X-ray attenuation (CT) image of a human head. (b) A partial segmentation into three seed classes of bone, soft tissue, air (white, light gray, and dark gray, respectively) with unallocated pixels shown as black. This is simply a rough multilevel thresholding of the gray-scale range (i.e., air $<30, 90 <$ tissue <100, bone > 130). (c) Completed segmentation obtained by SRG from the seeds in (b).

objects or classes to be delineated depend on the application, and the human interpreter is often the arbiter of which objects are significant and which are not. Thus, a minimal level of human input will always be required. Traditionally, the form of this control has been through a trial-and-error-based adjustment of parameters. The principle is to find a process which works for some data, often known as the training set, and then to apply it to a large quantity of similar images. The difference with seeded region growing is that the effect of the control, in the form of the seeds, is readily conceptualized.

In the simplest application of SRG, a user views the image and, based on personal judgment, chooses the seed points, for example, by a mouse-based point-and-click mechanism. Seeded region growing completes the job. We refer to this as semiinteractive image processing, as it is neither fully automatic nor fully manual. It has the advantage of having an inbuilt quality control—the human operator.

In a semiinteractive application of SRG, there are two factors we must keep in mind, namely, that results will go awry if a seed falls on a noisy point, and that users may make errors in the seed choice. The first problem is overcome by increasing the area of the seed to obtain a stable estimate of the statistics of that region. For the second problem, a simple corrective program can be built into the semiinteractive process. If, for example, too few seeds were initially chosen, then the result would include at least one region which should be subdivided. A user could be presented with this first segmentation with the option of splitting such regions. This is easily accomplished using a constrained form of SRG in which the implemented program restricts its processing to the regions being subdivided. As SRG does not require rectangular images for its operation, this corrective process will be very rapid. An example is shown in Fig. 4. Just as this example illustrates a semiinteractive corrective splitting, a corrective merging is also easy to encode.

Thus, we have developed a robust, three-step (seed selection, SRG, correction) semiinteractive image segmentation process applicable to a wide range of images without the need for any exploratory data analysis or modeling of the data.

V. AUTOMATED IMAGE SEGMENTATION

In Section II, we presented the algorithm of seeded region growing in terms of sets $A_1, A_2, \cdots, A_n$ whose initial states consisted of one manually selected seed point or area each. The obvious way to extend this method to automated image segmentation is simply to automate the process of seed selection. An example of this use is given in Fig. 5. Images (a) and (b) are X-ray images of the human chest in which the left and right lung fields are to be segmented. Note that these two images have quite different left lung fields. The seeds, for the left and right lung fields and the mediastinum (the region in between them) are found automatically using the converging squares algorithm [11]. The high-level knowledge being used here is that suitable seeds for the left and right lung fields are the minima of the two minimum density regions (subject to the constraint that these regions are not on the edge of the image), and a suitable seed for the mediastinum is the maximum of the region of maximum density. In this case, minimum is defined not as the point of minimum value, but as the point of minimum value within the region of minimum density (and similarly for the maximum). The seeds and SRG segmentations of (a) and (b) are shown as images (c) and (d) respectively.

The restriction of each seed leading to a distinct region is not necessary. The initial states of each of the A_i may consist of any positive number of states subject to the restriction that $A_i \cap A_j = \emptyset$ for $i \neq j$. The algorithm then proceeds as before.

We can use this formulation in treating SRG as a correction procedure as applied to a first attempt at segmentation obtained by some other automated method. The first process may only allocate some of the pixels whence SRG can be used to complete the segmentation. For example, the first process may form a rough approximation to the solution from which we extract a core by homotopic thinning [14] or other methods; this core is then extended to the whole image by seeded region growing. Fig. 6 gives an example of this use.

VI. CONCLUSION

We have presented here a description of seeded region growing as applied to gray-scale images. The extension to color or multispectral images is straightforward. SRG is a rapid, robust, easy-to-use image segmentation procedure requiring neither tuning parameters nor training sets. It is applicable to a wide range of image types. By itself, however, it is not a self-contained process, as it also requires the input of a few control points in the image known as seeds. These can be manually entered or can be the output of other image processing algorithms.

ACKNOWLEDGMENT

Many thanks to E. Breen for the use of his implementation of the Converging Squares [11] algorithm, and to P. Soille for his comments and suggestions.

REFERENCES

[1] R. Adams, "Two new methods for seeded or semi-interactive image processing," *Proc. APRS Conf. Digital Image Computing: Techniques Appl.,* 1991, pp. 459–465.

[2] D. H. Ballard and C. M. Brown, *Computer Vision.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[3] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-10, pp. 167–192, 1988.

[4] F. Cheevasuvit, H. Maitre, and D. Vidal-Madjar, "A robust method for picture segmentation based on split-and-merge procedure," *Comput. Vis. Graph. Image Process.,* vol. 34, pp. 268–281, 1986.

[5] S-Y. Chen, W-C. Lin, and C-T. Chen, "Split-and-merge image segmentation based on localized feature analysis and statistical tests," *CVGIP: Graphic. Models Image Processing,* vol. 53, no. 5, pp. 457–475, 1991.

[6] L. S. Davis, "A survey of edge detection techniques," *Comput. Graph. Image Process.,* vol. 4, pp. 248–270, 1975.

[7] D. Y. Hand, *Discrimination and Classification.* New York: Wiley, 1981.

[8] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vis. Graph. Image Process.,* vol. 29, pp. 100–132, 1985.

[9] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," *Proc. 2nd Int. Joint Conf. Pattern Recognit.,* 1974, pp. 424–433.

[10] F. Meyer and S. Beucher, "Morphological segmentation," *J. Vis. Commun. Image Represent.,* vol. 1, pp. 21–46, 1990.

[11] L. O'Gorman and A. C. Sanderson, "The converging squares algorithm: An efficient method for locating peaks in multidimensions," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 280–288, 1984.

[12] T. Pavlidis and Y-T. Liow, "Integrating region growing and edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 12, pp. 225–233, 1990.

[13] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vis., Graph. Image Process.,* vol. 41, pp. 233–260, 1988.

[14] J. Serra, *Image Analysis and Mathematical Morphology.* New York: Academic Press, 1982, pp. 390–423.

[15] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 13, pp. 583–598, 1991.

[16] S. W. Zucker, "Region growing: Childhood and adolescence," *Comput. Graph. Image Process.,* vol. 5, pp. 382–399, 1976.

Digital Planarity of Rectangular Surface Segments

Peter Veelaert

Abstract— We generalize the concept of evenness which has been developed for digital straight lines. Evenness is a necessary and sufficient condition for a digital arc segment to be a digital straight line segment. We prove that evenness is also a necessary and sufficient condition for a rectangular surface segment to be a digital plane segment. This is not true for surface segments of arbitrary shape. To clarify the relation between shape and evenness we introduce the notion of a regular shape and of an arbitrarily extendable even set.

Index Terms—Digital planes, digital straight lines, evenness, flatness, Fourier–Motzkin elimination method, regular shape, straightness.

I. INTRODUCTION

Digital geometry is the study of the geometric properties of sets of points that lie on an n-dimensional lattice or grid. The digital straight line is the curve that has been studied most in digital geometry. Rosenfeld has shown that a digital straight line satisfies the chord property [4]. Hung has stated an evenness criterion and shown that it is equivalent to the chord property [1]. Kim has shown that a digital straight line is a digital arc that is digitally convex [2]. Most of these properties have been proven by geometrical methods. An algebraic approach has been followed by Dorst and Smeulders to establish a correspondence between line segments and quadruples of integers [8].

Far less is known about digital planes. Kim has proven a chordal triangle property for digital planes [3]. This property only holds for infinite digital plane segments without a boundary. For plane segments of arbitrary shape, Kim has shown that the chordal triangle property is neither necessary nor sufficient.

In this correspondence, we shall apply Hung's notion of evenness to plane segments. For rectangular segments we shall prove that evenness is a necessary and sufficient condition for the segment to be part of a digital plane. We also show that the evenness criterion gives rise to an algorithm to detect flatness with time complexity $O(n^2)$, where n is the number of points of the segment. Although there exist algorithms to detect flatness in $O(n)$ time (see [9], [11]), the evenness criterion remains a valuable alternative because of its simplicity.

The derivation of the evenness property by Hung was based on the grid-intersect digitization scheme. In contrast, we shall use an algebraic approach and relate the digitization of a line or plane to a system of linear inequalities. By elimination of the unknown parameters in these inequalities we will obtain conditions that a set of points must satisfy to be on a digital plane. The method we shall use to eliminate the unknowns is the Fourier–Motzkin elimination method which in fact can be used for any digitization that is related to a system of linear inequalities.

In Section II, we define flatness and evenness. Section III covers the Fourier–Motzkin method. In Section IV, we examine the relation between flatness and evenness for infinite segments. In Section V, we investigate the evenness of regularly shaped segments. Finally, in Sections VI and VII, we discuss the application and meaning of these results.