Multi-Dimensional Image Segmentation Using Seed-Invariant Region Growing
非變異種子群域成長的影像切割技術之研究

Shu-Yen Wan (萬書言),[1,2] Cherng-Min Ma (馬成民),[2] and Eric Nung (農北辰)[2]

[1]Graduate Institute of Computer Science and Information Engineering (長庚大學資訊工程研究所)
[2]Department of Information Management, Chang Gung University (長庚大學資訊管理系)
sywan@mail.cgu.edu.tw; minma@mail.cgu.edu.tw; m8944002@stmail.cgu.edu.tw

Abstract

The goal of image segmentation is to partition a digital image into disjoint regions of interest. Of the many proposed image-segmentation methods, region growing has been one of the most popular. Research on region growing, however, has focused primarily on the design of feature measures and on growing and merging criteria. Most of these methods have an inherent dependence on the order in which the points and regions are examined. This weakness implies that a desired segmented result is sensitive to the selection of the initial growing points. We define a set of theoretical criteria for a subclass of region-growing algorithms that are insensitive to the selection of the initial growing points. This class of algorithms, referred to as Symmetric Region Growing, leads to a single-pass region-growing approach applicable to any dimensionality of images. Furthermore, they lead to region-growing algorithms that are both memory- and computation-efficient. Finally, by-products of this general paradigm are algorithms for fast connected-component labeling and cavity deletion. The paper gives complete theoretical results and 3-D image examples.

Keywords: *image segmentation, region growing, three-dimensional image analysis, connected-component analysis, region-based segmentation*

I. Introduction

The goal of image segmentation is to partition a digital image into disjoint regions of interest. Of the many proposed image-segmentation methods, region growing has been one of the most popular [1,6,8,10,11,20]. Region growing methods generally require the desired regions to be homogeneous with respect to certain pre-specified features. An example is the well--known split-and-merge approach [9,13]. This approach iteratively applies region splitting and merging operations to form a segmented image. The intermediate decisions on splitting and merging are governed by the homogeneity of the regions being constructed.

Research on region-based segmentation methods has focused on either: (*a*) the design of feature measures and growing/merging criteria [1,2,3,7,8,14,16,20] or (*b*) algorithm efficiency and accuracy [4,13,19]. Most of these methods, however, have an inherent dependence on the order in which the points and regions are examined [1,6]. This weakness implies that a segmented result is sensitive to the selection of the initial growing points (or *seeds*). A region-based segmentation method can have this problem because its measured feature information adaptively changes as the segmentation process progresses. For example, most seeded region-growing processes only add a new point to a region if its corresponding feature measures are similar to those of an adjacent existing region; after this new point is added to the region, the region's feature measures change. Therefore, different initial growing point assignments lead to different values for evolving region information.

Region-based methods often are also computation and memory intensive. For example, the three-dimensional (3D) algorithms of [7,16,17] operate as if they are *x*-, *y*-, and *z*-inseparable (hence requiring significant computation) and demand considerable memory (e.g., the entire image, plus another copy of an image buffer for storing region labels).

We propose the concept of Symmetric Region Growing (*SymRG*). Region-growing algorithms that abide by the theoretical criteria defining *SymRG* are insensitive to the initial growing points and initial conditions set forth for segmentation. These criteria, defined in Section III, lead to fast single--pass growing algorithms. Such algorithms can be built for any image dimensionality, as discussed in Section IV. Furthermore, as indicated in Section V, the *SymRG* paradigm leads to efficient algorithms for 3D connected--component labeling and 3D cavity deletion. Also, as shown in Section V, *SymRG* algorithms are both memory- and computation-efficient.

II. Notation And Problem Statement

Consider a digital image I defined on an *n*-dimensional discrete (digital) space Z^n; i.e., $I \in Z^n$. The goal of image segmentation is to partition the digital image I into M disjoint regions of interest R_i, $i = 1, \ldots, M$, where the final segmented image S takes the form [10]:

$$S = \bigcup_{i=1}^{M} R_i, \text{ where } R_i \cap R_j = \phi \text{ for } i \neq j \qquad (1)$$

Assume region R_M is reserved for the *background* (generally set to "0" in the final segmented image). Also, assume without loss of generality that each region of interest R_i, $i = 1, \ldots, M$, consists of one connected component. (In practice the individual regions in S are distinguished by region labels [5,15].) In the theory of relations, the segmentation S is formally called a *partition* of set I and each of the disjoint regions R_i constitute *blocks* of the partition [12].

Let lower-case quantities, such as *a*, *b*, *p*, and *q*, represent image points $\in I$. An image point is called a pixel in two-dimensional (2D) images and a voxel in 3D images [15,17]. Let upper-case quantities, such as R_i, *I*, *S*, *A*, and *B* denote sets of points in Z^n. The quantity $f(p)$ gives the intensity, or gray-level, value of image point $p \in I$.

If two image points *a* and *b* are connected, then at least one *path* (or ordered sequence of connected points) exists between them [5]. Let the notation P_{ab} represent such a path. Alternately, let the notation $(a, p_1, p_2, \ldots, p_n, b)$ represent a particular path between *a* and *b*, where point *a* is a neighbor of point p_1, p_1 is a neighbor of p_2, etc. For this paper, all points on a path must lie in the same region of *S*; i.e., if $a \in R_i$, then $p_1 \in R_i$, $p_2 \in R_i$, …, $b \in R_i$. In 2D images, connectivity and neighbors are defined using either 4-connectivity or 8-connectivity [5]. Analogously, for 3D images, 6-connectivity or 26-connectivity define such concepts [17].

Focusing the segmentation process to region growing, the segmented image (Equation 1) can be represented as

$$S(I, RG(\psi), \mathcal{S}) = \bigcup_{i=1}^{M} R_i \qquad (2)$$

where I is the image under consideration, $RG(\psi)$ denotes a region-growing algorithm governed by measure and growing criteria ψ, and $\mathcal{S}$ represents criteria for defining the initial growing points, or *seeds*, for regions. A seed is an image point that is known to belong to a particular region and begins the construction of the region. The collection of measure and growing criteria ψ can be viewed as consisting of two components: $\psi = <\mathcal{I}, \mathcal{X}>$. $\mathcal{I}$ specifies properties that non-seed points must have to be *included* in evolving segmented regions. $\mathcal{X}$ specifies criteria for *excluding* certain image points from all regions of interest.

In general each set of criteria $\mathcal{I}$, $\mathcal{X}$ and $\mathcal{S}$ consists of a predicate composed of Boolean operations of feature measures. Without loss of generality, the pair $(RG(\psi), \mathcal{S})$ constitutes a complete image-segmentation algorithm based on region growing. The operations or feature measures are combined to form a complete predicate for $\mathcal{I}$, $\mathcal{X}$ and $\mathcal{S}$, using the standard algebraic operators $\{\vee, \wedge, \sim\}$, where "$\vee$" is logical OR, "$\wedge$" is logical AND, and "~" is complementation. Thus, valid predicates for ψ and $\mathcal{S}$ are defined over a Boolean algebra. The exclusion criteria $\mathcal{X}$ can, of course, be easily translated into additional criteria for $\mathcal{I}$. But, as shown in the example below, the use of $\mathcal{I}$ leads to more intuitive segmentation algorithms.

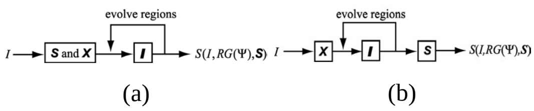


Figure 1. Processing flow for region growing: (*a*) general region-growing algorithm; (*b*) alternate flow possible for a symmetric region-growing algorithm. *I* is the input image, $\mathcal{S}$ specifies the seed criteria, $\psi = <\mathcal{I}, \mathcal{X}>$ specifies the region growing criteria, and $S(I, RG(\psi), \mathcal{S})$ is the final segmented image.

Figure 1a illustrates the flow for segmenting image *I* using the segmentation algorithm $(RG(\psi), \mathcal{S})$. Seeds are first defined for the regions R_i, $i = 1, \ldots, M$. Next, the region-growing criteria $\psi = <\mathcal{I}, \mathcal{X}>$ are iteratively applied to construct the evolving regions. The growing process terminates when application of the region-growing algorithm produces no further changes to the evolving segmented image. The final result is $S(I, RG(\psi), \mathcal{S})$. The following simple example illustrates a segmentation algorithm.

Example: Consider the problem of segmenting two regions of interests from an 8-bit digital image I. Suppose region R_1 contains points centered about gray-level value 100, R_2 contains points centered about gray-level value 200, and all remaining points are assigned to the background R_3. Then, a possible segmentation algorithm $(RG(\psi), \mathcal{S})$ is as follows:

1. Seed criteria $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2\}$, where

$\mathcal{S}_1$ ≡ "q is the first point in I such that $f(q)$ = 100"

$\mathcal{S}_2$ ≡ "q is the first point in I such that $f(q)$ = 200"

2. Growing Criteria $\psi = \langle \mathcal{I}, \mathcal{X} \rangle$:
   (a) Inclusion criteria $\mathcal{I} = \{\mathcal{I}_{11}, \mathcal{I}_{12}, \mathcal{I}_{21}, \mathcal{I}_{22}\}$, where
   $\mathcal{I}_{11} \equiv "|f(q) - 100| \leq 20"$,
   $\mathcal{I}_{12} \equiv$ "A path P_{qa_1} exists"
   $\mathcal{I}_{21} \equiv "|f(q) - 200| \leq 20"$,
   $\mathcal{I}_{22} \equiv$ "A path P_{qa_2} exists"
   (b) Exclusion criteria $\mathcal{X} \equiv "f(q) \leq 10"$.

3. $RG(\psi)$:
   (a) Find seed points $q_1 \in I$ satisfying $\mathcal{S}_1$ and $q_2 \in I$ satisfying $\mathcal{S}_2$. Assign q_1 to R_1 and q_2 to R_2 in $S(I, RG(\psi), \mathcal{S})$.
   (b) For each point $q \in I$,
   If $\mathcal{I}_{11} \wedge \mathcal{I}_{12} \equiv \text{TRUE}$, assign q to R_1 in $S(I, RG(\psi), \mathcal{S})$.
   Else if $\mathcal{I}_{21} \wedge \mathcal{I}_{22} \equiv \text{TRUE}$, assign q to R_2 in $S(I, RG(\psi), \mathcal{S})$.
   Else if $\mathcal{X} \equiv \text{TRUE}$, assign q to R_3 in $S(I, RG(\psi), \mathcal{S})$.
   (c) Iterate (b) on points in I until no further changes occur to the evolving $S(I, RG(\psi), \mathcal{S})$.

(Many other algorithms, of course, are possible for the example above.)

Since we are currently leaving open the algorithm flow for the region-growing algorithm $RG(\bullet)$, the pair $(RG(\psi), \mathcal{S})$ does indeed represent a general region-growing algorithm. Some region-based algorithms may not seem to fit the framework of $(RG(\psi), \mathcal{S})$ at first glance, but they can be transformed into $(RG(\psi), \mathcal{S})$. For example, the split-and-merge algorithm actually performs the process of iteratively searching the entire image for initial growing points or seeds (splitting) and then growing back regions of interest (merging) [9].

The seed criteria $\mathcal{S}$ can consist of a set of criteria that *implicitly* specify seed points for regions. Equivalently, $\mathcal{S}$ can also be specified as an *explicit* set of seed points, such as:

$$A = \{a_1, \ldots, a_{M-1}\} \subset I \qquad (3)$$

where, in general, set A contains one seed point per region of interest. Point a_1 acts as the initial growing point, or seed, for R_1, a_2 is the seed for R_2, ..., and a_{M-1} is the seed for R_{M-1}. No seed is needed for the background region R_M, as all points not assigned to a "true" region of interest $R_i, i = 1,2,\ldots, M-1$, are assumed to be "relegated" to the background. Each point of an explicitly defined seed set, such as A in (Equation 3), is known *a priori* to belong to a particular region. If A contains additional points beyond (Equation 3), then it is assumed that these points are already assigned to one of the evolving regions $R_i, i = 1,2,\ldots, M-1$. Using the seed criteria (Equation 3), the segmentation (Equation 2) can be stated equivalently as

$$S(I, RG(\psi), A) = \bigcup_{i=1}^{M} R_i \qquad (4)$$

For the remainder of this paper, we will assume that seed criteria $\mathcal{S}$ are converted to an equivalent seed set such as A.

Consider now a different set of initial growing points given by

$$B = \{b_1, \ldots, b_{M-1}\} \subset I \qquad (5)$$

where b_1 acts as a possible seed for R_1, b_2 acts as a possible seed for R_2, etc. Suppose this set produces the segmented image

$$S(I, RG(\psi), \mathcal{B}) = \bigcup_{i=1}^{M} R'_i \qquad (6)$$

R'_1 is the region grown from b_1, R'_2 is the region grown from b_2, etc. In general, for $i = 1,2,\ldots, M-1$ $a_i \neq b_i$ and $R_i \neq R'_i$. In this paper, the statement

$$S(I, RG(\psi), A) \equiv S(I, RG(\psi), B) \qquad (7)$$

means that $R_i = R'_i$ for $i = 1,2,\ldots, M-1$, per (Equation 4) and (Equation 6). If two different segmentation algorithms, $(RG(\psi), A)$ and $(RG(\psi), B)$, satisfy Equation (7), then they produce *equivalent (identical) segmentations* of image I. Figure 2 schematically illustrates many of the concepts defined thus far for a four-region problem.

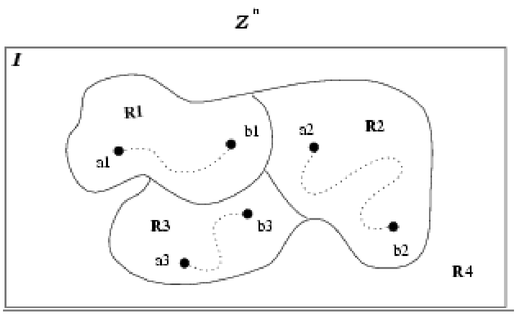


Figure 2. Depiction of the region-growing process for a 4-region segmentation problem. R_1, R_2, and R_3 are the segmented regions of interest and R_4 is the background. The points a_1 and b_1 are possible seeds for R_1, a_2 and b_2 are possible seeds for R_2, etc. The dotted lines give examples of valid paths $P_{a_i b_i}$ between corresponding points a_i and b_i. This figure illustrates the case where a_i and b_i lead to the "same" R_i; i.e., they produce equivalent segmentations S of I, per Equation (7). But, this is not necessarily the case in general.

The following important question arises. What are the requirements on region-growing algorithm $RG(\psi)$ so that $S(I, RG(\psi), A) \equiv S(I, RG(\psi), B)$? *That is, what constraints are required on a region-growing algorithm, so that the algorithm is guaranteed to give identical segmentations when starting with any valid seed set?* Section III answers this question and also provides the theoretical motivation for devising an efficient implementation of region growing.

III. Theoretical Development

Region-based algorithms build regions from the seeds by following a certain evolving growing sequence. If the seeds change, then the resulting growing sequence changes. Our question is whether different seed sets, Equation (3) and (5), and growing sequences lead to the *same* segmentation results. If not, what constraints can be placed on an algorithm, so that it generates the same segmentation regardless of the seed sets? That is, what constraints must a region-growing algorithm have to be invariant to changes in the seed set? We assume that the goal of image segmentation for image I is to form the partition of M regions per (Equation 1). We assume that any seed, such as A, used to achieve (Equation 1) must have M distinct seed points; such a set will be called a *valid seed set*. This section describes the constraints necessary to make a region-growing algorithm invariant to the seed set. These constraints lead to the concept of symmetric region growing (SymRG).

Subsection III-A introduces basic definitions and theoretical constraints. These constraints lead to the concept of symmetric region growing. Additional theoretical results of Section III-B give guidance on how to devise a symmetric region-growing algorithm and motivate the general *n*-dimensional memory- and computation-efficient implementation of symmetric region growing described in Section IV.

A. General Definitions and Theorems

DEFINITION 1: $\mathrm{P}_{ab}(I, RG(\psi))$ is defined as the set of all possible paths $\{P_{ab}, P_{ab}^2, P_{ab}^3, \ldots\}$ between points a and b, where $a, b \in I$, point a is a seed used to grow region $R \subset I$ using $RG(\psi)$ and $b \in R$. □

If seed a in conjunction with region-growing algorithm $RG(\psi)$ produces a region R that does not contain point b, then $\mathrm{P}_{ab}(I, RG(\psi)) = \phi$. Also, by the assumption that R consists of one connected component, if $b \in R$, then at least one path P_{ab} $ must exist from seed a to image point b. Within the context of relation theory, if a path exists from a to b, then a and b must be in the same block (region) of the partition S of I.

DEFINITION 2: $\mathrm{P}_{AB}(I, RG(\psi))$ is defined as the set of all possible paths from points in seed set A to points in set B:

$$\mathrm{P}_{AB}(I, RG(\psi)) = \begin{cases} \bigcup_{i=1}^{M-1} \mathrm{P}_{a_i b_i}(I, RG(\psi)), & \text{if } \forall i, \mathrm{P}_{a_i b_i}(I, RG(\psi)) = \phi \\ \phi, & \text{otherwise} \end{cases}$$

where A and B are given by Equations (3) and (5). □

The points of A and region-growing algorithm $RG(\psi)$ define a segmentation $S(I, RG(\psi), A)$. The set $\mathrm{P}_{AB}(I, RG(\psi))$ enumerates all paths from each point $ $a_i \in A$ to its corresponding point $b_i \in B$, provided that at least one path exists to each b_i. $\mathrm{P}_{AB}(I, RG(\psi)) = \phi$, if *any* point $a_i \in A$ (responsible for generating region R_i per Equation (3)) does not have at least one path $P_{a_i b_i}$ to its corresponding point $b_i \in B$. If for some point $a_i \in A$, no path $P_{a_i b_i}$ exists, then $b_i \notin B$. This immediately implies that $S(I, RG(\psi), A) \neq S(I, RG(\psi), B)$, because, per Equation (6), $b_i \in R_i^{'}$ and $R_i \neq R_i^{'}$.

DEFINITION 3: The notation

$$A \xrightarrow{RG(\psi)} B \text{ is equivalent to } \mathrm{P}_{AB}(I, RG(\psi)) = \phi$$

The quantity $A \xrightarrow{RG(\psi)} B$ is a *binary relation* from set A to set B over the region-growing operation $RG(\psi)$ [12]. □

The relation $A \xrightarrow{RG(\psi)} B$ implies that there is a way to form at least one path in $S(I, RG(\psi), A)$ between each initial growing point in A and its corresponding point in B. Otherwise, $A \xrightarrow{RG(\psi)} B$ is false. Note that $A \xrightarrow{RG(\psi)} B$ does not imply $B \xrightarrow{RG(\psi)} A$.

LEMMA 1: The binary relation $\xrightarrow{RG(\psi)}$ is reflexive and transitive. That is, for any seed set $A \subset I$,

$A \xrightarrow{RG(\psi)} A$ (Reflexivity). Also, for any seed sets $A, B, C \subset I$, if $A \xrightarrow{RG(\psi)} B$ and $B \xrightarrow{RG(\psi)} C$, then $A \xrightarrow{RG(\psi)} C$ (transitivity).

Proof: (Reflexivity) It is trivial that $A \xrightarrow{RG(\psi)} A$, because $\mathsf{P}_{AA}(I, RG(\psi))$ contains the trivial one-point paths $P_{a_i a_i}, i = 1,2,\ldots,M-1$.
(Transitivity) Given $A \xrightarrow{RG(\psi)} B$ and $B \xrightarrow{RG(\psi)} C$. Then, for all $i = 1,2,\ldots,M-1$, there exists $P_{a_i b_i} = (a_i,\ldots,b_i) \in \mathsf{P}_{a_i b_i}(I, RG(\psi))$ and $P_{b_i c_i} = (b_i,\ldots,c_i) \in \mathsf{P}_{b_i c_i}(I, RG(\psi))$. By concatenating paths $P_{a_i b_i}$ and $P_{b_i c_i}$, we have $P_{a_i c_i} = (a_i,\ldots,c_i)$. Thus, $\mathsf{P}_{AC}(I, RG(\psi)) = \phi$, or $A \xrightarrow{RG(\psi)} C$. □

Now, consider a general binary relation R on domain

D, such that $R : D \to D$. The binary relation R is said to be *symmetric* if $rRs \Leftrightarrow rRs$, $\forall r \subset D$ and $s \subset D$ [12].

The concept of a symmetric binary relation can be applied to region growing.

DEFINITION 4: Binary relation $\xrightarrow{RG(\psi)}$ is symmetric if,

$$\forall \text{ valid seed sets } A, B \subset I,\ A \xrightarrow{RG(\psi)} B \text{ implies } B \xrightarrow{RG(\psi)} A.$$

If $\xrightarrow{RG(\psi)}$ is symmetric, we denote it as $\xleftrightarrow{RG(\psi)}$ or $\xleftrightarrow{SymRG(\psi)}$. □

In general, the binary relation $\xrightarrow{RG(\psi)}$ is, of course, not symmetric [12]. However, if $RG(\psi)$ satisfies $A \xleftrightarrow{RG(\psi)} B$ for all valid seed sets $A, B \subset I$, then $RG(\psi)$ is called a *symmetric region-growing algorithm* and denoted as $SymRG(\psi)$. Furthermore, given $S(I, SymRG(\psi), A)$ in the context of the segmentation (Equation 4), DEFINITION 4 implies that we can arbitrarily choose sets $X = \{x_1,\ldots,x_{M-1}\}$ and $Y = \{y_1,\ldots,y_{M-1}\}$, where $x_i, y_i \in R_i \subset S(I, SymRG(\psi), A) \setminus R_M$ and form a bijection (or one-to-one and onto) relation between X and Y. Also, by LEMMA 1 and DEFINITION 4, $\xleftrightarrow{SymRG(\psi)}$ is an *equivalence relation* and the segmented regions $R_i, i = 1,2,\ldots,M$, \$ induced by $SymRG(\psi)$, are equivalence classes [12].

LEMMA 2: Let p and q be any pair of points in the same region $R_i \subset S(I, RG(\psi), A)$ for some $i = 1,2,\ldots,M-1$ per Equations (3) and (4). If $RG(\psi)$ is symmetric (i.e., $RG(\psi)$ can be replaced by $SymRG(\psi)$ in (4)), then $\mathsf{P}_{pq}(I, SymRG(\psi)) \neq \phi$.

Proof: Suppose $p, q \in R_i \subset S(I, RG(\psi), A)$ for some $i = 1,2,\ldots,M-1$. Then, for seed $a_i \in A$ (see (3)), $\mathsf{P}_{a_i p}(I, SymRG(\psi)) \neq \phi$ and $\mathsf{P}_{a_i q}(I, SymRG(\psi)) \neq \phi$. Because $SymRG(\psi)$ is symmetric, $\mathsf{P}_{p a_i}(I, SymRG(\psi)) \neq \phi$. Thus, by LEMMA 1, $\mathsf{P}_{pq}(I, SymRG(\psi)) \neq \phi$. □

LEMMA 2 implies that if a symmetric region growing algorithm is used, then any point p in a region can be used to reach (grow) any other point \$ q \$ in the same region. This leads to the following important result.

THEOREM 1: Consider a symmetric region growing algorithm $SymRG(\psi)$, such that $S(I, SymRG(\psi), A) = \bigcup_{i=1}^{M} R_i$ in the context of Equations (3) and (4). Suppose $a_i \in A$ is replaced by an arbitrary point $p \in R_i$ to form an alternate seed set $\hat{A}$. Then, in the resulting segmentation $R_i \subset S(I, SymRG(\psi), \hat{A})$, the region grown from p is R_i.

Proof: Replace a_i with $p \in R_i \subset S(I, SymRG(\psi), A)$ in A. This gives the new seed set $\hat{A} = \{a_1,\ldots,a_{i-1}, p, a_{i+1},\ldots,a_{M-1}\}$. Generate a segmentation with this new seed set: $S(I, SymRG(\psi), \hat{A}) = \bigcup_{i=1}^{M} R_i$, where a_1 produces $\hat{R}_1$, …, a_{i-1} produces $\hat{R}_{i-1}$, p produces $\hat{R}_i$, etc. Consider a point $r \in \hat{R}_i$. Then, $\mathsf{P}_{pr}(I, SymRG(\psi)) \neq \phi$. By LEMMA 2, $\mathsf{P}_{rp}(I, SymRG(\psi)) \neq \phi$. Further, $\mathsf{P}_{a_i p}(I, SymRG(\psi)) \neq \phi$. So, by LEMMA 1 (Transitivity), $\mathsf{P}_{a_i r}(I, SymRG(\psi)) \neq \phi$. Hence, at least one path exists from point $a_i \in R_i$ to $r \in \hat{R}_i$.

Therefore, $\hat{R}_i = R_i$. □

THEOREM 1 states that if a symmetric region-growing algorithm is used, then any point p in region R_i can be used as a seed to grow the region R_i and that the resulting grown region is always the same one. In fact, any and all seed points $a_i \in A$, $i = 1,2,\ldots,M-1$, can be replaced by any point $p_i \in R_i \subset S(I, SymRG(\psi), A)$ to form a new seed set X and the resulting segmentation $S(I, SymRG(\psi), X)$ will be equivalent to $S(I, SymRG(\psi), A)$.

THEOREM 2: Given $SymRG(\psi)$ and seed sets $A, B \subset I$, as in Equations (3) and (5).

$$P_{AB}(I, SymRG(\psi)) \neq \phi \Leftrightarrow S(I, SymRG(\psi), A) \equiv S(I, SymRG(\psi), B). \quad (8)$$

Proof: We use the definitions of A, B, $S(I, SymRG(\psi), A)$, and $S(I, SymRG(\psi), B)$, given in (3-6), with $RG(\psi)$ replaced by $SymRG(\psi)$ in (4,6).

($\Leftarrow$) Given $S(I, SymRG(\psi), A) \equiv S(I, SymRG(\psi), B)$, which is (7). From (4), (6), and (7), $R_i = R_i^{'}$, $i = 1,2,\ldots,M-1$. By LEMMA 2, for any pair of seed points (a_i, b_i), $i = 1,2,\ldots,M-1$, drawn from A and B, at least one $P_{a_i b_i}$ exists. Therefore, $P_{AB}(I, SymRG(\psi)) \neq \phi$, or $A \xrightarrow{SymRG(\psi)} B$

($\Rightarrow$) Given $P_{AB}(I, SymRG(\psi)) \neq \phi$. Consider an arbitrary point $p \in I$. There are two cases to consider: (1) foreground - for some $i = 1,2,\ldots,M-1$, $p \in R_i \subset S(I, SymRG(\psi), A)$; (2) background - $p \in R_M \subset S(I, SymRG(\psi), A)$.

Case (1): foreground - Suppose for some $i = 1,2,\ldots,M-1$, $p \in R_i \subset S(I, SymRG(\psi), A)$. Then, $P_{a_i p}(I, SymRG(\psi)) \neq \phi$, following the definition of seed point a_i in (3). Also, $P_{a_i b_i}(I, SymRG(\psi)) \neq \phi$ and $P_{b_i a_i}(I, SymRG(\psi)) \neq \phi$. By LEMMA 1, $\xrightarrow{SymRG(\psi)}$ is transitive. Hence, $P_{b_i p}(I, SymRG(\psi)) \neq \phi$. Therefore, $p \in R_i^{'}$ of $S(I, SymRG(\psi), B)$, per (6).

Case (2): background - Suppose $p \in R_M \subset S(I, SymRG(\psi), A)$. Suppose for some $i = 1,2,\ldots,M-1$, there exists $b_i \in B$, such that $P_{b_i p}(I, SymRG(\psi)) \neq \phi$; i.e., $p \in R_i^{'} \subset S(I, SymRG(\psi), B)$. As we know, $P_{a_i b_i}(I, SymRG(\psi)) \neq \phi$. Thus, by LEMMA 1 (transitivity), $P_{a_i b_i}(I, SymRG(\psi)) \neq \phi$, which implies that $p \in R_i$. This contradicts the assumption. Hence, $\forall b_i \in B$, $P_{b_i p}(I, SymRG(\psi)) \neq \phi$, which implies $p \in R_M^{'}$.

Thus, $\forall p \in I$, if $p \in R_i \subset S(I, SymRG(\psi), A)$, then $p \in R_i^{'} \subset S(I, SymRG(\psi), B)$, which implies (7). □

THEOREM 2 states that if a symmetric region growing algorithm produces a segmentation of image I of the form $S(I, RG(\psi), A) = \bigcup_{i=1}^{M} R_i$, then, for any of the *M-1* regions of interest $i = 1,2,\ldots,M-1$, any point $p \in R_i$ can be used as a seed point to produce the segmentation $S(I, RG(\psi), A)$. In fact, THEOREM 2 eliminates the importance of the set of initial growing points: the set A (or criteria $\mathcal{S}$) has no influence on whether a region-growing algorithm is symmetric or not. Further, for a symmetric region-growing algorithm, the order that points are visited during the growing process does not matter. The subsection below proposes corollaries that assert these points and helps bridge the gap from theory to practical implementation.

B. *Practical Conditions for Symmetric Region-Growing*

COROLLARY 1: Consider $SymRG(\psi)$ and A such that $S(I, SymRG(\psi), A) = \bigcup_{i=1}^{M} R_i$. Instead of using A to produce the segmentation $S(I, SymRG(\psi), A)$, consider using $B = \{b_1, \ldots, b_{M-1}\}$, where $b_i \in R_i$ and b_i is the *first* point of R_i encountered while scanning image *I*. Then, $S(I, SymRG(\psi), A) \equiv S(I, SymRG(\psi), B)$.

Proof: Follows immediately from THEOREM 2. □

COROLLARY 1 reveals that the first encountered point of a region (e.g., the extreme upper left corner point of the region) can be used to grow it with a symmetric region-growing algorithm. This concept helps in improving algorithm efficiency. Yet, before segmentation proceeds, no regions exist, and, thus, the first encountered point of each region is not necessarily known. The following corollary solves this problem.

COROLLARY 2: Consider $(SymRG(\psi), \mathcal{S})$, a complete segmentation algorithm based on symmetric region growing. Scan the digital image of interest, *I*, sequentially. Grow regions from each scanned point by applying criteria $\psi =< I, X >$, until all image points have been visited. Examine the resulting regions using S. If any point p of a region satisfies criteria S for region R_i, then assign the region to R_i; otherwise, relegate it to the background R_M. The resulting segmented image is $S(I, SymRG(\psi), S)$.

Proof: Let B represent the set of first encountered points $b_i \in R_i$, $i = 1,2,\ldots,M-1$, of the eventual regions of interest. From COROLLARY 1, $S(I, SymRG(\psi), B)$ = $S(I, SymRG(\psi), A)$ = $\bigcup_{i=1}^{M} R_i$. We will now instead segment I by applying $SymRG(\psi)$ use the seed criteria to do the final region labeling. Assume this produces results in preliminary regions $R_1',\ldots,R_{N-1}'$. The seed criteria S (or A) is now used. Denote the first point of each region R_i' as c_i, $i = 1,\ldots,N-1$. From COROLLARY 1, no pair of points in B are in the same region, so $N \geq M$. Also, because the c_i's are the first points of the regions R_i', $i = 1,\ldots,N-1$, and by THEOREM 1, we can re-label c_i and R_i' so that $c_i = b_i$ and $R_i = R_i'$, $i = 1,2,\ldots,M-1$. Furthermore, $R_M = \bigcup_M^N R_i'$. R_M does not contain any seeds, so does any of the regions $R_M',\ldots,R_N'$. We can therefore form R_M' by gathering regions $R_M',\ldots,R_N'$, and $R_M = R_M'$. □

If the region growing algorithm is symmetric, COROLLARY 2 states that one can scan and grow regions first; after the growing process, one then applies S to label the "useful" regions. All unlabelled regions are merged into the background. This idea, an attribute of symmetric region-growing algorithms, helps in computation efficiency, as shown in Figure 1b.

Because of THEOREM 2, the seed criteria S has no influence on whether a region-growing algorithm is symmetric or not. It is sufficient to focus on the properties of $\psi =< I, X >$ to define a *SymRG*. Recall that ψ is a composite of Boolean operations. ψ can be represented as a single predicate, per the definition below.

DEFINITION 5: For $p, q \in I$, let $g(p,q)$ be a predicate representing the growing criteria ψ. Then,

$$g(p,q) = \text{TRUE} \Rightarrow p \xrightarrow{RG(\psi)} q$$

Thus, for any point $p \in R_i \subset I$, a neighbor q will be included in R_i *iff* $g(p,q)$ = TRUE. □

THEOREM 3: (*Symmetric Criteria*) For $g(\bullet,\bullet)$ representing ψ of region-growing algorithm $RG(\psi)$, if $g(\bullet,\bullet)$ is symmetric – i.e., $g(p,q)$ = $g(q,p)$, $\forall p,q \in I$ - then $RG(\psi)$ is symmetric.

Proof: Consider sets A and B, per (3) and (5). Suppose $A \xleftrightarrow{RG(\psi)} B$; i.e., $g(a_i,b_i) = \text{TRUE}$, $\forall a_i \in A$ and $b_i \in B$ and $\mathrm{P}_{a_i b_i}(I, RG(\psi)) \neq \phi$. Assume $g(\bullet,\bullet)$ is symmetric. $\forall a_i \in A$, $b_i \in B$, and $g(b_i,a_i) = \text{TRUE}$, implying $\mathrm{P}_{b_i a_i}(I, RG(\psi)) \neq \phi$. Thus, by THEOREM 2, $RG(\psi)$ is symmetric. □

THEOREM 3 shows that if ψ is a symmetric function, the region-growing algorithm is symmetric. Since ψ can be denoted as $\psi = I \wedge \overline{X}$, then, by the properties of a Boolean algebra, ψ is symmetric if and only if both I and X are symmetric [12]. Similarly, each individual criterion of I and X must be symmetric.

Intuitively, for a symmetric region growing algorithm, I and X should only consist of symmetric operations. Also, the image features employed by I and X should not depend on the previous states of the features. Otherwise, the function employing the feature cannot in general be symmetric. Thus, the growing process does not depend on the order that points are scanned.

The region growing for a traditional $RG(\psi)$ implies an iterative or recursive process. It is not true for *SymRG* anymore, as the regions can validly grow sequentially as suggested by COROLLARY 2 and the algorithm collects region information incrementally therein for final region labeling in reference of the seed criteria.

Below are examples of common region-growing functions. The labels indicate whether or not they are symmetric.

$g(p,q) \equiv \sigma_1 \leq \lvert f(p) - f(q) \rvert \leq \sigma_2$	Symmetric
$g(p,\bullet) \equiv \sigma_3 \leq \sigma_4$	Symmetric
$g(p,\bullet) \equiv \lvert f(p) - \mu_{N(p)} \rvert \leq \sigma$	Symmetric
$g(p,q) \equiv \lvert f(q) - \mu_{R(p)} \rvert \leq \sigma$	Not symmetric

p and q are neighboring image points. σ_1, σ_2, σ_3, and σ_4 are parameters. $\mu_{N(p)}$ denotes the average gray-level value of point p's neighbors, and $\mu_{R(p)}$ denotes the average gray-level value of the points constituting p's member region. Clearly, functions of the form $g(p,\bullet)$ = $g(p)$, which only depend on one pixel, are symmetric.

Note that the region-growing algorithm given in the earlier example, with $\psi = I_{11} \wedge I_{12} \wedge I_{21} \wedge I_{22} \wedge \overline{X}$, is symmetric.

IV. General *SymRG* Algorithm

THEOREM 3 states that a region-growing algorithm is symmetric if and only if all criteria constituting ψ are symmetric functions. If the region-growing algorithm is symmetric, then COROLLARY 1and COROLLARY 2 suggest that the implementation of the *SymRG* can grow regions from the first region points scanned and then apply the seed criteria S afterward to label the final regions. This approach is invariant to which region point is scanned first. It also motivates a general *N*-dimensional *SymRG* algorithm that is computation- and memory-efficient. This algorithm appears below.

Assume that an *N*-dimensional image *I* has image points $(i, j, k, \ldots w, \ldots)$, where *i* is the index of a point along a row, *j* denotes row index, *k* denotes slice number (for 3D images), etc. The gray-level value of point $(i, j, k, \ldots w, \ldots)$ is given by $I(i, j, k, \ldots w, \ldots)$. Growing criteria ψ and seed criteria S are given. Two global data structures are necessary:

Region Table: Each entry in the region table contains *region ID*, *region bounding box*, *number of points*, *number of 0-to-1 crossings*, *number of seeds*, etc., for a region.

Equivalence Table: The equivalence table is incrementally constructed after two homogeneous regions merge. Each entry in the table represents a growing region and maintains a linked list of *region ID* of "equivalent" regions and composite region information gathered from the region table plus the status of this entry. The status of a region may be *growing*, *roi*, or *undesired*. The *growing* regions are pending for final labeling. The *roi* regions are those finished growing and contain seed points. The *undesired* regions on the other hand contain no seeds. The following functions are used:

Construct_1D_Regions(*j*, ψ)

Construct 1-D regions (actually 1-D line segments) on the j^{th} row by applying growing criteria ψ. The output is the updated **Region Table**.

Region_Merge(*n*, *w*, ψ)

Merge contiguous (*n*-1)--dimensional regions between the w^{th} and $(w-1)^{th}$ (*n*-1)-dimensional image, using ψ. The output is the updated **Equivalence Table**.

Label_Regions(S)

Assign final region labels to the regions that contain seeds satisfying S. The remaining regions are relegated to the background. The output **Equivalence Table** contains the final region labels.

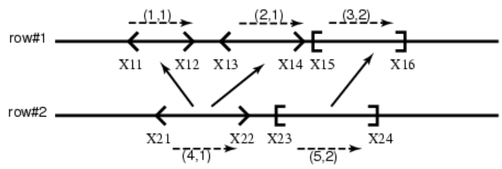


Figure 3. 2D *SymRG*. The region growing starts with row#1 and record regarding information in the region table. The dash arrow-headed lines represent 1D-growing process. Intermediate regions enclosed by '<' and '>' satisfy the criteria for region#1, while those by '[' and ']' satisfy the criteria for region#2. The solid-arrow-headed lines represent merging between regions on consecutive rows. The merging updates the equivalence table that associates equivalent regions. (*p*,*q*) above or below a dashed line denotes the intermediate region ID (*p*) and equivalence region ID (*q*). Please note that intermediate region#1 and #4 can merge because they have an overlapping segment [X21,X12] and satisfy the criteria for region#1. However, intermediate region#2 and #5 cannot merge because they satisfy criteria for different regions, although sharing overlapping segment [X23, X14]. The information of the final desired regions is stored in the equivalence table.

The algorithm shows that *SymRG* segmentation may sequentially scan through the image with two passes. The first pass performs region growing and merging, and the second pass defines the final region labels. It also shows the implementation of the algorithm can be *x*, *y*, *z*, …, etc., separable, and thus enables parallelism and faster computation. The other implementation issues have been addressed in Ref. [19,18]. Besides, because the visited point in the first pass won't be needed until the second pass, *SymRG* segmentation requires only a few rows of the image available plus a small amount of working buffer to maintain the region and equivalence tables. Most portion of the image can be stored in the disk media for later use, without suffering significant disk input/output overhead. We demonstrate *SymRG* efficiency in Section V.

```
Function   2DSymRG(I, RG, ψ, S)
   /* Perform SymRG on a 2-D image */
For row j = 0 to N_y − 1.
   /* Step through rows of 2-D image I. */
   Construct_1D_Regions(j, ψ)
   If j ≥ 1 Region_Merge(2, j, ψ)
EndFor
Label_Regions(S)
End
```

Figure 4. General 2-D Symmetric Region Growing algorithm.

```
Function   3DSymRG(I, RG, ψ, S)
  /* Perform SymRG on a 3-D volumetric image */
For slice k = 0 to N_z − 1.
  /* Step through each 2-D slice of 3-D image I. */
  Do 2DSymRG(I(k), RG, ψ, S)
     /* Perform 2-D region growing on k^th slice of I. */
  If k ≥ 1 Region_Merge(3, k, ψ)
EndFor
Label_Regions(S)
End
```

Figure 5. General 3-D Symmetric Region Growing algorithm.

```
Function   NDSymRG(I, RG, ψ, S)
  /* Perform SymRG on an (N-D)-dimensional image */
For (N-1)-dimensional image w = 0 to N_w − 1.
  /* Step through each (N-1)-dimensional image of I. */
  Do (N-1)DSymRG(I(w), RG, ψ, S)
     /* Perform (N-1)-D region growing on w^th (N-1)-dimensional image of I. */
  If w ≥ 1 Region_Merge(N, w, ψ)
Label_Regions(S)
End
```

Figure 6. General *N*-dimensional Symmetric Region Growing algorithm.

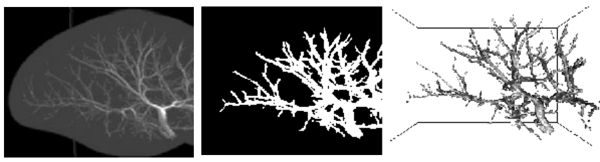
V. Experimental Results And Other Applications

We will also give an example of implementing a previous region--growing algorithm in a *SymRG* way [7,19,18].

SymRG proposes for various region-growing algorithms a designing paradigm that facilitates performance improvement. In this section, we implement the algorithm shown in Ref. [7] by means of *SymRG* approach, and demonstrate its time and memory efficiency. We will also show *SymRG* applications to the other image-processing modules: connected component labeling and cavity deletion.

A. Experimental Results

The experiments were performed on both a Sun™ machine (Solaris© 2.5.1, CPU: 250MHz) and a PC (Windows© NT 4.0, CPU: 400MHz). The human-liver image (Figure 7(a)) is an 8-bit 3D image from an EBCT scanner. The rat-liver images [19,18] are all 16--bit 3D Micro-CT images.



(a) Original image (b) Segmented (c) Surface rendering

Figure 7. Coronal (*x-z*) maximum-intensity projection (MIP) 3D human liver images (*24.4MB*) with their bile ducts selectively opacified with contrast agent. Dimensions: $302 \times 389 \times 218$. $\Delta x = \Delta y = \Delta z = 0.586mm$.

Figure 7(b) and (c) are the segmented result and its corresponding surface rendering. We compare the segmentation time for *SymRG* against previously proposed method [7] and demonstrate the quantitative results in Figure 8.

	humliv	**ctr01**	**ctr02**	**ctr03**
Past on Sun	63	86	127	147
***SymRG* (1)**	34	51	59	69
***SymRG* (2)**	4	8	8	9

Figure 8. Running time comparisons in seconds. The past approach is an implementation proposed in Ref. [7]. *SymRG*(1) is performed on a Sun™ machine that has one 250MHz CPU running Solaris© 2.5.1, while *SymRG*(2) on a PC that has a 400MHz CPU running Windows© NT 4.0.

One of the most significant strengths of *SymRG* is that it allows efficient memory usage. If the neighbors of the current voxel is within a $3 \times 3 \times 3$ cube. The proposed algorithm requires 3 original and 3 working slices of the image plus the memory needed by the region and equivalence tables.

Each entry of the region table requires 18 bytes to store region-related information. Each entry of the equivalence table uses 24 bytes for storing information plus 2 bytes for each of the corresponding equivalent regions. The number of entries in the region and equivalence table depends on the number of the intermediate regions during the process. For an *N*-bit image, we have set the upper bound as 2^{N-1} regions. It implies that the largest number of the equivalent regions is 2^{N-1}. Therefore, the approximate memory usage for performing a *SymRG* method on an *N*-bit image is 6 slices of the image plus $2^{N-1} \times (18+24) + 2\times 2^{N-1} = 44 \times 2^{N-1} = 21 \times 2^{N}$ bytes. On the contrary, the algorithm of Ref. [7] requires memory for 2 copies of the images plus the region table. Figure 9 depicts a quantitative comparison between these two approaches.

	humliv	ctr01	ctr02	ctr03
Past on Sun	49.36	146.56	160.96	229.36
***SymRG* (1)**	27.06	75.21	82.49	117.54
***SymRG* (2)**	2.66	2.21	2.29	3.14

Figure 9. Memory usage comparisons in megabytes. The past approach is an implementation as proposed in Ref. [7]. *SymRG* (1) retains a copy of the image in the memory to avoid I/O overhead, while *SymRG* (2) keeps only six slices of the image in the memory when the memory resource is limited.

B. Other Applications

The immediate applications of *SymRG* are efficient *N*-dimensional connected component labeling and 3D (or 2D) cavity deletion. Connected component labeling is a module that works on the binary image to form regions, and is a special case of the region--growing module. We can, therefore, alter the behavior of a *SymRG* algorithm, by changing its parameter settings, to yield an *N*-dimensional connected component labeling that performs single pass along the image and requires only partial of the image in the memory at a time.

The purpose of cavity deletion is to remove holes in desired regions. Holes or cavities are defined as background regions that do not touch the image boundary. Their generation is virtually inevitable unless the original image presents perfect contrast between foreground and background – in which case a simple thresholding method would be just as efficient. A 3D (or 2D) cavity deletion algorithm can also be obtained by adapting the connected-component labeling algorithm. We first compute 3D (or 2D) “background” connected-components. In this case, if the foreground is defined as 26- (or 8-) connected, the background is 6- (or 4-) connected, and vice versa. The background components that do not touch the boundary of the image are considered to be cavities and are then converted to the foreground. The final resulting image then contains solid regions.

VI. Summary

The contribution of this paper is that we define a family of region-based methods as *SymRG*. Their feature measures and growing criteria yield a growing process that is insensitive to the selection of the initial growing points. We demonstrated the general design of a *SymRG* method and its single-pass implementation by giving an example 3D seeded region-growing algorithm. Because of, for example in 3D, the *x*-, *y*-, and *z*-separable implementation, *SymRG* can gain more performance yields on pipelined or parallel machines.

By applying more *inclusion criteria* (I) that preserve the symmetric property, we can design more sophisticated *SymRG* algorithms. When coping with local discontinuity of a region caused by imperfect image formation, we can evaluate the neighborhood measure ($Nbr(p) = \sum_{q\in N(p), p\neq q} v(q)$, where *p* is current point of concern and *N*(*p*) is the set of *p*'s neighbors defined by the a given window size) such that neighboring points share similar measures to fill the local breaking on a region. We can also specify spatial information for *exclusion criteria* X, such that certain portion of the image is ignored during the region--growing process.

We have also shown the by-products of the *SymRG* design: efficient 3D connected component labeling and 3D cavity deletion. By adjusting the parameters, the proposed example *SymRG* algorithm in Section V can even perform as a threshold-based method.

The limitation of the proposed implementation of *SymRG* is that we store the region labels back to the original image. The maximum number of regions of an *M*-bit image is thus 2^{M}-1, the other one for the background.

Our future research includes (1) more studies on feature measures and growing criteria; (2) lift of the upper-bound of the number of regions; an *inactive* region's label could be assigned to a newly constructed region - an advantage of the single-pass implementation.

Acknowledgments

This work was partially supported by National Science Council, Taiwan, R.O.C., Grants NSC-90-2314-B-1812-093 and NSC-89-2314-B-1812-147-M08, Chang Gung Memorial Hospital Grant NARPD9032, NIH Grants R01-RR11800 and R01-CA74325, by the Whitaker Foundation, and by NSF Grant BIR-9317816 under the Instrument Development for Biological Research Program.

References

1. R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641-647, 1994.
2. T. Asano and N. Yokoya, "Image segmentation schema for low-level computer vision," *Pattern Recognition*, vol. 14, pp. 267-273, 1981.
3. Y.-L. Chang and X. Li, "Adaptive image region-growing," *IEEE Transactions on Image Processing*, vol. 3, no. 6, pp. 868-872, Nov. 1994.
4. B. P. Dobrin, T. Viero, and M. Gabbouj, "Fast watershed algorithms: analysis and extensions," in *Proceedings of SPIE Nonlinear Image Processing*, vol. 2180, pp. 209-220, 1994.
5. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison Wesley, 1992.
6. R. M. Haralick and L. G. Shapiro, "Survey: Image segmentation techniques," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 100-132, Jan. 1985.
7. W. E. Higgins, R. A. Karwoski, W. J. T. Spyra, and E. L. Ritman, "System for analyzing true three-dimensional angiograms," *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 377-385, June 1996.
8. S. A. Hojjatoleslami and J. Kittler, "Region growing: A new approach," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1079-1084, 1998.
9. S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," in *Proceedings of 2nd International Joint Conference Pattern Recognition*, pp. 424-433, 1977.
10. R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York: MIT Press and McGraw-Hill Inc., 1995.
11. H. Jiang, J. Toriwaki, and H. Suzuki, "Comparative performance evaluation of segmentation methods based on region growing and division," *Syst. Comput. Japan*, vol. 24, no. 13, pp. 28-42, 1993.
12. C. L. Liu, *Elements of Discrete Mathematics, 2nd Ed*. New York: McGraw-Hill International Editions, 1985.
13. I. N. Manousakas, P. E. Undrill, and T. W. Redpath, "A comparison of 3D split-and-merge segmentation with direct MRI determination of cerebral ventricule volume," in *Proceedings of SPIE Medical Imaging 1996: Image Processing*, vol. 2710, pp. 190-200.
14. T. Pavlidis and Y.T. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 225-233, Mar. 1990.
15. A. Rosenfeld and A. Kak, *Digital Picture Processing, 2nd Ed*. New York: Academic Press, 1982.
16. D. C. Taylor and W. A. Barrett, "Image segmentation using globally optimal growth in three dimensions with an adaptive feature set," in *Proceedings of SPIE 1994: Visualization in Biomedical Computing*, vol. 2359, pp. 98-107, R. A. Robb, Ed., 1994.
17. J. Toriwaki and S. Yokoi, "Basics of algorithms for processing three-dimensional digitized pictures," *Syst. And Comp. in Japan*, vol. 17, pp. 73-82, 1986.
18. S.-Y. Wan, A. Kiraly, E. L. Ritman, and W. Higgins, "Extraction of the hepatic vasculature in rats using 3D micro-CT images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 9, pp. 964-971, Sep. 2000.
19. S.-Y. Wan, E. L. Ritman, and W. E. Higgins, "Extraction and analysis of large vascular networks in 3D micro-CT images," in *Proceedings of SPIE Medical Imaging 1999: Physiology and Function from Multidimensional Images*, vol. 3660, pp. 322-334, A. Clough and C.-T. Chen, eds., 1999.
20. S.C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayer/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, 1996.