Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 467928, 14 pages
doi:10.1155/2008/467928

Research Article

Object Delineation by κ-Connected Components

Paulo A. V. Miranda, Alexandre X. Falcão, Anderson Rocha, and Felipe P. G. Bergo

Institute of Computing, University of Campinas, 13084-851 Campinas, SP, Brazil

Correspondence should be addressed to Alexandre X. Falcão, afalcao@ic.unicamp.br

Received 30 November 2007; Revised 27 March 2008; Accepted 2 June 2008

Recommended by Chein-I Chang

The notion of "strength of connectedness" between pixels has been successfully used in image segmentation. We present extensions to these works, which can considerably improve the efficiency of object delineation tasks. A set of pixels is said to be a κ-connected component with respect to a seed pixel, when the strength of connectedness of any pixel in that set with respect to the seed is higher than or equal to a threshold. We discuss two approaches that define objects based on κ-connected components with respect to a given seed set: with and without competition among seeds. While the previous approaches either assume no competition with a single threshold for all seeds or eliminate the threshold for seed competition, we show that seeds with different thresholds can improve segmentation in both paradigms. We also propose automatic and user-friendly interactive methods to determining the thresholds. The proposed methods are presented in the framework of the image foresting transform, which naturally leads to efficient and correct graph algorithms. The improvements are demonstrated through several segmentation experiments involving medical images.

1. INTRODUCTION

Image segmentation has been a challenge which involves object *recognition* and *delineation*. Recognition is represented by cognitive tasks that determine the approximate location of a desired object in a given image (object detection), verify the correctness of a segmentation result, and identify a desired object among candidate ones (object classification). Delineation is the task that completes segmentation by defining the precise spatial extent of the desired object in the image. Effective recognition requires object properties while accurate delineation usually depends on image properties to distinguish object and background.

In the context of interactive segmentation, a human operator performs the recognition tasks and the computer performs delineation. In order to make these approaches automatic, we must substitute the human operator by a mathematical model. Model-based approaches have used object properties to build numerical, geometrical, and statistical models for segmentation [1–3], and for simple object detection [4]. Since that a mathematical model usually acts worse than a human expert in the recognition task, it is important to develop interactive methods which minimize the user's time and involvement in the delineation process, such that their automation becomes feasible. For example, we are interested in reducing the user intervention to simple selection of a few pixels in the image.

Delineation methods are usually based on a functional of the arc-weights such as graph-cut approaches [5–9] or based on a connectivity functional in the form of a path-cost function [10–13]. This work advances the state-of-the-art of delineation methods based on connectivity functional, being the recognition tasks performed by human operators.

Fuzzy connectedness/watersheds are image segmentation approaches based on seed pixels, which have been successfully used in many applications [10, 14–18]. The relation between *relative-fuzzy connectedness* [11, 19, 20] and *watershed transform by markers* [12, 13] has been pointed out in [21] and formally proved in [22]. They are essentially the same method (one is the dual of the other), where the seeds are specified inside and outside the object, each seed defines an influence zone composed by pixels more strongly connected to that seed than to any other, and the object is defined by the union of the influence zones of its internal seeds. In *absolute-fuzzy connectedness* [23], a seed is specified inside the object, and the *strength of connectedness* of each pixel with respect to that seed is computed, such that the object is obtained by thresholding the resulting connectivity image. Clearly, these approaches represent two distinct region-based segmentation paradigms, with and without competition among seeds.

We present extensions to these works, using the framework of the *image foresting transform* (IFT) [21]—a general tool for the design, implementation, and evaluation of image processing operators based on connectivity. In the IFT, the image is interpreted as a graph, whose nodes are the image pixels and whose arcs are defined by an *adjacency relation* between pixels. The cost of a path in this graph is determined by an application-specific *path-cost function*, which usually depends on local image properties along the path—such as color, gradient, and pixel position. For suitable path-cost functions and a set of seed pixels, one can obtain an image partition as an *optimum-path forest* rooted at the seed set. That is, each seed is root of a *minimum-cost path tree* whose pixels are reached from that seed by a path of minimum cost, as compared to the cost of any other path starting in the seed set. The IFT essentially reduces image operators to a simple local processing of attributes of the forest [24–28].

The *strength of connectedness* of a pixel with respect to a seed is inversely related to the cost of the optimum path connecting the seed to that pixel in the graph. In absolute-fuzzy connectedness, the object can be obtained by selecting pixels reached from an internal seed by an optimum path whose cost is less than or equal to a number κ. In this case, the object is said to be a single *κ-connected component* (a minimum-cost path tree). The object can also be defined as the union of all κ-connected components created from each seed separately, which requires one IFT for each seed. In relative-fuzzy connectedness, seeds selected inside and outside the object compete among themselves, partitioning the image into an optimum-path forest, and the object is defined by the union of the optimum-path trees rooted at its internal seeds. The initial appeal for relative-fuzzy connectedness was the possibility to delineate multiple objects simultaneously, without depending on thresholds. However, the use of thresholding together with seed competition provides a hybrid approach which turns out to be more efficient than the previous ones in many situations. While the previous approaches either assume no competition with a single value of κ for all seeds or eliminate κ for seed competition, we show that seeds with different values of κ can considerably improve segmentation in both paradigms. Of course, this comes with the problem of finding the values of κ for each seed, but we provide automatic and user-friendly interactive ways to determine them.

Section 2 describes some definitions related to the IFT, making them more specific for region-based image segmentation. For the sake of simplicity, we will describe the methods for gray-scale and two-dimensional images, but they are extensive to multiparametric and multidimensional data sets. The proposed variants and their algorithms are presented in Sections 3 and 4. Section 5 demonstrates the improvements with respect to the previous approaches. Conclusion and future work are presented in Section 6.

2. BACKGROUND

An *image* $\hat{I}$ is a pair (D_I, I) consisting of a finite set D_I of *pixels* (points in $\mathbb{Z}^2$) and a mapping I that assigns to each pixel p in D_I a *pixel value* $I(p)$ in some arbitrary value space.

An *adjacency relation* A is a binary relation between pixels p and q of D_I. We use $q \in A(p)$ and $(p,q) \in A$ to indicate that q is adjacent to p. Once the adjacency relation A has been fixed, the image $\hat{I}$ can be interpreted as a directed graph (D_I, A) whose nodes are the image pixels in D_I and whose arcs are the pixel pairs (p,q) in A. We are interested in irreflexive, symmetric, and translation-invariant relations. For example, one can take A to consist of all pairs of pixels (p,q) in the Cartesian product $D_I \times D_I$ such that $d(p,q) \le \rho$ and $p \ne q$, where $d(p,q)$ denotes the Euclidean distance and ρ is a specified constant (i.e., 4-adjacency, when $\rho = 1$, and 8-adjacency, when $\rho = \sqrt{2}$).

A *path* is a sequence $\pi = \langle p_1, p_2, \ldots, p_n \rangle$ of pixels, where $(p_i, p_{i+1}) \in A$, for $1 \le i \le n-1$. The path is *trivial* if $n = 1$. Let $\mathrm{org}(\pi) = p_1$ and $\mathrm{dst}(\pi) = p_n$ be the origin and destination of a path π. If π and τ are paths such that $\mathrm{dst}(\pi) = \mathrm{org}(\tau) = p$, we denote by $\pi \cdot \tau$ the concatenation of the two paths, with the two joining instances of p merged into one. In particular, $\pi \cdot \langle p,q \rangle$ is a path resulting from the concatenation of its longest prefix π and the last arc $(p,q) \in A$.

A *predecessor map* is a function P that assigns to each pixel $q \in D_I$ either some other pixel in D_I, or a distinctive marker nil not in D_I—in which case q is said to be a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles—in other words, one which takes every pixel to nil in a finite number of iterations. For any pixel $q \in D_I$, a spanning forest P defines a path $P^*(q)$ recursively as $\langle q \rangle$, if $P(q) = \mathrm{nil}$, or $P^*(p) \cdot \langle p,q \rangle$ if $P(q) = p \ne \mathrm{nil}$ (see Figure 1(a)).

A pixel q is *connected* to a pixel p if there exists a path in the graph from p to q. In this sense, every pixel is connected to itself by its trivial path. Since A is symmetric, we can also say that p is connected to q, or simply p and q are connected. Therefore, a *connected component* is a subset of D_I wherein all pairs of pixels are connected.

A *path-cost function* f assigns to each path π a *path cost* $f(\pi)$, in some totally ordered set V of cost values, whose maximum element is denoted by $+\infty$. A path π is *optimum* if $f(\pi) \le f(\tau)$ for any other path τ with $\mathrm{dst}(\tau) = \mathrm{dst}(\pi)$, irrespective to its starting point. The IFT establishes some conditions applied to optimum paths, which are satisfied by only *smooth* path-cost functions. That is, for any pixel $q \in D_I$, there must exist an optimum path π ending at q which either is trivial, or has the form $\tau \cdot \langle p,q \rangle$, where

(C1) $f(\tau) \le f(\pi)$,

(C2) τ is optimum,

(C3) for any optimum path τ' ending at p, $f(\tau' \cdot \langle p,q \rangle) = f(\pi)$.

The IFT takes an image $\hat{I}$, a smooth path-cost function f and an adjacency relation A; and returns an *optimum-path forest*—a spanning forest P such that $P^*(q)$ is optimum for every pixel $q \in D_I$. In the forest, there are three important attributes for each pixel: its predecessor in the optimum

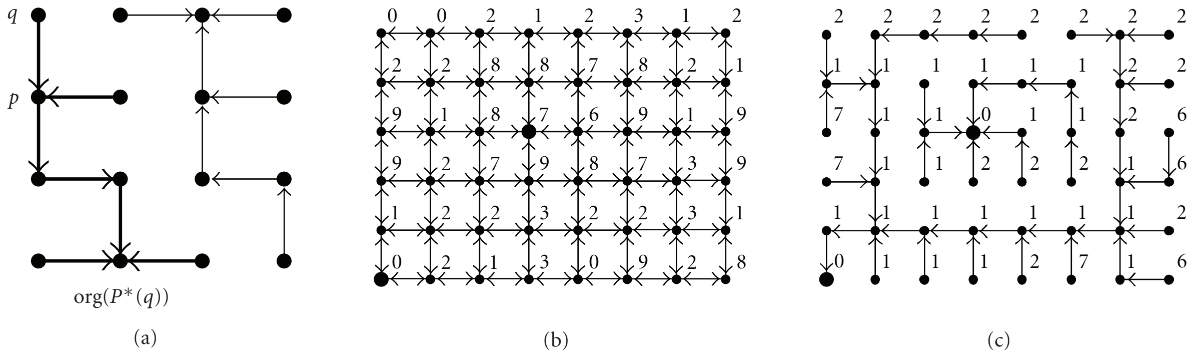


FIGURE 1: (a) The main elements of a spanning forest with two roots, where the thicker path indicates $P^*(q)$. (b) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dots indicate two seeds. One is inside the brighter rectangle and one is in the darker background outside it. Note that the background also contains brighter parts. (c) An optimum-path forest for $f_{\max}$, with $\delta(p,q) = |I(q) - I(p)|$. The integers are the cost values, and the rectangle is obtained as an optimum-path tree rooted at the internal seed.

path, the cost of that path, and the corresponding root (or some label associated with it). The IFT-based image operators result from simple local processing of one or more of these attributes.

For a given seed set $S \subset D_I$, the concept of *strength of connectedness* [23, 29] of a pixel $q \in D_I$ with respect to a seed $s \in S$ can be interpreted as an image property inversely related to the cost of the optimum path from s to q according to the max-arc path-cost function $f_{\max}$:

$$f_{\max}(\langle q \rangle) = \begin{cases} 0, & \text{if } q \in S, \\ +\infty, & \text{otherwise,} \end{cases} \qquad (1)$$

$$f_{\max}(\pi \cdot \langle p, q \rangle) = \max\{f_{\max}(\pi), \delta(p,q)\},$$

where $(p,q) \in A$, π is any path ending at p and starting in S, and $\delta(p,q)$ is a nonnegative *dissimilarity function* between p and q which depends on image properties, such as brightness and gradient (see Figures 1(b) and 1(c)).

One may think of smoothness as a more general definition for strength of connectedness. In this work, we discuss only $f_{\max}$ because the comparison with previous approaches and our practical experience in region-based segmentation, which shows that $f_{\max}$ often leads to better results than other commonly known smooth cost functions.

3. IMAGE SEGMENTATION BY κ-CONNECTIVITY

We assume given a seed set S either interactively, by simple mouse clicks, or automatically, based on some a priori knowledge about the approximate location of the object. The adjacency relation A is usually a simple 8-neighborhood, but sometimes it is important to allow farther pixels be adjacent. This may reduce the number of seeds required to label nearby components of a same object, such as letters of a word in the image of a text. Some examples of δ functions for $f_{\max}$ are given below:

$$\delta_1(p,q) = K\left(1 - \exp\left(-\frac{1}{2\sigma^2}\left(I(p) - I(q)\right)^2\right)\right), \qquad (2)$$

$$\delta_2(p,q) = G(q), \qquad (3)$$

$$\delta_3(p,q) = K\left(1 - \exp\left(-\frac{1}{2\sigma^2}\left(\frac{I(p)+I(q)}{2} - I(s)\right)^2\right)\right), \qquad (4)$$

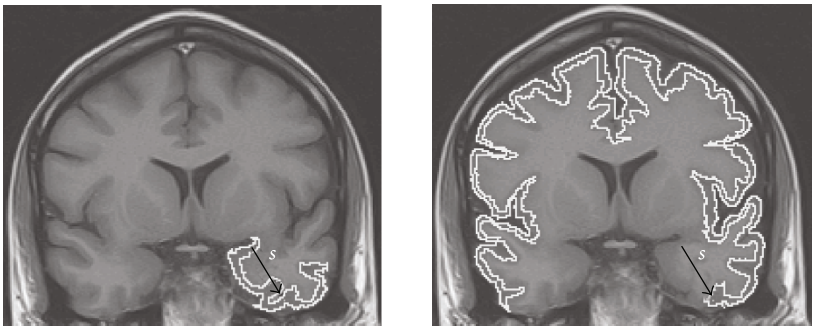
$$\delta_4(p,q) = \min_{\forall s \in S}\{\delta_3(p,q)\}, \qquad (5)$$

$$\delta_5(p,q) = a\delta_1(p,q) + b\delta_3(p,q), \qquad (6)$$

$$\delta_6(p,q) = \begin{cases} \delta_3(p,q)\left(1 + \vec{g}(p,q) \cdot \vec{\eta}(p,q)\right), \\ \qquad \text{if } E_r(p,q) > D_r(p,q), \\ K, \quad \text{otherwise,} \end{cases} \qquad (7)$$

where K is a positive integer (e.g., the maximum image intensity), σ is an allowed intensity variation, $G(q)$ is a gradient magnitude computed at q, and $I(s)$ is the intensity of a seed $s \in S$, such that $s = \text{org}(P^*(p))$ in δ_3 and δ_4 considers all seeds in S. The parameters a and b are constants such that $a + b = 1$, and $\vec{g}(p,q)$ is a normalized gradient vector computed at arc(p,q), $\vec{\eta}(p,q)$ is the unit vector of the arc(p,q), $E_r(p,q)$, and $D_r(p,q)$ are the pixel intensities at a distance r to the left and right sides of the arc(p,q), respectively.

The dissimilarity functions aim to penalize arcs that cross borders, by assigning higher arc weights to them. We are interested in using the above functions under two possible segmentation paradigms: with and without seed competition. Functions δ_1 and δ_2 assume low inhomogeneity within the object. They represent gradient magnitudes with different image resolutions and lead to smooth functions in both paradigms. In fact, $f_{\max}$ is smooth whenever $\delta(p,q)$ is fixed for any $(p,q) \in A$. Function δ_3 exploits the dissimilarity between object and pixel intensities, being the object represented by its seed pixels. Although $f_{\max}$ is smooth for δ_3 with no seed competition, it may not be smooth in the case of competition among seeds [21] (i.e., the IFT results in a spanning forest, but it may be non-optimal). This problem was the main motivation for δ_4 [11]. However, sometimes δ_3 with seed competition provides better segmentation results than δ_4 (see Section 5). Function δ_3 may also limit the influence zones of the seeds, when the intensities inside the object vary linearly toward the background. Function δ_5



(a) (b)

Figure 2: An MR-T1 image of the brain with one seed s inside the cortex. (a-b) The maximum influence zones of s within the cortex for $f_{\max}$ with δ_3 and with δ_6, respectively. The asymmetry of δ_6 favors segmentation in anticlockwise orientation, increasing the influence zone of s.

reduces this problem, and in the case of seed competition, one can also replace δ_3 by δ_4 in (6). The basic idea in function δ_6 stemmed from [30], where the intensities on the left and right sides of each arc are used to compute its weight, such that longer boundary segments are favored in only one orientation (either clockwise or anticlockwise). We are extending this idea to provide oriented region growing. Function δ_6 is suitable to objects, such as the cortex, composed by intermediary intensities with respect to the intensities on both of its sides. For MR-T1 images of the brain, the GM intensities in the cortex are expected to be higher than the intensities in one side (CSF) and lower than the intensities in the other side (WM). To grow regions in anticlockwise, we expect that the intensity $E_r(p,q)$ at a distance r to the left (WM) of an arc (p,q) be higher than the intensity $D_r(p,q)$ at the same distance r to the right (CSF) of the arc. We favor or penalize the arc dissimilarities based on this rule in δ_6. The term $\vec{g}(p,q)\cdot\vec{\eta}(p,q)$ also penalizes arcs which cross boundaries. The result is that the same seed s allows to delineate more pixels in the cortex with δ_6 (Figure 2(b)), following the anticlockwise orientation, than with δ_3 (Figure 2(a)). Other interesting ideas of dissimilarity functions for $f_{\max}$ are presented in [11, 19, 23, 31, 32].

The basic differences between the formulations proposed in [11, 19, 20] are that (i) the former assumes $\delta(p,q) = \delta(q,p)$ for all $(p,q) \in A$, and requires smooth path-cost functions, and (ii) the later allows asymmetric dissimilarity relations (e.g., δ_2), and nonsmooth cost functions (e.g., $f_{\max}$ with δ_3 and seed competition). The strength of connectedness between image pixels in (i) is a symmetric relation, while it may be asymmetric in (ii). The main theoretical differences between our formulation and these ones are presented next.

3.1. Object definition without seed competition

We say that a pixel p is *κ-connected* to a seed $s \in S$, if there exists an optimum path π from s to p such that $f(\pi) \leq \kappa$. This κ-connectivity relation will be asymmetric whenever the dissimilarity $\delta(p,q)$ is asymmetric.

An *object* is a maximal subset of D_I wherein all pixels p are at least κ-connected to one pixel $s \in S$. Similarly to the method presented in [23], the object is the union of all κ-connected components with respect to each seed $s \in S$, which must be computed separately. This makes $f_{\max}$ smooth for all dissimilarity functions described in (2)–(7).

The algorithm described in [23] assumes that the object can be defined by a single value of κ for all seeds in S. Figure 3(a) illustrates an example where this assumption works. However, a simple change in the position of a seed can fail segmentation (Figure 3(b)), because the influence zone of each seed inside the object is actually limited by a distinct value of cost κ (Figure 3(c)). Moreover, the choice of seeds with distinct values of κ usually reduces the number of seeds required to complete segmentation. This situation is better understood when we relate the concepts of minimum-spanning tree and minimum-cost path tree for $f_{\max}$ and symmetric κ-connectivity relations [33].

A *minimum-spanning tree* is a spanning forest P with a single arbitrary root, where the sum of the arc weights $\delta(p,q)$ for all pairs $(p,q) \in A$, such that $P(q) = p$, is minimum, as compared to any other minimum-spanning tree obtained from the original graph (D_I, A) (Figures 4(a) and 4(b)). If we remove the orientation of the arcs in Figure 4(b), every pair of pixels in P is connected by a path which is also optimum according to $f_{\max}$ (Figure 4(c)). That is, the minimum-spanning tree encodes all possible minimum-cost path trees for $f_{\max}$. A κ-connected object with respect to a seed s can be obtained by taking the component connected to s, after removing all arcs from P whose $\delta(p,q) > \kappa$. Suppose, for example, that the object is the brighter rectangle in the center of Figure 4(a). Figure 4(c) shows that only the left side of the rectangle is obtained with s_1 and $\kappa_1 = 3$. If $\kappa_1 = 4$, s_1 reaches the right side of the rectangle but invades the background. The rectangle can be obtained with three seeds and $\kappa = 2$. However, different values of κ reduce the number of seeds to two, s_1 with $\kappa_1 = 3$ and s_2 with $\kappa_2 = 2$ (Figure 4(d)).

There may be many arcs connecting object and background in a minimum-spanning tree. The choice of a single value of κ is equivalent to remove the arcs whose weight $\delta(p,q)$ is minimum among those connecting object and background. This usually divides the object into several κ-connected components (minimum-spanning trees) and the segmentation will require one seed for each component. When we allow different values of κ, the object components become larger, and consequently, the number of seeds is reduced.

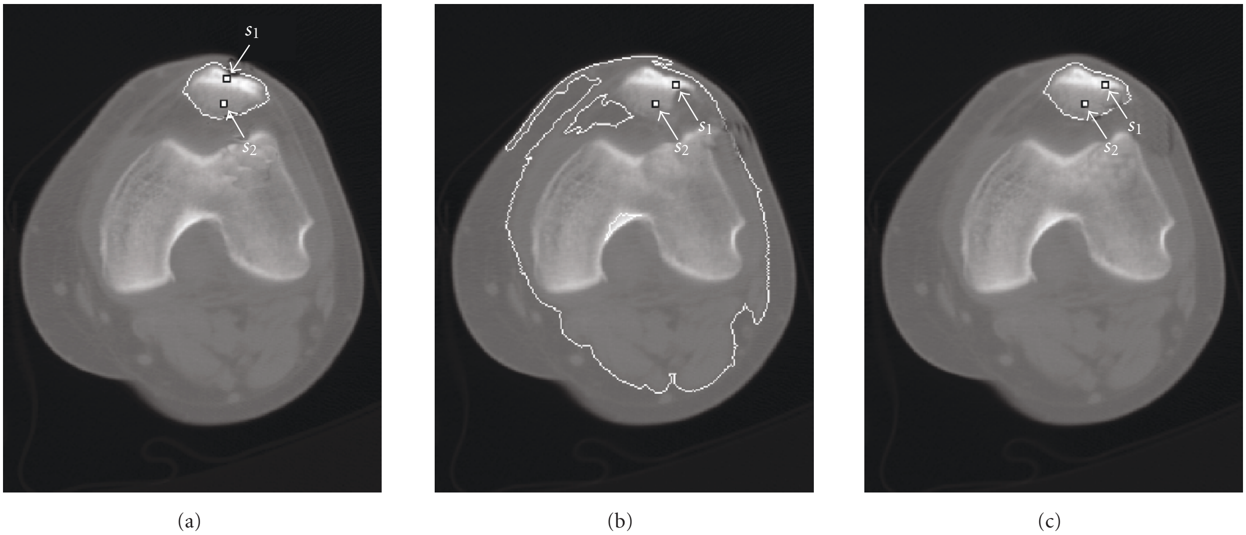


Figure 3: A CT image of a knee where the patella can be segmented with two seed pixels, s_1 and s_2, $f_{\max}$ with δ_3, and without seed competition. (a) The result with a single value of κ for both seeds. (b) The segmentation with a single value of κ fails when we change the position of s_1, because s_1 requires a higher value of κ to get the brighter part of the bone, and B invades the background at this higher value of κ. (c) The result can be corrected with distinct values of κ for each seed.

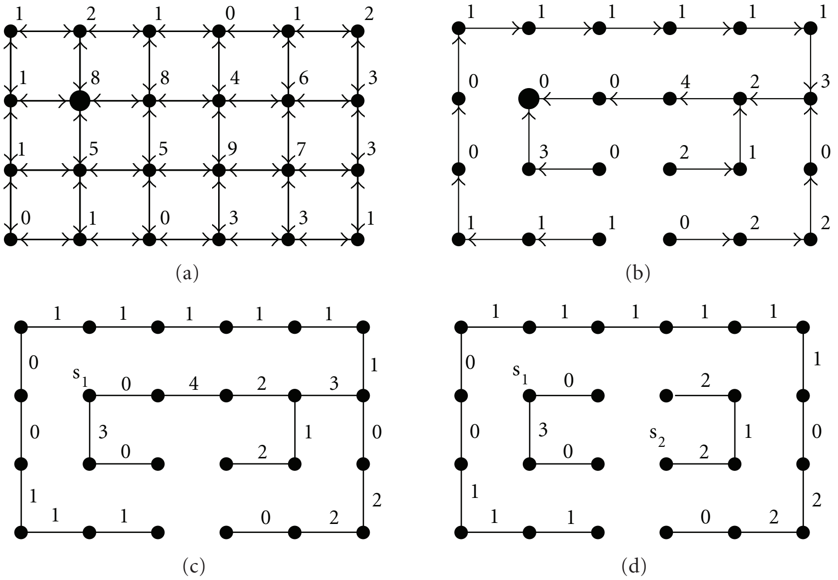


Figure 4: (a) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dot is an arbitrary pixel. The object of interest is the brighter rectangle in the center. (b) A minimum-spanning tree computed from the arbitrary pixel, where the integers for each pixel q are the arc weights $\delta(p,q) = |I(q) - I(p)|$, for $p = P(q)$. (c) The minimum-spanning tree without arc orientation. A single seed s_1 can not extract the rectangle for any value of κ. (d) The rectangle can be obtained with two seeds and distinct values of κ, s_1 with $\kappa_1 = 3$ and s_2 with $\kappa_2 = 2$.

3.2. *Object definition with seed competition*

In [11, 19], seeds are selected inside and outside the object, and the *object* is defined by the subset of pixels which are more strongly connected to its internal seeds than to any other. This is the same as removing the arcs of maximum weight from the paths that connect object and background in the minimum-spanning tree. For example, the rectangle in Figure 4(c) is obtained by changing the position of s_1 to any pixel in the background and selecting s_2 as shown in Figure 4(d). The main motivation for this paradigm was to eliminate the choice of κ, favoring the simultaneous segmentation of multiple objects.

We define the *object* as the subset of pixels which are more strongly κ-connected to its internal seeds than to any other. That is, the seeds will compete among themselves for pixels reached from more than one seed by paths whose costs are less than or equal to κ. In which case, the pixel is conquered

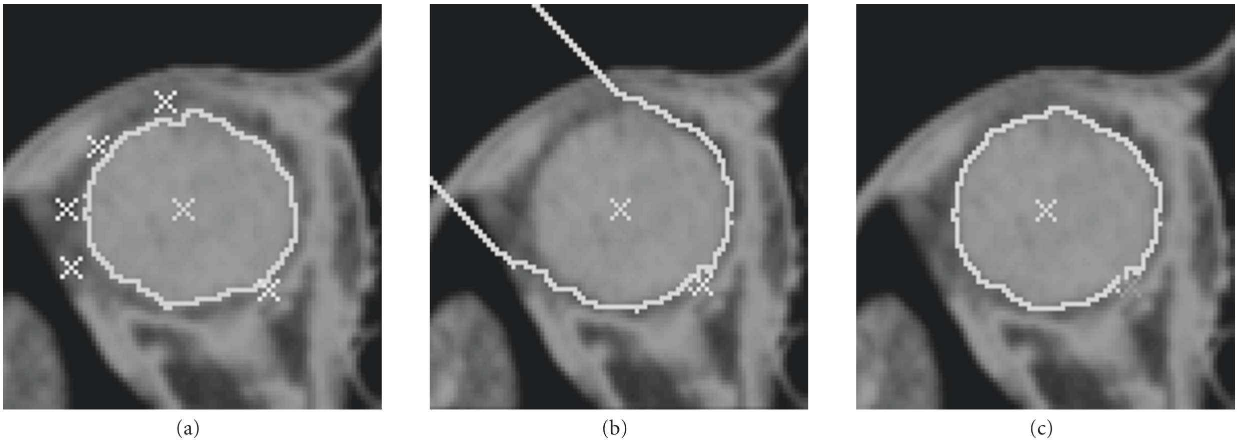


FIGURE 5: A CT image of the orbital region where the eye ball is obtained by seed competition. (a) One internal seed and many external seeds are required for segmentation, using $f_{\max}$ with δ_4. (b) The segmentation fails when some of the external seeds are removed. (c) A value of κ is used to limit the influence zone of the internal seed in parts, where the seed competition fails.

by the seed whose path cost is minimum. Note that even the internal seeds compete among themselves, and a distinct value of κ may be required for each seed. When the seed competition fails, these thresholds should limit the influence zones of the seeds avoiding connection between object and background, and the pixels, which are not conquered by any seed, should be considered as belonging to the background.

In general, the use of distinct values of κ together with seed competition reduces the number of seeds required to complete segmentation. Figure 5(a) shows an example where many seeds have to be carefully selected in the background to delineate the object. The segmentation fails when some of these seeds are removed (Figure 5(b)), but it works when we limit the extent of the internal seed to some value of κ (Figure 5(c)).

The algorithms and the problem of determining these thresholds for the internal seeds are addressed next.

4. ALGORITHMS

The IFT uses a variant of Dijkstra's algorithm [34] to compute three attributes for each pixel $p \in D_I$ [21]: its predecessor $P(p)$ in the optimum path, the cost $C(p)$ of that path, and the corresponding root $R(p)$. In the algorithms presented in this section, we do not need to create the predecessor map P and the root map R is only used in the case of seed competition.

The IFT with $f_{\max}$ propagates wavefronts W_{cst} of same cost cst around each seed, following the order of the costs $\text{cst} = 0, 1, \ldots, K$. By assigning higher values of $\delta(p,q)$ to arcs that cross the object's boundary, the wavefronts fill first the object and, when they leak to the background, a considerable increase in their areas can be observed (Figures 6(a) and 6(b)). That is, many pixels in the background are reached by optimum paths whose cost is the lowest value $\delta(p,q)$ among the dissimilarities of the arcs (p,q) that cross the boundary. This ordered region growing process is exploited to compute the values κ_s of each seed $s \in S$ automatically and interactively.

4.1. *Automatic computation of κ_s*

First consider the wavefronts around a seed s selected inside a given object. All pixels p in the wavefront W_{cst} around s have optimum cost $C(p) = \text{cst}$, $0 \leq \text{cst} \leq K$. If the object is a single κ-connected component with respect to s, then there exists a threshold κ_s, $0 \leq \kappa_s \leq K$, such that the object can be defined by the union of all wavefronts W_{cst}, for $\text{cst} = 0, 1, \ldots, \kappa_s$. We can specify a fixed κ_s for this particular application, but this is susceptible to intensity variations. Another alternative is to search for matchings between the shape of the object and the shape of the wavefronts. One drawback is the speed of segmentation, but this may be justified in some applications. A more complex situation occurs when the object definition requires more than one seed pixel. Each seed defines its own maximal extent inside the object and we need to match the shape of the object with the shape of the union of their influence zones.

The approach presented here is much simpler and yet effective. It stems from the previously mentioned observation about the areas of the wavefronts, when they invade the background. The ordered region growing process of a seed s must stop when the size of its wavefront of cost cst is greater than an area threshold $0\% < T < 100\%$, computed over the image size, and the value of κ_s is determined as $\max\{\text{cst} - 1, 0\}$. The choice of one value κ_s for each seed $s \in S$ is then substituted by the choice of T, which limits the maximum sizes of the wavefronts. This threshold can be verified by selecting internal seeds and setting $T = 99\%$. The total area of the wavefronts during propagation can be displayed as a curve. A peak on this curve indicates the maximum possible value for T at the instant of leaking. Some animations of this ordered region growing process are provided in http://www.liv.ic.unicamp.br/demo/miranda-kconnected.avi.

The algorithms are presented for single object delineation without seed competition (Algorithm 1) and multiple object definition with seed competition (Algorithm 2).

The priority queue Q can be implemented as described in [35, 36], such that each instance of the IFT will run in time proportional to the number $|D_I|$ of pixels. Note that the first

```
INPUT: Image $\hat{I} = (D_I, I)$, adjacency $A$, internal seeds $S$, and path-cost function $f_{\max}$, and the size threshold $T$.
OUTPUT: Binary image $\hat{L} = (D_I, L)$, where $L(p) = 1$, if $p$ belongs to the object, and $L(p) = 0$ otherwise.
Auxiliary: A priority queue $Q$, variables tmp, $\kappa$, cst and size, and cost map $C$ defined in $D_I$.

For every pixel $p \in D_I$, set $L(p) \leftarrow 0$.
While $S \neq \emptyset$, do
    For every pixel $p \in D_I$, set $C(p) \leftarrow +\infty$.
    Remove a seed $s$ from $S$.
    Set $C(s) \leftarrow 0$, size $\leftarrow 0$, cst $\leftarrow 0$, $\kappa \leftarrow +\infty$, and insert $s$ in $Q$.
    While $Q \neq \emptyset$ and $\kappa = +\infty$, do
        Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
        For every $q \in A(p)$, such that $C(q) > C(p)$, do
            Set tmp $\leftarrow \max\{C(p), \delta(p,q)\}$.
            If tmp $< C(q)$, then
                If $C(q) \neq +\infty$, then remove $q$ from $Q$.
                Set $C(q) \leftarrow$ tmp and insert $q$ in $Q$.
        If $C(p) \neq$ cst, then set size $\leftarrow 1$ and cst $\leftarrow C(p)$.
        Else, set size $\leftarrow$ size $+ 1$.
        If size $> T$ then set $\kappa \leftarrow \max\{\text{cst} - 1, 0\}$
    For every pixel $p \in D_I$, do
        If $C(p) \leq \kappa$, then set $L(p) \leftarrow 1$.
    Remove any remaining pixels from $Q$.
```

Algorithm 1: Single object definition without seed competition.

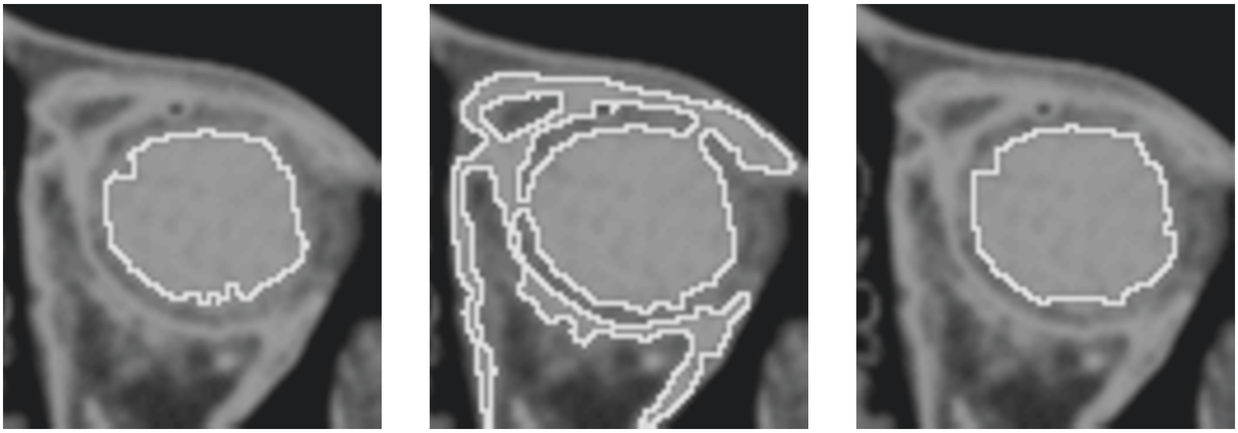
```
Input: Image $\hat{I} = (D_I, I)$, adjacency $A$, path-cost function $f_{\max}$, size threshold $T$, and a labeled image $\hat{L} = (D_I, L)$,
where $L(p) = i$, $0 \leq i \leq k$, if $p$ is a seed pixel selected inside object $i > 0$ among $k$ objects, being $i = 0$ reserved
for seeds in the background, and $L(p) = -1$ otherwise.
Output: A labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$, $0 \leq i \leq k$.
Auxiliary: Priority queue $Q$, variable tmp, and $C$, $R$, $\kappa$, size, and cst are maps defined in $D_I$ to store cost and root
of each pixel and threshold, wavefront size, and wavefront cost of each seed, respectively.

For every pixel $p \in D_I$, do
    Set $R(p) \leftarrow p$, size$(p) \leftarrow 0$, cst$(p) \leftarrow 0$, and $\kappa(p) \leftarrow +\infty$.
    If $L(p) = -1$, then set $C(p) \leftarrow +\infty$ and $L(p) \leftarrow 0$.
    Else, set $C(p) \leftarrow 0$ and insert $p$ in $Q$.
While $Q \neq \emptyset$, do
    Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
    If $\kappa(R(p)) = +\infty$ and $L(R(p)) \neq 0$, then
        If $C(p) \neq$ cst$(R(p))$, then set size$(R(p)) \leftarrow 1$ and cst$(R(p)) \leftarrow C(p)$.
        Else, set size$(R(p)) \leftarrow$size$(R(p)) + 1$.
        If size$(R(p)) > T$, then set $\kappa(R(p)) \leftarrow \max\{\text{cst}(R(p)) - 1, 0\}$.
    If $C(p) \leq \kappa(R(p))$, then
        For every $q \in A(p)$, such that $C(q) > C(p)$, do
            Set tmp $\leftarrow \max\{C(p), \delta(p,q)\}$.
            If tmp $< C(q)$, then
                If $C(q) \neq +\infty$, then remove $q$ from $Q$.
                Set $C(q) \leftarrow$ tmp, $R(q) \leftarrow R(p)$, and insert $q$ in $Q$.
For every pixel $p \in D_I$, do
    If $C(p) \leq \kappa(R(p))$, then set $L(p) \leftarrow L(R(p))$.
```

Algorithm 2: Multiple object definition with seed competition.

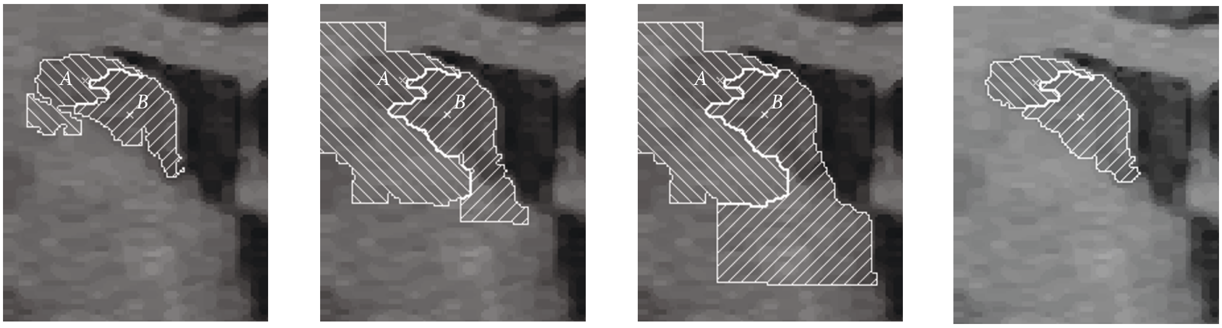
algorithm stops propagation when the value κ_s of a seed s is found. In the case of seed competition, the root map is used to find in constant time the root of each pixel in S. The influence zone of a seed $s \in S$ is limited either when it meets the influence zone of other seed at the same minimum cost or when the value κ_s of s is found.

One advantage of the presented algorithms as compared to classical segmentation methods based on seed competition



(a) (b) (c)

Figure 6: A CT image of the orbital region with one seed inside the eye ball. (a) A wavefront of cost κ which represents the maximum extent of this seed inside the eye ball. (b) The wavefront of cost $\kappa + 1$ shows a considerable augment in size when it invades the background. (c) The pixel propagation order provides more continuous transitions of the wavefronts to select κ, interactively.



(a) (b) (c) (d)

Figure 7: Segmentation of a caudate nucleus with two internal seeds, A and B. (a) The leaking occurs before the object be filled. (b) The moment when $\kappa_A = 324$ is detected. (c) The instant when κ_B is detected. (d) Final segmentation.

occurs when the object contains several background parts (holes) inside it. In this case, the use of κ_s usually eliminates the need for at least one background seed at each hole. On the other hand, some small noisy parts of the object may not be conquered by the internal seeds due to the use of κ_s. The labeled image can be postprocessed, such that holes with area below a threshold are closed [37, 38]. The area closing operator has shown to be a very effective complement for the presented algorithms. In many situations, the objects do not have holes and high area thresholds can be used to reduce the number of internal seeds.

The animations in http://www.liv.ic.unicamp.br/demo/miranda-kconnected.avi were created by using Algorithm 2. It is usually preferable with respect to Algorithm 1, because it allows faster multiple object segmentation. Note that a wavefront of one seed can leak to the background before the object be fully filled by the wavefronts of other seeds. Figure 7(a) illustrates an example where the leaking occurs for seed A before the object be filled. The moment when $\kappa_A = 324$ is detected is shown in Figure 7(b), and Figure 7(c) shows the instant when $\kappa_B = 770$ is detected. The figures show only a region of interest of the original image, where the segmentation was done with $T = 1\%$. The final segmentation is shown in Figure 7(d). Even when the dissimilarities are not higher for arcs that cross the object's boundary, Algorithm 2 can work either due to the seed competition among internal seeds (parts of the object can be filled without leaking) or due to the automatic κ_s computation, as shown in the example of Figure 7.

4.2. *Interactive computation of* κ_s

A first approach is to compute the IFT for every pixel $p \in D_I$, such that the cost $C(p)$ of the optimum path that reaches p from S is found. In the case of seed competition, the corresponding root $R(p) \in S$ is also propagated to each pixel $p \in D_I$. Then, the user moves the cursor of the mouse over the image, and for each position q of the cursor, the program displays the influence zone of the corresponding root $s = R(q) \in S$ defined by pixels $p \in D_I$, such that $C(p) \leq C(q)$ and $R(p) = R(q)$. This interactive process can be repeated until the user selects a pixel q to confirm the influence zone of s (i.e., $\kappa_s = C(q)$). The user can repeat this interactive process for each seed $s \in S$, in both paradigms.

```
Input: Image $\hat{I} = (D_I, I)$, adjacency $A$, internal seeds $S$, and path-cost function $f_{\max}$.
Output: Binary image $\hat{L} = (D_I, L)$, where $L(p) = 1$, if $p$ belongs to the object, and $L(p) = 0$ otherwise.
Auxiliary: Priority queue $Q$, variables tmp, ord, and cost map $C$ and propagation order map $O$ defined in $D_I$.

For every pixel $p \in D_I$, set $L(p) \leftarrow 0$.
While $S \neq \varnothing$, do
    For every pixel $p \in D_I$, set $C(p) \leftarrow +\infty$.
    Remove a seed $s$ from $S$.
    Set $C(s) \leftarrow 0$, ord $\leftarrow 0$, and insert $s$ in $Q$.
    While $Q \neq \varnothing$, do
        Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
        Set $O(p) \leftarrow$ ord + 1 and ord $\leftarrow$ ord + 1.
        For every $q \in A(p)$, such that $C(q) > C(p)$, do
            Set tmp $\leftarrow \max\{C(p), \delta(p,q)\}$.
            If tmp $< C(q)$, then
                If $C(q) \neq +\infty$, then remove $q$ from $Q$.
                Set $C(q) \leftarrow$ tmp and insert $q$ in $Q$.
The user selects a pixel $q$ on the image.
For every pixel $p \in D_I$, do
    If $O(p) \leq O(q)$, then set $L(p) \leftarrow 1$.
```

Algorithm 3: Single object definition without seed competition.

One drawback of the method above is the abrupt size variations of the wavefronts (Figures 6(a) and 6(b)), which makes the selection of pixel q sometimes difficult. We circumvent the problem by exploiting the *propagation order* $O(p)$ (a number from 1 to $|D_I|$) of each pixel p removed from Q during execution of the IFT. Note that a pixel p propagates before a pixel q (i.e., $O(p) < O(q)$) when it is reached by an optimum path from S, whose cost $C(p)$ is less than the cost $C(q)$ of the optimum path that reaches q. When $C(p) = C(q)$, we assume a *first-in-first-out* (FIFO) tie-breaking policy for Q. That is, among all pixels with the same minimum cost in Q, the one first reached by an optimum path from S is removed for propagation. Therefore, we also compute the propagation order $O(p)$ of each pixel $p \in D_I$. When the user moves the cursor to a position q, the program displays the influence zone of the corresponding root $s = R(q) \in S$ defined by pixels $p \in D_I$, such that $O(p) \leq O(q)$ and $R(p) = R(q)$. The rest of the process is the same. Note that although $\kappa_s = C(q)$, only the pixels p in the wavefront $W_{C(q)}$ which have $O(p) \leq O(q)$ are selected as belonging to the influence zone of s. This provides smoother transitions between consecutive wavefronts (Figure 6(c)) as compared to the first idea. See Algorithms 3 and 4.

5. EVALUATION

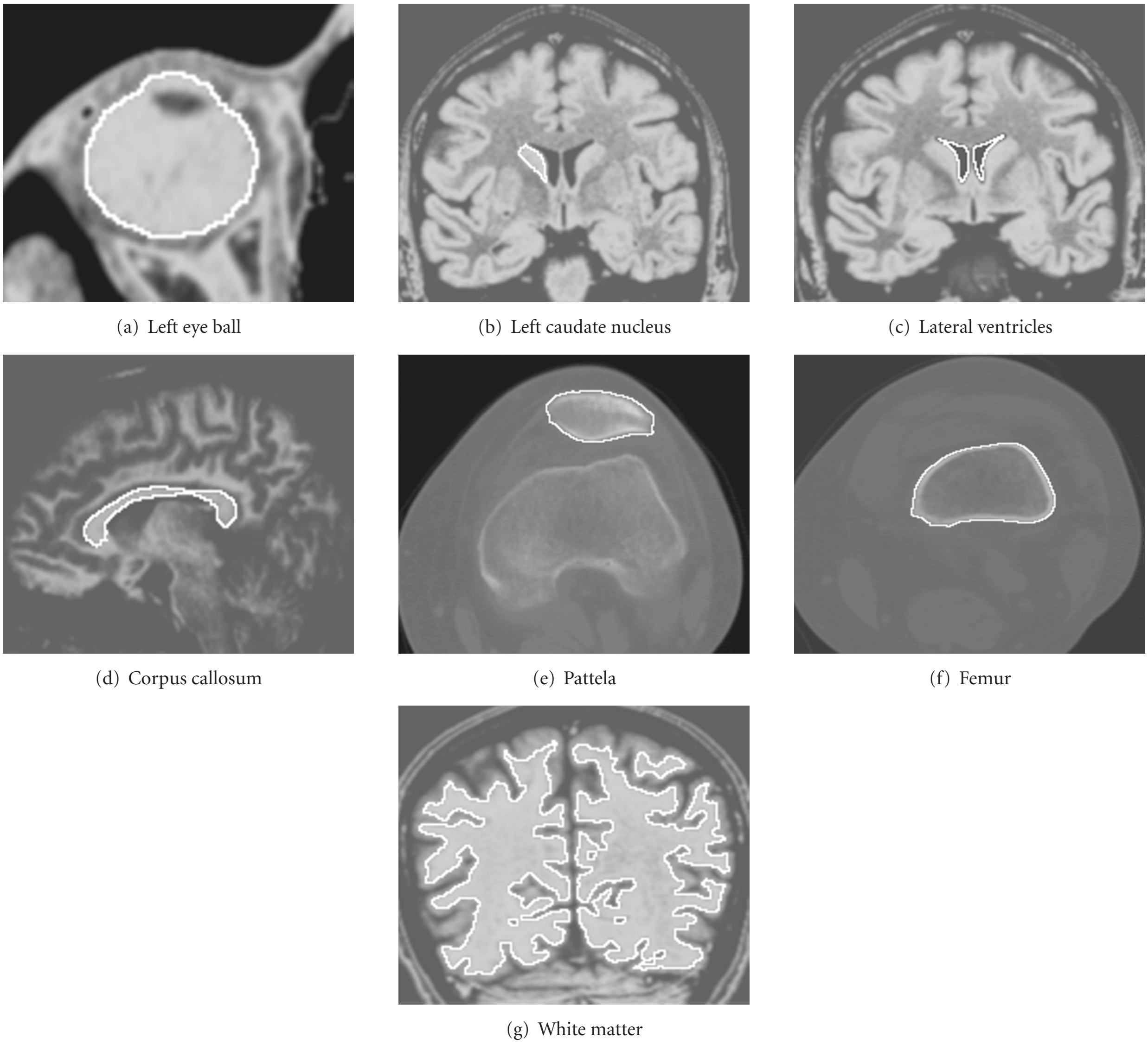
We have selected 100 images from magnetic resonance (MR) and computerized tomography (CT) data sets of 7 objects for evaluation (see Table 1 and Figure 8). Each object consists of some slices that represent different degrees of challenge for segmentation. The original images have been preprocessed to increase the similarities between pixels inside the objects and the contrast between object and background. Each of four users has performed segmentation over the 100 images using each of three methods, M1, M2, and M3, with interactive seed selection (mouse clicks).

Table 1: Description, imaging modality, and number of slices for each object used in the experiments.

Object	Description	Imaging modality	Number of slices
O1	Left eye ball	CT-orbit	15
O2	Left caudate nucleus	MR-brain	15
O3	Lateral ventricles	MR-brain	15
O4	Corpus callosum	MR-brain	10
O5	Patella	CT-knee	15
O6	Femur	CT-knee	15
O7	White matter	MR-brain	15

M1: Object delineation without seed competition and automatic/interactive computation of κ_s. This method uses Algorithms 1 and 3. When M1 requires a single κ_s for all seeds, it indicates that absolute-fuzzy connectedness (AFC) would work.

M2: Object delineation with seed competition and automatic κ_s computation. This method uses only Algorithm 2. We did not evaluate Algorithm 4, because preliminary tests indicated that user intervention to add external seeds in Algorithm 2 is simpler and more effective than to indicate κ_s in Algorithm 4.


(a) Left eye ball

(b) Left caudate nucleus

(c) Lateral ventricles

(d) Corpus callosum

(e) Pattela

(f) Femur

(g) White matter

Figure 8: (a)–(g) Results of slice segmentation of the objects from 1 to 7, respectively, overlaid with the preprocessed images.

M3: Object delineation with seed competition without κ_s computation. As mentioned in Section 1, relative-fuzzy connectedness (RFC) and watershed transform by markers (WT) are the same method [22] (one is the dual of the other), represented here by M3.

Therefore, the user can correct segmentation by adding/removing seeds in M1, M2, and M3, and in the case of M1, by pointing the mouse to the pixel, whose propagation order indicates the correct κ_s implicitly (Section 4.2).

M1 aims to show two aspects about AFC: (i) a single κ for all seeds is not sufficient in most cases and (ii) the problem of computing multiple κ_s thresholds can be easily solved by a wavefront area threshold $0\% < T < 100\%$, computed over the image size. Note that, being M1 an extension of AFC, there is no situation where AFC works and M1 would fail. M2 aims to reduce the number of required seeds with respect to M3 by automatic κ_s computation. When this automatic procedure fails, M2 becomes M3. Therefore, in the worst case, the efficiency of M2 should be the same of M3.

Given that M1 and M2 are extensions of AFC and RFC/WT, we expect that they do not affect the accuracy of the original approaches, which is assumed to be good from the results of several other works [10, 14–18]. The experiments then aimed to show that M1 works in situations where AFC would fail, M1 and M2 require less user interaction than M3, and the methods produce similar results.

The choice of parameters took a couple of minutes per object, by trying the methods in a first slice. Then, the parameters were fixed to the rest of the slices. Note that this can be done only once for any given application (object of interest and imaging protocol). We have chosen the best dissimilarity function for each object and method (Table 2). We used the 8-neighborhood as adjacency relation A and set the wavefront area threshold T to 1% of the image size (except for O2 where $T = 0.5\%$ in M1 and $T = 0.2\%$ in

```
Input: Image $\hat{I} = (D_I, I)$, adjacency $A$, path-cost function $f_{\max}$, and a labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$, $0 \leq i \leq k$, if $p$ is a seed pixel selected inside object $i > 0$ from $k$ objects, being $i = 0$ reserved for seeds in the background, and $L(p) = -1$ otherwise.
Output: A labeled image $\hat{L} = (D_I, L)$, where $L(p) = i, 0 \leq i \leq k$.
Auxiliary: Priority queue $Q$, variables tmp and ord, and $C$, $R$, $O$ are maps defined in $D_I$ to store cost, root and propagation order of each pixel, respectively.

Set ord $\leftarrow 0$.
For every pixel $p \in D_I$, do
    Set $R(p) \leftarrow p$.
    If $L(p) = -1$, then set $C(p) \leftarrow +\infty$ and $L(p) \leftarrow 0$.
    Else, set $C(p) \leftarrow 0$ and insert $p$ in $Q$.
While $Q \neq \emptyset$, do
    Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
    Set $O(p) \leftarrow$ ord + 1 and ord $\leftarrow$ ord + 1.
    For every $q \in A(p)$, such that $C(q) > C(p)$, do
        Set tmp $\leftarrow \max\{C(p), \delta(p,q)\}$.
        If tmp $< C(q)$, then
            If $C(q) \neq +\infty$, then remove $q$ from $Q$.
            Set $C(q) \leftarrow$ tmp, $R(q) \leftarrow R(p)$, and insert $q$ in $Q$.
While the user is not satisfied.
    The user can select a pixel $q$ on the image.
    For every pixel $p \in D_I$, do
        If $O(p) \leq O(q)$ and $R(p) = R(q)$, then set $L(p) \leftarrow L(R(p))$.
```

Algorithm 4: Multiple object definition with seed competition.

Table 2: The dissimilarity functions used for each combination of object and method.

Object	M1	M2	M3
O1	δ_3	δ_2	δ_2
O2	δ_3	δ_4	δ_2
O3	δ_3	δ_3	δ_3
O4	δ_3	δ_4	δ_2
O5	δ_3	δ_4	δ_4
O6	δ_3	δ_3	δ_2
O7	δ_3	δ_3	δ_3

M2). Since objects from O1 to O6 do not have holes, we set the area closing threshold to some arbitrary high value (e.g., 500 pixels). The only exception was O7, whose area closing threshold could not be higher than 3 pixels due to its holes. In function δ_2, we used the magnitude of the Sobel's gradient. The value of σ was 20 for all cases involving δ_3 and δ_4. Note also that δ_3 has been chosen in some situations involving seed competition, despite $f_{\max}$ is not smooth.

Each object was represented by a set of l binary slices $\hat{L}_i = (D_I, L_i)$, $i = 1, 2, \ldots, l$, where $L_i(p) = 1$ for object pixels and 0 otherwise. Let $\hat{L}_i$ and $\hat{L}'_i$ be the binary images resulting from the segmentation of a same object slice using different methods. The similarity between these results was measured by

$$1.0 - \frac{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) \oplus L'_i(p)}{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) + \sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L'_i(p)}, \qquad (8)$$

where $\oplus$ is the "exclusive or" operation (i.e., $L_i(p) \oplus L'_i(p) = 1$, if $L_i(p) \neq L'_i(p)$, and 0 otherwise). Given that we have four different users using three distinct methods, we may assume that similarity values around 0.90 represent good agreement in delineation (Figure 9).

The number of user interactions in M3 is the total number of seeds selected inside (NIS—*number of internal seeds*) and outside (NES—*number of external seeds*) the object. In M1, the amount of user interaction is represented by the total number of *interactive* κ_s *detections* (IKD) and NIS. The *automatic* κ_s *detections* (AKD) are chosen as much as possible in order to reduce the number of user interactions. In M2, the number of user interactions is computed as in M3, but the number of seeds is expected to be much less due to automatic κ_s detection.

Instead of quantifying the number of user interactions for a fixed value of κ, we decided to quantify the number of different κ_s values found in M1 for cases of multiple seeds. The *percentages of different* κ_s *values* (PDK) are presented in Table 3, together with the average number of interactions and similarity values among all users. Note that O3 was detected with a same value of κ, but the other objects required from 6.8% to 92.4% of different κ_s values. O5 did not count because it was segmented with only one seed per slice. Therefore, AFC would work only for O3 and O5. On average, M3 and M1 required 2.8 and 1.1 more user interactions than M2, respectively. The advantages of M1 and M2 over M3 increase in more complex situations, such as in the delineation of O7, where M3 required 6 times more user interactions than M1 and M2. Although the

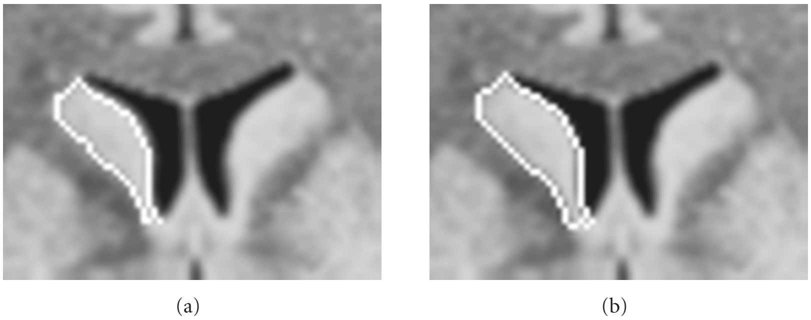


Figure 9: (a) A segmentation of the left caudate nucleus. (b) The result of dilating the binary image with a circular structuring element of radius 1. The similarity value between these two masks is 0.87. The differences in the experiments using distinct methods on this object (O2) are less significant than this small dilation.

Table 3: The percentages of different κ_s values (PDK) in M1, the average numbers of user interactions for each object and method, and the average similarity values between different methods for a same object.

	M1	PDK	M2	M3	M1, M2	M1, M3	M2,M3
O1	35.0	83.4%	29.5	77.6	0.965	0.969	0.962
O2	38.5	92.4%	29.3	38.8	0.904	0.890	0.915
O3	31.8	0.0%	31.3	61.3	0.992	0.932	0.935
O4	29.7	57.4%	27.5	46.8	0.922	0.914	0.918
O5	15.0	—	15.0	61.0	0.973	0.954	0.946
O6	26.3	27.1%	26.3	37.8	0.992	0.982	0.981
O7	47.5	6.8%	46.3	284.8	0.973	0.931	0.930

Table 4: Average numbers of internal seeds (NIS), interactive κ_s detections (IKD), external seeds (NES), and automatic κ_s detections (AKD).

	M1			M2			M3	
	NIS	IKD	AKD	NIS	NES	AKD	NIS	NES
O1	30.7	4.3	26.4	18.0	11.5	13.0	26.8	50.8
O2	30.0	8.5	21.5	25.3	4.0	24.3	18.8	20.0
O3	30.0	1.8	28.2	30.3	1.0	30.3	30.3	31.0
O4	24.5	5.2	19.3	22.3	5.2	19.8	22.3	24.5
O5	15.0	0.0	15.0	15.0	0.0	15.0	44.0	17.0
O6	26.3	0.0	26.3	26.3	0.0	15.5	22.8	15.0
O7	47.5	0.0	47.5	46.0	0.3	46.0	66.0	218.8

performances of M1 and M2 have been equivalent, M2 is preferable because its extension to 3D does not suffer from interactive κ_s indication and it can provide simultaneous segmentation of multiple objects.

Table 4 shows in detail the average values of NIS, NES, IKD, and AKD for each object and method. Note that, AKD varied from 59% to 100% of NIS, being on average 90% of NIS in M1 and 88% of NIS in M2. This demonstrates the effectiveness of the proposed approach for automatic κ_s detection and explains the reduction of user interactions in M1 and M2 with respect to M3. Note also that the number of external seeds was considerably reduced in comparison to M3. This is an important result for future automation, because seed competition is sensitive to the location of the external seeds due to the heterogeneity of the background.

6. CONCLUSIONS

We have presented four IFT-based algorithms for object delineation based on κ-connected components with and without seed competition. They differ from the previous approaches in the following aspects: computation of different values of κ for each seed, effective automatic κ_s detection, and user friendly κ_s computation, where the user moves the cursor of the mouse to indicate the pixel whose propagation order defines the object. The use of propagation order rather than the pixel cost is important to create smoother transitions between possible objects, facilitating the user's work. The new methods have considerably reduced the number of user interactions in medical image segmentation with respect to the previous approaches. We believe that these results are extensive to other image types by suitable choice of pre-processing and dissimilarity function.

The interactive κ_s detection counted with real time response for every position of the cursor, but this may not be feasible in 3D segmentation involving several slices. In this sense, the algorithms based on interactive κ_s detection are more adequate for 2D/3D segmentation in a slice by slice fashion, where seeds may be automatically propagated along the slices. In such a case, the interactive κ_s detection can be used to correct segmentation when the automatic detection of κ_s fails.

Seed competition with automatic κ_s detection (Algorithm 2) seems to be the most promising approach. We are currently investigating two approaches for 3D segmentation of medical images: (i) automatic segmentation with only internal seeds and automatic κ_s detection, and (ii) interactive segmentation with automatic κ_s detection, where the user can add/remove internal and external seeds, and subsequent IFTs are executed in a differential way [27].

ACKNOWLEDGMENTS

The authors thank FAPESP (Procs. 05/59808-0, 05/58103-3, 05/56578-4, and 03/13424-1), CNPq (Procs. 302617/2007-8 and 472402/2007-2), and CAPES for the financial support. Furthermore, the authors thank the reviewers and editors for their contributions.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[3] T. Cootes, G. Edwards, and C. J. Taylor, “Active appearance models,” in *Proceedings of the 5th European Conference on Computer Vision (ECCV ’98)*, vol. 2, pp. 484–498, Freiburg, Germany, June 1998.

[4] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’01)*, vol. 1, pp. 511–518, Kauai, Hawaii, USA, December 2001.

[5] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[6] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary &amp; region segmentation of objects in N-D images,” in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV ’01)*, vol. 1, pp. 105–112, Vancouver, Canada, July 2001.

[7] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[8] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[9] S. Wang and J. M. Siskind, “Image segmentation with minimum mean cut,” in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV ’01)*, vol. 1, pp. 517–524, Vancouver, Canada, July 2001.

[10] J. K. Udupa and P. K. Saha, “Fuzzy connectedness and image segmentation,” *Proceedings of the IEEE*, vol. 91, no. 10, pp. 1649–1669, 2003.

[11] P. K. Saha and J. K. Udupa, “Relative fuzzy connectedness among multiple objects: theory, algorithms, and applications in image segmentation,” *Computer Vision and Image Understanding*, vol. 82, no. 1, pp. 42–56, 2001.

[12] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[13] S. Beucher and F. Meyer, “The morphological approach to segmentation: the watershed transformation,” in *Mathematical Morphology in Image Processing*, chapter 12, pp. 433–481, Marcel Dekker, New York, NY, USA, 1993.

[14] T. Lei, J. K. Udupa, P. K. Saha, and D. Odhner, “Artery-vein separation via MRA-an image processing approach,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 8, pp. 689–703, 2001.

[15] G. Moonis, J. Liu, J. K. Udupa, and D. B. Hackney, “Estimation of tumor volume with fuzzy-connectedness segmentation of MR images,” *American Journal of Neuroradiology*, vol. 23, no. 3, pp. 356–363, 2002.

[16] H. T. Nguyen, M. Worring, and R. van den Boomgaard, “Watersnakes: energy-driven watershed segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 330–342, 2003.

[17] V. Grau, A. U. J. Mewes, M. Alcañiz, R. Kikinis, and S. K. Warfield, “Improved watershed transform for medical image segmentation using prior information,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, 2004.

[18] Y.-P. Tsai, C.-C. Lai, Y.-P. Hung, and Z.-C. Shih, “A Bayesian approach to video object segmentation via merging 3-D watershed volumes,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 175–180, 2005.

[19] G. T. Herman and B. M. Carvalho, “Multiseeded segmentation using fuzzy connectedness,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 460–474, 2001.

[20] J. K. Udupa, P. K. Saha, and R. A. Lotufo, “Relative fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1485–1500, 2002.

[21] A. X. Falcão, J. Stolfi, and R. A. Lotufo, “The image foresting transform: theory, algorithms, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[22] R. Audigier and R. A. Lotufo, “Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches,” in *Proceedings of the 20th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI ’07)*, pp. 61–70, Minas Gerais, Brazil, October 2007.

[23] J. K. Udupa and S. Samarasekera, “Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation,” *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 246–261, 1996.

[24] R. A. Lotufo and A. X. Falcão, “The ordered queue and the optimality of the watershed approaches,” in *Mathematical Morphology and Its Applications to Image and Signal Processing*,

vol. 18, pp. 341–350, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[25] A. X. Falcão, L. F. Costa, and B. S. da Cunha, "Multiscale skeletons by image foresting transform and its application to neuromorphometry," *Pattern Recognition*, vol. 35, no. 7, pp. 1571–1582, 2002.

[26] R. S. Torres, A. X. Falcão, and L. F. Costa, "A graph-based approach for multiscale shape analysis," *Pattern Recognition*, vol. 37, no. 6, pp. 1163–1174, 2004.

[27] A. X. Falcão and F. P. G. Bergo, "Interactive volume segmentation with differential image foresting transforms," *IEEE Transactions on Medical Imaging*, vol. 23, no. 9, pp. 1100–1108, 2004.

[28] A. X. Falcão, F. P. G. Bergo, and P. A. V. Miranda, "Image segmentation by tree pruning," in *Proceedings of the 17th Brazilian Symposium of Computer Graphic and Image Processing (SIBGRAPI '04)*, pp. 65–71, Curitiba, Brazil, October 2004.

[29] P. K. Saha and J. K. Udupa, "Fuzzy connected object delineation: axiomatic path strength definition and the case of multiple seeds," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 275–295, 2001.

[30] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo, "User-steered image segmentation paradigms: live wire and live lane," *Graphical Models and Image Processing*, vol. 60, no. 4, pp. 233–260, 1998.

[31] P. K. Saha, J. K. Udupa, and D. Odhner, "Scale-based fuzzy connected image segmentation: theory, algorithms, and validation," *Computer Vision and Image Understanding*, vol. 77, no. 2, pp. 145–174, 2000.

[32] P. K. Saha and J. K. Udupa, "Tensor scale-based fuzzy connectedness image segmentation," in *Medical Imaging 2003: Image Processing*, vol. 5032 of *Proceedings of SPIE*, pp. 1580–1590, San Diego, Calif, USA, February 2003.

[33] C. Allène, J. Y. Audibert, M. Couprie, J. Cousty, and R. Keriven, "Some links between min-cuts, optimal spanning forests and watersheds," in *Proceedings of the 8th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing (ISMM '07)*, pp. 253–264, Rio de Janeiro, Brazil, October 2007.

[34] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[35] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: live wire on the fly," *IEEE Transactions on Medical Imaging*, vol. 19, no. 1, pp. 55–62, 2000.

[36] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl, "Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest," *Computer Graphics Forum*, vol. 20, no. 3, pp. C26–C35, 2001.

[37] A. Meijster and M. H. F. Wilkinson, "A comparison of algorithms for connected set openings and closings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 484–494, 2002.

[38] L. Vincent, "Morphological area opening and closings for greyscale images," in *Proceeding of NATO Shape in Picture Workshop*, pp. 197–208, Springer, Driebergen, The Netherlands, September 1992.