Object definition by κ-connected components

A.X. Falcão *P.A.V. Miranda* *A. Rocha*
F.P.G. Bergo

Technical Report - IC-05-18 - Relatório Técnico

September - 2005 - Setembro

Object definition by κ-connected components

Alexandre X. Falcão[1] Paulo A.V. Miranda[1] Anderson Rocha[1] Felipe P.G. Bergo[1]

[1]Instituto de Computação, Universidade Estadual de Campinas
Av. Albert Einstein, 1251, CEP 13084-970, Campinas, SP, Brasil

Abstract

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 definition tasks. A set of pixels is said 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

Many image segmentation methods have been proposed, but their success usually depends on user intervention, being automatic segmentation only verified for some specific situations. In view of that, it is important to develop interactive methods, which minimize the user's time and involvement in the segmentation process, such that their automation becomes feasible under certain conditions. For example, we are interested in reducing the user intervention to simple selection of a few pixels in the image.

Fuzzy connectedness/watersheds are image segmentation approaches based on seed pixels, still in progress, and successfully used in many applications [1–6]. In *relative fuzzy connectedness* [7–9], as well as in watershed-based segmentations [10, 11], 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* [12], 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) [13]— 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 [14–18].

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 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 detect 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 determining them.

Section 2 describes some definitions related to the IFT, making them more specific for region-based image segmentation. For sake of simplicity, we will describe the methods for gray-scale and two-dimensional images, but they are extensive to multi-parametric and multi-dimensional 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

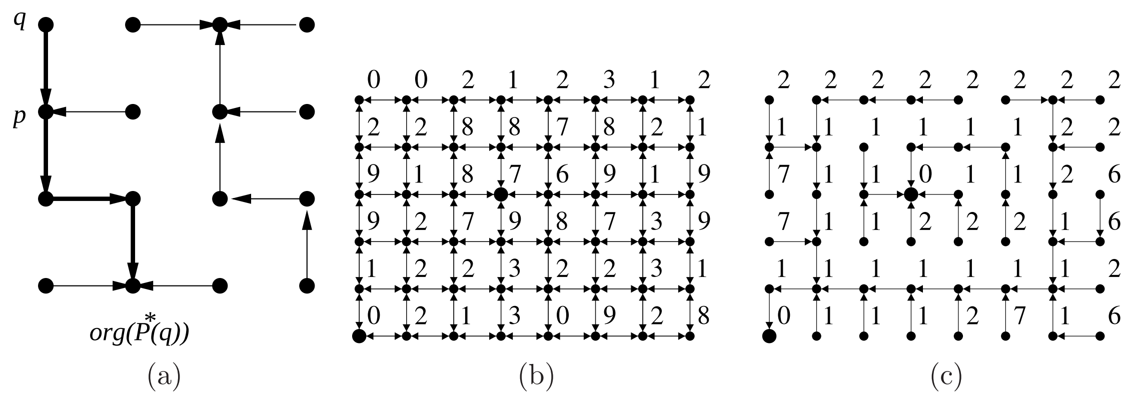


Figure 1: (a) The main elements of a spanning forest with two roots, where the thicker path indicates $P^*(q)$. (b) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dots indicate two seeds. One is inside the brighter rectangle and one is in the darker background 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.

$q \in A(p)$ 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 $(p,q) \in D_I \times D_I \setminus \{(0,0)\}$ such that $d(p,q) \leq \rho$, 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 $1 \leq i \leq n-1$. The path is *trivial* if $n = 1$. Let $org(\pi) = p_1$ and $dst(\pi) = p_n$ be the origin and destination of a path π. If π and τ are paths such that $dst(\pi) = org(\tau) = p$, 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 $P(q) = nil$, or $P^*(p) \cdot \langle p, q \rangle$ if $P(q) = p \neq nil$ (see Figure 1a).

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 $f(\pi) \leq f(\tau)$ for any other path τ with $dst(\tau) = dst(\pi)$, irrespective of 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) $f(\tau) \leq f(\pi)$,

(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 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 given seed set $S \subset D_I$, the concept of *strength of connectedness* [12, 19] 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}:

$$\begin{aligned} f_{\max}(\langle q\rangle) &= \begin{cases} 0, & \text{if } q \in S, \\ +\infty, & \text{otherwise.} \end{cases} \\ f_{\max}(\pi \cdot \langle p, q\rangle) &= \max\{f_{\max}(\pi), \delta(p, q)\}, \end{aligned} \tag{1}$$

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

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 detect 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:

$$\begin{aligned}
\delta_1(p,q) &= K\left(1-\exp\left(-\frac{1}{2\sigma^2}\left(I(p)-I(q)\right)^2\right)\right) && (2)\\
\delta_2(p,q) &= G(q) && (3)\\
\delta_3(p,q) &= K\left(1-\exp\left(-\frac{1}{2\sigma^2}\left(\frac{I(p)+I(q)}{2}-I(s)\right)^2\right)\right) && (4)\\
\delta_4(p,q) &= \min_{\forall s\in S}\{\delta_3(p,q)\} && (5)\\
\delta_5(p,q) &= a\delta_1(p,q)+b\delta_3(p,q) && (6)\\
\delta_6(p,q) &= \begin{cases} \delta_3(p,q)(1+\vec{g}(p,q)\cdot\vec{\eta}(p,q)), & \text{if } E_r(p,q)>D_r(p,q),\\ K, & \text{otherwise}\end{cases} && (7)
\end{aligned}$$

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 $s = org(P^*(p))$ 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.

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 $(p,q) \in A$. In the case of seed competition, the IFT with these functions becomes a watershed transform [14]. 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 [13] (i.e. the IFT results a spanning forest, but it may be non-optimal). This problem was the main motivation for δ_4 [7]. 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 reduces this problem, and in the case of seed competition, one can also replace δ_3 by δ_4 in Equation 6. Function δ_6 exploits situations where the intensities of the background on one side of the object region are higher than the object intensities and the other way around is true for the other side of the object (Figure 2). Other interesting ideas of dissimilarity functions for $f_{\max}$ are presented in [7, 8, 12, 20, 21].

The basic differences between the formulations proposed in [8] and [7, 9] 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 non-smooth 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.

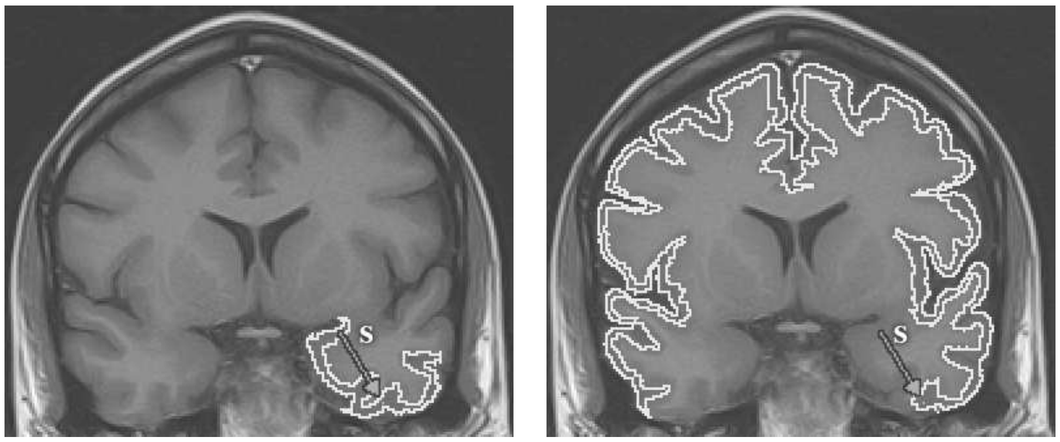


Figure 2: An MR-T1 image of the brain with one seed s inside the cortex. (a-b) The maximum influence zones of s within the cortex for $f_{\max}$ with δ_3 and with δ_6, respectively. The assymetry of δ_6 favors segmentation in anti-clockwise orientation, increasing the influence zone of s.

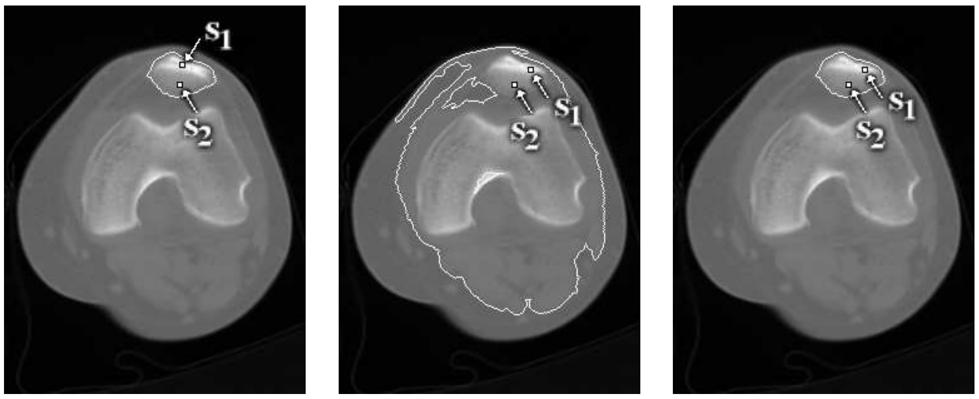
3.1 Object definition without seed competition

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 [12], 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 Equations 2- 7.

The algorithm described in [12] assumes that the object can be defined by a single value of κ for all seeds in S. Figure 3a illustrates an example where this assumption works. However, a simple change in the position of a seed can fail segmentation (Figure 3b), because the influence zone of each seed inside the object is actually limited by a distinct value of cost κ (Figure 3c). 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 [22] and minimum-cost path tree for f_{max} and symmetric κ-connectivity relations.

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 4a and 4b). If we remove the orientation of the arcs in Figure 4b, every pair of pixels in P is connected by a path which is also optimum according to $f_{\max}$ (Figure 4c). 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



(a) (b) (c)

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.

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 4a. Figure 4c 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 4d).

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.

3.2 Object definition with seed competition

In [7, 8], 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 4c is obtained by changing the position of s_1 to any pixel in the background and selecting s_2 as shown in Figure 4d. 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

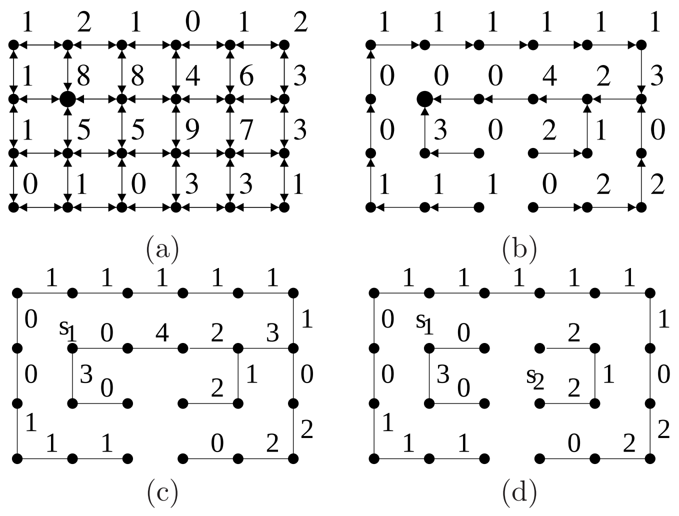


Figure 4: (a) An image graph with 4-adjacency, where the integers are the image values $I(p)$ and the bigger dot is an arbitrary pixel. The object of interest is the brighter rectangle in the center. (b) A minimum-spanning tree computed from the arbitrary pixel, where the integers for each pixel q are 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 $\kappa_2 = 2$.

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 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 do 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 5a shows an example where many seeds have to be carefully selected in the background to detect the object. The segmentation fails when some of these seeds are removed (Figure 5b), but it works when we limit the extent of the internal seed to some value of κ (Figure 5c).

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 [23] to compute three attributes for each pixel $p \in D_I$ [13]: its predecessor $P(p)$ in the optimum path, the cost $C(p)$ of that path,

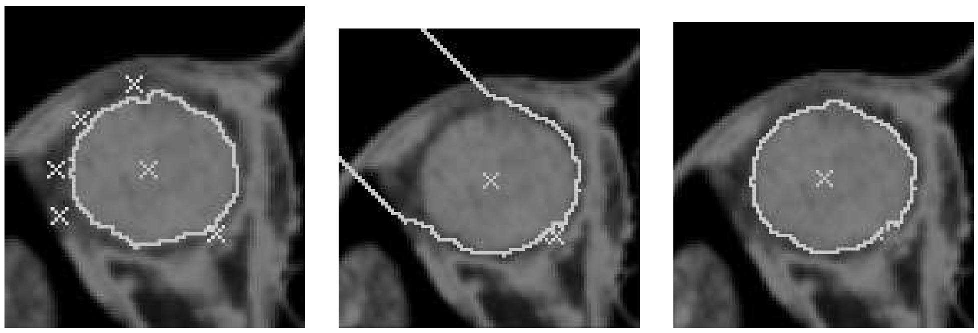


Figure 5: A CT image of the orbital region where the eye ball is obtained by seed competition. (a) One internal seed and many external seeds are required for segmentation, using $f_{\max}$ with δ_4. (b) The segmentation fails when some of the external seeds are removed. (c) A value of κ is used to limit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

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 propagates wavefronts W_{cst} of same cost cst around each seed, following the order of the costs $cst = 0, 1, \ldots, K$. This 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 $C(p) = cst$, $0 \leq cst \leq K$. 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 $cst = 0, 1, \ldots, \kappa_s$. 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 following observation.

The effectiveness of segmentation using $f_{\max}$ depends on assigning lower values of $\delta(p, q)$ to arcs inside (and outside) the objects and higher values to arcs across their boundaries. As consequence, the wavefronts usually present a considerable increase in *number of pixels* when they cross the object boundaries (Figures 6a and 6b). That is, many pixels in the background are reached by optimum paths whose cost is the value $\delta(p, q)$ of some arc (p, q) across the boundary.

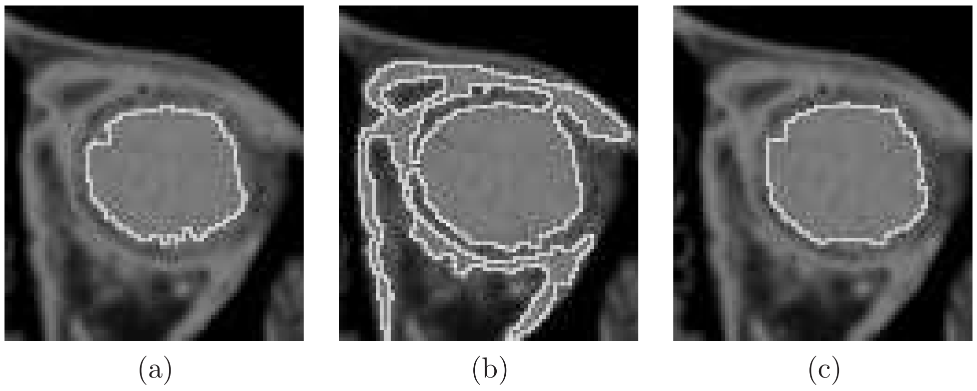


Figure 6: A CT image of the orbital region with one seed inside the eye ball. (a) A wavefront of cost κ which represents the maximum extent of this seed inside the eye ball. (b) The wavefront of cost $\kappa+1$ shows a considerable augment in size when it invades the background. (c) The pixel propagation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

We substitute the choice of one value κ_s for each seed $s \in S$ by a threshold T (i.e. a percentage of the total number of pixels), which limits the maximum size of their wavefronts. The region growing process of a seed s must stop when the size of its wavefront of cost cst is greater than T, and the value of κ_s is determined as $\max\{cst-1, 0\}$. The algorithms are presented for single object detection without seed competition and multiple object definition with seed competition.

Single object definition without seed competition:

Input: Image $\hat{I} = (D_I, I)$, adjacency A, internal seeds S, 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, κ, cst and $size$, and cost map C defined in D_I.

```
1.  For every pixel $p \in D_I$, set $L(p) \leftarrow 0$.
2.  While $S \neq \emptyset$, do
3.      For every pixel $p \in D_I$, set $C(p) \leftarrow +\infty$.
4.      Remove a seed $s$ from $S$.
5.      Set $C(s) \leftarrow 0$, $size \leftarrow 0$, $cst \leftarrow 0$, $\kappa \leftarrow +\infty$, and insert $s$ in $Q$.
6.      While $Q \neq \emptyset$ and $\kappa = +\infty$, do
7.          Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
8.          For every $q \in A(p)$, such that $C(q) > C(p)$, do
9.              Set $tmp \leftarrow \max\{C(p), \delta(p,q)\}$.
```

```
10.                 If $tmp < C(q)$, then
11.                     If $C(q) \neq +\infty$, then remove $q$ from $Q$.
12.                     Set $C(q) \leftarrow tmp$ and insert $q$ in $Q$.
13.             If $C(p) \neq cst$, then set $size \leftarrow 1$ and $cst \leftarrow C(p)$.
14.             Else, set $size \leftarrow size + 1$.
15.             If $size > T$ then set $\kappa \leftarrow \max\{cst - 1, 0\}$.
16.     For every pixel $p \in D_I$, do
17.             If $C(p) \leq \kappa$, then set $L(p) \leftarrow 1$.
18.     Remove any remaining pixels from $Q$.
```

Multiple object definition with 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 $i > 0$ 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, variable *tmp*, and C, R, κ, *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.

```
1.  For every pixel $p \in D_I$, do
2.      Set $R(p) \leftarrow p$, $size(p) \leftarrow 0$, $cst(p) \leftarrow 0$, and $\kappa(p) \leftarrow +\infty$.
3.      If $L(p) = -1$, then set $C(p) \leftarrow +\infty$ and $L(p) \leftarrow 0$.
4.      Else, set $C(p) \leftarrow 0$ and insert $p$ in $Q$.
5. While $Q \neq \emptyset$, do
6.      Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
7.      If $\kappa(R(p)) = +\infty$ and $L(R(p)) \neq 0$, then
8.          If $C(p) \neq cst(R(p))$, then set $size(R(p)) \leftarrow 1$ and $cst(R(p)) \leftarrow C(p)$.
9.          Else, set $size(R(p)) \leftarrow size(R(p)) + 1$.
10.         If $size(R(p)) > T$, then set $\kappa(R(p)) \leftarrow \max\{cst(R(p)) - 1, 0\}$.
11.     If $C(p) \leq \kappa(R(p))$, then
12.         For every $q \in A(p)$, such that $C(q) > C(p)$, do
13.             Set $tmp \leftarrow \max\{C(p), \delta(p, q)\}$.
14.             If $tmp < C(q)$, then
15.                 If $C(q) \neq +\infty$, then remove $q$ from $Q$.
16.                 Set $C(q) \leftarrow tmp$, $R(q) \leftarrow R(p)$, and insert $q$ in $Q$.
17. For every pixel $p \in D_I$, do
18.     If $C(p) \leq \kappa(R(p))$, then set $L(p) \leftarrow L(R(p))$.
```

The priority queue Q can be implemented as described in [24, 25], such that each instance of the IFT will run in time proportional to the number $|D_I|$ of pixels. Note that, the first algorithm above 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.

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.

One drawback of the method above is the abrupt size variations of the wavefronts (Figures 6a and 6b) 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 6c) as compared to the first idea. The algorithms are presented below.

Single object definition without seed competition:

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.

```
1. For every pixel $p \in D_I$, set $L(p) \leftarrow 0$.
2. While $S \neq \emptyset$, do
3.     For every pixel $p \in D_I$, set $C(p) \leftarrow +\infty$.
4.     Remove a seed $s$ from $S$.
5.     Set $C(s) \leftarrow 0$, $ord \leftarrow 0$, and insert $s$ in $Q$.
```

```
6.  While $Q \neq \emptyset$, do
7.      Remove a pixel $p$ from $Q$ such that $C(p)$ is minimum.
8.      Set $O(p) \leftarrow ord + 1$ and $ord \leftarrow ord + 1$.
9.      For every $q \in A(p)$, such that $C(q) > C(p)$, do
10.         Set $tmp \leftarrow \max\{C(p), \delta(p,q)\}$.
11.         If $tmp < C(q)$, then
12.             If $C(q) \neq +\infty$, then remove $q$ from $Q$.
13.             Set $C(q) \leftarrow tmp$ and insert $q$ in $Q$.
14. The user selects a pixel $q$ on the image.
15. For every pixel $p \in D_I$, do
16.     If $O(p) \leq O(q)$, then set $L(p) \leftarrow 1$.
```

Multiple object definition with seed competition:

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 $i > 0$ 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, 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.

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

One advantage of the presented algorithms as compared to classical segmentation methods based on seed competition occurs when the object contains several background parts

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

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

(holes) inside it. In this case, the use of κ_s usually eliminates the need for at least one background seed at each hole. One 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 post-processed, such that holes with area below a threshold are closed [26, 27]. 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. These algorithms are compared to the classical segmentation approach based on seed competition in the next section.

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 7). Each object consists of some slices that represent different degrees of challenge for segmentation. The original images have been pre-processed to increase the similarities between pixels inside the objects and the contrast between object and background. Each of four users have performed segmentation over the 100 images using each of three methods:

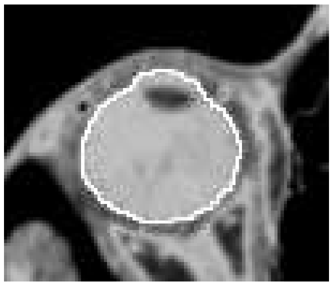
M1. Object detection without seed competition and automatic/interactive computation of κ_s;

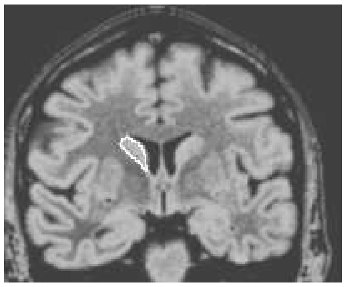
M2. Object detection with seed competition and automatic κ_s computation;

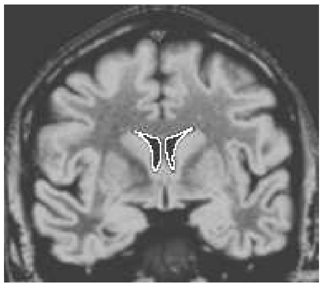
M3. Object detection with seed competition and without κ_s computation.

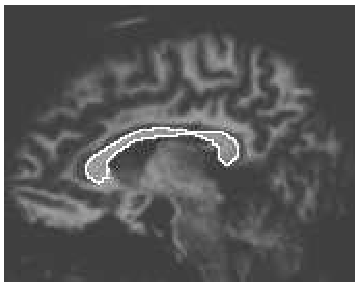
The experiments aimed to compare these methods with respect to the number of user interactions required to complete segmentation. Although interactive κ_s detection may reduce the number of external seeds in M2, we decided to avoid it in order to evaluate the combination of seed competition and automatic κ_s detection with respect to M3.

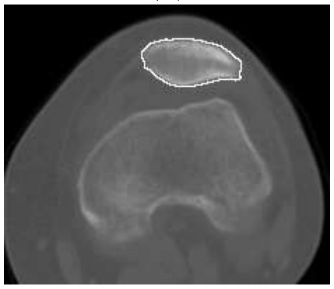
In order to show the robustness of these approaches, we have chosen the best dissimilarity function for each situation and fixed the parameters of segmentation. We used the 8-neighborhood as adjacency relation A. The size threshold T was set to 1% of the image

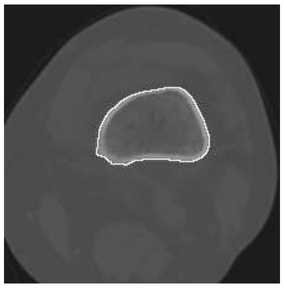

(a)


(b)


(c)


(d)


(e)


(f)

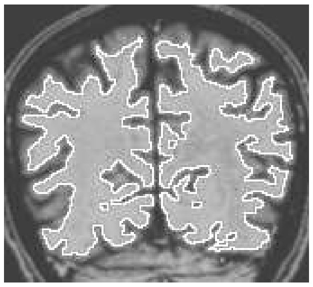

(g)

Figure 7: (a)-(g) Results of slice segmentation of the objects from 1 to 7, respectively, overlaid with the pre-processed images.

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

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

size, except for O2 where $T = 0.5\%$ in M1 and $T = 0.2\%$ in 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 threshold could not be higher than 3 pixels due to its holes. Table 2 shows the most suitable dissimilarity function found for each pair object and method. 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.

In Medical Image Analysis, it is common to use as ground truth the results of manual segmentation performed by an user expert. This methodology is questionable, because the experts usually make mistakes when they delineate the same object twice. In most cases, the results look like the same but there are small differences along the object boundaries. The small differences, however, seem to be acceptable in many applications. This similarity measure was defined as follows.

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:

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

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). In the case of manual segmentation by an user expert, it has been shown that the similarity values are around 0.96 [28]. Since none of the users is an expert, we required from them acceptable results from the expert's point of view with similarity values around 0.90 between distinct segmentations of a same object, using different methods.

Method M3 represents the classical approach based on relative fuzzy connectedness/watershed transform. 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. Methods M1 and M2 are the proposed variants of absolute fuzzy connectedness and relative fuzzy connectedness, respectively. In M1, the amount of user interaction is represented by the total number of *interactive κ_s detections* (IKD) and

	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 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.

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.

In Section 1, we showed that the use of a unique κ for all seeds may increase the number of seeds (user interactions). 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. 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 detection of O7 where M3 required 6 times more user interactions than M1 and M2. Although the performances of M1 and M2 have been equivalent, M2 is preferable because it is more general than M1 and M3.

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 detection 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 au-

	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

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

tomatic 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.

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 [17].

Acknowledgments

The authors thank CNPq (Proc. 302427/04-0), FAPESP (Proc. 03/09793-1), and CAPES for the financial support.

References

[1] T. Lei, J. K. Udupa, P. K. Saha, and D. Odhner. Artery-vein separation via MRA - An image processing approach. *IEEE Trans. on Medical Imaging*, 20(8), 2001.

[2] Gul Moonis, Jianguo Liu, Jayaram K. Udupa, and David B. Hackney. Estimation of tumor volume with fuzzy-connectedness segmentation of MR images. *American Journal of Neuroradiology*, 23:356–363, Mar 2002.

[3] J.K. Udupa and P.K. Saha. Fuzzy connectedness and image segmentation. *Proceedings of the IEEE*, 91(10):1649–1669, Oct 2003.

[4] H.T. Nguyen, M. Worring, and R. van den Boomgaard. Watersnakes: energy-driven watershed segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(3):330–342, Mar 2003.

[5] V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, and S.K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE Trans. on Medical Imaging*, 23(4):447–458, Apr 2004.

[6] 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 Trans. on Circuits and Systems for Video Technology*, 15(1):175–180, Jan 2005.

[7] 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*, 82:42–56, 2001.

[8] G.T. Herman and B.M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(5):460–474, 2001.

[9] J.K. Udupa, P.K. Saha, and R.A. Lotufo. Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1485–1500, 2002.

[10] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6), Jun 1991.

[11] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, 1993.

[12] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.

[13] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[14] 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*, volume 18, pages 341–350. Kluwer, Jun 2000.

[15] A.X. Falcão, L.F. Costa, and B.S. da Cunha. Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, 2002.

[16] R.S. Torres, A.X. Falcão, and L.F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004.

[17] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.

[18] A.X. Falcão, F.P.G. Bergo, and P.A.V. Miranda. Image segmentation by tree pruning. In *Proc. of the XVII Brazillian Symposium on Computer Graphics and Image Processing*, pages 65–71. IEEE, Oct 2004.

[19] 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*, 83:275–295, 2001.

[20] P.K. Saha, J.K. Udupa, and D. Odhner. Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation. *Computer Vision and Image Understanding*, 77(2):145–174, 2000.

[21] P.K. Saha and J.K. Udupa. Tensor scale-based fuzzy connectedness image segmentation. In *Proc. of SPIE on Medical Imaging*, volume 5032, pages 1580–1590, Feb 2003.

[22] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT, 1990.

[23] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[24] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, 2000.

[25] 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 (EUROGRAPHICS)*, 20(3):(C) 26–35, 2001.

[26] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):484–494, Apr 2002.

[27] L. Vincent. Morphological area opening and closings for greyscale images. In *Shape in Picture'92 - NATO Workshop*. Springer, Sep 1992.

[28] 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*, 60(4):233–260, Jul 1998.