Object detection by κ-connected seed competition

Alexandre X. Falcão, Paulo A.V. Miranda, Anderson Rocha and Felipe P.G. Bergo
Institute of Computing – State University of Campinas (UNICAMP)
CEP 13084-851, Campinas, SP, Brazil
{afalcao,paulo.miranda,anderson.rocha,felipe.bergo}@ic.unicamp.br

Abstract

The notion of "strength of connectedness" between pixels has been successfully used in image segmentation. We present an extension to these works, which can considerably increase 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. While the previous approaches either assume no competition with a single threshold for all seeds or eliminate the threshold for seed competition, we found that seed competition with different thresholds can reduce the number of seeds and the need for user interaction during segmentation. We also propose automatic and user-friendly interactive methods for determining the thresholds. 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 [11, 14, 15, 9]. These approaches have been used under two region growing paradigms, with and without competition among seeds. In object detection with seed competition [17, 10, 21, 1], 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 object definition without seed competition [20], 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.

We extend these methods using the framework of the *image foresting transform* (IFT) [5]— 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 [12, 4, 19, 2, 3].

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. A set of pixels is said a *κ-connected component* with respect to a seed, when they are reached by optimum paths whose costs are less than or equal to κ. In this sense, when a seed is selected inside an object, its maximal extent is a κ-connected component composed by only internal pixels. In [20], the object is defined without competition, as the union of all κ-connected components (minimum-cost path trees) created from each internal seed, separately (which requires one IFT for each seed). Clearly, the initial appeal for seed competition is the possibility to detect multiple objects with a single IFT and without depending on thresholds: external and internal seeds compete among themselves, partitioning the image into an optimum-path forest, and each object is de-

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)

fined by the union of the optimum-path trees rooted at its internal seeds. However, we found that seed competition with a threshold κ_s for each internal seed s can considerably increase the efficiency of object definition tasks. The method restricts seed competition into regions of pixels that are κ_s-connected to some internal seed s, pixels not reached by any seed are considered background, and external seeds are only needed when the extent of a seed is not contained in the object. Of course, this comes with the problem of finding the values κ_s for each seed s, 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 method and its algorithms with automatic and interactive k_s detection, respectively, are presented in Sections 3 and 4. Section 5 demonstrates the improvements with respect to the previous approaches and Section 6 states conclusions and discusses future work.

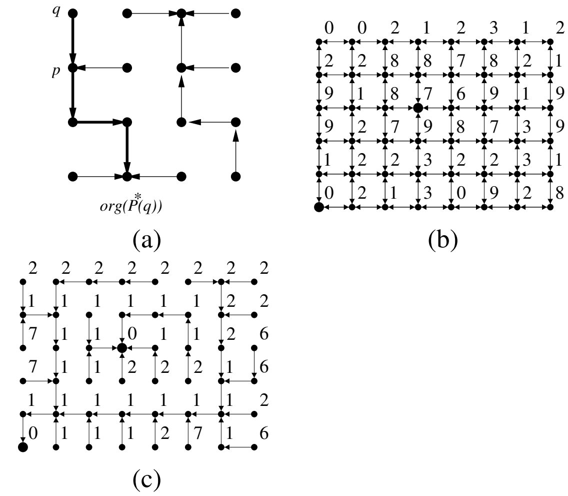

Figure 1. (a) A spanning forest with two roots. The bold path is $P^*(q)$. (b) An image graph with 4-adjacency. The numbers are the image values $I(p)$ and the bigger dots denote two seeds (One inside the brighter rectangle and one in the darker background around it). The background also has bright pixels. (c) An optimum-path forest for $f_{\max}$, with $\delta(p,q) = |I(q) - I(p)|$. The numbers are the cost values and the rectangle is obtained as a tree rooted at the internal seed.

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 $(p,q) \in D_I \times D_I \setminus \{(0,0)\}$ 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,

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)

(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 a given seed set $S \subset D_I$, the concept of *strength of connectedness* [20, 16, 10] 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}:

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

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. Seed competition with 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:

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

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

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

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

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

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 $a + b = 1$.

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 fact, $f_{\max}$ is smooth whenever $\delta(p, q)$ is fixed for any $(p, q) \in A$. The IFT with these functions becomes a watershed transform [12]. Function δ_3 exploits the dissimilarity between object and pixel intensities, being the object represented by its seed pixels, but $f_{\max}$ may not be smooth for the general case with multiple seeds [5] (i.e. the IFT results a spanning forest, but it may be non-optimal). This problem was the main motivation for δ_4 [17]. However, sometimes δ_3 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 δ_3 can be replaced by δ_4 in Equation 6 to ensure smoothness. Other interesting ideas of dissimilarity functions for $f_{\max}$ are presented in [20, 10, 18, 17].

The basic differences between the formulations proposed in [10] and [17] 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 multiple seeds). The strength of connectedness between image pixels in (i) is a symmetric relation, while it may be asymmetric in (ii).

In [17, 10], 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. 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 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 reduces the number of seeds required to complete segmentation. Figure 2a also illustrates 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

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)
1530-1834/05 $20.00 © 2005 IEEE

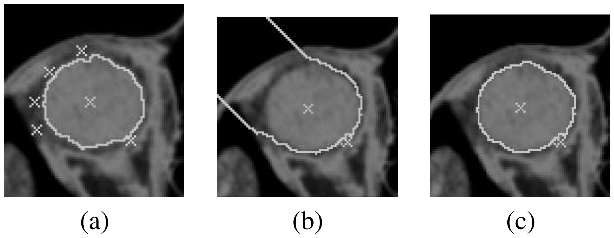

Figure 2. Segmentation by seed competition of the eye ball in a CT image of the eye orbit. (a) One internal seed and many external seeds are required for segmentation, using $f_{\max}$ with δ_4. (b) Segmentation fails when some external seeds are removed. (c) A value of κ limits the influence zone of the internal seed where the seed competition fails.

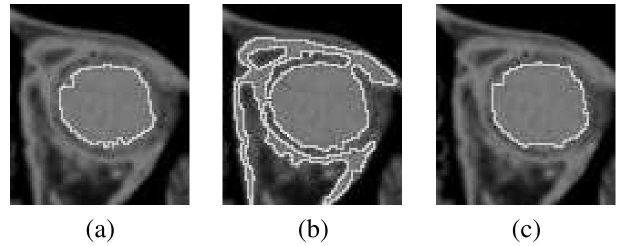

Figure 3. A CT image of the eye orbit with one seed inside the eye ball. (a) A wavefront of cost κ which leads to the maximum extent of this seed inside the eye ball. (b) The wavefront of cost $\kappa + 1$ shows a large area growth when it invades the background. (c) The pixel propagation order provides smoother wavefront transitions for interactive selection of κ.

removed (Figure 2b), but it works when we limit the extent of the internal seed to some value of κ (Figure 2c).

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

4. Algorithms

The IFT computes three attributes for each pixel $p \in D_I$ [5]: 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. 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 3a and 3b). 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.

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 divided by the number of internal seeds), 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 algorithm presented below computes κ_s for multiple object definition with seed competition.

Algorithm 1 OBJECT DEFINITION WITH AUTOMATIC κ_s DETECTION

INPUT: Image $\hat{I} = (D_I, I)$, adjacency A, size threshold T, and a labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$ for seed pixels inside object $0 < i \leq k$, $L(p) = 0$ for seeds in the background, $L(p) = -1$ otherwise.

OUTPUT: A labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$, $0 \leq i \leq k$.

AUXILIARY: Priority queue Q and maps C, R, κ, $size$, and cst defined in D_I to store cost and root of each pixel and threshold, wavefront size, and wavefront cost of each seed, respectively.

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)
1530-1834/05 $20.00 © 2005 IEEE

```
For every pixel p ∈ D_I, do
    R(p) ← p, size(p) ← 0, cst(p) ← 0, κ(p) ← +∞.
    If L(p) = −1, then set C(p) ← +∞ and L(p) ← 0.
    Else, set C(p) ← 0 and insert p in Q.
While Q ≠ ∅, do
    Remove a pixel p from Q such that C(p) is minimum.
    If κ(R(p)) = +∞ and L(R(p)) ≠ 0, then
        If C(p) ≠ cst(R(p)), then
            Set size(R(p)) ← 1 and cst(R(p)) ← C(p).
        Else, set size(R(p)) ← size(R(p)) + 1.
        If size(R(p)) > T, then
            Set κ(R(p)) ← max{cst(R(p)) − 1, 0}.
    If C(p) ≤ κ(R(p)), then
        For every q ∈ A(p), such that C(q) > C(p), do
            Set tmp ← max{C(p), δ(p, q)}.
            If tmp < C(q), then
                If C(q) ≠ +∞, then remove q from Q.
                Set C(q) ← tmp, R(q) ← R(p).
                Insert q in Q.
For every pixel p ∈ D_I, do
    If C(p) ≤ κ(R(p)), then set L(p) ← L(R(p)).
```

The priority queue Q can be implemented as described in [6, 8], such that each instance of the IFT will run in time proportional to the number $|D_I|$ of pixels. 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 initially compute the optimum cost $C(p)$ and root $R(p) \in S$ for 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 $s \in S$.

One drawback of the method above is the abrupt size variations of the wavefronts (Figures 3a and 3b) which makes the selection of pixel q sometimes difficult. We circumvent this 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 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 3c) as compared to the first idea. The algorithm is presented below.

Algorithm 2 OBJECT DEFINITION WITH INTERACTIVE κ_s DETECTION

INPUT:	Image $\hat{I} = (D_I, I)$, adjacency A, and a labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$ for seed pixels inside object $0 < i \leq k$, $L(p) = 0$ for seeds in the background, $L(p) = -1$ otherwise.
OUTPUT:	A labeled image $\hat{L} = (D_I, L)$, where $L(p) = i$, $0 \leq i \leq k$.
AUXILIARY:	Priority queue Q and maps C, R, O defined in D_I to store cost, root and propagation order of each pixel, respectively.

```
1.  Set ord ← 0.
2.  For every pixel p ∈ D_I, do
3.      Set R(p) ← p.
4.      If L(p) = −1, then set C(p) ← +∞ and L(p) ← 0.
5.      Else, set C(p) ← 0 and insert p in Q.
6.  While Q ≠ ∅, do
7.      Remove a pixel p from Q such that C(p) is minimum.
8.      Set O(p) ← ord + 1 and ord ← ord + 1.
9.      For every q ∈ A(p), such that C(q) > C(p), do
10.         Set tmp ← max{C(p), δ(p, q)}.
11.         If tmp < C(q), then
12.             If C(q) ≠ +∞, then remove q from Q.
13.             Set C(q) ← tmp, R(q) ← R(p).
14.             Insert q in Q.
15. While the user is not satisfied, do
16.     The user can select a pixel q on the image.
17.     For every pixel p ∈ D_I, do
18.         If O(p) ≤ O(q) and R(p) = R(q), then
19.             Set L(p) ← L(R(p)).
```

The selection of a pixel q in line 16 is done based on the propagation order as described above.

One advantage of the these algorithms as compared to classical segmentation methods based on seed competition 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 post-processed, such that holes

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)
1530-1834/05 $20.00 © 2005 IEEE

Object	Description	Modality	# 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.

with area below a threshold are closed [13]. 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 4). 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 two methods:

M1. Object detection with seed competition and automatic κ_s computation;

M2. 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 might reduce the number of external seeds in M1, we decided to avoid it in order to evaluate the combination of seed competition and automatic κ_s detection with respect to M2.

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%, except for O2 where $T = 0.2\%$ in M1. Since objects from O1 to O6 do not have holes, we set the area closing threshold to some arbitrary high value

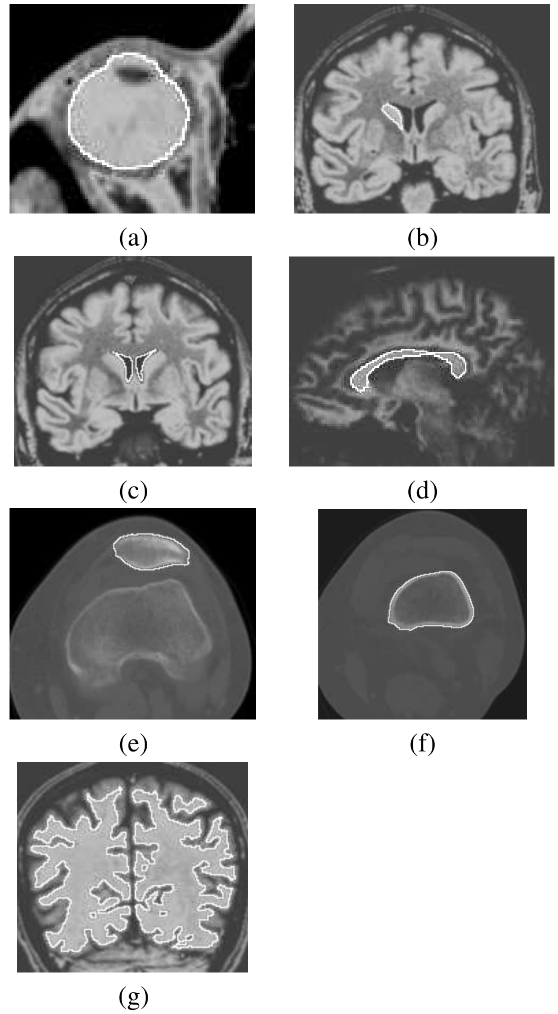

Figure 4. (a)-(g) Results of slice segmentation of the objects from 1 to 7, respectively, overlaid with the pre-processed images.

(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 of 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 that we chose δ_3 in some situations, despite $f_{\max}$ not being smooth.

In Medical Image Analysis, it is common to use as ground truth the results of manual segmentation performed by an expert user. 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. These small differences, however, seem to

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)
1530-1834/05 $20.00 © 2005 IEEE

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

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

	M1	M2	M1,M2
O1	29.5	77.6	0.962
O2	29.3	38.8	0.915
O3	31.3	61.3	0.935
O4	27.5	46.8	0.918
O5	15.0	61.0	0.946
O6	26.3	37.8	0.981
O7	46.3	284.8	0.930

Table 3. The average numbers of user interactions for each object and method, and the average similarity values between both methods for a same object.

be acceptable in many applications. This similarity measure was defined as follows.

Each object was represented by a set of l binary slices $\hat{L}_i = (D_I, L_i)$, $i = 1, 2, \dots, l$, 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:

$$1 - \frac{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) \oplus L'_i(p)}{\sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L_i(p) + \sum_{i=1}^{i=l} \sum_{\forall p \in D_I} L'_i(p)} \quad (7)$$

where $\oplus$ is the "exclusive or" operation. In the case of manual segmentation by an expert user, it has been shown that the similarity values are around 0.96 [7]. 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 M2 represents the classical approach based on relative fuzzy connectedness/watershed transform [17, 12]. The number of user interactions in both methods is the total number of seeds selected inside (NIS - *Number of Internal Seeds*) and outside (NES - *Number of External Seeds*) the object. Method M1 is the proposed variant of relative fuzzy connectedness. The number of seeds is expected to be much less with M1 than with M2, due to the automatic κ_s detection.

Table 3 shows the average number of interactions and similarity values among all users, for both methods. 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, M2 required 2.8 more user interactions than M1.

Table 4 shows in detail the average values of NIS, NES, and AKD for each object and method. Note that the number of automatic κ_s varied from 59% to 100% of NIS (88% on average). This demonstrates the effectiveness of the proposed approach for automatic κ_s detection and explains the

	M1			M2	
	NIS	NES	AKD	NIS	NES
O1	18.0	11.5	13.0	26.8	50.8
O2	25.3	4.0	24.3	18.8	20.0
O3	30.3	1.0	30.3	30.3	31.0
O4	22.3	5.2	19.8	22.3	24.5
O5	15.0	0.0	15.0	44.0	17.0
O6	26.3	0.0	15.5	22.8	15.0
O7	46.0	0.3	46.0	66.0	218.8

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

reduction of user interactions and external seeds in M1 with respect to M2. This is an important result for future automation, since seed competition is sensitive to the location of the external seeds due to the heterogeneity of the background.

6. Conclusions

We presented two IFT-based algorithms for object detection based on κ-connected components with 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 the 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 re-

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)

sults are extensive to other image types by suitable choice of pre-processing and dissimilarity function.

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 [2].

Acknowledgments

The authors thank CNPq (Proc. 302427/04-0), FAPESP (Proc. 03/09793-1 and Proc. 03/13424-1), and CAPES for the financial support.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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*, 23:356–363, Mar 2002.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)