Comparison of fuzzy connectedness and graph cut segmentation algorithms

Krzysztof Chris Ciesielski,[a,b] Jayaram K. Udupa,[b] A.X. Falcão,[c] and P.A.V. Miranda[c]

[a]Department of Mathematics, West Virginia University, Morgantown, WV 26506-6310

[b]Dept. of Radiology, MIPG, Univ. of Pennsylvania, Blockley Hall – 4th Floor, 423 Guardian Dr., Philadelphia, PA 19104-6021

[c]Institute of Computing, University of Campinas, Campinas, SP, Brazil

ABSTRACT

The goal of this paper is a theoretical and experimental comparison of two popular image segmentation algorithms: fuzzy connectedness (FC) and graph cut (GC). On the theoretical side, our emphasis will be on describing a common framework in which both of these methods can be expressed. We will give a full analysis of the framework and describe precisely a place which each of the two methods occupies in it. Within the same framework, other region based segmentation methods, like watershed, can also be expressed. We will also discuss in detail the relationship between FC segmentations obtained via image forest transform (IFT) algorithms, as opposed to FC segmentations obtained by other standard versions of FC algorithms.

We also present an experimental comparison of the performance of FC and GC algorithms. This concentrates on comparing the actual (as opposed to provable worst scenario) algorithms' running time, as well as influence of the choice of the seeds on the output.

1. INTRODUCTION

In this paper, we put a special emphasis on the delineation algorithms, that is, the segmentation procedures returning only one object region of interest at a time rather than multiple objects simultaneously. This makes the presentation clearer, even when a method can be easily extended to a multi-object version. In addition, the comparisons of different segmentation methods, both theoretical and experimental, is easier in this single-object setting. The general, multi-object segmentation algorithms will be also discussed here, but in a format of generalizations of the appropriate delineation methods and only at a theoretical level.

We will concentrate on the two families of algorithms, FC and GC. The leading theme will be the framework of *fuzzy connectedness, FC*, methods published since 1996 [1, 2, 3, 4, 5]. (For other extensions of FC, compare [19, 20].) We will also discuss in detail the family of *graph cut, GC*, methods [6, 7, 8, 9, 10, 11] (see also [12, 13]) and their relations to the FC family of algorithms. The GC methodology will be of special importance to our presentation, since we will formalize the FC framework in the language of graphs and graph cuts. Other segmentation methods can also be expressed in the presented framework [22, 14]. This, in particular, includes watershed, WS, [15, 16] and region growing level set methods from [17].

2. FRAMEWORK

In this section, we will point out the similarities between FC and GC algorithms and provide a common framework in which both of them (as well as WS) can be expressed. It should be stressed that the framework is very general and it will require some adjustments for the specific descriptions of the individual methods. Nevertheless, the adjustments will be small and the framework stresses the core similarities among the algorithms. At the same time, the subtle differences that do exist among them set them apart and cause their individual weaknesses/strengths. Due to the space limitation, the proofs of presented theoretical results will be deferred to a full, journal version of this paper.

The common feature of all presented algorithms is that the object to be segmented by them is indicated (by user, or automatically) by one or more spels (*spel* stands for a space element) referred to as *seeds*. In addition, if P is an object returned by such an algorithm, then any spel belonging to P is connected to at least one of the seeds indicating this object. The word "connected" indicates that the topological properties of the image scene play an important role in this class of segmentation processes. So, we will proceed with explaining what we mean by the image scene, its topology, as well as the notion of connectedness in this context.

For the rest of this article, $\mathfrak{n} \geq 2$ will stand for the dimension of the image we consider. In most medically relevant cases, $\mathfrak{n}$ is either 2 or 3, but a time sequence of 3D images is often considered as a 4D image.

Medical Imaging 2011: Image Processing, edited by Benoit M. Dawant, David R. Haynor,
Proc. of SPIE Vol. 7962, 796203 · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.872522

2.1. Digital image scene

A digital image scene C can be identified with any finite subset of the $\mathfrak{n}$-dimensional Euclidean space $\mathbb{R}^{\mathfrak{n}}$. However, we will concentrate here only on the case, most relevant for medical imaging, in which C is of the rectangular form $C_1 \times \cdots \times C_{\mathfrak{n}}$ and each C_i is identified with the set of integers $\{1, \ldots, m_i\}$.

A *topology* on a *scene* $\mathcal{C} = \langle C, \alpha \rangle$ will be given in terms of an *adjacency relation* α, which intuitively determines which spels $c, d \in C$ in a pair are "close enough to be considered connected." Formally, an adjacency relation α is a binary relation on C, which will be identified with a subset of $C \times C$, that is, spels $c, d \in C$ are α-adjacent, if and only if, $\langle c, d \rangle \in \alpha$. From the theoretical point of view, we need only to assume that the adjacency relation is symmetric (i.e., if c is adjacent to d, then also d is adjacent to c). However, in most medical applications, it is enough to assume that c is adjacent to d when the distance* $||c - d||$ between c and d does not exceed some fixed number. In most applications, we use adjacencies like 4-adjacency (for $\mathfrak{n} = 2$) or 6-adjacency (in the three-dimensional case), defined as $||c - d|| \leq 1$. Similarly, the 8-adjacency (for $\mathfrak{n} = 2$) and 26-adjacency (in 3D) relations can be defined as $||c - d|| \leq \sqrt{3}$.

The adjacency relation on C translates to the notion of connectivity as follows. A (connected) *path* p in a subset A of C is any finite sequence $\langle c_1, \ldots, c_k \rangle$ of spels in A such that any consecutive spels c_i, c_{i+1} in p are adjacent. The family of all paths in A is denoted by $\mathbb{P}^A$. Spels c and s are connected in A provided there exists a path $p = \langle c_1, \ldots, c_k \rangle$ in A from c to s such that $c_1 = c$ and $c_k = s$. The family of all paths in A from c to d is denoted by $\mathbb{P}^A_{cd}$.

2.2. Scene: duality of topological and graph-theoretical representations

The topological interpretation of the scene given above is routinely used in the description of many image segmentation algorithms. In particular, this is the case for FC, WS, and most of the level set methods. On the other hand, the algorithms for GC use the interpretation of the scene as a *directed graph* $G = \langle V, E \rangle$, where $V = C$ is the set of vertices† and E is the set of edges, which are identified with the set of pairs $\langle c, d \rangle$ of spels from $V = C$ for which c and d are joined by an edge. Notice that if we define E as the set of all adjacent pairs $\langle c, d \rangle$ from C (i.e., when $E = \alpha$), then the graph $G = \langle V, E \rangle$ and the scene $\mathcal{C} = \langle C, \alpha \rangle$ are identical structures (i.e., $G = \mathcal{C}$), despite their different interpretations. This forms the basis of the *duality between the topological and graph-theoretical views* of this structure: any topological scene $\mathcal{C} = \langle C, \alpha \rangle$ can be treated as a graph $G = \langle C, \alpha \rangle$, and, conversely any graph $G = \langle V, E \rangle$ can be treated as topological scene $\mathcal{C} = \langle V, E \rangle$.

Notice that, under this duality, the standard topological and graph theoretical notions fully agree. Specifically, a path p in C is connected in $\mathcal{C} = G$ in a topological sense, if and only if, it is connected in the graph $G = \mathcal{C}$. A subset P of C is connected, in a topological sense, in $\mathcal{C} = G$, if and only if, it is connected in the graph $G = \mathcal{C}$. Notice also that although the graph $G = \langle C, \alpha \rangle$ is defined as a directed graph, it can be also treated as an undirected graph, since any edge $\langle c, d \rangle$ in G can be reversed (i.e., if $\langle c, d \rangle$ is in $E = \alpha$, then so is $\langle d, c \rangle$).

2.3. Digital image

All of the above notions depend only on the geometry of the image scene and are independent of the image intensity function. Here, the *image intensity function* will be a function f from C into $\mathbb{R}^k$, $f \colon C \to \mathbb{R}^k$. The value $f(c)$ of f at any spel c is a k-dimensional vector of image intensities. A *digital image* will be treated as a pair $\langle \mathcal{C}, f \rangle$, where $\mathcal{C}$ is its scene (treated either as a topological scene or as a related graph) and f is the image intensity. We will often identify the image with its intensity function, that is, without explicitly specifying the associated scene adjacency. In case when $k = 1$, we will say that the image is scalar; for $k > 1$ we talk about vectorial images. Mostly, when giving examples, we will confine ourselves to scalar images.

2.4. Delineated objects and cost function

Fix an image $I = \langle \mathcal{C}, f \rangle$ and consider the task of delineating an object indicated by a non-empty set of seeds $S \subset C$. Suppose also that we have a set $T \subset C \setminus S$ of seeds (possibly empty) that indicate the background. This sets up the basic constrains on the object we seek—it will belong to the family $\mathcal{P}(S, T)$ of all sets $P \subset C$ containing S and disjoint with T. We will also write $\mathcal{P}(S)$ in place of $\mathcal{P}(S, \emptyset)$, which pertains to the case when there are no seeds indicating the background.

*In the examples, we use the Euclidean distance $|| \cdot ||$. But any other distance notion can be also used here.

†Actually, in the GC algorithms, usually V contains two additional vertices, but this will be discussed in detail later.

Now, assume that we have also an energy function ε associated with I, which, to every set $P \subset C$, assigns its energy value $\varepsilon(P) \in \mathbb{R}$. Assume also that we have a fixed energy threshold value $\theta \in \mathbb{R}$. Let $\mathcal{P}_\theta(S,T)$ be the family of all objects $P \in \mathcal{P}(S,T)$ such that $\varepsilon(P) \leq \theta$. Threshold θ will be always chosen so that the family $\mathcal{P}_\theta(S,T)$ is non-empty. We will write $\mathcal{P}_\theta(S)$ in place of $\mathcal{P}_\theta(S,\emptyset)$.

Any of the region-based algorithms we consider here will return, as a *delineated object*, a set $P_\theta(S,T) \in \mathcal{P}_\theta(S,T)$. Usually (but not always) $P_\theta(S,T)$ is the smallest (in set inclusion sense) element of $\mathcal{P}_\theta(S,T)$. In addition, in all essential cases, θ will be chosen, via an optimization process, as the minimum of all numbers $\varepsilon(P)$ over $P \in \mathcal{P}(S,T)$.

In the case of any of the methods FC, GC, WS, and LS, the value $\varepsilon(P)$ of the energy function is defined in terms of the boundary $\mathrm{bd}(P)$ of P, which is the set $K = \mathrm{bd}(P)$ of all edges $\langle c,d\rangle$ of a graph $\mathcal{C} = \langle C,E\rangle$ with $c \in P$ and d not in P. We often refer to this boundary set K as a *graph cut*, since removing these edges from $\mathcal{C}$ disconnects P from its complement $C \setminus P$. The actual definition of ε depends on the particular segmentation method, as indicated below.

Let $\kappa\colon E \to \mathbb{R}$ be a local cost function. For $\langle c,d\rangle \in E$ the value $\kappa(c,d)$ depends on the value of the image intensity function f at c, d, and (sometimes) at nearby spels. Usually, the bigger is the difference between the values of $f(c)$ and $f(d)$, the smaller is the cost value $\kappa(c,d)$. This agrees with the intuition that, often, the bigger the magnitude of the difference $f(c) - f(d)$ is, the greater is the chance that the "real" boundary of the object we seek is between these spels. More generally, the smallest $\kappa(c,d)$ may be corresponding to a difference $f(c) - f(d)$ that is expected for the object boundary. In the FC algorithms, κ is called the *affinity* function. In the GC algorithms, κ is treated as a weight function of the edges and is referred to as a *local cost* function. For the classical GC algorithms, the energy function $\varepsilon(P)$ is defined as the sum of the weights of all edges in $K = \mathrm{bd}(P)$, that is, as $\varepsilon^{\mathrm{sum}}(P) = \sum_{\langle c,d\rangle \in K} \kappa(c,d)$. The delineations for the FC family of algorithms are obtained with the energy function $\varepsilon(P)$ defined as the maximum of the weights of all edges in $K = \mathrm{bd}(P)$, that is, as $\varepsilon^{\max}(P) = \max_{\langle c,d\rangle \in K} \kappa(c,d)$. (See Section 3.2.) In other words, the delineations output by GC are the smallest sets whose total sum boundary cost is minimum, while FC outputs are the smallest sets such that the maximum boundary element cost (affinity) is the smallest. The same maximum function works also for the WS family with an appropriately chosen κ. The energy function for LS is a bit more complicated, as it depends also on the geometry of the boundary, specifically its curvature.

3. FUNDAMENTALS OF FUZZY CONNECTEDNESS

Let $I = \langle \mathcal{C}, f\rangle$ be a digital image, with the scene $\mathcal{C} = \langle C,E\rangle$ being identified with a graph. As indicated above, the FC segmentations require a local measure of connectivity κ associated with I, known as affinity function, where for a graph edge $\langle c,d\rangle \in E$ (i.e., for adjacent c and d) the number $\kappa(c,d)$ (edge weight) represents a measure of how strongly spels c and d are connected to each other in a local sense. The affinity functions are discussed in detail in the papers [19, 20]. Here we like to indicate only the examples of the most prominent affinities used in the applications so far [18], where $\sigma > 0$ is a fixed constant: (1) The *homogeneity based affinity*

$$\psi_\sigma(c,d) = e^{-||f(c)-f(d)||^2/\sigma^2}, \quad \text{where} \quad \langle c,d\rangle \in E, \tag{1}$$

with its value being close to 1 (meaning that c and d are well connected) when the spels have very similar intensity values; ψ_σ is related to the notion of directional derivative. (2) The *object feature based affinity* (single object case, with an expected intensity vector $m \in \mathbb{R}^k$ for the object) $\phi_\sigma(c,d) = e^{-\max\{||f(c)-m||,||f(d)-m||\}^2/\sigma^2}$, where $\langle c,d\rangle \in E$, with its value being close to 1 when both adjacent spels have intensity values close to m. The weighted averages of these two forms of affinities — either additive or multiplicative — have also been used.

The reader can keep these examples of the affinity functions in mind when reading the rest of the results on the FC methods. Nevertheless, we like to stress that these are just examples and a considerable variety of affinity functions can be used [19, 20]. In the algorithms presented below, we will assume that the values of affinity functions are in the interval $[0,1]$, as is the case of the examples given above. For the rest of this section, we will assume that a digital image $I = \langle \mathcal{C}, f\rangle$, a related affinity function κ, and the associated energy function $\varepsilon = \varepsilon^{\max}$ are fixed.

3.1. Absolute fuzzy connectedness, AFC, objects

In this section, we will define an AFC object, denoted $P_{S\theta}$ (notice that this notation is different from $\mathcal{P}_\theta(S)$), containing a non-empty set $S \subset C$ of seeds and indicated by a threshold θ. Our main goal then is to fulfill a promise from Section 2 by showing, in Theorem 3.1, that $P_{S\theta}$ is indeed the smallest element of the family $\mathcal{P}_\theta(S) = \mathcal{P}_\theta(S, \emptyset) = \{P \in \mathcal{P}(S)\colon \varepsilon(P) \leq \theta\}$.

We cannot define $P_{S\theta}$ as the smallest element of the family $\mathcal{P}_\theta(S)$, since such a definition is not well justified, unless it is proved that the family $\mathcal{P}_\theta(S)$ actually contains the smallest element. (In general, an element $P' = \arg\min_{P \in \mathcal{P}_\theta(S)} \varepsilon^{\max}(P)$ need not be the smallest element of $\mathcal{P}_\theta(S)$.) Therefore, we will use an alternative definition, indicated below, which is motivated by the actual implementation of the FC algorithm. Theorem 3.1 then proves the equivalence of both approaches.

The *strength* of a path $p = \langle c_1, \ldots, c_k\rangle$, $k > 1$, is defined as $\mu(p) = \min\{\kappa(c_{i-1}, c_i)\colon 1 < i \leq k\}$, that is, the strength of the κ-weakest link of p. For $k = 1$ (i.e., when p has length 1) we associate with p the strongest possible value: $\mu(p) = 1$. (For $k = 1$, when set $\{\kappa(c_{i-1}, c_i)\colon 1 < i \leq k\}$ is empty, we put $\mu(p) = 1$.) For $c, d \in A \subseteq C$, the (global) *κ-connectedness strength in A between c and d* is defined as the strength of a strongest path in A between c and d; that is, $\mu^A(c, d) = \max\{\mu(p)\colon p \in \mathbb{P}^A_{cd}\}$. Notice that $\mu^A(c, c) = \mu(\langle c\rangle) = 1$. We will refer to the function μ^A as a *connectivity measure* (on A) induced by κ. For $c \in A \subseteq C$ and a non-empty $D \subset A$, we also define $\mu^A(c, D) = \max_{d \in D} \mu^A(c, d)$. Then, we define the *absolute fuzzy connectedness, AFC, object* $P_{S\theta}$ as $\{c \in C\colon \theta < \mu^C(c, S)\}$.

THEOREM 3.1. *If $\emptyset \neq S \subset C$ and $\theta < 1$, then $P_{S\theta}$ is the smallest element of the family $\mathcal{P}_\theta(S)$.*

If a set of seeds S contains only one seed s, then we will write $P_{s\theta}$ for the object $P_{S\theta} = P_{\{s\}\theta}$. It is easy to see that $P_{S\theta}$ is a union of all objects $P_{s\theta}$ for $s \in S$, that is, $P_{S\theta} = \bigcup_{s \in S} P_{s\theta}$.

Notice, that $P_{s\theta}$ is connected, since for every $c \in P_{s\theta}$ there is a path $p = \langle c_1, \ldots, c_k\rangle$ from s to c with $\mu(p) = \mu^C(c, s) > \theta$, and such a path is contained in $P_{s\theta}$. Moreover, if $G_\theta = \langle C, E_\theta\rangle$ is a graph with E_θ consisting of the scene graph edges $\langle c, d\rangle$ with weight $\kappa(c, d)$ greater than θ, then $P_{s\theta}$ is a connected component of G_θ containing s, and $P_{S\theta}$ is a union of all components of G_θ intersecting S.

One of the most important properties of the AFC objects is known as *robustness*. Intuitively, this property states that the FC delineation results do not change if the seeds S indicating an object are replaced by another nearby set U of seeds. Formally, it reads as follows.

THEOREM 3.2. **(Robustness)** *For every digital image I on a scene $\mathcal{C} = \langle C, E\rangle$, every $s \in C$ and $\theta < 1$, if $P_{s\theta}$ is an associated FC object, then $P_{U\theta} = P_{s\theta}$ for every $U \subset P_{s\theta}$. More generally, if $S \subset C$ and $U \subset P_{S\theta}$ intersects every connected component of G_θ intersecting $P_{S\theta}$ (i.e., $U \cap P_{s\theta} \neq \emptyset$ for every $s \in S$), then $P_{U\theta} = P_{S\theta}$.*

The proof of this result follows easily from our graph interpretation of the object, as indicated above. The proof based only on the topological description of the scene can be found in [2, 4]. The robustness property constitutes one of the strongest arguments for defining the objects in the FC fashion. We note that none of the other algorithms discussed here (GC as well as briefly mentioned WS and LS) have this property.

The standard algorithm $\kappa\theta FOEMS$ that, given an image $I = \langle \mathcal{C}, f\rangle$, a set $S \subset C$ of seeds indicating the object, and a threshold $\theta < 1$, returns the AFC object $P_{S\theta}$ is described in [1]. It is easy to see that $\kappa\theta FOEMS$ runs in linear time with respect to the size n of the scene C. More precisely, it runs in time of order $O(\Delta n)$, where Δ is the degree of the graph $\mathcal{C}$ (i.e., the largest number of spels that can be adjacent to a single spel; e.g., $\Delta = 26$ for the 26-adjacency).

3.2. Optimization: Relative fuzzy connectedness RFC

The AFC delineation, beside a set S of seeds indicating the object, requires also a mysterious parameter — a threshold θ — which has no visible association with the object. In the *relative fuzzy connectedness, RFC*, delineation this requirement for the explicit input of the threshold θ is removed; it is replaced by a requirement of an input of a non-empty set T of seeds, disjoint with S, indicating the background of the object we seek. The actual RFC object $P_{S,T}$ is defined via competition of seed sets S and T for attracting a given spel c to their realms (see [2]): $P_{S,T} = \{c \in C\colon \mu^C(c, S) > \mu^C(c, T)\}$. Clearly, we would like for $P_{S,T}$ to belong to $\mathcal{P}(S, T)$. It is easy to see that for this to be true, it is necessary that the number $\mu^C(S, T) = \max_{s \in S} \mu^C(s, T)$ is less than 1. Therefore, we will always assume that the seed sets are chosen properly, that is, such that $\mu^C(S, T) < 1$. Notice also that $P_{S,T} = \bigcup_{s \in S} P_{\{s\},T}$, since $P_{S,T} = \{c \in C\colon (\exists s \in S)\mu^C(c, s) > \mu^C(c, T)\} = \bigcup_{s \in S} P_{\{s\},T}$, as

$\mu^C(c,S) = \max_{s\in S} \mu^C(c,s)$.

The fact that $P_{S,T}$ minimizes the energy ε in $\mathcal{P}(S,T)$ follows, in particular, from the following theorem. Notice also that its part (iii) indicates that $P_{S,T} = \bigcup_{s\in S} P_{\{s\},T}$ not only minimizes ε globally, but that each of its components $P_{\{s\},T}$ minimizes ε on $\mathcal{P}(\{s\},T)$ with its own version of the minimum, $\theta_s = \mu^C(s,T)$, which may be (and often is) smaller than the global minimizer $\theta_S = \mu^C(S,T)$. In other words, the object $P_{S,T}$ can be viewed as a result of minimization procedure used separately for each $s \in S$, which gives a sharper result than a simple minimization of global energy for the entire object $P_{S,T}$.

THEOREM 3.3. *Assume that $\mu^C(S,T) < 1$. Then $P_{S,T}$ minimizes the energy $\varepsilon = \varepsilon^{\max}$ on $\mathcal{P}(S,T)$. Moreover,*

(i) *The number $\theta_S = \mu^C(S,T)$ is the minimum of ε on $\mathcal{P}(S,T)$, that is, $\theta_S = \min\{\varepsilon(P)\colon P \in \mathcal{P}(S,T)\}$.*

(ii) *If S is a singleton, then $P_{S,T}$ is the smallest set in $\mathcal{P}_{\theta_S}(S,T)$.*

(iii) *For general S, let $\mathcal{P}^*_{\theta_S}(S,T)$ be the family of all sets of the form $\bigcup_{s\in S} P^s$, where each P^s belongs to $\mathcal{P}_{\theta_{\{s\}}}(\{s\},T)$. Then $\mathcal{P}^*_{\theta_S}(S,T) \subset \mathcal{P}_{\theta_S}(S,T)$ and $P_{S,T}$ is the smallest set in $\mathcal{P}^*_{\theta_S}(S,T)$.*

The above described delineation RFC procedure easily and naturally generalizes to the segmentation algorithm of $m > 1$ separate objects. More precisely, assume that for an image $I = \langle \mathcal{C}, f\rangle$ we have a family $\mathcal{S} = \{S_1, \ldots, S_m\}$ of pairwise disjoint non-empty sets of seeds, each S_i indicating an associated object P_i. If for each i we put $T_i = (\bigcup_{j=1}^m S_j) \setminus S_i$, then the RFC segmentation is defined as a family $\mathcal{P} = \{P_{S_i,T_i}\colon i = 1,\ldots,m\}$. It is easy to see that the different objects in $\mathcal{P}$ are disjoint. Moreover, each object P_{S_i,T_i} contains S_i provided the seeds are chosen properly, that is, when $\mu^C(S_i,S_j) < 1$ for every $j \neq i$.

It is worth to mention that while each P_{S_i,T_i} minimizes the energy $\varepsilon = \varepsilon^{\max}$ in $\mathcal{P}(S_i,T_i)$ with the energy value $\theta_i = \mu^C(S_i,T_i)$, the numbers θ_i's need not be equal when the number m of objects is greater than 2.

The robustness Theorem 3.2 can be modified to this setting as follows. (See [2, 4].)

THEOREM 3.4. **(Robustness for RFC)** *Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a family of seeds in a digital image I and let $\mathcal{P} = \{P_{S_i,T_i}\colon i = 1,\ldots,m\}$ be the associated RFC segmentation. For every i and $s \in S_i$ let $g(s)$ be in $P_{\{s\},T_i}$ and let $S'_i = \{g(s)\colon s \in S_i\}$. Then, for every i, if $T'_i = (\bigcup_{j=1}^m S'_j) \setminus S'_i$, then $P_{S_i,T_i} = P_{S'_i,T'_i}$.*

In other words, if each seed s present in $\mathcal{S}$ is only "slightly" shifted to a new position $g(s)$, then the RFC segmentation $\{P_{S'_i,T'_i}\colon i = 1,\ldots,m\}$ associated with the modified set of seeds is identical to the original one $\mathcal{P}$.

To find the RFC segmentation $\mathcal{P} = \{P_{S_i,T_i}\colon i = 1,\ldots,m\}$ for a given family $\mathcal{S} = \{S_1,\ldots,S_m\}$ of seeds, it is enough to use m-times an algorithm that for disjoint non-empty sets $S,T \subset C$ with $\mu^C(S,T) < 1$ returns the object $P_{S,T}$. In the experimental section we examine two versions of such an algorithm: RFC-standard and RFC-IFT. Each version follows the same simple procedure, as displayed. They differ only in a routine that, given a non-empty set $S \subset C$, returns $\mu^C(\cdot,S)$. So, their outputs are identical.

Algorithm RFC (-standard or -IFT)
Input: Affinity function defined on a scene $\mathcal{C} = \langle C, E\rangle$ and non-empty disjoint sets $S,T \subset C$.
Output: The RFC object $P_{S,T} = \{c \in C\colon \mu^C(c,S) > \mu^C(c,T)\}$.
begin
1. calculate $\mu^C(\cdot,S)$ and $\mu^C(\cdot,T)$ (running appropriate subroutine twice, once for S and once for T);
2. return $P_{S,T} = \{c \in C\colon \mu^C(c,S) > \mu^C(c,T)\}$;
end

The RFC-standard algorithm calculates function $\mu^C(\cdot,S)$ using the routine $\kappa FOEMS$ [1] (not presented in this paper) that runs in time of order $O(n^2)$, (or, more precisely, $O(\Delta^2 n^2)$), where n is the size of the scene $\mathcal{C}$. Thus, since line 2 of RFC runs in time $O(n)$, the RFC-standard algorithm stops in time of order $O(n^2)$.

The RFC-IFT algorithm calculates function $\mu^C(\cdot,W)$ using the IRFC-IFT routine described in the next section, which takes as an input a non-empty set $W \subset C$ and returns, in $O(n)$ time, the function $\mu^C(\cdot,W)$. Clearly, RFC-IFT runs in $O(n)$ time, since so does IRFC-IFT.

3.3. Iterative relative fuzzy connectedness IRFC

The RFC segmentation $\mathcal{P} = \{P_{S_i,T_i}\colon i = 1,\ldots,m\}$ of a scene, associated with a family $\mathcal{S} = \{S_i\colon i = 1,\ldots,m\}$ of seeds, can still leave quite a sizable "leftover" background set $B = B_{\mathcal{P}}$ of all spels c outside any of the objects wherein the strengths of connectedness are equal with respect to the seeds. The goal of the *iterative relative fuzzy*

connectedness segmentation, IRFC, is to find a way to naturally redistribute some of the spels from $B_{\mathcal{P}}$ among the object regions in a new generation (iteration) of segmentation. There are two FC delineation approaches that lead to the IRFC objects: the standard, bottom-up approach, in which the RFC object P_{S_i,T_i} is expanded to the "maximal" IRFC object $P^{\infty}_{S_i,T_i}$; and the IFT top-down approach, in which the IRFC object $P^{IFT}_{S_i,T_i}$ is chosen as the minimal among all S_i-indicated objects that result from different IFT $\mathcal{S}$-indicated segmentations of the scene. In this section we describe briefly both of these approaches and prove that indeed the objects $P^{\infty}_{S_i,T_i}$ and $P^{IFT}_{S_i,T_i}$ are identical. In addition, we show that this common object can be viewed as a result of the energy ε minimization, that is, it satisfies an analog of Theorem 3.3.

Historically, the first IRFC approach was bottom-up [3, 4], so we start with it. The idea is to treat the RFC delineated objects P_{S_i,T_i} as the first iteration $P^1_{S_i,T_i}$ approximation of the final segmentation, while the next step iteration is designed to redistribute some of the background spels $c \in B_{\mathcal{P}}$, for which $\mu^C(c,S_i) = \mu^C(c,T_i)$ for some i. Such a tie can be resolved if the strongest paths justifying $\mu^C(c,S_i)$ and $\mu^C(c,T_i)$ cannot pass through the spels already assigned to another object. In other words, we like to add spels from the set $P^* = \{c \in B\colon \mu^{B\cup P_{S_i,T_i}}(c,S_i) > \mu^{B\cup P_{S_j,T_i}}(c,S_j) \text{ for every } j \neq i\}$, to a new generation $P^2_{S_i,T_i}$ of $P^1_{S_i,T_i}$, that is, define $P^2_{S_i,T_i}$ as $P^1_{S_i,T_i} \cup P^*$. This formula can be taken as a definition. However, from the algorithmic point of view, it is more convenient to define $P^2_{S_i,T_i}$ as $P^2_{S_i,T_i} = P^1_{S_i,T_i} \cup \{c \in C \setminus P^1_{S_i,T_i}\colon \mu^C(c,S_i) > \mu^{C\setminus P^1_{S_i,T_i}}(c,T_i)\}$, while the equation $P^2_{S_i,T_i} = P^1_{S_i,T_i} \cup P^*$ always holds, as proved in [4, thm. 3.7]. Thus, the IRFC object is defined as $P^{\infty}_{S_i,T_i} = \bigcup_{k=1}^{\infty} P^k_{S_i,T_i}$, where sets $P^k_{S_i,T_i}$ are defined recursively by the formulas $P^1_{S_i,T_i} = P_{S_i,T_i}$ and

$$P^{k+1}_{S_i,T_i} = P^k_{S_i,T_i} \cup \{c \in C \setminus P^k_{S_i,T_i}\colon \mu^C(c,S_i) > \mu^{C\setminus P^k_{S_i,T_i}}(c,T_i)\}. \tag{2}$$

Notice, that formula (2) holds also for $k = 0$, where we define $P^0_{S_i,T_i}$ as the empty set $\emptyset$. The IRFC segmentation associated with the family $\mathcal{S}$ of seeds is defined as the collection $\mathcal{P}^I_{\mathcal{S}} = \{P^{\infty}_{S_i,T_i}\colon i = 1,\ldots,m\}$. Its members are still disjoint, as proved in [4]. More importantly, each IRFC object $P^{\infty}_{S_i,T_i}$ has the same energy value as its RFC counterpart P_{S_i,T_i}:

THEOREM 3.5. *Assume that $\theta = \mu^C(S,T) < 1$. Then $P^{\infty}_{S,T}$ minimizes the energy $\varepsilon = \varepsilon^{\max}$ on $\mathcal{P}(S,T)$, i.e., $P^{\infty}_{S,T} \in \hat{\mathcal{P}}_{\theta}(S,T) = \{P \in \mathcal{P}(S,T)\colon \varepsilon(P) = \theta\}$.*

The IRFC segmentation is robust in the sense of Theorem 3.4, where in its statement the objects P_{S_i,T_i} are replaced by the first iteration $P^1_{S_i,T_i}$ of $P^{\infty}_{S_i,T_i}$. This follows easily from Theorem 3.4, see [4, thm. 2.5]. The robustness result does not hold for the entire sets $P^{\infty}_{S_i,T_i}$, as discussed in [4]. (Specifically, see [4, example 3.15].) The original IRFC algorithm (see e.g. [4]) was not very efficient — it run in time of order $O(n^3)$. Therefore, we will not discuss it here.

Next, we will describe the IFT, top-down approach, which was originally developed in [13]. A *spanning forest* for a scene $\mathcal{C} = \langle C, \alpha\rangle$ is any family $\mathbb{F}$ of directed paths such that: (1) for every spel $c \in C$ there is exactly one path p_c in $\mathbb{F}$ which terminates at c; (2) for every path $p = \langle c_1,\ldots,c_k\rangle$ in $\mathbb{F}$, every initial segment of p (i.e., a path $\langle c_1,\ldots,c_j\rangle$ with $j = 1,\ldots,k$) also belongs to $\mathbb{F}$. This terminology agrees with the standard graph theory terminology, when $\mathcal{C}$ is treated as a graph. A spanning forest $\mathbb{F}$ is often (e.g. [21, 13]) identified with its *predecessor map* $Pr_{\mathbb{F}}\colon C \to C \cup \{nil\}$ defined as follows: if $p_c = \langle c_1,\ldots,c_k\rangle \in \mathbb{F}$ is the unique path with $c_k = c$, then $Pr_{\mathbb{F}}(c) = nil$ for $k = 1$ and $Pr_{\mathbb{F}}(c) = c_{k-1}$ for $k > 1$. The spanning forest $\mathbb{F}$ can be easily recovered from the predecessor function $Pr_{\mathbb{F}}$, so the objects $\mathbb{F}$ and $Pr_{\mathbb{F}}$ are often identified.

Any spanning forest $\mathbb{F}$ on $\mathcal{C}$ induces also its *root function* $R_{\mathbb{F}}$ from C onto $S_{\mathbb{F}} = \{c \in C\colon Pr_{\mathbb{F}}(c) = nil\}$ defined for any $c \in C$ as $R_{\mathbb{F}}(c) = c_1$, where $p_c = \langle c_1,\ldots,c_k\rangle$ is the unique path in $\mathbb{F}$ which terminates at c (i.e., with $c_k = c$). For an $S \subset C$ we also define $P(S,\mathbb{F})$ as the set of all $c \in C$ with $R_{\mathbb{F}}(c) \in S$. In particular, if $\mathcal{S} = \{S_i\colon i = 1,\ldots,m\}$ is a family of pairwise disjoint non-empty sets of seeds in $\mathcal{C}$ and $\mathbb{F}$ is a spanning forest on $\mathcal{C}$ for which $S_{\mathbb{F}} = \bigcup_{i=1}^{m} S_i$, then the family $\mathcal{P}_{\mathbb{F},\mathcal{S}} = \{P(S_i,\mathbb{F})\colon i = 1,\ldots,m\}$ is a partition of C to which we refer as the *segmentation indicated by $\mathbb{F}$ and $\mathcal{S}$*. Notice that $S_i \subset P(S_i,\mathbb{F})$ for every $P(S_i,\mathbb{F}) \in \mathcal{P}_{\mathbb{F},\mathcal{S}}$.

For a fixed non-empty $S \subset C$, we say that a *path $p = \langle c_1,\ldots,c_k\rangle$ is optimal* (with respect to S and a path cost function μ) provided that $c_1 \in S$ and $\mu(p) = \mu^C(c_k,S)$. A *spanning forest $\mathbb{F}$ is optimal* (with respect to a path cost function μ) provided every path in $\mathbb{F}$ is optimal with respect to $S_{\mathbb{F}}$. Following [21] (compare also [13]), we say that any partition $\mathcal{P}_{\mathbb{F},\mathcal{S}}$ of C for which $\mathbb{F}$ is optimal is an *IFT segmentation by Seed Competition, IFT-SC.* Such partitions $\mathcal{P}_{\mathbb{F},\mathcal{S}}$ are closely related to the IRFC partition $\mathcal{P}^I_{\mathcal{S}} = \{P^{\infty}_{S_i,T_i}\colon i = 1,\ldots,m\}$, as recognized in [21].

However, the segmentations $\mathcal{P}_{\mathbb{F},\mathcal{S}}$ are, in general, not unique and, usually, not equal to $\mathcal{P}^I_{\mathcal{S}}$, since $\mathcal{P}^I_{\mathcal{S}}$ is usually not a partition of C (i.e, there are spels in C belonging to no $P^{\infty}_{S_i,T_i}$). In addition, not all segmentations $\mathcal{P}_{\mathbb{F},\mathcal{S}}$ must minimize the energy function ε. Therefore, we need to modify slightly the IFT-SC approach to make the objects $P^{IFT}_{S_i,T_i}$ it generates equal to the IRFC objects $P^{\infty}_{S_i,T_i}$.

For a family $\mathcal{S} = \{S_i \colon i = 1, \ldots, m\}$ of seeds in $\mathcal{C}$, $i \in \{1, \ldots, m\}$, and $T_i = \bigcup_{j \neq i} S_j$, we define $P^{IFT}_{S_i,T_i}$ as the smallest set in the family $\mathcal{P}^{IRFC}(S_i, T_i) = \{P(S_i, \mathbb{F}) \colon \mathbb{F} \text{ is optimal and } S_{\mathbb{F}} = S_i \cup T_i\}$. Of course, for this definition to be correct, it needs to be argued that the family $\mathcal{P}^{IRFC}(S_i, T_i)$ indeed has the smallest element. This, and the fact that $P^{IFT}_{S_i,T_i} = P^{\infty}_{S_i,T_i}$, is proved in Theorem 3.6.

The proof of Theorem 3.6 and the effective construction of objects $P^{IFT}_{S_i,T_i}$ are based on the following IRFC-IFT algorithm, which is a version of Dijkstra's procedure for computing minimum-cost path from a single source in a graph. IRFC-IFT also constitutes a modification of the algorithm from [13] to the format that best suits our goals here. In the algorithm we will use a *dictionary linear order* relation defined on a set $\mathbb{R}^2$ as: $\langle r_1, r_2\rangle \preceq \langle s_1, s_2\rangle$ if, and only if, either $r_1 < s_1$ or both $r_1 = s_1$ and $r_2 \leq s_2$. We will write $\langle r_1, r_2\rangle \prec \langle s_1, s_2\rangle$ when $\langle r_1, r_2\rangle \preceq \langle s_1, s_2\rangle$ but $\langle r_1, r_2\rangle \neq \langle s_1, s_2\rangle$.

Algorithm IRFC-IFT	
Input:	Affinity function κ on a connected scene $\mathcal{C} = \langle C, E\rangle$ of size n with values in a set $Z \subset [0, 1]$. A non-empty set $W \subset C$ of seeds. A priority labeling map $\lambda \colon C \to \{0, 1\}$ such that $\emptyset \neq S_\lambda \subseteq W$, where $S_\lambda = \{c \in C \colon \lambda(s) = 0\}$.
Output:	Function $\mu^C(\cdot, W)$; an optimal spanning forest $\mathbb{F}$ with $S_{\mathbb{F}} = W$, indicated by its predecessor map Pr; and the object $P^{IFT}_{S,T} = P(S, \mathbb{F})$, where $S = S_\lambda$ and $T = W \setminus S$.
Auxiliary Data Structure:	Functions: $h \colon C \to \{-1\} \cup Z$ approximating $\mu^C(\cdot, W)$, $Pr \colon C \to C \cup \{nil\}$ eventually becoming the predecessor map $Pr_{\mathbb{F}}$, and $R \colon C \to C$ eventually becoming the root function $R_{\mathbb{F}}$. A priority queue Q of size n ordered such that: c_1 can precede c_2 in Q (denoted $c_1 \preceq c_2$) if, and only if, $\langle h(c_1), \lambda(R(c_1))\rangle \preceq \langle\, h(c_2), \lambda(R(c_2))\rangle$.

We will also use a data structure Q, ordered by $\preceq$ as indicated in the description of IRFC-IFT. Q will hold at most n elements, pointers to spels, and it can be defined as a priority queue like binary heap that allows insertion and deletion of any element in $O(\ln n)$ time. However, in medical practice, the set of possible values of an affinity function κ is usually restricted to a fixed set Z of a modest size, most frequently of a form $Z = \{i/D \colon i = 0, 1, \ldots, D\}$ for D of order $2^{12} = 4096$. In this case, Q can be defined as an array of buckets indexed by the set $V = (\{-1\} \cup Z) \times \{0, 1\}$ and ordered according to $\preceq$. Each bucket with an index $\langle z, \ell\rangle \in V$ consists of pointers to the spels with a current label $\langle z, \ell\rangle$. An advantage of Q to be represented in such an array format is that this allows $O(1)$-time insertion into Q and deletion from Q of any element c with a fixed label $\langle z, \ell\rangle$. Emptying Q in the priority order from the largest to the smallest indexed spel, as done when executing line 4 of the algorithm, may require $O(D)$ time during the complete execution of IRFC-IFT. Since we consider $O(D)$ as not exceeding $O(n)$, we use Q in an array format in our implementation of IRFC-IFT, and, in theoretical investigation, estimate the running time of IRFC-IFT as $O(n)$. However, if facing running IRFC-IFT in the situation when $O(D)$ is lager than $O(n \ln n)$, it makes sense to use as Q a simple priority queue (like binary heap), which will then lead to IRFC-IFT running time of order $O(n \ln n)$.

THEOREM 3.6. *The algorithm IRFC-IFT runs in $O(n)$ time and its output is as indicated in the algorithm. Moreover, if $T \neq \emptyset$ and $\mu^C(S, T) < 1$, then (i) $P^{IFT}_{S,T} = P^{\infty}_{S,T}$; (ii) $P^{IFT}_{S,T}$ is the smallest element of the family $\mathcal{P}^{IRFC}(S, T) \subset \mathcal{P}(S, T)$; (iii)$P^{IFT}_{S,T}$ minimizes the energy ε on $\mathcal{P}(S, T)$.*

4. GC DELINEATION

For the GC algorithms, a graph $G^I = \langle V, E\rangle$ associated with the image $I = \langle \mathcal{C}, f\rangle$, where $\mathcal{C} = \langle C, \alpha\rangle$, is a slight modification of the graph $\langle C, \alpha\rangle$ discussed above. Specifically, the set of vertices V is defined as $C \cup \{s, t\}$, that is, the standard set C of image spels considered as vertices is expanded by two new additional vertices s and t called *terminals*. Individually, s is referred to as *source* and t as *sink*. The set of edges is defined as $E = \alpha \cup \{\langle b, d\rangle \colon$ one of b, d is in C, the other in $\{s, t\}\}$. So, the set of edges between vertices in C remains as in $\mathcal{C}$, while we connect each terminal vertex to each $c \in C$.

The simplest way to think about the terminals is that they serve as the seed indicators: s for seeds $S \subset C$ indicating the object; t for seeds $T \subset C$ indicating the background. The indication works as follows. For each

edge connecting a terminal $r \in \{s,t\}$ with a $c \in C$, associate the weight: ∞ if either $r = s$ and $c \in S$, or $r = t$ and $c \in T$; and 0 otherwise. This means, that the source s has infinitely strong connection to any seed c in S, and the weakest possible to any other spel $c \in C$. (We assume that all weights are nonnegative, that is, in $[0,\infty]$.) Similarly, for the sink t and seeds c from T.

Now, assume that for every edge $\langle c,d\rangle \in \alpha$ we give a weight $\kappa(c,d)$ associated with the image $I = \langle \mathcal{C}, f\rangle$. Since the algorithm for delineating RFC object uses only the information on the associated graph (which includes the weights given by the affinity κ), we can delineate RFC object $P^*_{\{s\},\{t\}} \subset V$ associated with this new graph G^I. It is easy to see that the RFC object $P_{S,T} \subset C$ associated with I is equal to $P^*_{\{s\},\{t\}} \cap C$. Similarly, for $\theta < 1$, if $P^*_{s\theta} \subset V$ is an AFC object associated with the graph G^I, then the AFC object $P_{S\theta} \subset C$ associated with I is equal to $P^*_{S\theta} \cap C$. All of this proves that, from the FC framework point of view, replacing the graph $G = \langle C, \alpha\rangle$ with G^I is only technical in nature and results in no delineation differences. Historically, the rationale for using in GC framework's graphs G^I, with distinctive terminals, is algorithmic in nature. More precisely, for a weighted graph $G = \langle V, E\rangle$ with positive weights and two distinct vertices s and t indicated in it, there is an algorithm returning the smallest set P_G in $\mathcal{P}_{\min} = \{P \in \mathcal{P}(s,t)\colon \varepsilon^{\mathrm{sum}}(P) = \varepsilon_0\}$, where $\mathcal{P}(s,t) = \{P \subset V \setminus \{t\}\colon s \in P\}$, $\varepsilon_0 = \min\{\varepsilon^{\mathrm{sum}}(P)\colon P \in \mathcal{P}(s,t)\}$, $\varepsilon^{\mathrm{sum}}(P) = \sum_{e\in \mathrm{bd}(P)} w_e$, and w_e is the weight of the edge e in the graph.

Now, let $G^I = \langle C \cup \{s,t\}\rangle, E\rangle$ be the graph associated with an image I as described above, that is, weights of edges between spels from C are obtained from the image I (in a manner similar to the affinity numbers) and weights between the other edges by seed sets S and T indicating foreground and background. In this setting we can restate the above comments in a format similar to that of Theorems 3.3 and 3.5:

THEOREM 4.1. *The GC object $P^{\Sigma}_{S,T} = C \cap P_{G^I}$ minimizes the energy $\varepsilon^{\mathrm{sum}}$ on $\mathcal{P}(S,T)$ and $P^{\Sigma}_{S,T}$ is the smallest set in $\mathcal{P}(S,T)$ with this energy.*

4.1. GC vs FC algorithms

In spite of similarities between the GC and RFC methodologies as indicated above, there are also considerable differences between them. There are several theoretical advantages of the RFC framework over GC in this setting.

Speed: The FC algorithms run faster than those for GC. The theoretical estimation of the worst case run time of the two main FC algorithms, RFC-IFT and IRFC-IFT, is $O(n)$ (or $O(n \ln n)$) with respect to the scene size n (see Section 3), while the best theoretical run time for delineating $P^{\Sigma}_{S,T}$ is of order $O(n^3)$ (for the best known algorithms) or $O(n^{2.5})$ (for the fastest currently known), see [8]. The experimental comparisons of the running time also confirm that FC algorithms run faster. (See Section 5, Figure 1.)

Robustness: The outcome of FC algorithms is unaffected by reasonable (within the objects) changes of the position of seeds. (See Theorems 3.2 and 3.4.) On the other hand, the results of GC delineation may become sensitive for even small perturbations of the seeds. See Section 5, Figure 1.

Multiple objects: The RFC framework handles easily the segmentation of multiple objects, retaining its running time and robustness property from the single object case. (See Section 3.) In the multiple object setting, GC leads to an NP-hard problem (see [6]); so all existing algorithms for performing the required precise delineation run in exponential time, rendering them impractical. (However, there are algorithms that render approximate solutions for such GC problems in a practical time [6].)

GC shrinking problem: The GC algorithms have a tendency of choosing the objects with very small size of the boundary, even if the weights of the boundary edges are very high. (See e.g. [9, 12].) This may easily lead to the segmented object being very close to either the foreground seed set S, or to the background seed set T. Therefore, unless sets S and T are already good approximations for the desired delineation, the object returned by GC may be far from desirable. This problem has been addressed by many authors, via modification of the GC method. The best known among these modifications is the method of normalized cuts (see [12]), in which the energy $\varepsilon^{\mathrm{sum}}$ is replaced by another "normalized" measure of energy cost. However, finding the resulting delineation minimizing this new energy measure is NP-hard as well (see [12]), and so only approximate solutions can be found in practical time. Notice that neither RFC nor the IRFC method has any shrinking problem.

Iterative approach: The FC framework allows an iterative refinement of its connectivity measure μ^A (leading to the iterative relative FC), which in turn makes it possible to redefine ε as we go along. From the viewpoint of algorithm, this is a powerful strategy. No such methods exist for GC at present.

All this said, it should be noted that GC has also some nice properties that FC does not possess. First notice that the shrinking problem is the result of favoring shorter boundaries over the longer; that is, GC has a smoothing effect on the boundaries. This, in many (but not all) cases of medically important image delineations, is perhaps a desirable feature. There is no boundary smoothing factor built into the FC basic framework, and, if desirable, boundary smoothing must be done at the FC post processing stage.

Another nice feature of GC graph representation G^I of an image I is that the weights of edges to terminal vertices naturally represent the object-feature type of affinity, while the weights of the edges with both vertices in C are naturally connected with the homogeneity type of affinity (1). This is the case, since homogeneity based affinity (an image intensity derivative concept) is a binary relation in nature, while the object-feature based affinity is actually a unary relation. Such a clear cut distinction is difficult to achieve in the FC framework, since it requires only one affinity relation in its setting.

Now, let $P^{IFT}_{S,T}(\kappa)$ and $P^{\Sigma}_{S,T}(\kappa)$ be the IRFC and GC objects, respectively, determined by the set of seeds S and T, while using the same affinity/cost function κ (with $\kappa(c,d) \geq 0$). For $m > 0$ let κ^m be the mth power of κ, that is, $\kappa^m(c,d) = (\kappa(c,d))^m$. In [13] it was proved that, under some assumptions, the objects $P^{IFT}_{S,T}(\kappa^m)$ and $P^{\Sigma}_{S,T}(\kappa^m)$ converge to the same set, as m goes to ∞. Notice also that, by [19, theorems 3 and 5], $P^{IFT}_{S,T}(\kappa^m) = P^{IFT}_{S,T}(\kappa)$ for every $m > 0$, since function x^m is increasing. We can use this fact in our version of this convergence theorem from [13]:

THEOREM 4.2. *Let $S, T \subset C$ be such that $\theta = \mu^C(S,T) < 1$. Then, there exists a number m_0 such that $P^{\Sigma}_{S,T}(\kappa^m)$ belongs to the family $\hat{\mathcal{P}}_\theta(S,T) = \{P \in \mathcal{P}(S,T)\colon \varepsilon^{\max}(P) = \theta\}$ for every $m > m_0$. In particular, if $\hat{\mathcal{P}}_\theta(S,T)$ has only one element, then $P^{\Sigma}_{S,T}(\kappa^m) = P^{IFT}_{S,T}(\kappa)$. Moreover, m_0 can be expressed by a formula $m_0 = \log_\delta N$, where $\delta = \min\{q > 1\colon q = \eta/\hat{\theta}$ for some $\eta, \hat{\theta} \in Z\}$, N is the size of α, and Z is the range of κ in $\mathcal{C}$, i.e., $Z = \{\kappa(c,d)\colon \langle c,d\rangle \in \alpha\}$.*

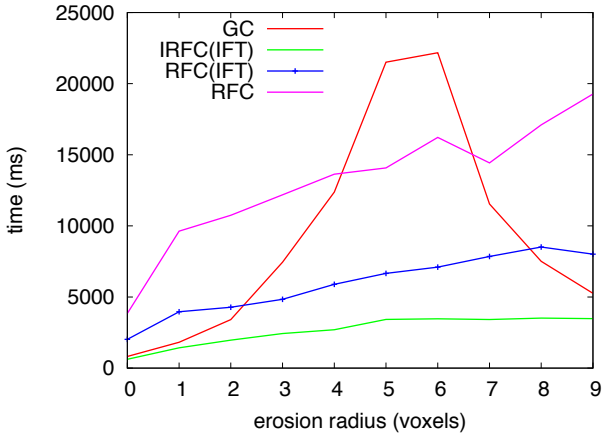
5. EXPERIMENTAL COMPARISON OF FC AND GC ALGORITHMS

In this section we describe the experiments that were designed to verify and demonstrate the main differences between FC and GC delineation algorithms discussed in Section 4.1: speed and robustness of FC versus GC and the GC shrinking problem.

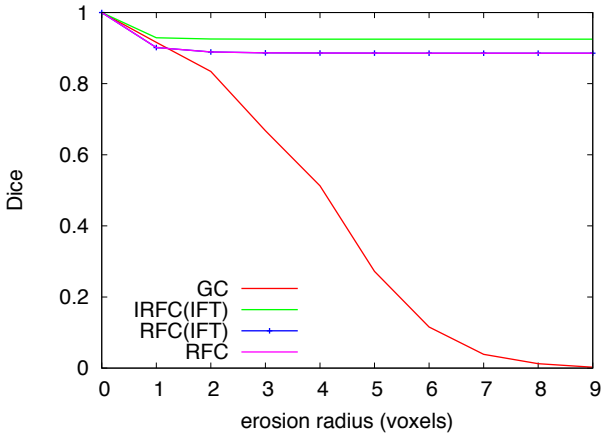
We compared four algorithms: GC using the min-cut/max-flow algorithm [11]; RFC algorithm, standard "old-fashioned" implementation [1]; RFC-IFT, implemented by using the IFT approach [13]; IRFC-IFT, iterative RFC algorithms, which iteratively refine the choice of output among all energy minimizers from $\mathcal{P}(S,T)$, [4, 13]. The last two algorithms are described in detail in Section 3.

Simulated MR image phantom data from the BrainWeb repository pertaining to 20 different normal patient anatomies were utilized for our evaluation. We used the T1 data sets, since separation of white matter (WM) and grey matter (GM) tissue regions is less challenging in these images than in images of other protocols such as T2 or PD. The parameters for the simulated T1 acquisition were as follows: spoiled FLASH sequence with TR=22ms and TE=9.2ms, flip angle = 30°, voxel size = $1 \times 1 \times 1\text{mm}^3$, noise = 3%, and background non-uniformity = 20%. In these simulated data sets, true segmentations are known, since the simulations were done starting with known anatomy. In the experiments we used a PC with an AMD Athlon 64 X2 Dual-Core Processor TK-57, 1.9 GHz, 2×256 KB L2 cache, and 2 GB DDR2 of RAM.

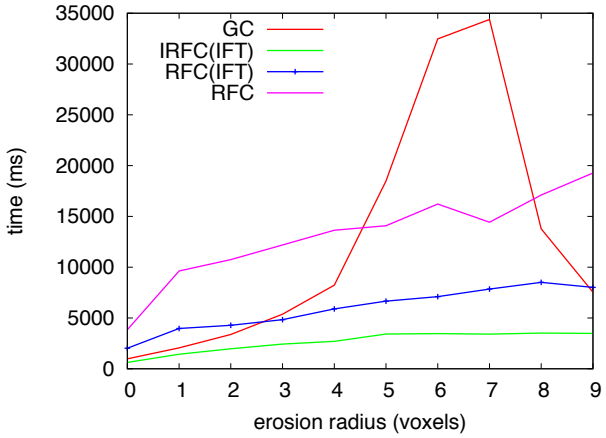
The affinity function $\kappa(c,d)$ was defined as follows. Each given image $I = \langle \mathcal{C}, f\rangle$ was filtered by a Gaussian function G_k with mean μ_k and standard deviation σ_k, where $k \in \{\text{WM,GM}\}$, separately for WM and GM, to produce two new images $I_k = \langle \mathcal{C}, f_k\rangle$, where $f_k(c) = G_k(f(c))$ for any $c \in C$. Parameters μ_k and σ_k were chosen appropriately separately for WM and GM. For each image I_k, the appropriate tissue k was segmented by using each algorithm. For the GC algorithm, the weight function $w(c,d) = (\kappa(c,d))^m$ was used with $m = 1, 5$, and 30. The rationale for this choice of w was that, according to Theorem 4.2 (compare also [13]), the output of GC converges to the output of IRFC when m goes to infinity. Changing m does not influence the output of FC algorithms [19, theorems 3 and 5]. Thus, for large m, the outputs of these two forms of algorithms should



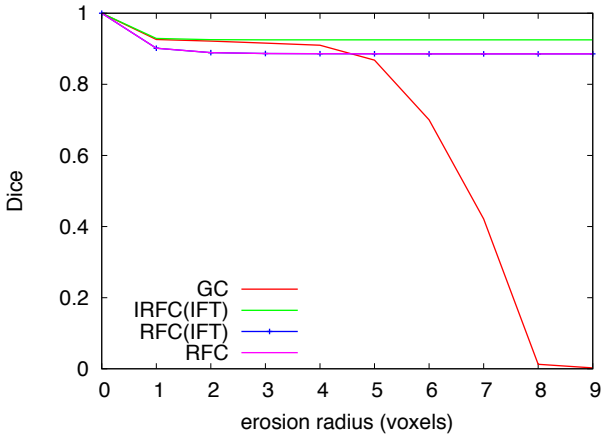
(a) Running time for $w(c,d) = (\kappa(c,d))^m$ with $m = 1$



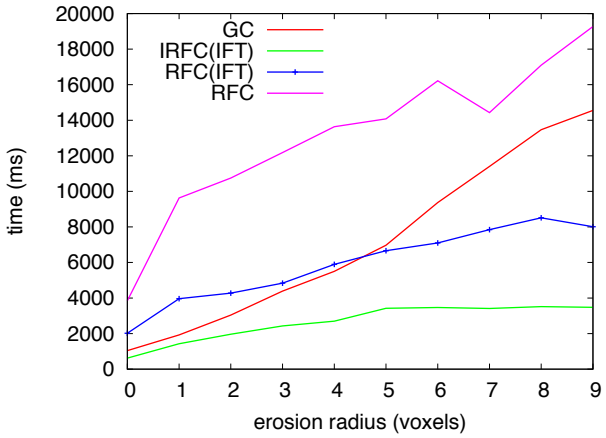
(b) Accuracy for $w(c,d) = (\kappa(c,d))^m$ with $m = 1$



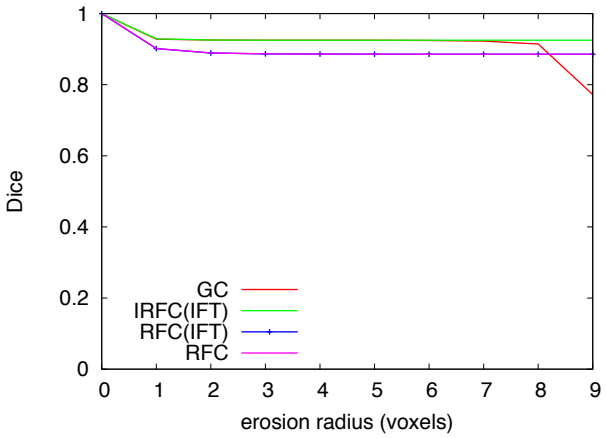
(c) Running time for $w(c,d) = (\kappa(c,d))^m$ with $m = 5$



(d) Accuracy for $w(c,d) = (\kappa(c,d))^m$ with $m = 5$



(e) Running time for $w(c,d) = (\kappa(c,d))^m$ with $m = 30$



(f) Accuracy for $w(c,d) = (\kappa(c,d))^m$ with $m = 30$

Figure 1. Time and accuracy graphs for segmenting WM for GC algorithm and FC methods

be similar. Different sets of seeds were generated, and fed automatically as input to algorithms, by applying different degrees of erosion operations to the known true segmentation binary images. We express the degree of erosion by erosion radius — the larger the radius, the greater is the degree and smaller is the seed set. As the radius of erosion increases, we may therefore expect lower delineation accuracy.

The graphs in Figure 1 summarize our experimental results. (Only the graphs for WM are shown. The graphs for GM are similar.) Figures 1(a), (c), and (e) display the run time of each algorithm, as function of the erosion radius, averaged over the 20 images for $m = 1, 5$, and 30, respectively. Similarly, Figures 1(b), (d), and (f) demonstrate the accuracy of the algorithms expressed in Dice coefficient as a function of the erosion radius, for $m = 1, 5$, and 30, respectively.

As to the *efficiency* of delineations, as described earlier, the theoretical worst case run times for GC, standard RFC, RFC-IFT, and IRFC-IFT are $O(n^{2.5})$ (or $O(n^3)$), $O(n^2)$, $O(n)$ (or $O(n \ln n)$), and $O(n)$ (or $O(n \ln n)$), respectively. These are borne out and supported by the graphs in Figures 1(a), (c), and (e). The time curve of GC should be properly interpreted in conjunction with its accuracy curve. It shows an unstable time behavior — the peak represents the high computational cost of GC and its drop off denotes the shrinkage problem or finding small cuts when larger erosions (and hence small seed sets) are used that lead to highly inaccurate segmentations. in many situations in these data sets we need to specify seed sets that are close to the true segmentation.

As to the *robustness* of delineations, GC is sensitive to the position of the seed sets, not just their size, as it has a bias toward small boundaries unlike the FC family. This implies that, in an interactive setup, its precision may suffer owing to intra- and inter-operator variability in seed specification. One way to circumvent this problem is to increase the value of m, which will bring the accuracy of GC close to that of IRFC. However, this will add to the computational cost of GC.

As to the *accuracy* of delineations, it is clear from Figures 1(b), (d), and (f) that, as the seed set becomes smaller, the delineations by GC become less accurate, mainly due to the shrinkage issue. Even when $m = 30$, this effect is noticeable.

6. CONCLUDING REMARKS

Focusing on FC and GC algorithms, we have presented a unified mathematical theory for describing these approaches as energy optimization methods. We have taken a graph and topological approach to present these combinatorial optimization analyses. The unifying treatment has helped us in delineating the similarities and differences among these methods. We have also demonstrated the forecast theoretical behavior via experiments conducted on the 20 BrainWeb MR image data sets.

The results demonstrate that, while the theoretical underpinning for GC and FC are similar, the subtle differences that remain between them are vitally responsible for their different behavior. The major differences are the dependence of GC's results on the size and position of the seed set compared to a relative independence of FC's results of these parameters. Traceable exactly to those characteristics, GC suffers in computational efficiency (time and storage), precision (repeatability), and accuracy compared to FC algorithms. Also due to these characteristics, there is a complex interplay among GC's efficiency, precision, and accuracy.

REFERENCES

1. J.K. Udupa and S. Samarasekera, Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing* **58**(3) (1996), 246–261.
2. P.K. Saha and J.K. Udupa, Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Comput. Vis. Image Understand.* **82**(1) (2001), 42–56.
3. P.K. Saha and J.K. Udupa, Iterative relative fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. In *Proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, Hilton Head, South Carolina 2002, 28–35.
4. K.C. Ciesielski, J.K. Udupa, P.K. Saha, and Y. Zhuge, Iterative Relative Fuzzy Connectedness for Multiple Objects, Allowing Multiple Seeds. *Comput. Vis. Image Understand.* **107**(3) (2007), 160–182.
5. Y. Zhuge, J.K. Udupa, and P.K. Saha, Vectorial scale-based fuzzy connected image segmentation. *Comput. Vis. Image Understand.* **101** (2006), 177–193.

6. Y. Boykov, O. Veksler, and R. Zabih, Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.* **23**(11) (2001), 1222–1239.
7. Y. Boykov and V. Kolmogorov: Computing geodesics and minimal surfaces via graph cuts, International Conference on Computer Vision, I: 26-33, 2003.
8. Y. Boykov and V. Kolmogorov: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision, IEEE Trans. Pattern Anal. Mach. Intell., 26:1124-1137, 2004.
9. Y. Boykov and G. Funka-Lea: Graph Cuts and Efficient N-D Image Segmentation, International Journal of Computer Vision, 70: 109-131, 2006.
10. Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong: An Integral Solution to Surface Evolution PDEs via Geo-Cuts, International Conference on Computer Vision, LNCS 3953, vol.III, 409-422, 2006.
11. Y. Boykov and O. Veksler: Graph cuts in vision and graphics: Theories and applications. In: *Handbook of Mathematical Models in Computer Vision*, Springer-Verlag (2006), 79–96.
12. J. Shi and J. Malik: Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell., 22:888-905, 2000.
13. P.A.V. Miranda and A.X. Falcão: Links Between Image Segmentation based on Optimum-Path Forest and Minimum Cut in Graph, Journal of Mathematical Imaging and Vision, 35:128-142, 2009.
14. K.C. Ciesielski and J.K. Udupa: "Region-based segmentation: fuzzy connectedness, graph cut, and other related algorithms," *Recent Advances in Biomedical Image Processing and Analysis*, Springer-Verlag, 2011.
15. S. Beucher, The watershed transformation applied to image segmentation. In: *10th Pfefferkorn Conf. Signal and Image Processing in Microscopy and Microanalysis* (1992), 299–314.
16. J. Park, J. Keller: Snakes on the Watershed, *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**:1201-1205, 2001.
17. R. Malladi, J. Sethian, and B. Vemuri: Shape modeling with front propagation: A level set approach, IEEE Trans. Pattern Anal. Mach. Intell., 17:158-175, 1995.
18. P.K. Saha, J.K. Udupa, and D. Odhner, Scale-Based Fuzzy Connectedness Image Segmentation: Theory, Algorithms, and Validation. *Comput. Vis. Image Understand.* **77** (2000), 145–174.
19. K.C. Ciesielski and J.K. Udupa, Affinity functions in fuzzy connectedness based image segmentation I: Equivalence of affinities, *Comput. Vis. Image Understand.* **114** (2010), 146–154.
20. K.C. Ciesielski and J.K. Udupa, Affinity functions in fuzzy connectedness based image segmentation II: Defining and recognizing truly novel affinities, *Comput. Vis. Image Understand.* **114** (2010), 155-166.
21. A.X. Falcão, J. Stol, and R.A. Lotufo: The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **26(1)**, 19-29 (2004).
22. K.C. Ciesielski and J.K. Udupa, A general theory of image segmentation: level set segmentation in the fuzzy connectedness framework. Medical Imaging 2007: Image Processing, SPIE Proceedings 6512, 2007.