# Duality between the Watershed by Image Foresting Transform and the Fuzzy Connectedness Segmentation Approaches

Romaric Audigier*, Roberto Lotufo
School of Electrical and Computer Engineering
State University of Campinas – UNICAMP
C.P. 6101, 13083-852, Campinas (SP), Brazil
audigier@dca.fee.unicamp.br, www.dca.fee.unicamp.br/∼lotufo

## Abstract

*This paper makes a rereading of two successful image segmentation approaches, the fuzzy connectedness (FC) and the watershed (WS) approaches, by analyzing both by means of the Image Foresting Transform (IFT). This graph-based transform provides a sound framework for analyzing and implementing these methods. This paradigm allows to show the duality existing between the WS by IFT and the FC segmentation approaches. Both can be modeled by an optimal forest computation in a dual form (maximization of the similarities or minimization of the dissimilarities), the main difference being the input parameters: the weights associated to each arc of the graph representing the image. In the WS approach, such weights are based on the (possibly filtered) image gradient values whereas they are based on much more complex affinity values in the FC theory. An efficient algorithm for both FC and IFT-WS computation is proposed. Segmentation robustness issue is also discussed.*

## 1 Introduction

This paper deals with two successful methods of semi-automatic segmentation. On the one hand, the segmentation by *fuzzy connectedness* (FC) comes from the fuzzy set theory and was proposed by Udupa and Samarasekera [20] and has been improved [10, 17, 18, 19] and is still being developed [22]. It is based on the hangingness among elements within an object. So, this region-based technique detects "similarity" between pixels and return quite "homogeneous" regions. Each object is represented by a fuzzy set where a membership degree is associated to each element.

On the other hand, the segmentation by *watershed* (WS) comes from the mathematical morphology theory [8]. First proposed by Beucher and Lantuéjoul [4], the watershed transform has been successfully applied to image segmentation and successive variants and improvements of the original paradigm have been proposed [5, 6, 11, 12, 13, 14, 15, 16, 21]. The WS approach detects frontier (the watershed lines) between regions of little dissimilarity (the catchment basins). It can be viewed as a computation of influence zones and achieved by successive morphological dilations. The most famous implementations [14, 21] adopt a flooding strategy of the image viewed as a topographic surface.

The WS can be obtained by other paradigms like the Image Foresting Transform (IFT). First introduced by Lotufo and Falcão [11], this graph-based paradigm can model a WS transform by computing a forest of minimum-cost paths. It returns segmented regions, the trees, where the "dissimilarity" between each node was "minimized".

The main contribution of this paper is to show the duality between the IFT-WS and the FC segmentation methods, and give an algorithm that solves both problems. The following concepts make the duality possible: (1) the interpretation of WS as a graph optimization problem in IFT framework [11] (strictly speaking, the other WS transforms cannot be seen like dual of FC methods); (2) the necessary existence of tie-zones [2] in this optimization problem; and (3) the use of symmetric arc weights in the IFT-WS. This last condition is apparently not satisfied [11]. However, we show in this paper that the asymmetric arc weight generally used in IFT-WS can be seen as a symmetric one. Consequently, the robustness properties demonstrated in the FC framework [19] are valid for the IFT-WS approach too.

The paper is organized as follows. The IFT notations, definitions and algorithms are recalled in section 2. Section 3 presents the WS in the IFT framework, possible symmetric arc weights and a new algorithm for IFT-WS that takes into account the tie-zones. Section 4 presents the FC segmentation, shows the duality between the FC and the IFT-WS approaches and discusses the segmentation robust-

*R. Audigier is funded by CAPES.

ness issue. Note that due to space limitation, the duality is presented without all the necessary demonstrations. An extended paper with the required formalism is in preparation.

# 2 The Image Foresting Transform

The IFT is a general framework based on graph theory in which an image is interpreted as a graph and pixels as its nodes. To simplify, we call *pixel* or *spel* any spatial element: point, pixel or voxel. The key idea of this transform is to obtain, according to a path-cost function, a *shortest path forest* from an input image-graph. Depending on the path-cost function and some other input parameters (adjacency, arc weights), the IFT can compute different image processing operations [9]: distance transforms, connected filters, interactive object delineation ("live-wire") and both WS and FC segmentations as shown in sections 3 and 4.

## 2.1 Notation and definitions

Under the IFT framework, an *image* is interpreted as a weighted graph $G = (V, A, I)$ consisting of a set of *nodes* or *vertices* $V$ that represent the image pixels, a set of arcs $A$ and an *intensity map* $I$, that is, a function from $V$ to some scalar or vectorial domain (e.g., $\mathbb{Z}$ in the case of grayscale digital images, $\mathbb{Z}^3$ for RGB color images). The intensity of vertex $v$ is denoted by $I(v)$.

An *adjacency relation* $A$ is a binary irreflexive relation between vertices of $V$ (e.g., 4− or 8−adjacency in 2D and 6− or 26−adjacency in 3D). The set of vertices adjacent to or neighbor of vertex $u$ is denoted by $N(u)$. If vertices $u$ and $v$ are adjacent, then the *arc* $\langle u, v\rangle$ belongs to $A$.

A *path* $\pi(u, v)$ from a node $u$ to a node $v$ in a graph $(V, A, I)$ is a sequence $\langle u = v_1, v_2, \ldots, v_n = v\rangle$ of nodes of $V$ such that $\forall i = 1 \ldots n-1$, $\langle v_i, v_{i+1}\rangle \in A$. A path is said *simple* if all its nodes are different from each other. A path with terminal node $v$ is denoted by $\pi_v$. The path $\pi_v$ is *trivial* when it consists of a single voxel $\langle v\rangle$. Otherwise, it can be defined by a path resulting from the concatenation $\pi_u \cdot \langle u, v\rangle$ of its longest prefix $\pi_u$ with terminus $u$ and the last arc $\langle u, v\rangle \in A$. A *path-cost function* $f$ assigns to each path $\pi$ a path cost $f(\pi)$, in some totally ordered set of cost values, whose maximum element is denoted by $+\infty$.

A graph $G' = (V', A')$ is *subgraph* of $G$ if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$. A *forest* $F$ of $G$ is an acyclic subgraph $F$ of $G$. *Trees* are connected components of the forest.

Let $S \subseteq V$ be a set of particular nodes $s_i$ called *seeds*. For a given weighted graph $G = (V, A, I)$ and a set $S$ of seeds, the *Image Foresting Transform* returns a forest $F$ of $G$ such that (i) there exists for each node $v \in V$ a unique and simple path $\pi(s_i, v)$ in $F$ from a seed node $s_i \in S$ to $v$ and (ii) each such path is *optimum*, i.e., has a minimum cost for linking $v$ to some seed of $S$, according to the specified path-cost function $f$.

In other words, the IFT returns a shortest (cheapest in fact) path forest where each minimum-cost tree is rooted to a seed (but possibly not all seeds are roots of a tree). In this paper, the shortest-path forest is also referred to as an *optimum-path forest* or an *optimal forest*. In general, there may be many paths of minimum cost leading to a given pixel and then, many optimal forests; only the path costs are uniquely defined.

Observe that, if we independently pick an optimum path for each pixel, the union of those paths may not be a forest. Moreover, as claimed in [9], certain graphs and cost functions may not even admit any optimum-path forest. Sufficient condition for the existence of the IFT is to use a *smooth* path-cost function, i.e., a function $f$ satisfying the three following conditions, given that for any pixel $v \in V$, there is an optimum path $\pi_v$ ending at $v$ which either is trivial, or has the form $\pi_u \cdot \langle u, v\rangle$: (i) $f(\pi_u) \leq f(\pi_v)$; (ii) $\pi_u$ is optimum, and (iii) for any optimum path $\pi'_u$ ending at $u$, $f(\pi'_u \cdot \langle u, v\rangle) = f(\pi_v)$.

Usually, the path cost depends on local properties of the image–such as color, gradient, and pixel position– along the path. Two popular examples of smooth functions are the *additive* path-cost function $f_{sum}$ and the *max-arc* path-cost function $f_{\max}$, defined by:

$$\begin{aligned} f_{sum}\left(\langle v\rangle\right) &= h(v) \\ f_{sum}\left(\pi_u \cdot \langle u, v\rangle\right) &= f_{sum}(\pi_u) + w(u, v) \end{aligned}$$

$$\begin{aligned} f_{\max}\left(\langle v\rangle\right) &= h(v) \\ f_{\max}\left(\pi_u \cdot \langle u, v\rangle\right) &= \max\left\{f_{\max}(\pi_u), w(u, v)\right\} \end{aligned} \quad (1)$$

where $\langle u, v\rangle \in A$, $h(v)$ is a fixed but arbitrary handicap cost for any paths starting at pixel $v$, and $w(u, v)$ is a fixed nonnegative weight assigned to the arc $\langle u, v\rangle$.

## 2.2 Algorithms

The efficient ordered queue-based algorithm for IFT proposed in [9, 11] is essentially Dijkstra's algorithm [7], extended for multiple sources and a more general path-cost function. It computes three attributes for each vertex $v \in V$: its *predecessor* $P[v]$ in the optimum path, the *cost* $C[v]$ of that path from the tree root to $v$, and the corresponding *root label* $L[v]$.

Note that other algorithms are able to compute the IFT. For example, the ordered queue is not necessary. One can process the image data in raster-scan and anti raster-scan order alternatively until stability of the result (algorithm not presented here and based on Berge's one [3]).

## 3 Watershed segmentation under the IFT framework

The watershed approach is often compared to a flooding simulation. If the pixel intensity represents the altitude, the image corresponds to a topographic surface. Holes are punched at some marked places. Then, the topography is inundated in water. Water springs from the holes (markers) and create catchment basins corresponding to objects. Watershed lines are dams built for separating the catchment basins growing from different sources (markers).

In order to define objects using the IFT, we assign a distinct label to each object and select at least one seed pixel per object (including background). After that, the IFT outputs an optimum-path forest where each object is represented by a set of trees rooted at seeds with the same label.

The path-cost function should be such that pixels of a given object are "more strongly" connected to its internal seeds than to any other. A suitable example is the function $f_{\max}$ when the $w(u, v)$ is a *dissimilarity* function between $u$ and $v$, usually computed based on properties of the input scene $I$. Ideally, function $w(u, v)$ must be higher on the object boundaries and lower inside the objects.

### 3.1 Watershed transforms by IFT

Many watershed transforms can be obtained by IFT according to the arc weight function. For example, the so-called *watershed by dissimilarity* [11] uses a symmetric arc weight function: $w_1(u, v) = |I(u) - I(v)|$.

Other example is the *IFT-watershed on gradient* [9, 11]. In this watershed transform, the arc weight that should be equal to some gradient intensity is: $w_2(u, v) = G(v)$, where $G(v)$ is the *morphological gradient* [8] of image $I$ at pixel $v$. This algorithm simulates the flooding by the use of an ordered queue, and looks like the Beucher and Meyer's ordered queue algorithm [5, 14]. With this arc weight function, the max-arc path-cost function of Equation 1 can be simplified into:

$$f_{\max}(\langle v_1, v_2, \ldots, v_n \rangle) = \max\{h(v_1), G(v_2), \ldots, G(v_n)\} \quad (2)$$

In this case, the IFT computes a region watershed transform where the trees of the forest, i.e., the disjoint sets of the forest, correspond to the catchment basins. Observe that all vertices are covered by this forest and no watershed line is returned, but only regions. Note also that the final cost map is unique and corresponds to the *morphological superior reconstruction* [8] of the gradient image from the seeds using a flat structuring element. However, the forest and then the labeling may be multiple. Indeed, the IFT can result in many optimal forests because many paths of minimum cost are sometimes possible.

It may be desirable to deal with symmetric arc weight, as in the watershed by dissimilarity. Instead of using the previous asymmetric arc weight, we can use the symmetric weight $w_3(u, v) = \max\{G(u), G(v)\}$. In this case, the path-cost function of Equation 1 becomes:

$$\begin{aligned} f_{\max}(\langle v_1, v_2, \ldots, v_n \rangle) &= \max\{h(v_1), \max\{G(v_1), G(v_2)\}, \\ &\quad \ldots, \max\{G(v_{n-1}), G(v_n)\}\} \\ &= \max\{h(v_1), G(v_1), \ldots, G(v_n)\} \quad (3) \end{aligned}$$

To get equivalent results for watersheds on gradient with asymmetric and symmetric arc weights (Equations 2 and 3), we can define the handicap function as: $h(s) = G(s)$ for any seed $s$ and $h(v) = \infty$ elsewhere. The role of the handicap function can be found in [12].

### 3.2 Tie-zone watershed

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image-graph. The *tie-zone watershed* (TZWS), introduced by Audigier, Lotufo and Couprie [2], returns a unique partition by definition. Indeed, this novel watershed paradigm avoids the introduction of bias due to a specific implementation [1]. We recall the definition for completeness of the paper.

A node is included in a specific catchment basin $CB_i$ when it is linked by a path to a same seed $s_i$ in all the optimal forests, otherwise it is included in the tie-zone $T$:

$$CB_i = \{v \in V, \ \forall F \in \Phi, \ \exists \pi(s_i, v) \text{ in } F\} \quad (4)$$

$$T = V \setminus \bigcup_i CB_i \quad (5)$$

where the set of all optimal forests is denoted by $\mathbf{\Phi}$.

Considering all the optimal forests, the nodes of the tie-zone can be linked to different seeds by paths of equal minimum cost. The TZWS labeling solution is defined without ambiguity because it synthesizes all the possible labelings due to optimal forests.

The algorithm presented in [2] and based on Dijkstra's computes the TZWS in the case of a *lexicographic path-cost function*. This function has two components of decreasing priority: the max-arc path-cost function and the distance to the flat zone border. This lexicographic function has the effect of reducing part of the tie-zone that occurs on flat-zones. In the case of the max-arc path-cost function without lexicographic component, the cited algorithm cannot manage the flat zones correctly and cannot return the TZWS.

We propose here two algorithms for computing the TZWS in the case of the max-arc path-cost function without lexicographic second component. First, it is possible to obtain the TZWS from any optimal forest by means of a sequential algorithm. Scan all the pixels and for each pixel, try to propagate each neighbor's cost to it. If the proposed cost is equal to the previous one and if the pixels have different labels, then assign the pixel to the tie-zone. Repeat this sequential scanning until stabilization of the labeling.

The second algorithm (cf. Pseudocode 1) is based on Dijkstra's and utilizes a *union-find* strategy to deal with the flat zones. Indeed, the Dijkstra-IFT algorithm without union-find would not manage the following case: when two pixels of different labels meet together in a flat zone of the cost map, the entire flat zone should belong to the tie-zone. But the ordered queue used does not handle already processed pixels. It is why the union-find is added to the Dijkstra-IFT algorithm: each pixel needs to point at the root of the flat zone it belongs to. When pixels meet in a flat zone (lines 15-19) and form a tie-zone, it is sufficient to merge ("union", line 19) their respective flat zone roots ("find") and label the resulting root as a tie-zone. When the ordered queue is empty (each pixel has been processed only once), a scanning of the entire image is necessary (line 22) to get the final labeling.

**Pseudocode 1: TZWS algorithm by union-find**

*Inputs*: image $(V, A, I)$, neighborhood $N$ (derived from $A$), seeds $S$, handicap $h$ (infinity for non-seed nodes), labeling $\lambda$ and arc weight $w$ functions.
*Outputs*: cost $C$, label $L$, predecessor $P$ maps.
*Auxiliary Data*: empty ordered queue $Q$, state flag $done$, cost variable $c$, flat zone root map $R$.

```
1.∀p ∈ V,  C[p] ← h(p);  R[p] ← p;  done(p) ← FALSE;
2.∀p ∉ S,  L[p] ← NIL;  P[p] ← NIL;
3.∀s ∈ S,  L[s] ← λ(s);  P[s] ← s;  Insert(s, Q, C[s]);

4.while IsEmpty(Q) = FALSE,
5.   v ← RemoveMin(Q);  done(v) ← TRUE;
6.   if L[Find(P[v])] = TZ,  L[v] ← TZ;
7.   ∀p ∈ N(v),  and  done(p) = FALSE,
8.        c ← max {C[v], w(v, p)};
9.        if c < C[p],
10.            C[p] ← c;  L[p] ← L[v];  P[p] ← v;
11.            if p in Q,  Remove(p, Q);
12.            Insert(p, Q, c);
13.            if C[p] = C[v],  R[p] ← Find(v);
14.       else, if c = C[p],
15.            if C[p] = C[v],  /* flat zone */
16.              if Find(p) ≠ Find(v),
17.                 if L[Find(p)] ≠ L[Find(v)],
18.                    L[Find(v)] ← TZ;
19.                 R[Find(p)] ← Find(v);  /* union */
20.            else, if L[Find(p)] ≠ L[Find(v)],
21.              L[Find(p)] ← TZ;  L[p] ← TZ;

22. ∀p ∈ V, L[p] ← L[Find(p)];

Find(v):  /* with path compression */
while R[R[v]] ≠ R[v],  R[v] ← Find(R[v]);
return R[v];
```

# 4 Fuzzy Connectedness Segmentation under the IFT Framework

First, we report the key ideas of the fuzzy connectedness (FC) approach [20] and relate to the IFT formalism (section 4.1). Different FC-based segmentation methods are presented (section 4.2). The segmentation robustness is discussed in section 4.3. Finally, we illustrate the concepts presented (section 4.4).

The segmentation by FC aims to create fuzzy sets from an image to represent the objects. In this case, a fuzzy set $\mathcal{O}$ is a set $O$ of spatial elements (called *spels*; i.e. points, pixels or voxels depending on the data dimension) assigned by the degree of belongingness (membership) $\mu_{\mathcal{O}}$ to an object. This degree is a real number between 0 and 1. When it is 0, it means that the spel does definitely not belong to the object. When it comes close to 1, it certainly belongs to the object. In case of multiple object segmentation, a degree of belongingness to each object can be assigned to each spel. And, the final object extraction can be obtained by several ways: by applying a minimum threshold to this degree or by simply assigning the pixel to the object with highest degree.

The idea of the FC is that the objects correspond to sets of close spels (in intensity and position), i.e. spels that are "more strongly connected" or have a "greater fuzzy connectedness". In general, seeds representative of each object are given as an input of the method that returns sets of spels hanging together with the seeds.

## 4.1 Key ideas on fuzzy connectedness

In this section, we sum up several basic concepts of FC approach that are fully developed in [19] and reinterpret them from the IFT point of view.

A *fuzzy digital space* is defined as $(Z^n, \alpha)$ where $Z^n$ is a set of spels in an $n$-dimensional space (but it could be any set of elements) and $\alpha$ is a *fuzzy spel adjacency*. The adjacency is a reflexive and symmetric relation. It associates to each pair of spels a value $\mu_\alpha$ bounded by 0 and 1 and generally given by a nonincreasing function of their Euclidean distance. Therefore, a spel is adjacent to itself and has an adjacency value of 1 (reflexivity). And any two spels whose fuzzy adjacency value is not zero are said *adjacent*. Note that the adjective "fuzzy" is used when a real value bounded by 0 and 1 is assigned to a spel of a set or to a pair of spels.

Like IFT, FC also uses the concept of vertices (spels) and arcs if we consider that –by the symmetry of the adjacency– there exists an undirected arc between any two adjacent (or neighbor) spels. Note that the arcs are weighted by the adjacency value that only depends on spatial configuration between spels. But this positive weight does not matter –once we know that it is positive– if we are only interested in scanning all neighbors of a spel.

FC methods are based on two fuzzy relations:
• A local fuzzy relation $\kappa$ called *affinity* on the set of spels. The affinity value $\mu_\kappa$ assigned to a pair of spels is based on the nearness of spels in space and in intensity (or in features derived from intensities). Affinity represents local "hanging togetherness" of spels.
• A global fuzzy relation $K$ called *connectedness*, on the set of spels, based on affinity $\kappa$. The connectedness value $\mu_K$ assigned to a pair of spels is the strength of the strongest path linking these spels. The strength of a path is the lowest affinity along the path.

The affinity is a reflexive and symmetric fuzzy relation whose valuing function may depend on: (i) the fuzzy adjacency (spatial distance) $\alpha$ between the spels; (ii) the homogeneity $\psi$, whose value is bounded by 0 and 1 and is greater when both spels belong to the same homogeneous region; (iii) intensity-based features $\phi$, (when the intensity feature is closer to an expected value of the features for a given object); (iv) the actual location of the spels, a shift-variant value not used in practice. A typical functional form for the affinity value $\mu_\kappa$ is: $\mu_\kappa = \mu_\alpha \sqrt{\mu_\psi \mu_\phi}$ where $\mu_\alpha$ is 1 for the hard $4-$ or $6-$adjacency relation. Note that $\mu_\psi$ and $\mu_\phi$ are evaluated in the neighborhood of each pixel and that neighborhoods may depend on space-variant local scales. Details of the design of such affinity functions to correctly represent the nearness or similarity between spels are given in [18].

The affinity value between two spels can be viewed as the weight of a virtually added arc shared by these spels. Imagine now that, instead of valuing the similarity between spels, we value their dissimilarity (with the complement or inverse of the affinity value for example). This dual arc weight can be seen as a cost or penalty to pixels that are not sufficiently near. It is exactly the case of the arc weight used in IFT. For the special case of WS by IFT, this weight is given by some gradient function (i.e., heterogeneity measure) of the image to segment. And for non-adjacent vertices, no arc is considered or, virtually, the arc has an infinite weight.

In FC methods, the *strength of a path* is defined as the minimum affinity value of any pair of spels in the path (see the so-called "fuzzy $\kappa-$net" from [19] that assigns this strength to each possible path). In IFT framework, a *path-cost* is assigned to each path. In the case of WS by IFT, this cost corresponds to the maximum arc, which is exactly the dual of the path strength.

Associated to a given affinity $\kappa$, the fuzzy connectedness $K$ assigns to a pair of spels the value $\mu_K$ defined as the maximum path strength when all paths linking these two spels are considered. Dually, in the IFT case, paths of minimum cost are computed.

The duality between FC and IFT-WS concepts is summarized in the upper part of Table 1. Correspondences between main notations used in IFT-WS and in [19] are given. The lower part shows the dual segmentation methods presented in the next section. Note that all the FC (and IFT-WS) methods have variants if other affinities (arc weights) are used (e.g., a scale-based affinity, other gradient).

**Table 1. Duality in concepts and segmentation methods.**

| Fuzzy connectedness (FC) | Watershed (WS) by IFT |
|---|---|
| fuzzy spel affinity $\mu_\kappa(c,d)$ | arc weight $w(c,d)$ |
| path $p_{cd}$: $\langle c=c_1,c_2,\ldots,c_L=d\rangle$ | path $\pi(c,d)$: $\langle c=c_1,c_2,\ldots,c_L=d\rangle$ |
| path strength: $\min_{1<l\leq L}\mu_\kappa(c_{l-1},c_l)$ | (max-arc) path cost: $\max_{1<l\leq L} w(c_{l-1},c_l)$ |
| strength of FC $\mu_K(c,d)$: $\max_{\forall p_{cd}}$ {path strength} | optimal path cost $C[d]$: $\min_{\forall\pi(c,d)}$ {path cost} |
| FOE (threshold $\theta_{\min}$) | reconstruction (threshold $\theta_{\max}$) (IFT cost map) |
| RFOE (strict max FC) | independent reconstructions from each seed (strict min cost) $\Longrightarrow$ *CORES* |
| IRFOE (iterative blocking) | TZWS: one synchronous IFT (seeds compete together) $\Longrightarrow$ *CATCHMENT BASINS* |

## 4.2 Segmentation by fuzzy connectedness

### 4.2.1 Fuzzy object extraction by connectedness thresholding

Now FC has been defined, many fuzzy object extractions are possible. Initially, Udupa and Samasekera [20] proposed a fuzzy object extraction, FOE, based on a threshold $\theta$. For a given object seed $o$, after computing for each spel the FC with the seed, set a minimum threshold on the strength of connectedness to obtain the set of object spels. The *fuzzy $\kappa\theta-$object* $\mathcal{O}$ is defined by:

$$\begin{aligned} \mathcal{O} &= (O_{\theta_\kappa},\ \mu_{\mathcal{O}}) \\ O_{\theta_\kappa} &= \{v\in V,\quad \mu_K(o,v)\geq\theta\} \qquad (6) \\ \mu_{\mathcal{O}}(v) &= \begin{cases}\eta(I(v)), & \text{if } v\in O_{\theta_\kappa},\\ 0, & \text{otherwise}\end{cases} \qquad (7)\end{aligned}$$

Note that the extracted object spels have an associated value given by a function $\eta$ of their intensity while the remaining spels have value 0. In the case of a *hard* segmentation (by *fuzzy* connectedness), the function $\eta$ is a constant (1 for binary segmentation or the object label $\lambda$ for N-ary segmentation, e.g.). Observe also that the strength of connectedness between any two object spels is greater or equal than the threshold value; and the strength of connectedness between an object spel and any other spel out of the object is less than the threshold.

In IFT context, FOE is equivalent to (i) computing the IFT from a unique object seed on the "dual graph" (composed by the same vertices and arcs but with "dual" weight–complement to 1, e.g.), then (ii) setting the "dual" maximum $\theta_{\max}$ threshold on the cost map (i.e., the superior reconstruction of the image) and finally (iii) applying the function $\eta$ to the selected spels if a fuzzy object is expected.

### 4.2.2 Relative fuzzy object extraction

Udupa, Saha and Lotufo [19] proposed then another object extraction method: the relative fuzzy object extraction (RFOE). In this method, strength of connectedness to the object and also to all co-objects are considered. Indeed, the objects are let to compete among themselves in having spels as their members. The spels will belong to the object that has the highest strength. Therefore, the object definition depends on how the spels hang together among themselves relative to others. This method does not need any threshold.

The *relative fuzzy* $\kappa-$*object* $\mathcal{O}$ is defined as the set of spels more strongly connected to the object seed $o$ than to the background (or co-object) seed $b$:

$$\begin{aligned} \mathcal{O} &= (P_{ob_\kappa},\ \mu_{\mathcal{O}}) \\ P_{ob_\kappa} &= \{v \in V,\quad \mu_K(o,v) > \mu_K(b,v)\} \\ \mu_{\mathcal{O}}(v) &= \begin{cases} \eta(I(v)), & \text{if } v \in P_{ob_\kappa}, \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

In the IFT framework, the RFOE is dually equivalent to independently compute IFT-WS from each seed $t \in S$ (one-by-one) and finally assign to each pixel $v$ the label $\lambda(s)$ corresponding to the seed $s$ that links it with a *strict* minimum path-cost $C_s[v]$, considering all cost maps $C_t$ (superior reconstructions). Each labeled region we obtain is called *core* $K_s$ (just as in [19]) of the seed $s$:

$$K_s = \{v \in V,\quad C_s[v] < C_t[v],\quad \forall t \in S,\quad t \neq s\} \tag{9}$$

where $C_s$ corresponds to the superior reconstruction of the image from seed $s$. Note that there can be pixels with no label. They do not belong to any object because there are many paths of same (non-strict) minimum cost linking to different seeds.

So, instead of using the RFOE algorithm of [19], we can follow this procedure: (i) For the extraction of $n$ objects (background included), compute $n$ superior reconstructions of the "dual" image-graph from seeds of different objects. (ii) Then compute the strict minimum of the reconstructions at each pixel. The cores are obtained.

### 4.2.3 Iterative relative fuzzy object extraction

We saw that some spels can have same connectedness with two different objects and therefore do not belong to any object. Sometimes the path, where the connectedness to an object was computed, passes through the other object (core). For the RFOE method, these two paths tie together. In fact, one of these paths invades the other object for trying to link a disconnected region to the core it comes from. The idea of the iterative relative fuzzy object extraction (IRFOE) proposed in [19] is to avoid this "passing through" by *blocking* any paths that try to pass through an object but come from other objects. This is exactly the same concept implicitly used in IFT. Indeed, the blocking is inherent to the forest of optimal paths: when a node is assigned to a particular tree, it cannot be used by any other growing tree.

The effect of the IRFOE is to expand iteratively the initial objects, the cores, obtained by RFOE, by reducing the chances of ties thanks to the blocking strategy. Nevertheless, not all ties are untied because some regions are linked to many seeds with same connectedness but without any path overlapping. Under the IFT framework, they correspond to the tie-zone defined in Equation 5 while the objects defined by IRFOE correspond to the CBs of Equation 4.

In practice, the IRFOE algorithm extracts first the cores and at the second iteration, any path from other seeds passing through a core is penalized by a low connectedness value (zero). In fact, affinity between any spel of an object obtained at iteration $i$ is automatically set to zero at the next iteration $j$. This constitutes a barrier to protect the objects of iteration $i$. In other words, pixels that have been conquered in previous iteration cannot be used by paths from other object seeds in the subsequent iterations.

It is shown that the objects iteratively defined are non-contracting: object of iteration $i < j$ is included in object of iteration $j$. Moreover, the objects maintain their disjoincy at every iteration. Observe that these properties are valid in the TZWS-IFT case: the cores are always included in the catchment basins and these are disjoint.

Thus, alternatively to the IRFOE presented in [19], the algorithm of TZWS by union-find proposed in section 3.2 finds the optimal paths without overlapping in an ordered way and returns directly the catchment basins without needing iterative object extraction.

### 4.2.4 Variants of the fuzzy connectedness approach

First, note that the RFOE of [19] extracts only one object relative to a background (the co-objects). It must theoretically be applied $n$ times if $n$ objects have to be segmented. The generalization of this method was proposed in [17]: the multiple relative fuzzy object extraction (MRFOE) processes many objects together. Let $S$ be the set of object seeds and $o$ a particular seed. The set of seeds different from $o$ is denoted by $b(o) = S\backslash\{o\}$. Therefore, the definition of relative fuzzy object in Equation 8 becomes:

$$P_{ob(o)_\kappa} = \{v \in V,\quad \mu_K(o,v) > \mu_K(o',v),\quad \forall o' \in b(o)\}$$

This does not change anything in the correspondence with multiple reconstructions where strict minima among them determine which object is better linked to a spel.

In addition, the MRFOE method [17] allows individual affinity for each object. Indeed, one drawback of the RFOE method is that the same affinity relation must be used for different objects. “This restriction somewhat compromises the effectiveness of the segmentation that can be achieved” [17]. On the other hand, if different affinities are used for different objects, then most of the properties we report in the next section will not hold. It is why the MRFOE method combines the multiple object affinities in a single affinity so that the theoretical results are valid as well as more effective practical segmentation is achieved.

Saha and Udupa [18] proposed scale-based affinity functions to improve the segmentation results of the fuzzy connectedness methods. Zhuge, Udupa and Saha [22] have recently introduced the vectorial scale-based relative fuzzy object extraction (VSRFOE) where the method of MRFOE with scale-based affinities is generalized from scalar images to vectorial images.

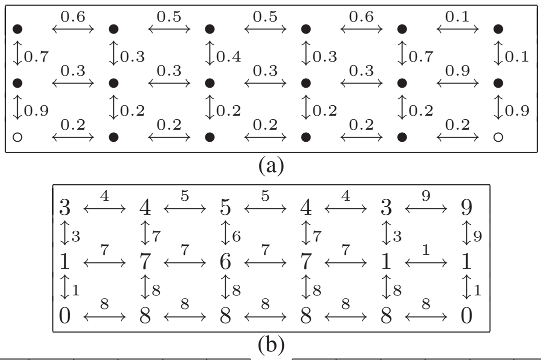
Finally, we can cite the approach of Herman and Carvalho [10] that proposes a multiseeded segmentation by FC. This method is somewhat different from those reported before because multiple affinities are indeed allowed and spels may belong to many objects with same maximum FC.

## 4.3 Robustness of the segmented objects

A series of useful properties and theorems are demonstrated in [19] and are valid for symmetric affinities (like in RFOE) or arc weights (like $w_1$ or $w_3$ of section 3.1). We recall only two of them and translate them in IFT-WS terms.

Any pair of spels in the defined object is linked by a best path entirely contained in the object. This property (Prop.3.2 of [19]) is also valid in the case of IFT-WS for both cores and catchment basins because the trees that represent them are composed of optimal paths by definition.

More interestingly, Theorem 3.4 of [19] guarantees the *robustness* of the objects (iteratively defined by the IRFOE) with respect to their respective seeds. Indeed, if the user designates other object seeds but still in their respective cores, the same objects will be obtained at any iteration of the IRFOE. Observe that the seeds must belong to the cores to guarantee the robustness of the objects. In IFT-WS framework, the robustness of the cores and the catchment basins is also guaranteed as long as the seeds are picked in the respective cores and, obviously, the max-arc path-cost function and a symmetric arc weight function are used. Note that the lexicographic cost (used in [2, 11]) could not be used because it would untie flat zones (of the tie-zone) differently if seeds were chosen in different places, and so, robustness would not be valid anymore.



| | | | | | |
|---|---|---|---|---|---|
| $\underline{\mathbf{3}}$ | $\underline{\mathbf{4}}$ | *5* | $\underline{\mathbf{4}}$ | $\underline{\mathbf{3}}$ | **9** |
| $\underline{\mathbf{1}}$ | *7* | *6* | *7* | $\underline{\mathbf{1}}$ | $\underline{\mathbf{1}}$ |
| $\underline{\mathbf{0}}$ | *8* | *8* | *8* | *8* | $\underline{\mathbf{0}}$ |

(c)

| | | | | | |
|---|---|---|---|---|---|
| **3** | **4** | *5* | **4** | **3** | **9** |
| $\underline{\mathbf{1}}$ | *7* | *6* | *7* | $\underline{\mathbf{1}}$ | $\underline{\mathbf{1}}$ |
| $\underline{\mathbf{0}}$ | *2* | *2* | *2* | *2* | $\underline{\mathbf{0}}$ |

(d)

$\underline{\textbf{cores}}$ - **catchment basins** - *tie-zone*

**Figure 1. Images and object extractions.**

## 4.4 Illustration

Figure 1(a) presents an example of image with fuzzy affinities between the spels “•”. Two seeds “○” were picked. Figure 1(b) shows a gradient image with the respective symmetric arc weights $w_3$ (section 3.1). We could also see the weights of the arcs of this image-graph as the “dual” weights of those from image (a). In this example, we applied $w(c, d) = \lceil k(1 - \mu_\kappa(c, d)) \rceil$ with $k = 10$ to compute the complement of the affinity and get an integer dissimilarity measure. Figure 1(c) is the output of the TZWS algorithm. The tie-zone is in italic whereas the two catchment basins (CB) are boldfaced. They correspond to two seeds placed in “0”. But two any other seeds chosen in the respective cores (underlined) would lead to the same cores and CBs (robustness). Disregarding the numbers, (c) corresponds also to the segmentation of (a) (duality). Observe the “blocking” applied by the right-hand core on the paths to the spel “9” coming from the left-hand seed and having an equal cost of 9. Figure 1(d) presents the segmentation of another image, which differs from (b) only in the flat zone of “8”. The cores are reduced to 2 and 3 spels while the CBs are the same. Now, the blocking is also applied for paths passing by the flat zone “2” and going to “3” and “4”.

# 5 Conclusion

In this paper, we talked about two successful segmentation methods. We showed the duality between the watershed by image foresting transform (IFT-WS) and the fuzzy connectedness (FC) segmentation approaches using the IFT framework. The IFT, an optimal forest computation, is a sound framework for understanding both. As the IFT-WS method finds the paths that link spels of an object by minimizing the dissimilarity among them, the FC method finds the paths that link spels of an object by maximizing the similarity (connectedness) among them. The methods solve dual problems.

Only parameters of the problems really differ because the methods were historically developed in different contexts. For example, FC works on fuzzy sets (fuzzy adjacency and fuzzy segmented object) that are in general not used in IFT-WS but could be. The outstanding difference of parameters is the design of the arc weight functions (IFT-WS) and their analog affinity functions (FC). In WS approach, the image is generally filtered and a gradient or dissimilarity operator is applied. In the FC approach, no preprocessing is applied but, instead, complex affinity functions determine the arc affinity. Examples of the sophistication of such affinity functions are given in [18] where the affinity can be scale-based. Observe that these sophistications could be modeled and integrated in the WS approach as preprocessing steps for computing other arc weights than the usual.

In addition, we saw that the duality is established when the WS is used in its IFT formulation with tie-zone (TZ). The tie-zone, that includes spels that have same connectedness/cost to at least two objects, must be taken into account to guarantee a unique segmentation. So, the new efficient algorithm of TZWS with union-find is another contribution of this paper: it can compute both FC and WS segmentation methods.

As a last contribution, we showed that the usual asymmetric arc weight for IFT-WS can be viewed as a symmetric one, so that all properties demonstrated for FC –in particular, the segmentation robustness– hold for IFT-WS too. In future, we think that both research domains –FC and WS– can benefit from the other one's advance using the duality.

## References


[1] R. Audigier and R. Lotufo. Tie-zone watershed, bottlenecks and segmentation robustness analysis. In *XVIII Brazilian Symp. on Comp. Graph. and Image Proc. (SIBGRAPI'05)*, pages 55–62, Natal, Brazil, Oct. 2005. IEEE Press.

[2] R. Audigier, R. Lotufo, and M. Couprie. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of IEEE Int. Conf. on Image Processing (ICIP'05)*, volume 2, pages 654–657, Genova, Italy, Sept. 2005.

[3] C. Berge. *Théorie des graphes et ses applications*. Dunod, Paris, France, 1958.

[4] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.

[5] S. Beucher and F. Meyer. The Morphological Approach to Segmentation: The Watershed Transform. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, Inc., New York (NY), USA, 1993.

[6] M. Couprie and G. Bertrand. Topological grayscale watershed transformation. In *SPIE Vision Geometry V Proceedings*, volume 3168, pages 136–146, 1997.

[7] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959.

[8] E. Dougherty and R. Lotufo. *Hands-on Morphological Image Processing*. SPIE – The International Society for Optical Engineering, Bellingham (Washington), USA, Aug. 2003.

[9] A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):19–29, Jan. 2004.

[10] G. T. Herman and B. M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(5):460–474, 2001.

[11] R. Lotufo and A. Falcão. The Ordered Queue and the Optimality of the Watershed Approaches. In *5th International Symposium on Mathematical Morphology*, pages 341–350, Palo Alto (CA), USA, June 2000. Kluwer Academic.

[12] R. Lotufo, A. Falcão, and F. Zampirolli. IFT-watershed from gray-scale marker. In *XV Brazilian Symp. on Computer Graph. and Image Proc.*, pages 146–152, Fortaleza (CE), Brazil, Oct. 2002. IEEE Computer Society.

[13] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, (38):113–125, 1994.

[14] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Comm. and Image Repr.*, 1(1):21–46, 1990.

[15] L. Najman and M. Schmitt. Watershed of a continuous function. *Signal Processing*, 38(1):99–112, 1994.

[16] J. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1-2):187–228, Jan. 2000. Special issue on mathematical morphology.

[17] P. Saha and J. Udupa. Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82(1):42–56, April 2001.

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

[19] J. Udupa, P. Saha, and R. Lotufo. Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11):1485–1500, November 2002.

[20] J. K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graph. Models Image Process.*, 58(3):246–261, May 1996.

[21] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, 1991.

[22] Y. Zhuge, J. Udupa, and P. Saha. Vectorial scale-based fuzzy-connected image segmentation. *Computer Vision and Image Understanding*, 101(3):177–193, March 2006.