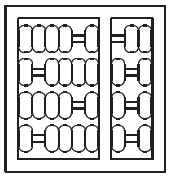

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

# IFT–Watershed From Gray-Scale Marker

*R.A. Lotufo* *A.X. Falcão* *F. Zampirolli*

Technical Report - IC-02-012 - Relatório Técnico

September - 2002 - Setembro

# IFT–Watershed From Gray-Scale Marker

Roberto de A. Lotufo[*], Alexandre X. Falcão[†], Francisco Zampirolli[‡]

**Abstract**

The watershed transform and the morphological reconstruction are two of the most important operators for image segmentation in the framework of mathematical morphology. In many situations, the segmentation requires the classical watershed transform of a reconstructed image. In this paper, we introduce the *IFT–watershed from gray scale marker* - a method to compute at the same time, the reconstruction and the classical watershed transform of the reconstructed image, without explicit computation of any regional minima. The method is based on the *Image Foresting Transform* (IFT) - a unified and efficient approach to reduce image processing tasks to a minimum-cost path forest problem in a graph. As additional contributions, we demonstrate why other reconstruction algorithms are not watersheds, and present a family of simple, yet efficient IFT-based watershed algorithms: classical, watershed from labeled marker, from binary marker with on-the-fly labeling, and from gray scale marker with on-the-fly labeling and bounded cost.

**keywords:** watershed transform, image foresting transform, graph-search algorithms, minimum-cost path problems, image segmentation, image analysis, morphological reconstruction.

## 1 Introduction

We have developed a method, called *image foresting transform* (IFT) [1], to reduce image processing problems to a minimum-cost path forest problem in a graph. This approach provides a mathematically sound common framework for many image-processing operations, often with considerable efficiency gains over published algorithms. It establishes the correctness of many algorithms that have been published and used without proof; and offers guidelines for modifying and generalizing existing operators without loosing their properties. Thus new image operators can be created by changing the parameters of the IFT and/or by simple modifications in its general algorithm. Examples of successful applications of the IFT are boundary tracking, watershed transforms, morphological reconstructions, multi-scale skeletonization, fast binary morphological operations, multi-scale shape filtering, Euclidean distance transform, geodesic path computation, multi-scale fractal dimension, and shape

[*]lotufo@dca.fee.unicamp.br,Faculty of Electrical and Computing Engineering, State University of Campinas, Campinas, SP, Brazil

[†]afalcao@ic.unicamp.br,Institute of Computing, State University of Campinas, Campinas, SP, Brazil

[‡]fz@dca.fee.unicamp.br,Faculty of Electrical and Computing Engineering, State University of Campinas, Campinas, SP, Brazil

salience extraction [1, 2, 3, 4, 5]. In this paper, we propose a variant of the IFT–watershed from labeled markers presented by Lotufo and Falcão [2], called the *IFT–watershed from gray scale marker*.

The classical watershed transform of an input image $I$ usually creates an over-segmented partition, where the catchment basins are influence zones of the regional minima of $I$, as pointed by Beucher and Meyer [6]. In order to reduce the number of catchment basins in the segmentation, we usually apply to $I$ a morphological reconstruction before the watershed transform. In [2], we proposed the IFT–watershed from labeled markers, which computes the watershed-from-markers transform without requiring minima imposition (see Figure 1). We presented in [3], the relation between watershed-from-markers transform and morphological reconstruction. In this paper, we demonstrate that the cost map of the IFT is identical to the output of the gray scale image sup-reconstruction. As a consequence, we show that the proposed method computes at the same time the morphological reconstruction and the classical watershed transform of the reconstructed image, without explicit computation of regional minima (see Figure 2). There are interesting advantages of this approach as compared to the current classical watershed approach. It is conceptually solid with a higher abstraction level, where the marker is specified by a gray scale image; it is faster and simpler by combining many steps in a single algorithm; and it does not require explicit computation of any regional minima [1].

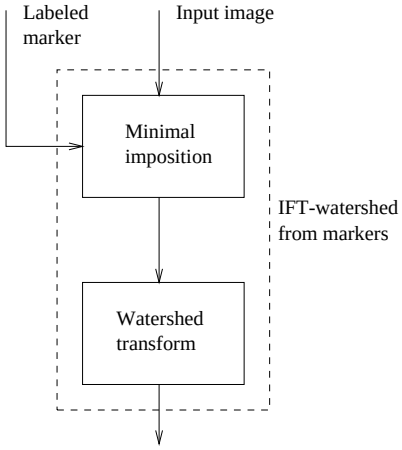


Figure 1: The IFT–watershed from markers does not require minima imposition.

In order to illustrate the advantages of using the IFT to the design of image operators, particulary watershed transforms, this paper presents a family of simple and efficient variations of the IFT-watershed transform: (i) the watershed from labeled markers [2], which is being used here to demonstrate that the cost map of the IFT is identical to the output of the gray scale image sup-reconstruction; (ii) the watershed from gray scale marker, the new proposed approach; (iii) the classical watershed, as a particular case of the watershed from gray scale marker; (iv) the watershed from gray scale marker with on-the-fly labeling

[1] A shorter version of this article has been published at [7]. The present version includes new algorithms, applications, and experiments for evaluation in 2D and 3D.

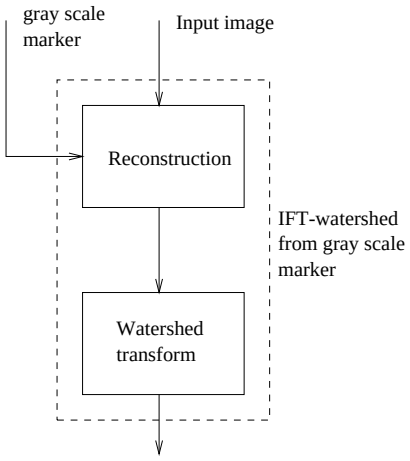


Figure 2: The IFT–watershed from gray scale marker computes the reconstruction and the watershed transform of the reconstructed image, simultaneously.

of catchment basins and bounded cost; and (v) the watershed from binary marker with on-the-fly labeling of catchment basins. Particularly (iv) is a very useful approach to avoid cost and label propagation in non-relevant regions of the image. This algorithm is used to cut touching binary blobs in the negation of the Euclidean distance transform and provides considerable speed-up gains over the traditional method. The paper also presents illustrative applications of (ii) and (iv); evaluates the efficiency gains of the IFT-based algorithms (i)-(v) in 2D and 3D, using as reference the traditional approach based on immersion simulations [12]; and demonstrates why other reconstruction algorithms are not watersheds.

This paper is organized as follows. We first review the IFT–watershed from labeled marker (Section 2) and demonstrate that its *cost map* is identical to the output of the sup-reconstruction algorithm (Section 3). We then introduce the IFT–watershed from gray scale marker and its efficient algorithm in Section 4, and present a further improvement to this algorithm for on-the-fly labeling of the catchment basins with bounded cost. We summarize all variations of the presented IFT-watershed, giving time measurements and illustrative examples in 2D and in 3D. In Section 5, we demonstrate why other reconstruction algorithms are not watersheds. The conclusions are stated in Section 6.

## 2 The IFT–watershed from Labeled Markers

Many problems in image processing can be interpreted as an image partition problem based on a given set of *root pixels*, where each root defines an *influence zone* consisting of the pixels that are "more closely connected" to that root than to any other, in some appropriate sense. The *image foresting transform* (IFT) reduces such problems to the computation of a *minimum-cost path forest* in a directed graph whose nodes are the image pixels and whose arcs are defined by an *adjacency relation* $N$ between pixels. We denote by $N(p)$ the set of pixels adjacent to the pixel $p$ (i.e. its neighbors in the graph).

The cost of a path in this graph is determined by a suitable *path-cost function* $f$, which usually depends on local image properties along the path — such as color, gradient, and pixel position. For suitable path-cost functions, one can choose the minimum-cost paths so that their union is an oriented forest, spanning the whole image. The nodes of a given tree in the forest are by definition the influence zone of the corresponding root.

The result of the IFT is an *annotated image*, that assigns to each pixel three attributes: its predecessor in the optimum path, the cost of that path, and the corresponding root. Its solution is usually obtained in linear time and requires a variant of the Dijkstra/Moore/Dial's shortest-path algorithm [8, 9, 10].

In this framework, the watershed-from-markers [6] can be computed by a single IFT [2], without minima imposition, where the labeled markers are root pixels. In this case, the root map can be replaced by a label map which corresponds to the catchment basins. The path-cost function $f_m$ is given by:

$$\begin{aligned} & f_m(< p_1, p_2, \ldots, p_n >) = \\ & = \begin{cases} max\{I(p_1), I(p_2), \ldots, I(p_n)\} & \text{if } p_1 \text{ is a marker pixel} \\ +\infty & \text{otherwise,} \end{cases} \end{aligned} \tag{1}$$

where $p_{i+1} \in N(p_i)$ and $I(p_i)$ is the value of the pixel $p_i$ in the image $I$.

The *IFT–watershed from labeled markers* may be simplified to the following algorithm:

**Algorithm 1** *Watershed from labeled markers using hierarchical FIFO queue (HFQ).*

INPUT *I: input image,*
*L: labeled marker image;*
OUTPUT *L: watershed catchment basins;*
AUXILIARY *C: cost map, initialized to infinity;*

INITIALIZATION
1. **for all pixels** $L(p) \neq 0$
2. $C(p) \leftarrow I(p)$, *insert* $p$ *in HFQ with cost* $C(p)$;

PROPAGATION
3. **while** *HFQ not empty*
4. $p \leftarrow$ *remove from HFQ;*
5. **for each** $q \in N(p)$
6. **if** $C(q) > \max\{C(p), I(q)\}$
7. $C(q) \leftarrow \max\{C(p), I(q)\}$;
8. *Insert* $q$ *in HFQ with cost* $C(q)$;
9. $L(q) \leftarrow L(p)$;

Two points are worth discussing in the above algorithm. Note that the pixels in the queue are never re-evaluated as required by the original Dijkstra/Moore/Dial algorithm [8, 9, 10]. This was explained by Lotufo and Falcão in [2] and the reason is that all the incident arcs to a node have the same weight. The other particularity of this algorithm is the use

of a FIFO in the hierarchical queue. This is desirable if one wants to have the inclusion of a slight modification of the path-cost to make the catchment basin boundaries be in the medial line of plateaus regions in the image. The FIFO property of the hierarchical queue can be modeled by a lexicographic path-cost function as pointed out in [2].

## 3 Morphological Reconstruction

Morphological reconstruction is a powerful operator in the framework of mathematical morphology. It is the building block of connected operator design. In this paper, we will be interested in the sup-reconstruction, also called dual reconstruction, which is related to the watershed. The *gray scale sup–reconstruction* of image $I$ from marker image $J$, $J \geq I$ is given by:

$$\phi_J(I) = \epsilon^{\infty}_{I,B}(J), \tag{2}$$

where $\epsilon_{I,B}(J) = \epsilon_B(J) \vee I$ is the geodesic erosion, $\epsilon_B(I)$ is the gray scale erosion of $I$ by the structuring element $B$ and $\vee$ is the pointwise maximum operator. The composition $\psi^n(I)$ is the successive application of the operator $n-1$ times: $\psi(\psi(\ldots(\psi(I))))$. The sup-reconstruction is a successive sequence of erosions of $J$ by $B$, pointwise maximum with $I$ until stability.

An important property, proved by Vincent [11], is that the sup-reconstruction of $I$ from $J$ is the same as sup-reconstruction of $I$ from $R_J$, where $R_J$ is given by:

$$R_J(p) = \begin{cases} J(p) & \text{if } p \text{ is regional minima of } J, \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

The *regional minima* $M$ of a gray scale image is the largest connected component of pixels with same gray scale (plateau), such that every neighbor pixel of $M$ has a strictly higher value.Fig 3 shows the relationship between $R_J$ and $J$.

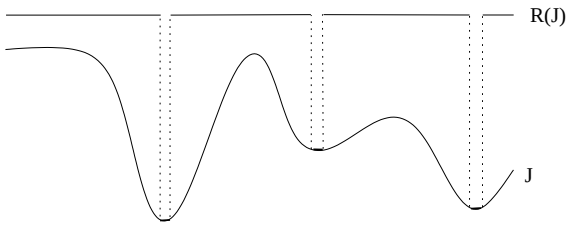


Figure 3: Functions $R_J$ and $J$.

Vincent [11] has described several morphological reconstruction algorithms. The one based on a FIFO queue is reproduced below.

**Algorithm 2** *Grayscale reconstruction using a queue.*

INPUT *I: gray scale mask image,*
*J: gray scale marker image,* $J \geq I$*;*

OUTPUT *J: sup-reconstruction of I from J*

INITIALIZATION
1. *Compute $R_J(p)$ by Eq. 3;*
2. *Insert the pixels in the regional minima of J in queue;*

PROPAGATION
3. **while** *queue not empty*
4. $p \leftarrow$ *remove from queue;*
5. **for each** $q \in N(p)$
6. **if** $J(q) > J(p)$ *and* $I(q) \neq J(q)$
7. $J(q) \leftarrow \max\{J(p), I(q)\}$;
8. *Insert q in queue;*

Note that algorithms 1 and 2 are very similar. We claim that the cost map $C$ of algorithm 1 is the sup-reconstruction of $I$ from any gray scale marker function $J$ such that:

$$R_J(p) = \begin{cases} I(p) & \text{if } L(p) \neq 0, \\ \infty & \text{otherwise.} \end{cases} \tag{4}$$

We can state:

**Proposition 1** *The cost map C in the algorithm 1 is the morphological reconstruction of I from any marker image J such that $J \geq I$ and $R_J$ is given by Eq. 4 (See Fig. 4).*

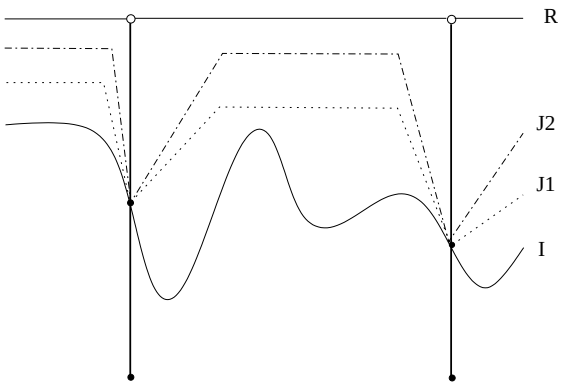


Figure 4: Functions $I$, $R_J$ and two possible $J$: $J1$, $J2$.

**Proof 1** *Both initialization algorithms are equivalent, the pixels that are inserted in the queue are the regional minima of J. In the watershed, the inserted pixels have value $I(p)$ at the markers. The others pixels not inserted in the queue have cost infinity. So, both regional minima are equivalent and inserted in the queue. According to Vincent [11], the*

*propagation order of the pixels does not matter in the reconstruction. Therefore, we can replace the FIFO in algorithm 2 by the HFQ of algorithm 1.*

*Line 6 of algorithm 2 may be rewritten as: If $J(q) > J(p)$ and $I(q) < J(q)$, since $J \geq I$. So $J(q)$ must be greater than $J(p)$ and greater than $I(q)$, which is equivalent to line 6 of algorithm 1, where $C$ plays the role of $J$.* ■

The consequence of this fact is that the morphological reconstruction, when implemented by the HFQ, can simultaneously output the result of the reconstruction and the catchment basins of $I$ that were propagated from the regional minima of the marker $J$. In the IFT framework, the output of the reconstruction is given by the cost map and the watershed catchment basins by the root map. This result is not new, as we have presented in [2] that the maximum cost function of the IFT–watershed from markers intrinsically includes the minima imposition, and recently, and in more detail, with practical illustrative examples, we showed that the watershed-from-markers and the superior reconstruction are obtained at the same time by a single IFT [3]. Here we prove this result by reducing the algorithm 1 to algorithm 2.

A more important consequence of the above conclusion is that we can show that the morphological reconstruction of $I$ from $J$, $I \leq J$, can be modeled as the cost map of an IFT, where the path-cost function $f_r$ is given by:

$$f_r(< p_1, p_2, \ldots, p_n >) = max\{J(p_1), I(p_2), \ldots, I(p_n)\}. \quad (5)$$

Note that $J(p)$ is the initial cost of any path starting at $p$ and all pixels $p$ are root candidates (we usually call them *seed pixels*). Such a marker function can be seen as an initial restriction on the maximum path cost. However, since the cost of a path from a pixel $p$ to a pixel $q$ may be less than $J(q)$, only some candidate pixels will remain as root pixels in the minimal-cost path forest. Moreover, we wish to use this formulation to compute simultaneously the morphological reconstruction and the watershed transform, without explicit computation of any regional minima. This is addressed next.

## 4 The IFT–watershed From Gray Scale Marker

We can define the *IFT–watershed of $I$ from gray scale marker $J$* as the watershed transform of $I$ from the markers given by the regional minima of $I$ sup-reconstructed from $J$.

Before presenting the IFT–watershed from gray scale marker algorithm, we will introduce a further improvement in the algorithm by avoiding the explicit regional minima detection of the marker function, which is normally used to insert the pixels in the HFQ in the initialization phase of the IFT algorithm. This is also a contribution of this work that is not published elsewhere.

The IFT–watershed from gray scale marker has the same input as the morphological sup–reconstruction: $I$ as the image and $J$ as the marker, both gray scale images. Our simplification stems from the fact that we actually do not need to compute the regional minima, but only one pixel per regional minima. This pixel will be the representative of the catchment basins (i.e. the root of a tree in the minimum-cost path forest). In the

initialization step, we set the initial root map such that all pixels are potential candidates to be root pixels. Instead of initializing the initial cost to infinity, we insert all pixels in the HFQ with the initial cost given by $J+1$ and root pointer to $p$ (i.e. $R(p) \leftarrow p$). The addition of one to the initial cost is the essential part of the technique. When a pixel never visited is removed from the HFQ, it becomes a root pixel. This pixel is the first of its regional minima (called representative), and as such, its cost value is corrected by subtracting one from the current cost. The other pixels that belong to the same regional minima of the reconstructed image will have their cost updated in the HFQ and will have their root value pointed to the representative of the regional minima. The same happes with the other pixels that belong to the same catching basins of the root pixel. Note that the root pixels are representatives of the regional minima of the sup-reconstructed image and not of the regional minima of $J$.

The technique of adding one makes the behavior of plateaus to be dealt with correctly according to the FIFO property of the HFQ, as all pixels in the HFQ will be updated given that they were inserted in the queue with a higher initial cost. This removal from and reinsertion in the HFQ for every pixel, except the representative one of each regional minima, is the price to pay for not computing the regional minima explicitly. When the HFQ is implemented efficiently, it is well worth to adopt the approach just described.

The IFT–watershed from gray scale marker algorithm is presented below.

**Algorithm 3** *IFT–watershed from gray scale marker*

INPUT *I: gray scale input image,*
*J: gray scale marker image, $J \geq I$;*
OUTPUT *J: reconstruction of I from J,*
*R: catchment basins of the watershed;*

INITIALIZATION
1. **for all pixels** $R(p) \leftarrow p$
2. *Insert all pixels p in HFQ with cost* $J(p) \leftarrow J(p) + 1$;

PROPAGATION
3. **while** *queue not empty*
4. $p \leftarrow$ *remove from HFQ;*
5. **if** $R(p) = p$ *then* $J(p) \leftarrow J(p) - 1$;
6. **for each** $q \in N(p)$
7. **if** $J(q) > \max\{J(p), I(q)\}$
8. $J(q) \leftarrow \max\{J(p), I(q)\}$;
9. $R(q) \leftarrow R(p)$;
10. *Update q in the HFQ with cost* $J(q)$;

## 4.1 Illustrative Examples

We illustrate algorithm 3 with an example of using the IFT–watershed from gray scale marker for cell segmentation.

Fig. 5a shows the original image $K$. The morphological gradient of $K$ is computed and shown in Fig. 5b, which is the input image $I$ for the watershed. Fig. 5c shows the gray

scale marker image $J$ which is computed by the area closing of $I$ by 200. This removes the number of regional minima to avoid the oversegmentation. The resultant partitioned image is shown in Fig 5d which is the watershed transform of $I$ from the gray scale marker $J$. The classical way of doing this would be first computing image $I_1$ as a reconstruction of $I$ from $J$ and then computing the watershed transform of $I_1$.

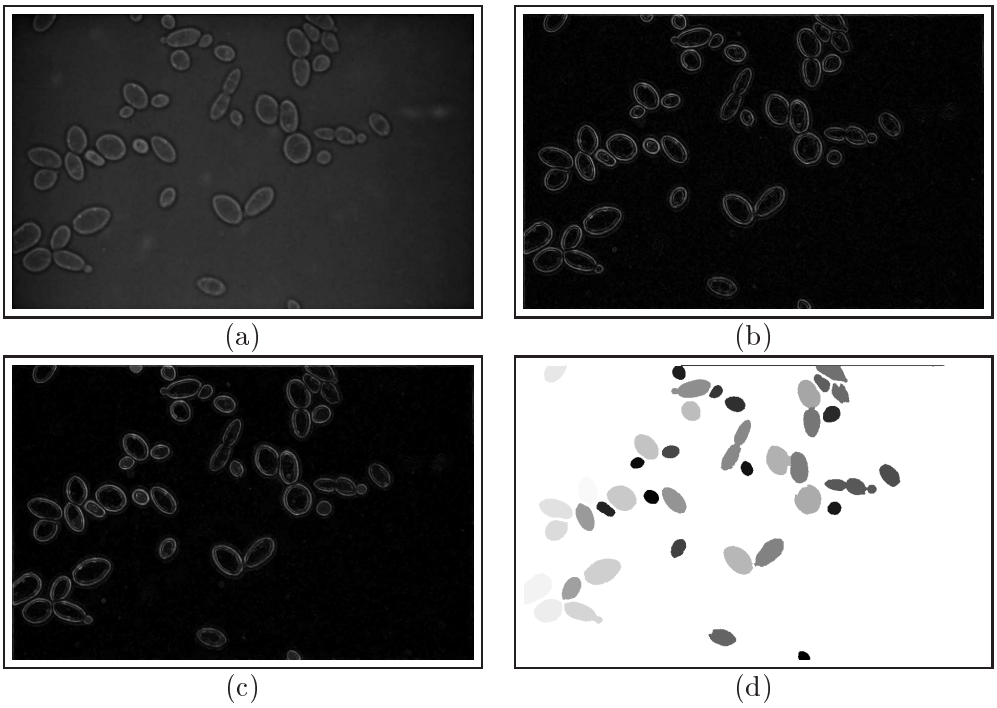


Figure 5: Watershed from gray-scale marker. a) input image, b) morphological gradient ($I$), c) marker image $J$ as the area closing of $I$, d) watershed partitions.

Another typical application of the watershed transform is to split touching cells in binary images, like some of the ones detected in the previous example. Note that there are few cells that are connected. The classical solution to split these cells is to find the watershed catchment basins of the negation of the distance transform, where the markers are associated to each cell. In the illustration shown in Fig. 6, part (a) is the input binary image of touching cells, part (b) is the inverse of the Euclidean distance transform. We compute the watershed partitions using the IFT-watershed of algorithm 3 of the negated Euclidean distance transform from its gray scale area closing as marker. The watershed partitions are shown in part (c). For illustration purpose we overlaid the binary image on the watershed partition to show the split of touching cells. Finally, in part (d) we show the intersection of the (a) and (c) with all detected cells.

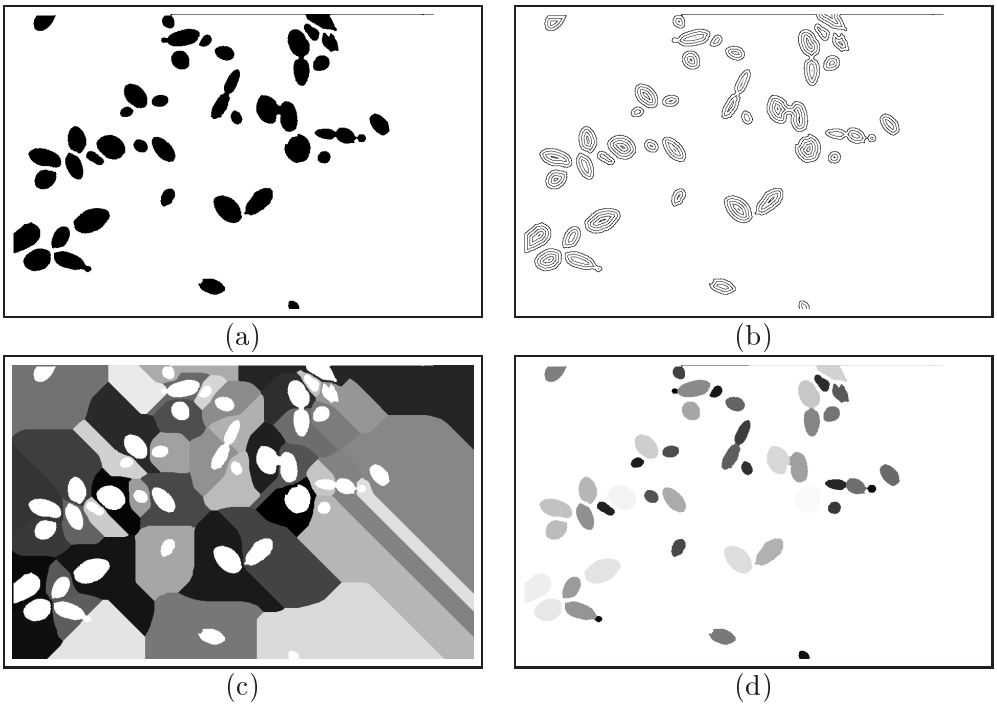


Figure 6: Binary cell segmentation a) Input binary image with touching cells, b) Inverse of the Euclidean distance transform of (a), c) watershed catchment basins of (b) from the gray scale marker used as the area closing of (b), superposed by the input binary image (a), d) intersection of the watershed catchment basins and the binary image.

### 4.2 On-the-Fly Labeling and Bounded Cost

The last application motivated us to include a further improvement to the IFT–watershed algorithm. The maximum pixel value in the negation of the distance transform is normally the maximum value in the data type used, called $k$. It is not necessary to propagate labels and costs in this region, which is outside the binary cells. This requires a bounded path-cost function:

$$f_b(< p_1, p_2, \ldots, p_n >) = \\ = \begin{cases} max\{J(p_1), I(p_2), \ldots, I(p_n)\} & \text{if } J(p_1) \text{ and } I(p_i) < k, \text{ for } i = 1, 2, \ldots, n \\ +\infty & \text{otherwise.} \end{cases} \quad (6)$$

This function can be simply and efficiently implemented by not inserting the pixels of $J$ with value $k$ in the queue. In addition to this modification, we can process the label map instead of the root map. Initially the label map is zero for all pixels. When a pixel, whose label is zero, is removed from the HFQ, it means it was not propagated yet and it is the representative of a new catchment basin, so a new label is assigned to it. In this way, the output of the watershed is the label map, which is often the desirable output of

the watershed transform. By applying these modifications to the algorithm 3, we introduce algorithm 4.

**Algorithm 4** *IFT–watershed from gray scale marker with on-the-fly labeling and bounded cost*

INPUT *I: gray scale input image,*
*J: gray scale marker image,* $J \geq I$*;*
OUTPUT *J: reconstruction of I from J,*
*L: catchment basin labels of the watershed, initialized to zero;*
AUXILIARY *tag: variable initialized to zero;*
INITIALIZATION
1. **for all pixels**
2. **if** $J(p) < k$ **then**
3. *Insert p in HFQ with cost* $J(p) \leftarrow J(p) + 1$*;*
PROPAGATION
4. **while** *queue not empty*
5. $p \leftarrow$ *remove from HFQ;*
6. **if** $L(p) = 0$ *then*
7. $J(p) \leftarrow J(p) - 1$*;*
8. $tag \leftarrow tag + 1$*;* $L(p) \leftarrow tag$*;*
9. **for each** $q \in N(p)$
10. **if** $J(q) > \max\{J(p), I(q)\}$
11. $J(q) \leftarrow \max\{J(p), I(q)\}$*;*
12. $L(q) \leftarrow L(p)$*;*
13. *Update q in the HFQ with cost* $J(q)$*;*

Using algorithm 4, the image shown in Fig. 6d is obtained directly without the computation of the full watershed partitions of Fig. 6c.

### 4.3 Watershed from Binary Marker Image

For completeness, we present another variant of algorithm 3 to compute the watershed from a binary marker using the on-the-fly labeling technique introduced in algorithm 4. In this case, the cost map ($J$) is initialized to infinity and in the initialization phase of the algorithm, only the marker pixels are inserted in the HFQ with cost one. To mimic the minima imposition, $I$ is set to zero for pixels under the marker and the zeros of $I$ are replaced by ones. The propagation phase is exactly the same as in algorithm 4, except for line 13 which is replaced by removing the pixel from HFQ only if the pixel is under the marker, followed by the insertion of the pixel in the HFQ.

**Algorithm 5** *IFT–watershed from marker with on-the-fly labeling*

INPUT *I: gray scale input image,*
*M: binary marker image;*

OUTPUT *J: reconstruction of I, initialized to infinity,*
*L: catchment basin labels of the watershed, initialized to zero;*
INITIALIZATION
*1.* **for all pixels**
*2.* **if** $M(p)$ **then**
*3.* *Insert p in HFQ with cost* $J(p) \leftarrow 1$*;*
*4.* $I(p) \leftarrow 0$*;*
*5.* **else** $I(p) \leftarrow max\{I(p), 1\}$*;*
PROPAGATION
{ *Same as in algorithm 4. except for line 13*}*:*
*13.* **if** $M(p)$ **then** *Remove q from HFQ;*
*14.* *Insert q in the HFQ with cost* $J(q)$*;*

### 4.4 Summary of IFT-watersheds

We finish this section by summarizing all variations of the presented IFT–watershed transforms:

**LM - watershed from labeled marker** : (algorithm 1) This is the classical hierarchical queue-based watershed algorithm. The important point in the IFT framework is the intrinsic sup-reconstruction of the minima imposition integrated with the watershed transform.

**BM - watershed from binary marker** : (algorithm 5) We presented a fast and simple algorithm to compute the watershed from markers with on-the-fly labeling.

**GM - watershed from gray scale marker** : (algorithm 4) This is the main contribution of this work. It combines sup-reconstruction and watershed transform as they are the same algorithm. Algorithm 4 detects and labels the catchment basins on-the-fly.

**CW - classical watershed** : (algorithm 4) If the marker image is the same as input image, the result of the watershed from gray scale marker is the classical watershed, where each regional minima is associated to each catchment basin.

We report in Table 1 the speed of each algorithm, when using 8 neighbors connectivity. The image used in the experiment is the morphological gradient of the image shown in Fig 5a which has $576 \times 902$ pixels. We compare three scenarios: in the first, the image is oversegmentated, in the second, the gray scale image is properly segmented, and the third, the binary cells is further segmented to cut touching cells. In all three applications we compare our algorithms to Vincent and Soille (VS) [12], based on immersion simulations. This algorithm is neither based on the hierarchical queue, nor requires the explicit computation of regional minima beforehand. The computer used in the experiments was a Pentium III, 750 MHz.

| Scenario 1: oversegmentation | |
|---|---|
| Vincent and Soille (VS) | 533 ms |
| algorithm 4 (CW) | 657 ms |
| algorithm 1 (LM) | 489 ms |
| algorithm 5 (BM) | 492 ms |
| Scenario 2: gray scale segmentation | |
| algorithm 4 (GM) | 627 ms |
| sup-reconstruction and VS | 764 ms |
| Scenario 3: binary segmentation | |
| algorithm 4 (GM) | 258 ms |
| sup-reconstruction and VS | 769 ms |

Table 1: Speed evaluation

In the first scenario, the image is oversegmented in 30309 catchment basins. From Table 1 we can see that the speed of algorithms 1 and 5 are a little faster than VS, despite their simplicity when compared to VS algorithm. Algorithm 4 is a little slower as it requires all pixels, except the representatives, to be inserted in the HFQ twice.

In the second scenario, we compute the watershed catchment basins of the gradient image using as gray scale marker the gray scale area closing by 200 pixels of the gradient image. The resultant image is shown in Fig. 5d. We see that our solution is faster than the classical approach to first sup-reconstruct the gradient image by the marker and then compute the watershed.

For the third scenario, the touching cells of Fig. 6a are segmented by the watershed trasform from gray scale marker, where the mask image is the negation of the Euclidean distance transform and the marker image is the result of its area closing by 6. Resultant image is shown in Fig. 6d. In this case our algorithm is about 3 times faster than the classical solution. This is due the fact the pixels are not propagated outside the binary cells because of the bounded cost implementation.

The extension of the methods discussed here to 3D or higher dimensions is straightforward. The improvements of the IFT–watershed from gray scale marker presented in algorithm 4 become more pronounced for higher dimensions. In a 3D application with touched binary spheres in an $80 \times 80 \times 80$ volume, algorithm 4 took 0.33 seconds to process while the separate sup-reconstruction and VS watershed algorithms took 5.33 seconds, a speed up factor of more than 16.

# 5 Why Other Reconstruction Algorithms Are Not Watersheds

Using the IFT framework, we have shown that the morphological reconstruction of $I$ from $J$, with $I \leq J$, is the cost map of a minimal-cost path forest with the path-cost function $f_r$ given by Eq. 5.

One of the simplest and most general shortest path algorithms is the one introduced by Berge [13], which was designed for the classical distance path-cost function $f_{sum}$:

$$f_{sum}(< p_1, p_2, \ldots, p_n >) = J(p_1) + I(p_2) + \ldots + I(p_n). \quad (7)$$

Berge's algorithm is based on a repetition until stability of the following situation that must be true in the shortest path solution. For any two neighboring nodes, $J(q) \leq J(p) + I(q)$. So this algorithm is a repetition in any scanning order of neighboring pixels of the above relation. This algorithm can also compute the root map of the forest partitioning.

**Algorithm 6** *Berge shortest path.*

INITIALIZATION
$R(p) \leftarrow p$;

**repeat** *until stability, using any scanning order*
**if** $J(q) \geq J(p) + I(q), q \in N(p)$
$J(q) \leftarrow J(p) + I(q)$;
$R(q) \leftarrow R(p)$;

For the reconstruction algorithm, we have shown that the path-cost function is given by $f_r$ which is based on the maximum pixel value in the path. Interesting is the fact that the Berge algorithm in this case can compute the cost map correctly, but not the root map.

Rewriting the Berge algorithm using the path-cost function $f_r$ and computing only the cost map, we arrived at the following algorithm.

**Algorithm 7** *Berge minimal-cost path with $f_r$.*

INITIALIZATION
$R(p) \leftarrow p$;

**repeat** *until stability, using any scanning order*
**if** $J(q) \geq \max\{J(p), I(q)\}, q \in N(p)$
$J(q) \leftarrow \max\{J(p), I(q)\}$;

which is the morphological reconstruction reported by Vincent [11] when the pixels can be updated in any scanning order.

Unfortunately, it is not possible to incorporate the root map updating in the Berge algorithm, as we can demonstrate with a simple illustrative example. Suppose we have the small numeric image illustrated in Fig. 7a. The image has two basins and each one has a non-infinity marker pixel with a small dot and a small square below the pixel value, as shown in the figure. If we scan the pixels in raster order, from left to right and up to down, using the Berge algorithm, the result of the first scan is presented in Fig. 7b. After the anti-raster scan order (left and up neighbors), note that the two top-right pixels with values 5 and 7 remained rooted at the squared dot marker, as their costs are already correct after

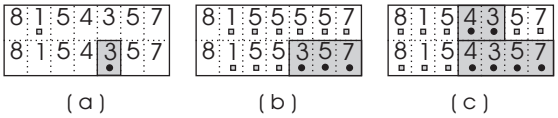


Figure 7: Berge algorithms are not watersheds: (a) initial image two markers; (b) raster scan right-down; (c) anti-raster left-up.

the first scan. Only the scanning order of the hierarchical queue propagates the root in the increasing path-cost order and as such can guarantee the correct computation of the root map in the path-cost function $f_r$.

It is possible to compute the watershed transform algorithm using the Berge algorithm only if the path-cost function is in the form of a distance function of type $f_{sum}$. This is exactly the case presented by Meyer in [14] with the definition of the topographic distance.

In summary, the morphological reconstruction and the watershed transform are well explained by the minimal-cost path forest using the $f_r$ path-cost function. The Dijkstra / Moore / Dial shortest path algorithm can compute the cost map (reconstruction) and the root map (watershed) of the minimal-cost path forest. The Berge algorithm using the $f_r$ path-cost function can only compute the cost map (reconstruction). This latter algorithm belongs to the class of the parallel reconstruction algorithms. If it is desirable to compute the watershed using the Berge algorithm because of its suitability to hardware implementation, the topographic distance must be used.

## 6 Conclusions and Comments

In this paper, we have shown several results. We had previously pointed out that the morphological reconstruction algorithm based on the hierarchical queue is also a watershed transformation [3]. In this work, we have demonstrated this result by showing the equivalence of the morphological reconstruction algorithm using a FIFO queue and the cost map of the IFT–watershed-from-markers algorithm.

We have consolidated the importance of the IFT framework, where the image is modeled by a graph and the minimal-cost path forest is computed resulting in three maps: cost, root, and predecessor maps. In this context, the morphological reconstruction is the cost map of a minimal-cost path forest using the path-cost function $f_r(p)$ of the maximum pixel values in the path and the pixel value of the gray scale marker function at $p$. We have shown that the Berge shortest path algorithm works for the computation of the minimal path-cost (reconstruction), but does not work for the computation of the root map (catchment basins), for this kind of path-cost. This is the reason why non-ordered queue based reconstruction algorithms are not watersheds.

We have introduced the IFT–watershed from gray scale marker which is a new concept where the watershed is not computed from a set of labeled markers but from a gray scale function. This concept puts together several steps used in classical watershed-based segmentation strategies in a single algorithm. We have shown that the classical watershed is a

special case of the watershed from gray scale marker.

We have presented a family of efficient and simple IFT–watershed algorithms which computes simultaneously the watershed and the sup-reconstruction and avoids the explicit computation of the regional minima of the marker function and generate on-the-fly labeling of the catchment basins.

We introduce a variation based on bounded cost where the labels are not propagated within undesirable regions, making the watershed much faster, specially in the popular application of cutting touching binary blobs. This efficiency becomes higher for high-dimensional image processing.

## 7 Acknowledgments

A.X. Falcão is grateful to CNPq (Proc. 300698/98-4) for the financial support. Francisco A. Zampirolli is supported by FAPESP Ph.D. scholarship under process n. 98/06641 − 6.

## References

[1] A.X. Falcão, R.A. Lotufo, and J. Stolfi,. The image foresting transform: theory, algorithms, and applications. submitted.

[2] R.A. Lotufo and A.X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer Academic Publishers, Palo Alto, USA, Jun 2000.

[3] A.X. Falcão, B. S. da Cunha, and R. A. Lotufo. Design of connected operators using the image foresting transform. In *Proceedings of SPIE on Medical Imaging*, volume 4322, pages 468–479, San Diego, CA, Feb 2001.

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

[5] R.S. Torres, A.X. Falcão, and L.F. Costa. Shape description by image foresting transform. In *14th International Conference on Digital Signal Processing*, Santorini, Greece, pp. 1089–1092, Jul 2002.

[6] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In Edward R. Dougherty, editor, *In Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, Inc., New York, NY, 1993.

[7] R.A. Lotufo, A.X. Falcão, and F. Zampirolli. IFT-Watershed From Gray-Scale Marker. In *SIBGRAPI-2002, Brazilian Symposium on Computer Graphics and Image Processing*, Oct 2002.

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

[9] E.F. Moore. The shortest path through a maze. In *International Symposium on the Theory of Switching Proceedings*, pages 285–292, Cambridge, Massachusetts, Apr. 1959. Harvard University Press.

[10] R.B. Dial. Shortest-path forest with topological ordering. *Communications of the ACM*, 12(11):632–633, Nov 1969.

[11] L. Vincent. Morphological grayscale reconstruction in image analysis. *IEEE Transactions on Image Processing*, 2(2):176–201, Apr 1993.

[12] L. Vincent, and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, Jun 1991.

[13] C. Berge. *Theorie des graphes et ses applications.* Dunod, 1958.

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