Analysis of the watershed algorithms based on the Breadth-First and Depth-First exploring methods

André Körbes, Roberto de Alencar Lotufo
Department of Computer Engineering and Industrial Automation - DCA
School of Electrical and Computer Engineering - FEEC
University of Campinas - UNICAMP
Campinas, São Paulo, Brazil
{korbes,lotufo}@dca.fee.unicamp.br

***Abstract*—In this paper, fifteen watershed algorithms are reviewed. For clarity, first we expose two graph exploring methods modified to be guidelines for understanding the approaches taken by these algorithms: the breadth-first watershed and the depth-first watershed. Both paradigms rely on the visiting order applied by the algorithms. The breadth-first is more recognisable as a seed region growing or marker expansion process, grouping both methods based on flooding and hierarchical queue. The depth-first groups the algorithms based on the drop of water simulation, forming a simple path until a regional minimum is found. We analyse and classify fifteen algorithms, and two of them were better characterised. Along with this, some useful information (i.e. use of markers and line over pixel) is organised, in order to facilitate the choice of an algorithm.**

***Keywords*-watershed transform, image segmentation, watershed algorithms**

I. Introduction

Introduced by Digabel and Lantuéjoul [1], the watershed transform was later used by Beucher and Lantuéjoul for contour detection [2]. These introductory works, along with Meyer and Beucher work on morphological segmentation [3] settled the concept of image segmentation based on a gradient image, where the grey levels are altitudes forming a surface with catchment basins submitted to a flooding process. When two basins touch, a barrier (a watershed line) is raised. Intuitively, these segmentation lines are points where a drop of water may slide to two different regional minima.

Many definitions and algorithms for the watershed transform exist in literature. However, in this paper, we show that most of the algorithms can be described by the two basic graph exploring methods: breadth-first and depth-first. On the watershed transform, these methods may also be seen respectively as a marker expansion or seed region growing procedure and by the drop of water simulation. These paradigms are depicted on the course of this paper, so that one may better analyse the algorithms and their motivation. We also intend to better classify them and to understand which types of algorithms are used for which problems. Also, the use of algorithm methods from graph theory for classification of the watershed algorithms described in the literature allow a better understanding and overview of these algorithms under the graph theory framework.

On the theoretical field, the watershed transform can be classified in five major definitions that outstand [4]: (1) the immersion by influence zones, that in fact is a generalisation of the SKIZ - Skeleton by Influence Zones (Flooding-WT), (2) the topographic distance watershed transform (TD-WT) and (3) its derivation on the local condition (LC-WT), and (4) the image foresting transform watershed (IFT-WT) along with (5) the tie-zone transform (TZ-IFT-WT), that formalises the shortest-path forest (SPF) with a max-arc path cost function .These five watershed transform definitions can be implemented by many algorithms. In this paper, we analysed fifteen algorithms described in the literature, pointed out its definition, similarities, differences and the way the pixels are scanned.

This paper is organised as follows: in Sec. II we present our proposal for the paradigms that embrace several other proposals, in order to achieve better understanding of them, Sec. III briefly reviews the major watershed transform definitions, Sec. IV presents fifteen watershed algorithms of the literature, exposing its definitions when the authors are not clear on it, as well as other features, such as the paradigm that fits best on the proposal and requirements for equivalence, as lower completion and seed input. Sec. V revisits the work on theoretical relationships [4] incorporating details from the algorithm analysis. Finally Sec. VI gives the conclusions and future work.

II. Algorithm Paradigms

Over the years, several algorithms of watershed have been proposed, according to different formal definitions and applying different strategies. This section introduces the two major paradigms of algorithms identified in the literature. The criterion for separating these is mainly the pixel scanning: the breadth-first and depth-first strategies. An important issue is that these paradigms are used independently from the watershed definition, that is, a particular definition can be implemented by any of the two pixel scanning paradigms.

A. Breadth-First Watershed

Breadth-first search methods are well known on computer science literature, as they are the basis of several procedures, such as Dijkstra's SPF and Prim's minimum spanning forest (MSF) graph search algorithms. It is so named because of its expansion ordering, always from the last known border and uniformly across its breadth. In the sense of distance from the original seed, all the vertices at distance k are visited before visiting any vertex at distance $k+1$. [5]

On the watershed transform algorithms field, one can see the similarity between a breadth-first search and a flooding from markers. In this sense, the first and very influential works on fast watershed transforms are those by Vincent and Soille with a simulated immersion [6] and the work on hierarchical queues by Beucher and Meyer [7]. These proposals differ on a series of features and use different definitions, but both preserve the neighbourhood visitation order, that is a modified breadth-first scanning, where the aforementioned distance k is not simply the number of edges, but is in fact dependant of the definition.

The algorithms that use hierarchical queues are special cases of a generalisation of Dijkstra's algorithm [8] for SPF, called Image Foresting Transform (IFT) using a cost function that is the maximum weight of the edges on the path [9], [10]. Another approach for the path cost function was given by Meyer [11] in the form of the topographic distance. Algorithm 1 generalises these algorithms on the breadth-first watershed.

Algorithm 1: Breadth-First Watershed

1: Define the markers to be expanded
2: Calculate the connections of the pixels on the current iteration to the previous
3: Label the pixels according to their connections. Go to step 2 with a new set of pixels (e.g. neighbours, next threshold), iterate until each pixel is visited.
4: Label pixels that are connected to multiple basins as one of them or as watershed lines

As a general case, the breadth-first watershed procedures suffer from a drawback, that is to find the regional minima, either by ordering the pixels or by scanning the image and detecting connected components. This preprocessing is a costly operation, and the watershed transform, even though becoming a simple neighbourhood processing, it is usually slower than its equivalent based on the depth-first paradigm [12]. Another type of forest, the MSF was also proposed as a watershed transform [13], and is also a special case of a SPF for the max-edge cost function, as shown in ref. [4]. When these algorithms are based on Prim's breadth-first procedure, they may be classified into this category.

B. Depth-First Watershed

As breadth-first algorithms, depth-first ones are also very common in the graph literature. The strategy is to search always from the most recently visited vertex on the graph, whenever possible, and to backtrack to explore the edges left behind. That is, in contrary to breadth-first, depth-first algorithms prioritises vertices at distance $k+1$ as soon as they are discovered, instead of visiting every vertex at distance k. [5]

As easily seen, a drop of water falling over a topographic surface imitates this procedure, once it follows a path along a surface, without the backtracking. On the watershed transform, the depth-first paradigm is the result of the evolution of the *arrowing* technique of Bieniek and Moga and the union-find technique used by Meijster and Roerdink [14]–[16]. Recently, several algorithms based on these preliminary works have been proposed, using variations of the previous procedures, achieving considerable speedups without loss of precision [12], [17]–[19]. These algorithms are all based on evaluating the neighbourhood, to identify which neighbour will be visited next, in a depth-first ordering, until a regional minimum is found, stopping the process for this pixel. The paths that end at the same minimum are all labelled the same, such as drops of water that fall into the same catchment basin. Algorithm 2 generalises these algorithms on the depth-first watershed.

Algorithm 2: Depth-First Watershed

1: Connect every pixel to its neighbours with the lowest grey level
2: Make sets containing pixels with paths that end at the same regional minimum
3: Label the remaining pixels (not in any set) as watershed lines or add them to the sets according to some rule
4: Propagate the labels of regional minima to the pixels of its sets

In comparison with the breadth-first paradigm, this one presents more variations at each step. In step 1, the most common approaches are to represent this connection through a graph (as in union-find and watershed cut) or through a number indicating a direction (as in arrowing). Steps 2, 3 and 4 are not in general seen as clear procedures in the algorithms, and usually are represented by path compression and traversing the path in reverse direction. The most distinctive step of this paradigm is the first one, where only local information is used and in fact mimics the drop of water simulation. However, this procedure is rarely implemented as a strict depth-first algorithm, as to process the watershed line in the plateau, a breadth-first algorithm has to be used.

III. Watershed Transform Definitions

In this section we briefly review the major watershed transform definitions: flooding (Flooding-WT), topographic distance (TD-WT), local condition (LC-WT), image foresting transform with max-arc path cost function (IFT-WT), tie-zone on the IFT-WT (TZ-IFT-WT) and the watershed cut.

The Flooding-WT is an iterative process of thresholding from the minor to the major grey level, where regions grow according to their influence zones calculated by the geodesic distance to the connected component of the previous level. Watershed lines are determined as the pixels that do not belong to any region. This process could be seen as a generalised SKIZ. [6]

The TD-WT is given in the form of a cost function between two points of an image, as the sum of geodesic distances between middle points with its costs related to their grey level difference, determining the slope inclination. From this cost function between any two points, it is determined which points belong uniquely to each minimum, forming the catchment basins. Points that belong to more than one basin form the watershed. One of the most important contributions of the TD-WT is the concept of downstream, defining a neighbourhood relation on the pixels with lowest grey level and less than the current, that is the basis of several algorithms. [11], [20]

The LC-WT is a derivation of the TD-WT, where the constraint of uniqueness of the solution is removed. This way, the watershed pixels no longer exists, as they are attributed to the neighbouring basins. Also, this means that the dividing line between basins is set between the pixels, and not over them. [14], [15], [21]

The IFT-WT is a SPF approach to the problem. Starting from a seed set, and considering the image a weighted graph, the max-arc path cost function is used to determine the minimum cost paths from each pixel to its nearest seed. However, inside a plateau these costs would be all the same. For this matter, it is introduced a secondary cost, the lexicographic cost, that determines the minor distance from the plateau border. Also, as the LC-WT, no pixel is labelled as watershed. [9], [10]

The tie-zone on the IFT-WT is a transform that unifies solutions. As the IFT-WT may produce a set of optimal solutions, the TZ-IFT-WT unifies them on a single solution, where the catchment basins are now defined as those pixels that have an optimal path to the same seed set in every possible solution. [4], [22]

The watershed cut is a graph cut transform where every operation is defined on the edges. Its formal definition is stated over the drop of water simulation, by means of a weighted graph. The cut is made whenever an edge satisfies the conditions that the vertices that it connects leads to two different minima, and that its value is greater than any other edge on these paths [19]. However, the watershed cut does not produce a unique solution, as many cuts may satisfy this definition. The proposed Algorithm 2 for the depth-first paradigm is very similar to the watershed cut.

One important consideration is that regardless of the algorithm, its results will be dependent of the definition that it implements. This means that algorithms under one definition must all be able to produce the same set of results. Sample results for each definition, with regional minima depicted in bold, are given on Fig. 1.

0	0	2	1	0
0	4	6	5	1
9	7	4	4	3
9	6	3	5	3
0	8	2	7	0
0	0	1	0	0

(a)

A	**A**	*W*	B	**B**
A	A	*W*	B	B
A	*W*	B	B	B
W	B	B	*W*	D
C	*W*	B	*W*	**D**
C	**C**	*W*	**D**	**D**

(b)

A	**A**	A	B	**B**
A	A	A	B	B
A	A	C	B	B
C	C	C	C	D
C	C	C	D	**D**
C	**C**	C	*D*	**D**

(c)

A	**A**	A	B	**B**
A	A	C	B	B
C	C	C	B	B
C	C	C	C	D
C	C	C	C	**D**
C	**C**	C	**D**	**D**

(d)

A	**A**	A	B	**B**
A	A	A	B	B
A	*W*	*W*	B	B
C	*W*	*W*	*W*	D
C	C	*W*	D	**D**
C	**C**	*W*	**D**	**D**

(e)

A	**A**	*TZ*	B	**B**
A	A	*TZ*	B	B
TZ	*TZ*	*TZ*	B	B
TZ	*TZ*	*TZ*	*TZ*	D
C	*TZ*	*TZ*	*TZ*	**D**
C	**C**	*TZ*	**D**	**D**

(f)

A	**A**	A	B	**B**
A	A	A	B	B
A	A	C	B	B
C	C	C	C	**C**
C	**C**	C	**C**	**C**
C	**C**	**C**	**C**	**C**

(g)

Figure 1. Sample results of definitions. (a) Image, (b) Flooding-WT, (c) LC-WT, (d) IFT-WT, (e) TD-WT, (f) TZ-IFT-WT [4], (g) Watershed cut.

IV. Analysis of Algorithms

This section surveys the literature of watershed transform algorithms, presenting them in a classification according to the paradigms previously depicted. Along this, other analysis are performed, like requirements, result type, implementation features, etc., composing a profile of the algorithms. The listing is given in chronological order of appearance.

A. Immersion

The immersion algorithm proposed by Vincent and Soille is the first fast watershed transform known in the literature, and its fundamentals are described by the breadth-first paradigm and the Flooding-WT definition [6]. Its implementation is based on a four step algorithm. Firstly, the pixels are ordered by its grey level. Secondly, iterating on the grey levels of the image, mask the pixels and insert those with a neighbour already labelled into a queue. Thirdly, the pixels on the queue are analysed and labelled appropriately, considering splitting plateaus as even as possible. Fourthly, set new labels for the newly detected regional minima.

To apply these steps, the algorithm makes use of a First-In First-Out (FIFO) data structure, henceforth called a queue. However, results may not be as expected, as the example given on Fig. 2 (a)-(b) shows that no line (watershed label) separates the minima, which is a problem related to the plateau behaviour of the definition, as in (c)-(e) a line is settled, even though not according to the definition. This problem has been previously noted on refs. [20], [23].

8	6	7	9
6	4	5	7
8	2	3	9
0	1	1	**0**

(a)

A	A	A	A
A	A	A	A
A	A	W	W
A	A	B	**B**

(b)

5	5	5
5	5	5
0	5	**0**

(c)

W	W	W
A	W	A
A	W	**B**

(d)

A	W	B
A	W	B
A	W	**B**

(e)

Figure 2. Sample results of algorithm of Vincent and Soille showing the difference between definition and algorithm results - (b) is the definition and algorithm result for image with 4-neighbourhood (a), whilst (d) is the definition and (e) is the algorithm result for image (c) with 8-neighbourhood.

B. Hierarchical Queue

Proposed by Beucher and Meyer [7], the use of the hierarchical queue for a watershed transform was only later shown to be an algorithm for generating a SPF for the max-edge path cost function (an IFT-WT) [9]. This algorithm requires only an initialisation and a working step. Nevertheless, the initialisation requires an input of seeds to insert in the queue (e.g. regional minima) from which to build the SPF, which in case of the regional minima demands a different algorithm. The working step consists of visiting, queueing and labelling the neighbours of the highest priority pixel from the queue (the one with lowest grey level with FIFO policy), until it is empty.

Also, this algorithm does not have a unique solution, as the insertion on the queue depends on the scanning order of the pixels, as shown on Fig. 3, where (a) and (c) are the images, (b) is the result of (a), (d) and (e) results of (c) on raster and anti-raster scanning order. However, for image (a) the scanning order would not interfere on the final result. Examples (a)-(b) use a 4-neighbourhood and examples (c)-(e) use an 8-neighbourhood.

8	6	7	9
6	4	5	7
8	2	3	9
0	1	1	**0**

(a)

A	A	B	B
A	A	B	B
A	A	B	B
A	A	B	**B**

(b)

5	5	5
5	5	5
0	5	**0**

(c)

A	A	A
A	A	B
A	A	**B**

(d)

B	B	B
A	B	B
A	B	**B**

(e)

Figure 3. Sample results of algorithm of Beucher and Meyer, exposing multiple possible solutions based on scan order.

C. Shortest-Path, Hill Climbing and Sequential Scanning

Following the work on hierarchical queues, Meyer proposes three algorithms for the watershed transform based on his TD-WT definition, revised by Roerdink and Meijster for generating the watershed lines [11], [20]. The first one calculates a SPF taking as path cost function the topographic distance previously mentioned and modifying Moore's [8] algorithms. The second one is called a hill-climbing algorithm, which uses the definition of upstream (the inverse of downstream), and the third one is a modification of Berge's algorithm for SPF [24].

Clearly, the modification of Moore's algorithm is related with the breadth-first watershed paradigm, as well as the hill-climbing as it analyses the upstream of the lastly labelled pixels. However, Berge's algorithm do not imply any ordering on the pixels, as it may scan randomly until stabilisation. Of the algorithms analysed, this one is the only that does not fit within any category proposed.

As the three algorithms adhere to the definition proposed, they present the same results, with uniqueness of the solution. Nevertheless, the image must be preprocessed for lower completion, changing pixel values to remove non-minima plateaus of an image [20], as it must not have these, for inside them the topographical distance is zero and that would result in multiple solutions [11], also the minima (or markers) must be supplied for input. One of the properties of these algorithms is that it is the watershed line on the pixels that ensure the uniqueness of the solution, and that means that in some cases this line will be more than one pixel thick, generating extensive areas that do not belong to any catchment basin. Fig. 4 shows that for image (a), the result in (b) will not contain a separating line between the catchment basins, and that for (c), firstly a lower completion must be done, presented in (d) using the algorithm in ref. [20], and that the results in (e) are independent of scanning order. Examples (a)-(b) use a 4-neighbourhood and examples (c)-(e) use an 8-neighbourhood.

8	6	7	9
6	4	5	7
8	2	3	9
0	1	1	**0**

(a)

A	A	B	B
A	A	B	B
A	A	B	B
A	A	B	**B**

(b)

5	5	5
5	5	5
0	5	**0**

(c)

11	11	11
10	10	10
0	10	**0**

(d)

W	W	W
A	W	B
A	W	**B**

(e)

Figure 4. Sample results of implementation of algorithm of Meyer by Roerdink and Meijster.

D. Connected Components

Based on the work of Meyer [11], Bieniek and Moga define and implement the LC-WT [14]. The technique proposed by these authors is based on the definition of downstream, as the algorithm works on 4 steps, the first being a scan for detecting the lowest neighbour of each pixel, as is the first step of the depth-first watershed paradigm, which fits best for this approach. Following this step, plateaus are detected and regional minima are labelled. The last step is to label the remaining pixels, traversing the path and compressing it for each pixel, on an operation called FIND, resembling the Union-Find algorithm presented next [15].

Clearly, the consequence of LC-WT is that the solution depends on the scanning order, as shown on Fig. 5 where (a) is the image, (b) and (c) are solutions on raster and anti-raster scan order with 8-neighbourhood. When no watershed lines were produced by the TD-WT definition, the result is exactly the same on the LC-WT, as in Fig. 4 (a)-(b).

5	5	5
5	5	5
0	5	**0**

(a)

A	A	A
A	A	B
A	A	**B**

(b)

B	B	B
A	B	B
A	B	**B**

(c)

Figure 5. Sample results presenting the multiple solutions of algorithm of Bieniek and Moga.

E. Union-Find

As a strict implementation of the definitions of TD-WT, the algorithm based on the Union-Find for disjoint sets of Meijster and Roerdink is based on a graph working on three main steps [16]. Avoiding to deal with plateaus, the first step is the lower completion of the image. Next, a directed graph is built, based on the downstream concept, accepting that a vertex is connected with more than one vertex. This graph is the basis for the last step which is fundamentally a path compression, where addresses along a path are all simplified directly to the root, detecting whether the paths lead to one single minimum for every pixel and labelling them appropriately. This algorithm is very similar to the previous one, though they differ in order that this one generates the same solutions as those proposed by Meyer. In fact, the algorithm of Bieniek and Moga is a variant of this one, where the results of the former could be generated by the latter only modifying the graph construction step [20]. Sample results are seen in Fig. 4, as it is a TD-WT.

F. Toboggan

The proposal of Mortensen and Barrett [25] is very similar to the depth-first watershed paradigm in the sense of the steps performed on the algorithm. The procedure is based on visiting every pixel in the image, trace a path through the lower neighbours until a minimum is found, label it and the path. However, the algorithm equivalence in comparison with the others depends on the lower completion of the image, as the plateau segmentation is dependant of scanning order. The algorithm implementation is rather simple, as no data structure for storing pixels is necessary. In terms of theoretical definitions, this algorithms is a LC-WT, as no pixel is set as watershed line and it follows the downstream definition. Therefore, results are also the same as in Fig. 5.

G. Image Foresting Transform

The algorithm of IFT-WT computes a SPF based on a hierarchical queue for selection of the next pixel to be processed. The use of a queue also provides inherent management of the plateau behaviour through a lexicographical cost [9], whereas the cost function itself is represented by a binary max operator. As the other breadth-first approaches, the IFT-WT input also consists of seeds from which the forest is generated, which may be the regional minima or markers. As aforementioned, the sample results of Fig. 5 also apply for the IFT-WT. However, as seen on ref. [4], the set of solutions for IFT-WT contains the set of solutions for LC-WT.

H. Chain Code

The algorithm of chain-code of Sun, Yang and Ren [17] extends the common use on contour description to indicate the direction of the arrowing. The algorithm operates on 4 steps: set the direction of the lowest neighbour for each of the pixels, set the directions on the plateaus traversing from the borders to the inside using a queue, label the minima and lastly label the paths starting from them. This approach adheres to the LC-WT definition and to the depth-first watershed paradigm, but its difference is to use a number to indicate a direction to the lower neighbour instead of its address, as proposed by the algorithm of Bieniek and Moga [15]. According to the authors, the advance of this algorithm is the possibility to generate other data related to the regions on the process of labelling, e.g. size. As a LC-WT algorithm, the sample results of Fig. 5 also applies.

I. Tie-Zone

In order to implement the TZ-IFT-WT, the algorithm of IFT-WT is modified to explicitly calculate the lexicographic distance and verify if a pixel has multiple solutions, what was not necessary on the IFT-WT. Among the features directly taken from the IFT-WT algorithm (e.g. hierarchical queue and requiring minima) the tie-zone transform adds a new type of label to represent those pixels that belong to multiple solutions. Nevertheless, these pixels are not watershed lines, as they indeed belong to the solutions and are optimal according to the definition, and this special label allows the algorithm to return always a unique solution, regardless of processing order of pixels. Sample results are presented on Fig. 6, with the label **TZ** denoting that a pixel belongs to the tie-zone.

As presented in the theoretical relations work by Audigier and Lotufo [4], the results of Fig. 6 are as expected, as for (a) the result in (b) is equal to the one produced by the local condition watershed transform, that is the same produced by

8	6	7	9
6	4	5	7
8	2	3	9
0	1	1	**0**

(a)

A	A	B	B
A	A	B	B
A	A	B	B
A	A	B	**B**

(b)

5	5	5
5	5	5
0	5	**0**

(c)

TZ	TZ	TZ
A	TZ	B
A	TZ	**B**

(d)

Figure 6. Sample results presenting the solutions of Tie-Zone watershed transform.

the topographic distance, as there are no watershed pixels. The same occurs for (c) and (d), as the pixels that are watershed lines by topographic distance must belong to the tie-zone.

J. Order Invariant Toboggan and Immersion

Lin *et al.* proposes two algorithms for watershed that produce equal results by an ordered processing and introduces a special label, called **RIDGE**, that is used to denote a line that divides distinct catchment basins [12]. Clearly, the immersion and toboggan algorithms are implementations of the breadth-first and depth-first watershed paradigms, respectively. However, in terms of theoretical definition, these procedures are not specific as to which they adhere. Analysing the toboggan algorithm, the first two steps are in fact a directed graph construction based on the downstream concept. The next two steps then manage the labelling of minima and resolution of the path labelling. As for the immersion, the concept is the same, with the difference that the graph is built iteratively for every grey level on the image. This leads to a conclusion that these algorithms indeed adhere to the TD-WT definition, with the toboggan and immersion resembling the union-find and the hill-climbing watershed transforms aforementioned respectively. Considering that the **RIDGE** label is equivalent to the **W** label, sample results of Fig. 4 also apply for these algorithms.

K. Shortest Paths

The algorithm of Osma-Ruiz *et al.* improves on the algorithm of Chain Code for the computation of the shortest paths for the watershed transform [18]. In general, the algorithms are similar, applying the depth-first watershed paradigm on an implementation of arrowing for the LC-WT definition. However, the most significant difference is the heavy use of queues on this proposal to reduce the unnecessary scan of the image, visiting the pixels more than once only when necessary. In fact, full scans of the image are performed only twice, with the other visits to the pixels decided by queue processing. Mainly, only two steps may be clearly identified as the arrowing and the labelling, performed on the full scans of the image. The sample results of Fig. 5 apply as the results are the same for both algorithms.

L. Watershed Cut

The work by Cousty *et al.* introduced the watershed cut [19]. The algorithm itself relies on the fact that the cut is a MSF relative to the regional minima. It computes the transform by visiting each pixel twice, one for determining its catchment basin and the other to label it. The procedure used to determine the catchment basin resembles the toboggan one, though with better management of the sets, as it switches from depth-first to breadth-first on plateaus. Moreover, for accomplishing equivalent results as the other algorithms, the image must be first lower complete, for this algorithm does not take into consideration an even division of plateaus. As a consequence of the definition of regional minima on the edges, the watershed cut algorithm identifies a single minimum for Fig. 2 (a) and (b) presented on this paper.

V. Relations

In a previous paper, Audigier and Lotufo presented the theoretical relationships on various watershed transform definitions [4]. In this section, these relations are recalled briefly and then extended to the algorithms, with special concern on the results that each one produce.

A. Theoretical relationships

The aforementioned work revisits the definitions of watershed transform by flooding (Flooding-WT) [6], topographic distance (TD-WT) [11], local condition (LC-WT) [15], image foresting transform (IFT-WT) [10], tie-zone (TZ-IFT-WT) [22], and minimum spanning forest (MSF-WT) [13]. Among these, the only definition that is not related with the others is the Flooding-WT, as its calculation is based on influence zones and is more likely a generalisation of the SKIZ on grey level images. The other definitions are graph based, where relationships are stronger and the proofs are direct by using special graphs, as the multipredecessor optimal graph (MOG) and the lower complete graph (LCG) [4]. With respect only to the catchment basins, Fig. 7 summarises the relationships between these definitions.

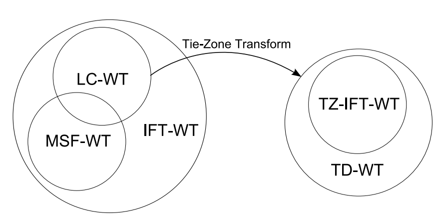

Figure 7. Theoretical relationships between watershed transform definitions with respect to catchment basins.

For clarity, the following statements are reproduced exactly as in ref. [4]:

1) Any LC-WT is also an IFT-WT
2) TD-WT is the tie-zone transform of LC-WT

3) The catchment basins defined by TZ-IFT-WT are subsets of the corresponding basins defined by TD-WT
4) Given a weighted graph and a seed set, any MSF (MSF-WT) is also a SPF-max (IFT-WT)

Aside from these relationships, Cousty *et al.* shows that, in a framework of graph cut operations, other relations are established, and some of those become equivalent, differently that in this analysis, where most relations are given as subsets. As an example, the IFT-WT allows to calculate a SPF cut, that is equivalent to the watershed cut and to the MSF cut [26]. Next, these relationships are applied and extended on the algorithms.

B. Algorithms Equivalence

As shown on Sec. IV every algorithm adhere to a watershed definition, with the exceptions of Vincent and Soille algorithm and the original proposals of Meyer for the topographic distance. This leads to the conclusion that the algorithms, classified into one of the watershed transform definitions aforementioned, also apply on the relationships exposed on Fig. 7. For clarity, Table I presents the watershed definition of each of the algorithms and the paradigm that explains the approach. Also, we expose the input of markers for the algorithms, the positioning of the watershed label, and the type of queue necessary for implementation.

There are 4 notes on Table I: (1) even though the three algorithms of Meyer are cited, the considered versions of them are those provided by Roerdink and Meijster that strictly implement the TD-WT definition [20], (2) as discussed on details previously, the sequential scanning algorithm does not fit on any of the paradigms, (3) the tie-zone algorithm produces a special label **TZ** with a meaning of ambiguity between solutions of the IFT-WT, and is not a watershed label in fact, and (4) according to Cousty *et al.* in ref. [26], the watershed cut is equivalent to a SPF cut, however, outside of the graph cut framework there are no relationships established.

Extending these theoretical results to the algorithms, it is direct that every algorithm on the same definition is able to produce the same output, regardless of software implementation, as it must follow the algorithm, whatever programming strategy used (e.g. static vs. dynamic queues). Clearly the algorithms of breadth-first are more suited for watershed from markers, as they require a set of seeds as input, regardless of these being regional minima or preprocessed markers. The other algorithms to achieve the same results would require a grey-scale morphological reconstruction, which in fact is equivalent to a watershed transform [27]. Lastly, even though some of the algorithms produce a label **W** on the pixels to indicate a watershed line, none of them ensure that this line divides the catchment basins, as the label only indicates ambiguity of distance to a minimum, in the case of TD-WT in the form of the topographic distance path cost function or the downstream and in the case of Flooding-WT in the form of influence zones.

VI. Conclusions and Future Work

In this paper, we proposed the classification of watershed algorithms in two graph exploring paradigms: the depth-first and the breadth-first watersheds. We relate fifteen of the algorithms described in the literature and point which graph exploring better characterise it, which definition it uses, whether the algorithm works with markers or from regional minima, and the positioning of the watershed line over the pixels. None of the algorithms that produce the watershed (**W**) label ensure that the line completely separates distinct catchment basins. We expect that this paper evaluation may be used for better understanding of the watershed algorithms described in the literature helping in the decision of which one to use and implement. Continuing this line of work, we are preparing a study on the speed performance of the many watershed algorithms.

References

[1] H. Digabel and C. Lantuéjoul, “Iterative algorithms,” in *Proc. Second European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, J.-L. Chermant, Ed. Stuttgart, Germany: Riederer Verlag, 1978, pp. 85–99.

[2] 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, September 1979.

[3] F. Meyer and S. Beucher, “Morphological segmentation,” *Journal of Visual Communication and Image Representation*, vol. 1, no. 1, pp. 21–46, September 1990.

[4] R. Audigier and R. A. Lotufo, “Watershed by image foresting transform, tie-zone, and theoretical relationships with other watershed definitions,” in *ISMM'2007 Proceedings*, vol. 1, Universidade de São Paulo (USP). São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), October 2007.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge , Massachusetts: The MIT Press, 2001.

[6] L. Vincent and P. Soille, “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[7] S. Beucher and F. Meyer, *Mathematical morphology in image processing*, ser. Optical Engineering. New York: M. Dekker, 1993, ch. The Morphological Approach to Segmentation: The Watershed Transformation.

[8] E. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959.

Table I
EQUIVALENCE BETWEEN THE ALGORITHMS

Algorithm	Definition	Paradigm	Markers	W Label	Queue
Immersion of Vincent and Soille (1991)	Flooding	Breadth-First	no	yes	Simple
Hierarchical Queue of Beucher and Meyer (1993)	IFT-WT	Breadth-First	yes	no	Hierarchical
Shortest-Path of Meyer (1994) - *see (1)*	TD-WT	Breadth-First	yes	yes	no
Hill Climbing of Meyer (1994) - *see (1)*	TD-WT	Breadth-First	yes	yes	no
Sequential Scanning of Meyer (1994) - *see (1)*	TD-WT	*see (2)*	yes	yes	no
Connected Components of Bieniek and Moga (1998)	LC-WT	Depth-First	no	no	Simple
Union-Find of Meijster and Roerdink (1998)	TD-WT	Depth-First	no	yes	Simple
Toboggan of Mortensen and Barrett	LC-WT	Depth-First	no	no	no
Image Foresting Transform of Lotufo and Falcao (2000)	IFT-WT	Breadth-First	yes	no	Hierarchical
Chain Code of Sun, Yang and Ren (2005)	LC-WT	Depth-First	no	no	Simple
Tie-Zone of Audigier, Lotufo and Couprie (2005)	TZ-IFT-WT	Breadth-First	yes	*see (3)*	Hierarchical
Order Invariant Toboggan of Lin *et al.* (2005)	TD-WT	Depth-First	no	yes	Simple
Order Invariant Immersion of Lin *et al.* (2005)	TD-WT	Breadth-First	no	yes	Simple
Shortest Paths of Osma-Ruiz *et al.* (2006)	LC-WT	Depth-First	no	no	Simple
Watershed Cut of Cousty *et al.* (2008)	*see (4)*	Depth-First	no	no	no

[9] R. Lotufo and A. Falcão, "The ordered queue and the optimality of the watershed approaches," in *Proceedings of the 5th International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, vol. 18. Kluwer Academic Publishers, June 2000, pp. 341–350.

[10] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: theory, algorithms, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 19–29, 2004.

[11] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.

[12] Y. Lin, Y. Tsai, Y. Hung, and Z. Shih, "Comparison between immersion-based and toboggan-based watershed image segmentation," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 632–640, 2006.

[13] F. Meyer, "Minimum spanning forests for morphological segmentation," in *Mathematical morphology and its applications to image processing*, J. Serra and P. Soille, Eds., vol. 2. Kluwer Academic Publishers, 1994, pp. 77–87.

[14] A. Bieniek and A. Moga, "A connected component approach to the watershed segmentation," in *ISMM '98: Proceedings of the fourth international symposium on Mathematical morphology and its applications to image and signal processing*. Norwell, MA, USA: Kluwer Academic Publishers, 1998, pp. 215–222.

[15] ——, "An efficient watershed algorithm based on connected components," *Pattern Recognition*, vol. 33, no. 6, pp. 907–916, 2000.

[16] A. Meijster and J. B. T. M. Roerdink, "A disjoint set algorithm for the watershed transform," in *Proc. IX European Signal Processing Conf EUSIPCO '98*, 1998, pp. 1665–1668.

[17] H. Sun, J. Yang, and M. Ren, "A fast watershed algorithm based on chain code and its application in image segmentation," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1266–1274, 2005.

[18] V. Osma-Ruiz, J. I. Godino-Llorente, N. Sáenz-Lechón, and P. Gómez-Vilda, "An improved watershed algorithm based on efficient computation of shortest paths," *Pattern Recognition*, vol. 40, no. 3, pp. 1078–1090, 2007.

[19] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Minimum spanning forests and the drop of water principle," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1362–1374, 2009.

[20] J. B. T. M. Roerdink and A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies," *Fundam. Inf.*, vol. 41, no. 1-2, pp. 187–228, 2000.

[21] A. Bieniek, H. Burkhardt, H. Marschner, M. Nölle, and G. Schreiber, "A parallel watershed algorithm," in *Proceedings of 10th Scandinavian Conference on Image Analysis (SCIA97)*, 1997, pp. 237–244.

[22] R. Audigier, R. de A. Lotufo, and M. Couprie, "The tie-zone watershed: Definition, algorithm and applications," in *In Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, vol. 2, 2005, pp. 654–657.

[23] R. Audigier and R. Lotufo, "Uniquely-determined thinning of the tie-zone watershed based on label frequency," *J. Math. Imaging Vis.*, vol. 27, no. 2, pp. 157–173, 2007.

[24] C. Berge, *The Theory of graphs and its applications*. Wiley, 1964.

[25] E. N. Mortensen and W. A. Barrett, "Toboggan-based intelligent scissors with a four-parameter edge model," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, p. 2452, 1999.

[26] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Thinnings, shortest-path forests and topological watersheds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, to appear.

[27] R. de Alencar Lotufo, A. X. Falcão, and F. A. Zampirolli, "Ift-watershed from gray-scale marker," in *SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 146–152.