On Watershed Transform: Plateau Treatment and Influence of the Different Definitions in Real Applications

André Körbes, Roberto de Alencar Lotufo
School of Electrical and Computer Engineering
University of Campinas - Unicamp
Campinas, Brazil
{korbes,lotufo}@dca.fee.unicamp.br

Abstract—In this paper we review and discuss two important aspects of the watershed transform: treatment of plateaus and the influence of different watershed transform definitions on the applications results. For the treatment of plateau we analyze and compare two methods for plateau division, the lower completion and the lexicographic path cost function. For the difference among watershed definitions, we test six different definitions on a real application and compare their results.

Keywords-watershed transform, plateaus, applications.

I. INTRODUCTION

The watershed transform is usually taken as a black box to perform segmentation, simulating a flooding process of an image viewed as a surface, to obtain closed contours and/or labelled regions. However, rarely is given special attention to details such as definition adherence, plateau treatment and implementation issues, and whether those have any influence on the final results. The purpose of this article is to highlight these points by demonstrative examples, showing the importance of each one. For that matter, the techniques used for plateau treatment are analyzed and shown to have differences on certain definitions, and the variation of solutions between watershed transforms is exemplified on a real application.

This paper is organized as follows: Sec. II gives background for the watershed transform definitions used throughout the work, Sec. III discusses the techniques for plateau management, Sec. IV applies the different watershed transforms to a real application and lastly Sec. V presents the conclusions and future works.

II. WATERSHED DEFINITIONS

The watershed transform has several definitions on the literature that define different sets of solutions. Some of the differences between the definitions are the existence or not of watershed pixels, the uniqueness of solution and the way plateaus are managed, with secondary cost or image transformation. In this paper we use the taxonomy of [11], with the addition of the work of Cousty *et al.* [8], the watershed cut, called WC-WT. For reference, the immersion of Vincent and Soille [2] is called Flooding-WT, the topographic distance of Meyer [3] is called TD-WT, and its derivation on the local condition by Bieniek and Moga [5] is called LC-WT. The image foresting transform [6] and its tie zone [7] are called respectively IFT-WT and TZ-IFT-WT.

One important concept on the context of this paper is the plateau. A plateau is a connected component where the values of the pixels are the same on every pixel belonging to it. A plateau is divided on edge and inner pixels. Edge pixels are those on the connected component that have at least one neighbouring pixel with lower value. The inner pixels are those where the neighbours have either greater or equal value. For the watershed transform, plateaus that require special management are those that are not regional minima. Other conditions may apply, such as having at least one inner pixel, though these are dependent of the definition chosen.

As how the plateaus are managed, one important concept is the lexicographic cost, henceforth called lex cost, defined on [6], which calculates geodesic distances for plateaus with respect to its edges. Also, FIFO (first-in first-out) data structures, or queues, are used to uniformly propagate labels and path costs to inner pixels. The lower completion is defined by Roerdink and Meijster [4], as a transformation on the image, where its range of values is modified through a process that searches for the greatest geodesic distance from an edge to an inner pixel on every plateau. The following section studies how these approaches are related.

III. PLATEAU TREATMENT

Plateau splitting is an important step of the watershed transform calculation, once these are regions where the paths' basic costs become equal and demand a different action from the algorithms. These actions may be classified as: pre-processing for lower completion; FIFO propagation of labels uniformly from the edges to the inner pixels of the plateau; lex cost calculation and usage as secondary cost component; and random division. In this section, we propose an experiment to evaluate the equivalence of these methods.

The lower completion operation is used on image pre-processing with the purpose of removing the plateaus, ensuring that for each pixel there is a lower neighbour, except on regional minima, thus simplifying the following algorithm of watershed [4]. When this process is not accomplished, the technique most used by algorithms is to propagate labels from the edges using queues, uniformly splitting the region, disregarding the cost of the definition that it implements. However, the use of a FIFO is implicitly the lex cost taken as a secondary cost component, for tie breaking the first, whichever it is, simulating the constant speed of water propagation on flat zones. The equivalence between FIFO propagation and lex cost is discussed in depth on [6].

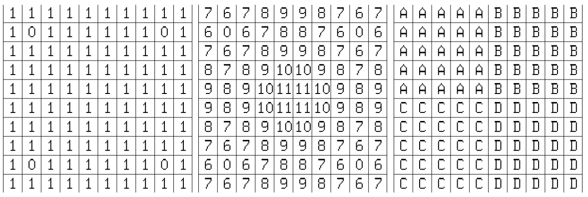
In the literature there is a common misconception related to the use of these methods for correct plateau division, where they are taken as equivalent, when they are not, as the counter-examples below will show. Also, the hierarchical queue, used as basis for some algorithms, is usually seen as a data structure that implicitly embeds only the max cost, where the FIFO policy is responsible for the lex cost [9]. However, discarding the FIFO policy, this structure still implements a more complex cost than just the max, leading to results that differ from the expected. This result is exemplified below, with a special case.

With regards to the TD-WT definition, the use of lower completion or the lex cost achieve the same results, as seen on several algorithms that implement both LC-WT and TD-WT. Especially considering the approaches of Meijster and Roerdink on the Union-Find algorithm [4] where the lower completion is a prerequisite, and Lin *et al.* algorithm' of Order Invariant Toboggan [10], where plateaus are processed using queues - implicitly the lex cost. Given that both algorithms render the same results [1], one concludes that both ways may be considered equivalent.

Nevertheless, these are not equivalent when the path cost function is the max, basis of the IFT-WT definition, once only the lex cost gives the expected results. In order to expose this problem accurately both max and lex cost must be explicitly calculated, to avoid bias from data structures, such as the hierarchical queue or the FIFO queue. Thus, the Berge algorithm [3] seems the most appropriate since it does not rely on any condition of the image or any data structure. This algorithm was modified to explicitly calculate the max of the path, called the Berge-Max algorithm, and both max and lex cost, called the Berge-MaxLex algorithm, generating shortest-path forest based on these costs. Also, to check for the cost that the hierarchical queue embeds, this structure is modified to change the FIFO policy onto a random tie breaker. Replacing this on the IFT algorithm it is then called the IFT-Random. None of these algorithms are included here for space restrictions, however, are easily constructed from the originals.

A. Experiment

We propose an experiment, using the algorithms mentioned above, to demonstrate that the lower completion pre-processing with the max path cost function renders incorrect results compared to the lex cost as a tie breaker for the max. This is shown by counter-example, as both methods should produce the same results on any image. On this experiment, a 10x10 pixels image, constant with value 1, with four regional minima of value 0, on coordinates (2,2), (2,9), (9,2), (9,9), is used. The lower completion is accomplished using the algorithm proposed in [4]. Fig. 1 shows in (a) the original image *f*, in (b) the lower complete image *lc*, and in (c) the expected result for a watershed of image *f*.



1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1

(a)

7	6	7	8	9	9	8	7	6	7
6	0	6	7	8	8	7	6	0	6
7	6	7	8	9	9	8	7	6	7
8	7	8	9	10	10	9	8	7	8
9	8	9	10	11	11	10	9	8	9
9	8	9	10	11	11	10	9	8	9
8	7	8	9	10	10	9	8	7	8
7	6	7	8	9	9	8	7	6	7
6	0	6	7	8	8	7	6	0	6
7	6	7	8	9	9	8	7	6	7

(b)

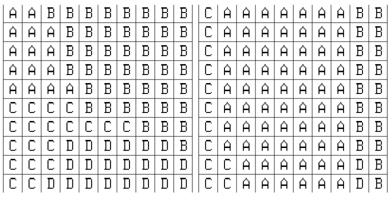
A	A	A	A	A	B	B	B	B	B
A	A	A	A	A	B	B	B	B	B
A	A	A	A	A	B	B	B	B	B
A	A	A	A	A	B	B	B	B	B
A	A	A	A	A	B	B	B	B	B
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D

(c)

Figure 1. Experiment images. (a) Original *f*, (b) Lower Complete *lc*, (c) Expected solution of *f*.

The IFT algorithm [6] does not explicitly calculate the lex cost, since the FIFO policy ensures the appropriate behaviour. However, the Berge-MaxLex algorithm needs this calculation, and must process those costs until stabilization with minimal costs. Thus, with explicit components, Berge-MaxLex algorithm produces the same results as IFT. That way, both algorithms produce the solution of Fig. 1 (c).

Applying these algorithms on image *lc* the same results are obtained that of Fig. 1, indicating the lower completion does not influence on the result when both components are used. Though removing the lex cost and applying the algorithms of max cost – Berge-Max and IFT-Random – on image *f*, the obtained results are strongly biased by the scanning order as seen on Fig. 2.



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

(a)

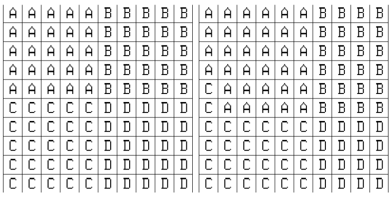
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	A	A	A	A	A	A	A	B	B
C	C	A	A	A	A	A	A	D	B
C	C	A	A	A	A	A	A	D	B

(b)

Figure 2. Max cost watershed. (a) IFT-Random, (b) Berge-Max.

On Fig. 2 two distinct solutions are generated, both valid by a max path cost function. The (a) solution depends on the organization of the hierarchical queue, itself dependent on memory organization. The solution on (b) is clearly biased by a raster scan order, as a consequence of the Berge-Max algorithm using the same matrix for processing and storing results, so that the first label found on minimum (2,2) is propagated to almost every pixel on the image. Alternating the scan order differing results may be found, each of which valid.

Applying these algorithms on image *lc* produces the desired effect for demonstration. In order to lower completion equate lex cost on a max cost definition the result of Berge-Max on image *lc* must be equal to that of Berge-MaxLex on image *f*. However, that is not observed, thus showing a counter-example for the assumption. Fig. 3 presents the results of algorithms (a) IFT-Random and (b) Berge-Max applied on image *lc*. It is interesting to note that IFT-Random renders the same result of IFT, showing that hierarchical queue's embedded cost is not simply the max, whereas Berge-Max generates different regions than Fig. 1(c).



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

(a)

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

(b)

Figure 3. Max cost on image *lc*. (a) IFT-Random, (b) Berge-Max.

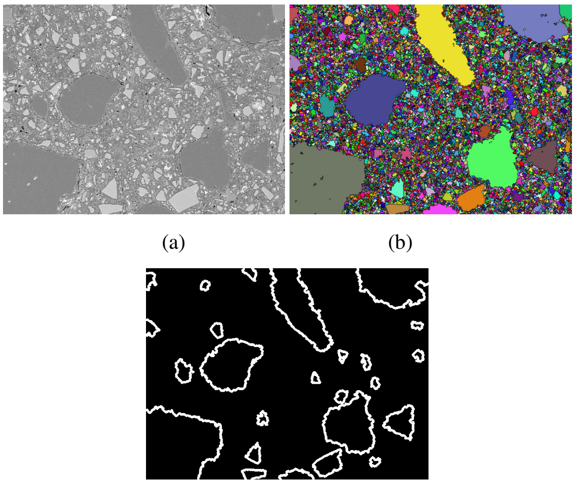
It is important to highlight that the main result of this experiment is that the equality between lower completion and lex cost for plateau treatment depends on the primary path cost function that is being minimized. The design of an algorithm must consider this result on the technique used for plateau treatment in order to produce consistent solutions. The choice of the method depends, in general, of the definition chosen and on the architecture of the algorithm, whether the costs are all calculated explicitly such as Berge-MaxLex, or the FIFO queue is used, such as on Lin's toboggan, or the hierarchical queue is used with a FIFO policy such as on IFT.

Each method has its implication. A queue's efficient implementation is not a trivial task. However, the lower completion process also demands a queue to calculate distances and change pixels values. The explicit calculation does not depend on data structures, but several scans are necessary until stabilization. Thus, a balance between the alternatives must be done, in order to choose the best method for the algorithm's features, as it is considered essential for one to manage plateaus correctly to achieve the desired results of the watershed transform.

IV. SEGMENTATION RESULTS

In this section the watershed transform definitions results are compared on a real application, extending the question of plateau treatment. This way, it is intended to highlight the subject that the definition implemented by an algorithm may impact on the application output, thus it must be considered when choosing one. In this experiment, the definitions mentioned on Sec. II are tested on an application to measure some regions of interest (ROI's). A special consideration must be taken regarding the Flooding-WT definition, once there is no algorithm consistent with it [4], the immersion of Vincent and Soille is used as the closest approach [2].

The application consists in detecting homogeneous grains of thin-section microscopic image of a sample of concrete, our regions of interest. The watershed transform is used as a texture detector for homogeneous regions that are filtered by area. Fig. 4 presents the steps taken to extract the ROI's: (a) input image *csample*; (b) labelled watershed transform applied on the morphological gradient of image (a), filtered by dynamics for noise removal; (c) dilated internal edges of the obtained regions after filtering by area to select regions and remove further noise.



(a) (b)

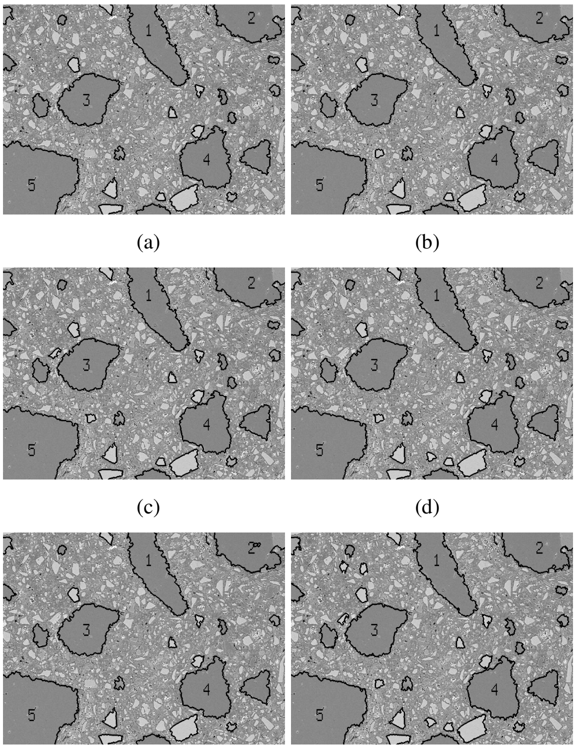
(c)

Figure 4. Concrete application steps. (a) Input image. (b) Labelled watershed regions. (c) Sample of dilated borders after filtering.

Alternating the watershed transform applied on step (b) for the six definitions, we measure the final result of the application: the number of homogeneous regions detected, and the area and variation of the five greater regions. Fig. 5 presents the results produced by the algorithms with the internal edge highlighted and dilated. Fig. 5 (a)-(f) shows the five ROI's numbered and present the solutions given by the algorithms of Vincent and Soille for Flooding-WT [2], Lin's *et al.* for TD-WT [10], Bieniek and Moga's for LC-WT [5], Beucher and Meyer's for IFT-WT [9, 6], Audigier, Lotufo and Couprie's for TZ-IFT-WT [7] and Cousty's *et al.* for WC-WT [8], respectively.

As mentioned, the purpose of this application is to measure homogeneous regions and the number of regions detected. For images (a)-(f), it was found 26, 27, 28, 28, 23 and 30 regions respectively. Considering that the variation of the areas of the ROI's are correlated to its perimeter, and therefore to its radius, the measure calculated here is $\left(A-\overline{A}\right)/\sqrt{\overline{A}}$, where $\overline{A}$ is the average area. This ratio gives a normalised measure that shows the variation independent of the size of the region being analysed. These measures are shown on the chart of Fig. 6, where each bar corresponds to one definition, and each number to a region. Positive ratios indicate the area for the result was larger than average, whilst negative indicate smaller areas.

In general, the analysis of the results points to consistency amongst solutions. As seen on Fig. 6, the main variations come from definitions TZ-IFT-WT and WC-WT. Also, one may see that the average size of the region being evaluated, acting as a normalisation for the ratio calculated, does not imply on larger variations, once region 4 has much more variation than region 5. This might be attributed to a noisier neighbourhood, where the differences of definition of TZ-IFT-WT and WC-WT are more effective.



(a) (b)
(c) (d)
(e) (f)

Figure 5. Concrete application watershed ROI's and comparison between definitions.

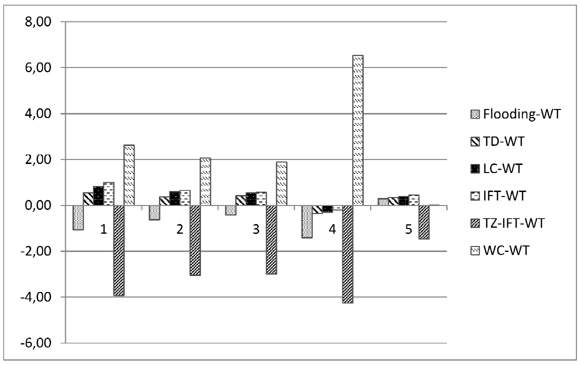


Figure 6. Chart of variations of measured areas of each algorithm with respect to the average area.

These conclusions, even though obtained from a very specific type of image, highlight that for the recent watershed algorithms a direct equivalence might incur on errors that are higher than those accepted on the application. On some problems a deeper analysis might be needed, in order to determine the most adequate approach, especially in concern with the definition behaviour, given that algorithms are tightly bound to these. However, in cases where there is a pre-processing with severe image simplification, that is the elimination of over segmentation by filtering of regional minima, where the watershed transform produces a limited number of regions, these differences tend to be smaller.

V. CONCLUSION

In this paper we analyze two aspects of the watershed transform: the plateau treatment, investigating the behaviour of lower completion, lexicographical cost and FIFO queues and the influence of the different definitions on a real application. We demonstrate via examples that there are differences on how plateaus are treated and that the possible strategies may produce different results. The real application was tested against six definitions to verify that, though approximate, results vary and this might be an issue.

We conclude that not always the implementations follow strictly the algorithms, once these may have different requirements. Also, the plateau treatment strategy depends on which definition is implemented and the architecture of the algorithm, implying on the final result. Finally, when developing an application that is highly dependent on the results of the watershed transform, an evaluation should be performed with special attention in order to determine if the results of the algorithm chosen, and by consequence its definition, are appropriate.

REFERENCES

[1] Körbes, André and Lotufo, Roberto de Alencar, "Analysis of the watershed algorithms based on the Breadth-First and Depth-First exploring methods," in Computer Graphics and Image Processing, Brazilian Symposium on, IEEE Computer Society, 2009.

[2] 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.

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

[4] 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.

[5] A. Bieniek and A. Moga, "An efficient watershed algorithm based on connected components," Pattern Recognition, vol. 33, no. 6, pp. 907–916, 2000.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] 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.