A parallel, O(n), algorithm for unbiased, thin watershed

Theodore Chabardes, Petr Dokladal, Matthieu Faessel, Michel Bilodeau

▶ To cite this version:

A PARALLEL, O(N) ALGORITHM FOR UNBIASED, THIN WATERSHED

T. Chabardès, P. Dokládal, M. Faessel, M. Bilodeau

PSL Research University - MINES ParisTech
CMM, Center for Mathematical Morphology
35 rue Saint Honoré - Fontainebleau, France

ABSTRACT

The watershed transform is a powerful tool for morphological segmentation. Most common implementations of this method involve a strict hierarchy on gray tones in processing the pixels composing an image. Those dependencies complexify the efficient use of modern computational architectures. This paper aims at answering this problem by introducing a new way of simulating the waterflood that preserves the locality of data to be processed. We propose a region growth algorithm based on arrowing graphs that is strictly linear despite the valuation domain of input images. Simultaneous and disorderly growth is made possible by using a synchronization mechanism coded directly on the weight of nodes. Experimental results show that the algorithm is accurate and by far outperforms common watershed algorithms.

Index Terms— Segmentation, Watershed, Local implementation, Linear time.

1. INTRODUCTION

The watershed transformation is a common tool for morphological segmentation. It is based on region growth and edge detection. One definition reflects the water flood on a topographic relief. This definition creates thin watershed lines and was chosen for the most used algorithms. Considering the topographic surface as pierced at the location of regional minima, it is immersed in water, so that the water rises from those holes and creates lakes. Those different lakes, each associated with a minimum, are not allowed to mix and their intersections will form dams. The set of all dams is called the watershed lines and represents the desired contours.

Algorithms that replicate this principle do not meet the speed required by the ever-growing amount of data to be processed. The queue-driven propagation is very sequential and only efficient for mono-processors. The current limitation of the clock rate limited by physical constraints brought the need for multi-processor architectures and adaptation of usual methods to the new specifications. The sequential nature of most algorithms makes impossible to benefit of their parallel processing capabilities. This motivates the work presented in this study.

In this article, we present a new method to calculate watershed lines using an oriented graph or digraph. Compared to traditional methods, the proposed method preserves the locality of data and allows for optimized implementation without compromising the quality of the extracted contours. Time complexity is still linear with respect to the number of pixels and the use of complex data structures such as hierarchical queues is avoided. We compare ourselves with the two following methods: a hierarchical queue-driven propagation [1] and successive geodesic thickenings of the regional minima [2].

2. RELATED WORK

Early work involved morphological operators. Reconstruction of successive thresholds of the image, by means of skeleton by geodesic influence zones, has been developed in [3] by Beucher and Lantuejoul. Iterations are calculated until idempotence to obtain the watershed.

Another iterative approach based on arrowing has been introduced in 1990 by Beucher [4]. A more recent method using successive pruning of the arrowing graph has been developed by Meyer in 2014 [5].

Other approaches involving the use of hierarchical queues were first introduced by Vincent and Soille in 1991 [6] and by Meyer [7, 1] and developed by Beucher in a non-biased algorithm[8]. Splitting the image into smaller images was studied in 1996 [9] by Bieniek and al., with a preprocessing of the overlapping areas and distribution of those images to multiple processing units. A study was realized on the parallelization of the hierarchical queue approach by Beucher in 1997 [10]. Further works on splitting by Gillibert and Jeulin involved iterative processing of smaller images and a merging procedure of the results until idempotence [11].

An alternative definition of the watershed transform based on topologic distance was proposed in 1997 by Couprie et Bertrand [12]. Futher works lead to the creation of the power watershed by Couprie, Grady, Najman and Talbot in [13]. Parallelization of this method was studied in [14] by van Neerbos, Najman and Wilkinson. Addionally, a first massively parallel algorithm in pseudo $O(n)$ in continuous space was proposed by Dejnozkova [15, 16].

3. TOPOGRAPHIC STRUCTURES

We define a gray-level image by the mapping $f : \mathscr{D} \to \mathscr{V}$, where $\mathscr{D}$ is the set of pixels and $\mathscr{V}$ the valuation domain. We associate this function with its corresponding topographic relief, where all gray values can be seen as elevations on the relief. This topographic relief contains a various number of topographic structures such as domes, valleys, ridges, thalwegs, regional minima, plateaus, and so on. Among those, a few interest us as flooding paths, plateaus, regional minima and catchment zones. The absolute heights of the pixels are not needed for defining the topographic structures and they depend only on the relative heights of neighboring pixels.

In the following, $\sim$ denotes the neighborhood relationship on $\mathscr{D}$ defined by the considered connectivity. Any usual connectivity can be used, *e.q.* 4,6 or 8 in 2D and 6,18 or 26 in 3D. If a,b are neighbors, we write $a \sim b$. For $a, b \in \mathscr{D} \mid a \sim b$:

$$\begin{cases} f(a) < f(b) & \Leftrightarrow a \text{ is a } \textit{lower neighbor} \text{ of } b, \text{ noted } a \prec b \\ f(a) > f(b) & \Leftrightarrow a \text{ is a } \textit{higher neighbor} \text{ of } b, \text{ noted } a \succ b \\ f(a) = f(b) & \Leftrightarrow a, b \text{ are } \textit{level-neighbors}, \text{ noted } a \simeq b \end{cases}$$

We introduce a graph $\mathscr{G}(N, E)$, a non-oriented graph where $N \leftrightarrow \mathscr{D}$ and the set E is generated by the connexity $\sim$. We can encode the relative height of neighboring nodes in an oriented graph built from the previous graph, by defining edge sets derived from E. We note $E_\prec$, $E_\succ$ and $E_\simeq$ the edge sets generated by the relation $\prec, \succ, \simeq$. Using these new edge sets, we can define the increasing graph $\mathscr{G}(N, E_\succ)$, the decreasing graph $\mathscr{G}(N, E_\prec)$ and the level-neighbor graph $\mathscr{G}(N, E_\simeq)$. Considering $a \prec b \Leftrightarrow b \succ a$, The two following properties can be deduced:

$$\mathscr{G}(N, E_\succ \cup E_\prec) \text{ is non-oriented,}$$

$$\mathscr{G}(N, E_\simeq) \text{ is non-oriented.}$$

The relations $\simeq$, $\succ$ and $\prec$ are transitive. If $a \simeq b \simeq c$ then $f(a) = f(b)$, $f(b) = f(c)$ and also $f(a) = f(c)$ and a is path-connected to c by a level-constant path. Similarly, if $a \succ b \succ c$ then also $f(a) > f(c)$ and a is connected to c by a strictly decreasing path.

For a point a the transitive closure $\simeq^+$ of a defines a plateau p, a connected region of a constant-level, with $a \in p$. From now on, let $P = \bigcup_i p_i$ denote the set of all plateaus and p_i its elements.

A plateau from where one cannot reach a lower altitude with a non-increasing path is called *regional minimum*. We can define it using the following equation:

$$m \in P, m \text{ is a } \textit{regional minimum} \Leftrightarrow$$
$$\forall a \in m, \nexists b \in N, a \prec b.$$

We call the set $M = \bigcup_i m_i$ the set of all minima.

Given some minimum m_i, the transitive closure $\succ^+$ defines the catchment basin of m_i: $CB(m_i) = \{x \mid x \succ^+$

$y, y \in m_i\}$ as the union of points connected to m_i by a strictly decreasing path.

The basins are not disjoint. Their intersection is usually considered as the watershed. However, this intersection does not have the necessary properties, namely it is neither thin nor contiguous. We show below how to efficiently generate a thin, continuous and well centered watershed line, and define the subset later as an union of nodes and edges.

4. BUILDING THE WATERSHED USING THE FLOODING GRAPH

In the next section, we show how a flooding graph can be used to generate the watershed lines. We describe the three main components of the method in sections 4.1, 4.2 and 4.3. Reconstructing the catchment basins requires the extraction of minima and the creation of the flooding graph. However, those two last components are independent. Figure 1 shows how they relate to each other. The result $l(f)$ holds both partial catchment basins and watershed lines.
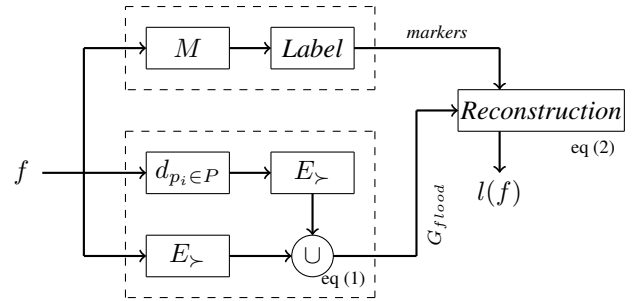

Fig. 1. Overview of the method.

4.1. Minimum detection

The set of regional minima U is obtained by removing from P all plateaus p_i that are not minima by

$$U = P \setminus \{p_i \mid \exists b, b \prec a, a \in p_i\}, \text{ with } a, b \in N$$

Note that this operation is an opening by reconstruction of P with a marker set $V = \{a \mid \exists b, b \prec a\}$ which is the decreasing border (provided it exists) of any plateau which is consequently not a regional minimum.

Given that P contains plateaus, U does not contain single-pixel minimum. The set of single-pixel minimum is denoted by S, and is efficiently detected using $S = \{a \mid \nexists b, a \succ b \text{ or } a \simeq b\}$ The set of all minima of an image, denoted by M, is the union of both regional and singleton minima, *i.e.* $M = U \cup S$. Figure 2 shows an example of this process.

4.2. Generating the flooding graph

Water flooding is a bottom-up process. A natural choice to simulate this behavior would be to use the increasing graph
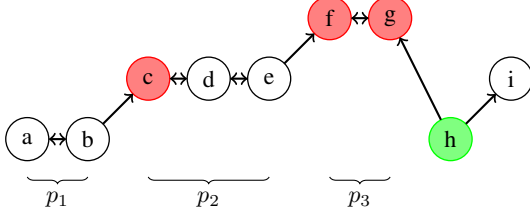
Fig. 2. Minima detection. $E_{\simeq}$(double-sided arrows), $E_{\succ}$(single-sided arrows). $P = \{p_1, p_2, p_3\}$. $V = \{c, f, g\}$, decreasing borders. $U = P \setminus \{p_1, p_2\} = \{a, b\}$. $S = \{h\}$. $M(f) = U \cup S = \{a, b, h\}$.

$\mathscr{G}(N, E_{\succ})$, since for any $a, b \in N$ such that $f(a) > f(b)$, there is $e_{b,a} \in E_{\succ}$. Note that this graph contains no arcs on plateaus to indicate how to flood plateaus. On plateaus, water flows from decreasing borders towards increasing borders. In the following we create $E_{\succ}$ arcs on plateaus to simulate this behavior as well. Consider some plateau p and its decreasing border V, $V \subset p$. Let $d_p(a, V)$, $a \in p$, denote a geodesic distance in p from a to V. We redefine the set of increasing edges in the following way. Let $a, b \in N \mid a \sim b$:

$$e_{b,a} \in E_{\succ} \text{ if } \begin{cases} f(a) > f(b) \text{ or} \\ d_p(a, V) > d_p(b, V) \text{ when } a, b \in p \end{cases} \quad (1)$$

Figure 3 shows how an increasing graph of the initial image is completed with an increasing graph of the geodesic distance of the plateau.
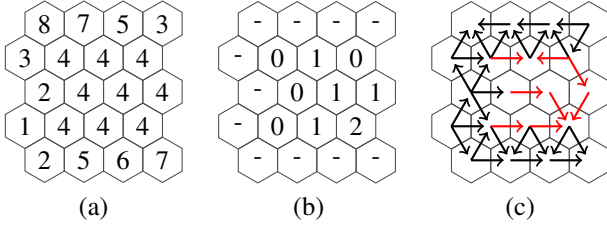

Fig. 3. Generating a flooding graph. (a) gray-level image. (b) geodesic distance on a plateau. (c) flooding graph, ascendancy graph (black), ascendancy graph on the geodesic distance (red).

4.3. Catchment basin reconstruction

Each basin is associated to a minimum, the minima are labeled. For every node, we associate a value in $\mathscr{L} = \{\emptyset, W\} \cup \mathbb{N}^*$, where $\emptyset$ is an undetermined value, associated to every non-minimum node at initialization, W is the watershed line value. Thus, we define the application $l : \mathscr{D} \to \mathscr{L}$.

As the water rises from bottom to top, a node takes its label from the nodes below it. If the same label appears on all lower neighbors, this label propagates. Otherwise, if different

labels are available, the value W is assigned. W itself does not propagate.

We note $Q(a) = \{l(b) \mid b \succ a\} \setminus \{W\}$ the set of labels on lower neighbors of a, and we propose the following rule of propagation:

> **repeat**
> **if** $Q(a) = \{l\}, l \neq \emptyset,$ **then** $l(p) \leftarrow l$
> **else if** $|Q(a) \setminus \emptyset| > 1$ **then** $l(p) \leftarrow W$.
> **until** stability
$$(2)$$

The resulting image contains both the catchment basins associated to the minima, and the watershed lines. Depending on the chosen flooding graph, the shape and location of watershed lines can vary.

4.4. Thin watershed lines

Equation (2) suffices to produce a watershed segmentation. However, thick watershed lines appear whenever every lower neighbor are labeled W, *i.e.* $Q(a) = \{\}$ in (2). Figure 4 shows an example of such a situation. We introduce a method
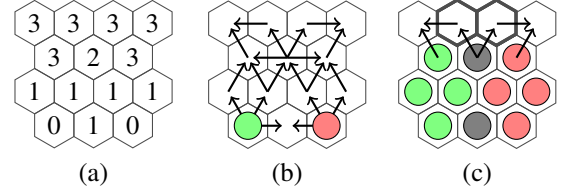

Fig. 4. Occurence of blocking nodes. (a) gray level image. (b) flooding graph, see eq (1), labeled minima (red, green). (c) reconstruction of basins get stuck at this stage. edges unexplored (black). blocking nodes (thick)

to modify the graph after a pass of basin reconstruction so that the unexplored nodes are flooded. Such nodes n, with $Q(n) = \{\}$ are called blocking nodes. We note $B = \{n \mid Q(n) = \{\}\}$ the set of all blocking nodes. We define the influence zones of each blocking node as $Z(n) = \{a \mid a \simeq^+ n, l(a) = \emptyset\}$. We suppress the edges from lower neighbors to nodes of $Z(n)$.

$$E_{\succ}^1 \leftarrow E_{\succ} \setminus \{e_{b,a} \mid a \in Z(n), b \succ a\} \quad (3)$$

We define the lowest reachable neighbors of a influence zone:

$$V' = \arg\inf_a f(a), \; l(a) \in \mathbb{N}, b \in Z(n), a \sim b.$$

Then we replace the missing graph on every influence zones so that water flows from the lowest reachable nodes:

$$E_{\succ}^2 \leftarrow E_{\succ}^1 \cup \{e_{b,a} \mid a \in Z(n), b \in V', a \sim b\}. \quad (4)$$

We complete the flooding graph on each $Z(n), n \in B$, similarly as equation (1):

$$E_{\succ}^3 \leftarrow E_{\succ}^2 \cup \\ \{e_{b,a} \mid d_{Z(n)}(a, V') > d_{Z(n)}(b, V'), a \sim b\} \quad (5)$$

Basin reconstruction can now explore the remaining nodes using $\mathscr{G}(N, E_{\succ}^3)$. Figure 5 shows an example of such modifications of the flooding graph.
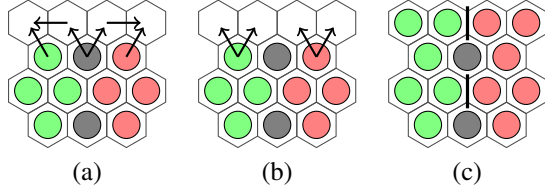

(a)　　　　　(b)　　　　　(c)

Fig. 5. An example of a modified graph. (a) result of one pass of basins reconstruction. (b) modified flooding graph, edges to lowest reachable nodes. (c) final basin reconstruction with nodes at value W (gray). Note that obtained watershed is discontinuous, but continuous if edges are considered as watershed (thick edges).

5. EVALUATION

The proposed method produces result equal to the unbiased watershed as described in [8]. The watershed line is not necessarily contiguous in a discret grid, as no biased choice is done. However, it is contiguous if we consider watershed to be on edges and nodes, see Figure 5(c).

　　Computations of minima, distances, and reconstructions can be realized using simple queues as a propagation mean. Parallelism can be achieved for reconstructing basins using an atomic synchronization on each node. Minimum detection can also be parallelized and it requires no synchronization in a shared memory context.

5.1. Time measurement

We ran experiments on an Intel(R) Xeon(R) CPU E5-2650 v2 based shared memory parallel computer with 8 cores, running at a 2.60Ghz clock frequency. We performed time measurements using synthetic images generated by a distance function on random seeds. We compared our method, the Beucher's hierarchical queue method [8] and the geodesic thickening Meyer [2]. Figure 6 shows the obtained results. The proposed method is almost 10 times faster than [8], and 100 times faster than [2]. The obtained speed-up is high due to the use of atomic synchronisation.

6. CONCLUSION

We propose a method to calculate the watershed lines in $O(n)$ that requires no sorting of pixels with respect to their elevations. It produces identical results to the unbiased version of watershed using the hierarchical queue proposed by Beucher in [8]. This method is faster than conventional methods. Removing the sequential nature of the transform allows us to
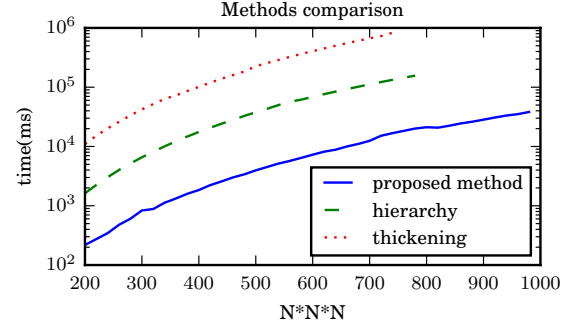
(a)
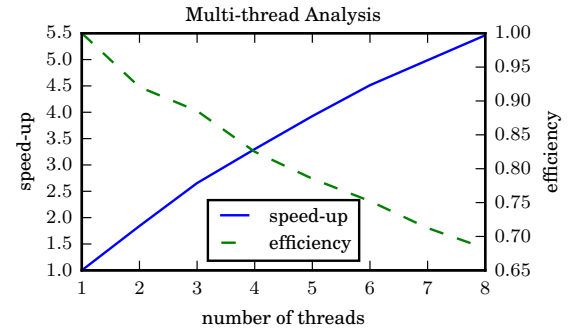
(b)

Fig. 6. Performances of the proposed method. (a) Comparison to the state-of-the-art methods. (b) Multi-thread Analysis.

reach higher performances by exploiting parallel computation such as MIMD and SIMD paradigms. This approach shows that transformations of morphological mathematics that were considered difficult to parallelize efficiently can be optimized for recent architectures using simple operators such as arrowing graphs and atomic propagation.

7. REFERENCES

[1] S. Beucher and F. Meyer, "The morphological approach to segmentation : the watershed transformation," *Optical engineering-New York-Marcel Dekker Incorporated*, vol. 34, pp. 433–433, 1993.

[2] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, pp. 113–125, jul 1994.

[3] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," *International Workshop on Image Processing : Real-time Edge and Motion Detection/Estimation*, sep 1979.

[4] S. Beucher, *Segmentation d'Images et Morphologie Mathématique*, Ph.D. thesis, Ecole Nationale Supérieure des Mines de Paris, jun 1990.

[5] F. Meyer, "Construction du watershed par fléchage et elagage des chemins d'inondations," Tech. Rep., Centre de Morphologie Mathématique, oct 2014.

[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] F. Meyer, "Un algorithme optimal de ligne de partage des eaux," *Proc. Huitime Congrs AFCET*, pp. 847–859, 1991.

[8] S. Beucher, "Algorithmes sans biais de ligne de partage des eaux," Tech. Rep., Centre de Morphologie Mathématique, fev 2002.

[9] A. Bieniek, H. Burkhardt, H. Marshner, M. Nelle, and G. Schreiber, "A parallel watershed algorithm," Tech. Rep., Technische Universitat Hamburg-Harburg, sep 96.

[10] S. Beucher, F. Lemonnier, and R. Sasportas, "Réalisation de la ligne de partage des eaux par file d'attente hierarchique parallèle," Tech. Rep., Centre de Morphologie Mathématique, jun 97.

[11] L. Gillibert and D. Jeulin, "Stochastic multiscale segmentation constrained by image content," *ISMM'11 Proceedings of the 10th international conference on Mathematical morphology and its application to image and signal processing*, pp. 132–142, 2011.

[12] M. Couprie and G. Bertrand, "Topological grayscale watershed transformation," *SPIE Vision Geometry VI Proceedings*, vol. 3168, pp. 136–146, 1997.

[13] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *Pattern Analysis and Machine Intelligence, IEE Transactions*, vol. 33, no. 7, pp. 1384–1399, jul 2011.

[14] J. Van Neerbos, L. Najman, and M. H. F. Wilkinson, "Towards a parallel topological watershed: first results," *Mathematical morphology and its applications to image and signal processing, lecture notes in computer science*, pp. 248–259, 2011.

[15] E Dejnozkova, *Architecture dédiée au traitement d'image base sur les équations aux dérivées partielles*, Theses, École Nationale Supérieure des Mines de Paris, 2004.

[16] E. Dejnozkova and P. Dokládal, "A parallel algorithm for solving the eikonal equation," *Acoustics, Speech and Signal Processing, Proceedings. (ICASSP '03). 2003 IEEE*, vol. 3, pp. III–325–8, apr 2003.