THE TIE-ZONE WATERSHED: DEFINITION, ALGORITHM AND APPLICATIONS

*Romaric Audigier**, *Roberto de A. Lotufo*

Dept. of Computer Engineering and Industrial Automation
FEEC – State University of Campinas, SP, Brazil
audigier@dca.fee.unicamp.br, www.dca.fee.unicamp.br/∼lotufo

Michel Couprie

Laboratoire A^2SI – ESIEE
Noisy-Le-Grand, France
coupriem@esiee.fr

ABSTRACT

In this work, a new type of watershed transform is introduced: the Tie-Zone WaterShed (TZWS). This region-based watershed transform does not depend on arbitrary implementation and provides a unique and optimal solution. Indeed, many solutions are sometimes possible when segmenting an image with a watershed algorithm. In this case, the TZWS assigns each pixel to a catchment basin (CB) if in all solutions it belongs to this CB. Otherwise, the pixel is said to belong to a tie-zone (TZ). We propose an efficient algorithm based on Image Foresting Transform (IFT) which computes the TZWS transform as a shortest-path forest. Finally, two applications of this TZWS are presented: bounding intervals for segmented objects' extensions and a progressive segmentation procedure.

1. INTRODUCTION

The watershed (WS) transform is a well-known and powerful segmentation tool for morphological image processing. It was first introduced by Beucher and Lantuéjoul [1] for contour detection and applied in image segmentation by Beucher and Meyer [2]. Nowadays, there are many definitions and algorithms of watershed transforms in literature. Roerdink and Meijster [3] give a comparison of some of them. The algorithm of Vincent and Soille [4] is based on immersion simulation: the image is represented by a topography inundated by water that springs from regional minima. The watershed lines are dams constructed for separating the growing catchment basins (CB) corresponding to minima. The algorithm of Meyer [5] computes the WS transform by solving a shortest paths problem with respect to a topographical distance function.

In the numerous WS algorithms, variations may first occur in the input: all regional minima; only imposed minima to avoid oversegmentation (WS from markers [2]); or grayscale markers [6] to specify the depth (handicap) of some imposed minima. Then, the output may be of different types: 'line-algorithms' return separating WS lines that are sometimes valued (as in the topological watershed [7, 8] which conserves the saliency between minima) whereas 'region-algorithms' return labeled regions (the CBs) that form a partition of the image.

0	1	1	0

(a)

A	A	W	B

(b)

A	W	B	B

(c)

0	1	1	1	0

(d)

A	A	A	B	B

(e)

A	A	B	B	B

(f)

Fig. 1. Original images **(a)(d)** and two possible labeled WS outputs (raster or anti-raster scan) of a line-algorithm **(b)(c)** or a region-algorithm **(e)(f)**. W represents the WS line.

In lots of (line- or region-) algorithms, the result varies with implementations (scanning order and other arbitrary processing order) or may be inconsistent with the WS definition as observed in [3]. This variation due to implementation can be insignificant in some cases (1 pixel bias for the line/region position, see Fig. 1) but in other cases, it becomes considerable: in some images an entire region is reached passing by a 'bottleneck' pixel and consequently included to the first (or last) CB that invades the bottleneck (like in Fig. 2(k)-(n)). Thus, the problem does not occur only on plateaux. Furthermore, it is of theoretical interest having a unique solution for the WS transform. These arguments encouraged the investigation of a WS definition that would result in a unique and consistent solution.

The Image Foresting Transform (IFT), introduced by Falcão, Lotufo and Stolfi [9] and based on Dijkstra algorithm [10], provides a sound framework for the efficient implementation of many image processing operators [11]. For instance, the WS transform is computed as a problem of trees of minimal paths.

This paper is organized as follows. In section 2, an overview of the IFT framework is given to define in this context the Tie-Zone Watershed (TZWS) that results in a unique solution, regardless of implementation. An efficient algorithm is presented in section 3 and finally two applications are described in section 4: a bounding interval for

*R. Audigier is funded by CAPES.

each segmented object's extension and a progressive segmentation procedure.

2. DEFINITION

2.1. Overview of the Image Foresting Transform (IFT)

Under the IFT framework, an image is seen as a weighted graph $G = (V, A, I)$ where each pixel (or voxel in 3D) is represented by a node or vertex $v \in V$ with intensity $I(v)$ (I is a map from V to $\mathbb{Z}$ for digital image). An arc $\langle u, v\rangle \in A$ exists between vertices u and v when the corresponding pixels are adjacent according to the defined adjacency (usually 4- or 8-adjacency in 2D and 6- or 26-adjacency in 3D). A path from a node u to a node v in a graph (V, A, I) is a sequence $\langle u = v_1, v_2, \dots, v_n = v\rangle$ of nodes of V such that $\forall i = 1 \dots n-1$, $\langle v_i, v_{i+1}\rangle \in A$. A path is said simple if all its nodes are different from each other. Let $S \subseteq V$ be a set of particular nodes s_i called seeds or markers.

For a given weighted graph $G = (V, A, I)$ and a set $S = \{s_i\}$ of seeds, an IFT is a *directed forest* F of G, i.e. a directed acyclic subgraph[1] F of G, such that (i) there exists for each node $v \in V$ a *unique and directed simple path* $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path has a *minimum cost* for linking v to any seed of S, according a specified path cost function f_C.

Assume that the arcs $\langle u, v\rangle$ are weighted with the gray-level $I[v]$ of the pixel corresponding to v. Assume that the seed nodes correspond to the regional minima of the image (or to imposed minima, i.e. markers). If the path cost function is defined as the 'maximum arc' function f_{max},

$$f_{\max}(\langle v_1, v_2, \dots, v_n\rangle) = \max\{h(v_1), I(v_2), \dots, I(v_n)\}$$

where h is a fixed but arbitrary *handicap cost* [6], the IFT computes a region-WS transform where each tree of the forest [2] corresponds to a CB. Note that all vertices (pixels) are covered by this forest. The IFT can result in many optimal forests because many paths of minimum cost are sometimes possible. The set of all optimal forests is denoted by $\mathbf{\Phi}$.

The optimality of the WS by IFT was proved in [12] where a two-component lexicographic cost function f_{LC} was proposed to mimic the flooding process and handle with plateaux too: $f_{LC} = (f_{\max}, f_d)$. The first component, of highest priority, is the max-arc function and represents the flooding process. The second one makes different waters propagate on plateau at a same speed rate:

$$\begin{aligned} f_d(\langle v_1, v_2, \dots, v_n\rangle) &= \max_{k\in[0,n-1]} \{k, C[v_n] = C[v_{n-k}]\} \\ C[v_n] &= f_{\max}(\langle v_1, v_2, \dots, v_n\rangle) \end{aligned}$$

[1] The graph $G' = (V', A')$ is a subgraph of G if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$.

[2] A tree of the forest F is a connected component of F.

This lexicographic path cost, inspired from Meyer's topographical distance strategy [5], is very simple to compute, avoids a prior lower completion on image with plateaux, and provides partitions that seem to be more equitable (on plateaux) than when only the maximum cost is used.

2.2. The Tie-Zone Watershed (TZWS) transform

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image-graph. We propose then a new definition of watershed transform in the IFT context which results in a unique partition.

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

$$\begin{aligned} CB_i &= \{v \in V,\ \ \forall F \in \Phi,\ \ \exists \pi(s_i, v) \text{ in } F\} \\ T &= V \setminus \bigcup_i CB_i \end{aligned}$$

If a node is in the tie-zone, it means that it could be included in different CBs without affecting the forest optimality. CBs are only the common part of all optimal solutions whereas differing parts are considered TZ. Therefore, the tie-zone existence prevents from making any arbitrary choice between optimal solutions. Consequently, the TZWS solution is defined without ambiguity.

Note that this definition does not produce watershed lines but only regions: catchment basins and tie zone. They form together a *unique* optimal partition of the image. If all pixels are assigned in a catchment basin, the tie zone will be empty. This situation can occur when the lexicographic path-cost function unties growing CBs on plateaux. So, the watershed transform possibly does not contain any tie zone.

Unlike in the WS by IFT, each CB corresponds to a tree *or* part of it, while the TZ is composed of many terminal parts of trees as in the example of Fig. 2(b).

3. ALGORITHM

In this section, we present an efficient algorithm that labels the image in order to obtain a TZWS. It is based on Dijkstra's shortest path algorithm [10] and utilizes an ordered queue Q where each bucket has a FIFO policy. Note that the second component C_2 of the lexicographic cost is *not* intrinsically computed by the FIFO policy and must be explicit in the TZWS by IFT in order to prevent 1-pixel bias.

The algorithm input is: the image (or scene) as a weighted graph $G = (V, A, I)$, the seed node set S with associated labeling function λ and handicap function h. The priority queue Q is initially empty: `DequeueMin` removes from Q and returns the node of minimum cost; `Enqueue`(p, c) inserts node p in Q at priority (cost) c bucket. We denote the neighborhood of a node $p \in V$ by:

$N_G(p) = \{q \in V, \langle p, q\rangle \in A\}$. Label map L corresponds to the TZWS result, map P gives each node's predecessor in the tree and maps C_1,C_2 give the lexicographic cost of an optimal path from a seed to each node.

The beginning of the algorithm (lines 1 to 11) is identical with the IFT algorithm in [9]. Lines 12-16 are TZWS-specific. In line 12, the second component of lexicographic cost is incremented, as water propagates on plateau. Lines 13 to 16 detect the nodes where paths from (at least) two seeds ($L[p] \neq L[v]$) tie, i.e. have same costs (C_1 and C_2).

Algorithm 1: TZWS by IFT with lexicographic path cost.

1. $\forall p \in V,\ C_2[p] \leftarrow 0;\quad done(p) \leftarrow$ FALSE;
2. $\forall p \notin S,\ C_1[p] \leftarrow \infty;\quad L[p] \leftarrow$ NIL; $\quad P[p] \leftarrow$ NIL;
3. $\forall p \in S,\ C_1[p] \leftarrow h(p); L[p] \leftarrow \lambda(p);\ \ P[p] \leftarrow p;$ `Enqueue`$(p, h(p))$;
4. **while** `QueueNotEmpty`,
5. $\quad v \leftarrow$ `DequeueMin`; $done(v) \leftarrow$ TRUE;
6. $\quad \forall p \in N_G(v)$ **and** $done(p) =$ FALSE,
7. $\qquad c \leftarrow \max\{C_1[v], I[p]\}$;
8. $\qquad$ **if** $c < C_1[p]$,
9. $\qquad\quad$ **if** p in Q, `Dequeue`(p);
10. $\qquad\quad C_1[p] \leftarrow c$; $L[p] \leftarrow L[v]$; $P[p] \leftarrow v$;
11. $\qquad\quad$ `Enqueue`$(p, C_1[p])$;
12. $\qquad\quad$ **if** $c = C_1[v]$, $C_2[p] \leftarrow C_2[v] + 1$;
13. $\qquad$ **else**, **if** $c = C_1[p]$ **and** $L[p] \neq L[v]$,
14. $\qquad\quad$ **if** $c = C_1[v]$,
15. $\qquad\qquad$ **if** $C_2[p] = C_2[v] + 1$, $L[p] \leftarrow$ TZ;
16. $\qquad\quad$ **else** $L[p] \leftarrow$ TZ;

This algorithm is fast and has the same speed performance as the IFT-WS [9]. The solution of TZWS is optimal because it is based on IFT, it keeps therefore the optimality of the shortest-path forest solution as demonstrated in [12, 9]. Besides, the other algorithms generally depend on arbitrary decisions in processing order (which pixel is removed first from a priority bucket of the queue?) that are not in the strict definition of WS and that introduce bias (see the different solutions of Fig. 2(i)-(n)). Note that the case of buttonhole shown in Fig. 2 is similar to configurations found in real images, as refered by [8]. Observe that the bias problem does not occur only on plateaux and may be unacceptable for some applications (e.g. precise measures on segmented objects).

4. APPLICATIONS

The Label Merging Algorithm (LMA) is a useful variation of the previous algorithm. When waters from different minima are merging, it assigns a new blended label to the region invaded by these waters. Substitute in algorithm 1 TZ label (lines 15-16) by `MergeLabels`$(L[p], L[v])$. So, the final labeled image allows a traceability on tie zones: it informs exactly which and how many trees of different labels tied together at each node. There is no more one TZ label but so many as the distinct label mergings (see Fig. 2(c)). This merging information can be used in several applications.

4.1. Maximum extension of segmented objects

As we saw, the proposed TZWS gives sets of nodes that are unequivocally associated with specific seeds, the CBs: they represent the minimum extensions E^i_{min} of segmented objects O^i and form, with the tie zone T, a partition of the image (V) (see Fig. 2(b)): $\forall i \neq j, E^i_{min} \cap E^j_{min} = \emptyset$, $E^i_{min} \cap T = \emptyset$ and $V = \bigcup_i E^i_{min} \cup T$.

The LMA allows to obtain the maximum extension E^i_{max} of each segmented object, composed by the corresponding E^i_{min} and all nodes assigned with merged labels where label i tied. Therefore, one can be sure that with a defined set of markers/minima, all WS algorithms (with same adjacency and cost definitions) will result in segmented objects O^i in these limits: $E^i_{min} \subseteq O^i \subseteq E^i_{max}$ (see Fig. 2(d)-(f)). Consequently, measures on segmented objects can be bounded. Moreover, in interactive segmentation procedures, these boundaries can help the user to correct or determine new markers for further resegmentations.

4.2. Progressive segmentation

As the final goal of segmentation is to partition images into objects, it is desirable not to have large undefined regions. A thinning procedure can therefore be used. We propose now a progressive segmentation in order to reduce the thick tie-zones without making implementation-dependent choices.

The number of labels that tied together at each node can be seen as the degree of liability of a single-label assignement. More labels tied at a node, more unliable would become a single-label assignment. The number of tied labels NTL is computed during the LMA at each call of the `MergeLabels` function. The map of NTL can be used to resegment the result of a first TZWS. This step is repeated until stability of the segmentation result. In most of cases, the thick tie zones will be progressively thinned or desappear. It is another possible option for the segmentation task.

In this new procedure, all regions segmented with certitude (minimum extensions) are definitively labeled whereas the tie zones have to be conquered in the next step by these labeled regions. The tie zones' original topography is then modified to allow a new segmentation by TZWS: it is modified into plateaux whose level is proportional to the number of candidate labels (Fig. 2(g)). Consequently, the proximity criterion progressively unties the tie-zones (Fig. 2(h)).

5. CONCLUSIONS AND FUTURE WORK

In this work, we introduced a new definition of watershed transform under the IFT framework: the Tie-Zone Watershed. It provides an optimal, unique and, thereby, unbiased solution: it is not implementation-dependent. We presented an efficient algorithm based on Dijkstra's one and a simple lexicographic cost, and proposed finally two possible applications of the TZWS. Firstly, it provides limits for the segmented objects' extensions which could guide the user in interactive segmentation process or bound measures on objects. Secondly, a procedure of progressive segmentation based on TZWS is described: the number of merging CBs is the key to this tie-zone thinning proposal.

As the TZWS is the consensus of all IFT-based solutions, the resulting partition is unique and unbiased. In future work, we plan to characterize the robustness of segmentation with tie-zone features. For example, the tie-zone size can be a warning of potential bias or unreliability in segmentations. We also intend to compare the TZWS with hierarchical WS approach and to investigate the relation of the bottleneck phenomenon (in bold in the Fig. 2(a)) with pass-value and saliency concepts, as well as similarities with the topological watershed.

6. REFERENCES

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

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

[3] J.B.T.M. Roerdink and A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies," *Fund. Inf.*, vol. 41, pp. 187–228, Jan. 2000.

[4] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. on PAMI*, vol. 13, no. 6, pp. 583–598, 1991.

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

[6] R.A. Lotufo, A.X. Falcão, and F.A. Zampirolli, "IFT-watershed from gray-scale marker," in *SIBGRAPI'02*, Fortaleza, Brazil, Oct. 2002, pp. 146–152, IEEE CS.

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

[8] L. Najman and M. Couprie, "Watershed algorithms and contrast preservation," in *Discr. Geom. Comp. Imagery*. 2003, vol. 2886 of *LNCS*, pp. 62–71, Springer.

[9] A.X. Falcão, J. Stolfi, and R.A. Lotufo, "The IFT: theory, algorithms, and applications," *IEEE Trans. on PAMI*, vol. 26, no. 1, pp. 19–29, Jan. 2004.

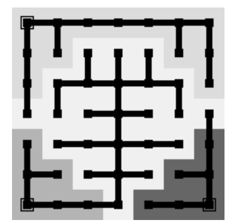
[10] E.W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, 1959.

[11] A.X. Falcão, B.S. da Cunha, and R.A. Lotufo, "Design of connected operators using the IFT," *SPIE on Medical Imaging*, vol. 4322, pp. 468–479, Feb. 2001.

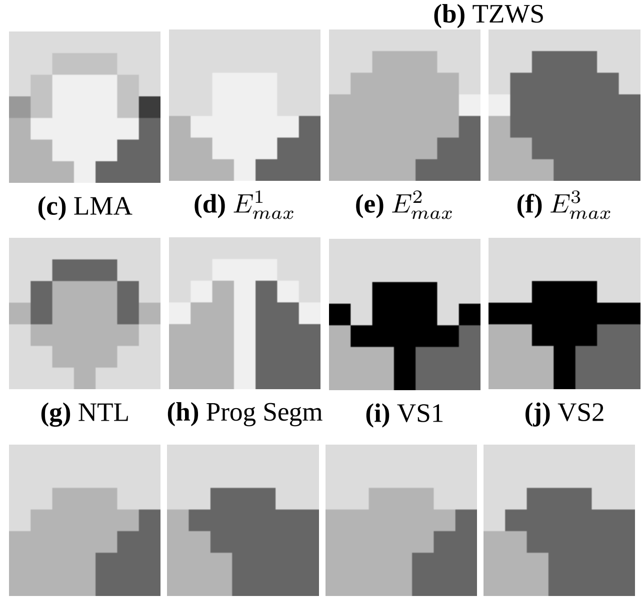
[12] R.A. Lotufo and A.X. Falcão, "The ordered queue and the optimality of the watershed approaches," in *5th ISMM*, Palo Alto, USA, June 2000, pp. 341–350.

0	1	1	1	1	1	1
1	20	**20**	**20**	**20**	20	1
1	**20**	16	14	17	**20**	1
30	30	14	13	15	30	**30**
1	30	13	12	14	30	1
1	30	30	11	30	30	1
0	1	1	**10**	1	1	0

(a)



(b) TZWS



(c) LMA **(d)** E^1_{max} **(e)** E^2_{max} **(f)** E^3_{max}

(g) NTL **(h)** Prog Segm **(i)** VS1 **(j)** VS2

(k) D1 **(l)** D2 **(m)** D3 **(n)** D4

Fig. 2. **(a)**: Original grayscale image (3 minima, 8 bottlenecks). **(b)**: TZWS using 4-adjacency: 3 CBs (gray), TZ (white), forest (black). **(c)**: Result of the Label Merging Algorithm. **(d)-(f)**: Each image extends a specific CB_i to its E^i_{max} while remaining CB_j have only minimum extensions. Remaining TZ in white. **(g)**: Map of Number of Tied Labels. **(h)**: Result of the progressive segmentation: TZ (in white) is thinned. **(i)-(j)**: Watersheds (black) by Vincent and Soille's algorithm (raster or anti-raster scan). **(k)-(n)**: Watersheds by Dijkstra-IFT varying scanning-order.