International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 11 Issue 7, July-2022, Impact Factor: 7.957

Work Flow of Immersion Based Watershed Algorithm

Ramya B K

M. Tech, Department of CSE, UBDTCE, Davangere, Karnataka, India

ABSTRACT

One of the primary topics of research in the discipline of topography is the identification of watersheds. It is essential in various applications, such as assessing flood risk and sustainability. The watershed method has shown to be a very helpful and potent tool in many various applications aside than topography, such as picture segmentation, since its first creation. In this article, we examine the most significant research on watershed algorithms, including studies on parallel techniques and issue over-segmentation. Future research and unresolved issues are also examined.

Keywords: Topography, watersheds, over-segmentation, flooding.

INTRODUCTION

Watersheds are the basic themes in the subject of topography. The division of contiguous catchment basins is termed as a watershed. The approach was reportedly proposed by S. Beucher and C. Lantuejoul in 1979. The vital plan was to install a water source at each regional low point of the relief, flood the whole relief with water, and construct barriers where several water sources converged. By flooding, the resultant group of obstacles creates a watershed. a collection of upgrades referred to as Priority-Flood.

Watershed segmentation can be implemented using either the immersion method or the rainfall method. Conceptually, the rainfall method begins at a high height and descends to a low altitude, whereas the immersion approach begins at a low altitude and ascends to a high altitude. Recently, the former appeared to have more followers (such as Vincent and Soille), while the latter also had its adherents (e.g., Mortensen and Barrett). It was unclear whether strategy was more effective and whether the two procedures could produce the exact same segmentation results.

WORK FLOW OF IMMERSION APPROACH

A. Immersion approach

The first tactic that will be examined is immersion (also called flooding). Beucher and Lantuejoul initially created it and introduced it in 1979 for contour detection in photographs. The fundamental plan was to place a water source at each regional low point of the relief, flood the whole relief with water, and construct barriers where several water sources converged. By flooding, the resultant group of obstacles creates a watershed. Since then, this algorithm has undergone a series of enhancements together referred to be Priority-Flood.

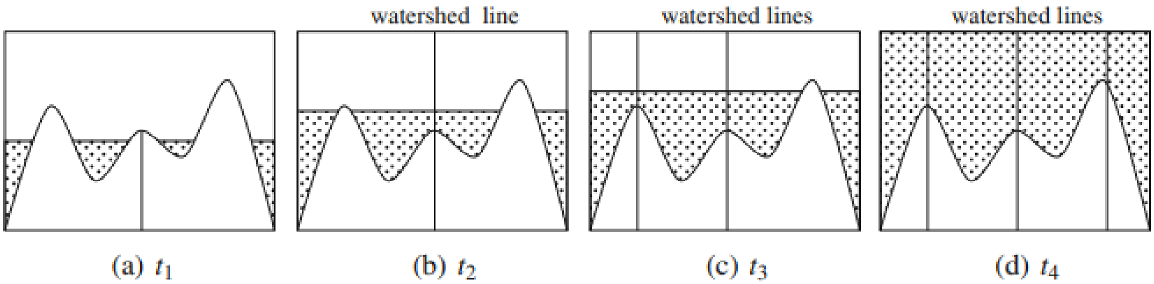

Figure 1: Immersion-based watersheds

B. Algorithm

define mask -2 /* initial value of a threshold level */
define wshed 0 /* value of the pixels belonging to the watersheds */
define init -1 /* initial value of im0 */
define inqueue -3 /* value assigned to the pixels when they are put into the queue */

- input: im_i decimal image;

output: im_0, image of the labeled watersheds; the labels are 1, 2, 3, etc.

- Initializations:

Value init is assigned to each pixel of im_0: Vp E D_{imo}, $im_0(p)$ = init;
int_label→0;
flag: boolean variable;

- Sort the pixels of im_i in the increasing order of their gray values (in the range [h_{min}, h_{max}]).
- For h← h_{min} to h_{max} { /* geodesic SKIZ of level h-1 inside level h */ }

for every pixel p such that $im_i(p)$ = h {
$m_o(p)$→mask;
if there exists p' € N(p) such that $im_0(p')$ > 0 or $im_o(p')$ = wshed { /* ($N_G(p)$ is the set of neighbors of p) */
$im_o(p)$← inqueue;
fifo_add(p); }
}
while fifo_empty() = false {
p←fifo.first();
For every pixel p' € NG(p) {
if im0(p') > 0 { /* i.e., p' belongs to an already labelled basin */
if (imo(p) = inqueue or (imo(p) = wshed and flag = true)) {
$im_o(p)$←$im_0(p')$;
}
else if (imo(p) > 0 and im0(p) ≠ imo(p')) {
im0(p)←wshed; flag ←false;
} }
else if imo(p') = wshed {
if im0(p) = inqueue {
im0(p)←wshed; flag←true;
} }
else if im0(p') = mask {
im0(p')←inqueue; fifo_add(p');
} } }
For every pixel p such that $im_i(p)$ = h { /* check whether new minima have been discovered */
if im0(p) = mask {
current_label←current_label+1;
fifo_add(p); im0(p)←current_label;
while fifo_empty() = false {
p' ←fifo_first();
For every pixel p" € NG(p') {
if im0(p") = mask { fifo_add(p"); imo(p") ←current_label; }
} }
} }
}

C. Work Flow

Step1: Gray scale image is given as input to the watershed algorithm, let the input image is.

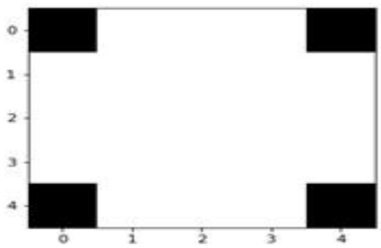

Figure 2: Input image

```
[[ 0 60 60 60  0]
 [60 60 60 60 60]
 [60 60 60 60 60]
 [60 60 60 60 60]
 [ 0 60 60 60  0]]
```

Figure 3: corresponding pixel values

- Initialize all values to -1 in separate matrix

	0	1	2	3	4
0	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	-1	-1	-1	-1	-1

Figure 4: Initializing all the pixels to -1.

- Then reshape and sort the pixel with increasing intensity, in the given image 0 intensity pixels comes first then pixels with intensity 60 are arranged.

	0
0	0
1	0
2	0
3	0
4	60
5	60
6	60
7	60
8	60
9	60
10	60
11	60
12	60
13	60
14	60
15	60
16	60
17	60

Figure 5: Sorted pixels in increasing intensity

- In the given image there are two level of values:

Level1 - 0
Level2 - 60

Step2: Take each level of values process them label them.

- First level values are taken-i.e, level1 (0,0,0,0) and they are marked as -2.

	0	1	2	3	4
0	-2	-1	-1	-1	-2
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	-2	-1	-1	-1	-2

Figure 6: Marking first level pixels

Since these values are different minima's, so they are given with different labels(1,2,3,4) .

	0	1	2	3	4
0	1	-1	-1	-1	2
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	3	-1	-1	-1	4

Figure 7: Labeling all the minima's with different labels.

- Processing of first level values is complete, now second level (60,60) values are taken and they need to be processed.
- For each pixel it marks as -2, then it checks whether the neighbors of that pixel is already labeled or not, if any of its neighbors are labeled, then that pixel is made as -3(it is kept in queue).
- If (0,1) coordinate pixel is considered then its neighbors are (0,0) (0,2) (0,1) (1,1) (1,2).
- Here (0,1) pixel neighbor is (0,0) which is already labeled as 1, so (0,1) pixel is made as -3(kept in queue) for further process.

	0	1	2	3	4
0	1	-2	-1	-1	2
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	3	-1	-1	-1	4

Figure 8: Marking second level pixels

	0	1	2	3	4
0	1	-3	-1	-1	2
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	3	-1	-1	-1	4

Figure 9:

	0	1	2	3	4
0	1	-3	-2	-3	2
1	-3	-3	-2	-3	-3
2	-2	-2	-2	-2	-2
3	-3	-3	-2	-3	-3
4	3	-3	-2	-3	4

Figure 10:

- The pixels which are -3(in queue) are labeled first Fig (10).
- For these pixels it checks for neighbors
  — If any neighbor is labeled(1,2,3 or 4) then it gives same label to it.
  — After labeling each pixel, it again check for its neighbors and make them as - 3(keep them in queue) for future decision.

Now all the pixels which are neighbors of labeled pixels they are given with same label Fig(12).

	0	1	2	3	4
0	1	1	-3	-3	2
1	-3	-3	-3	-3	-3
2	-2	-2	-2	-2	-2
3	-3	-3	-2	-3	-3
4	3	-3	-2	-3	4

Figure 11:

	0	1	2	3	4
0	1	1	-3	2	2
1	1	1	-3	2	2
2	-3	-3	-3	-3	-3
3	3	3	-3	4	4
4	3	3	-3	4	4

Figure 12:

- Now the pixels which are in middle of two or more labels Fig 11, they are considered as watershed and they are labeled as 0.

— For this process for each pixel it check its neighbors

	0	1	2	3	4
0	1	1	1	2	2
1	1	1	-3	2	2
2	-3	-3	-3	-3	-3
3	3	3	-3	4	4
4	3	3	-3	4	4

Figure 13:

	0	1	2	3	4
0	1	1	0	2	2
1	1	1	-3	2	2
2	-3	-3	-3	-3	-3
3	3	3	-3	4	4
4	3	3	-3	4	4

Figure 14:

	0	1	2	3	4
0	1	1	0	2	2
1	1	1	0	2	2
2	0	0	0	0	0
3	3	3	0	4	4
4	3	3	0	4	4

Figure 15:

For the pixel (0,2) Fig 14, while labeling it initially check for its first neighbor (0,1) which is labeled as 1 Fig 15, so the same label is given to (0,2) pixel Fig 13, but while checking with (0,3) neighbor ,which is labeled as 2.Hence at this condition this pixel (0,2) has equal probability to label 1 and 2. Hence these kinds of pixels are made as watershed –i.e., they are labeled as 0 Fig 15.

LITERATURE SURVEY

A short background of the watershed algorithm was provided by Fernand Meyer[1] to explain the watershed notion and its use in segmentation. The preferred method for morphological segmentation is the watershed. Minimizing the size, geodesic distances, and skeletons by zone of influence are some of its main benefits.

Recent developments in watershed algorithms presented by R. Romero-Zaliz et al. [2] centre on the usage of various data structures to increase the proposed algorithms' efficiency. The majority of the work done has focused on developing image segmentation techniques, particularly for biological and medical pictures. Parallel strategies also appear to be an area that is constantly developing. Strangely, rather of being published in journals, many of the more intriguing concepts were instead presented at conferences. In the most recent years of study, this pattern appears to be changing.

Gonzalez, Mariela Azul, and others[3] Clustering is not only automated but also more accurate in segmenting the trabeculae than the fuzzy logic-based approach. Additionally, it has lower processing costs than the clustering-based technique. The technique is a useful tool for automatically obtaining internal markers for the watershed transform. It may be used on highly textured pictures, such as microscopic images, whose processing is difficult and incompatible with standard techniques.

A technique for segmenting the picture based on mathematical morphology was put out by Beucher (1991)[4]. Image segmentation can be done in one of two ways: region-based or boundary-based. The region-based technique of "watershed segmentation" finds commonalities between pixels and regions. The flooding procedure is used to apply watershed modification to the grayscale pictures. By adopting marker controlled watershed segmentation, the issue of excess segmentation is minimised. Hierarchical segmentation is a different approach that has been suggested. Beginning with the graph representation of the pictures based on the mosaic image transform, this technique is specifically effective for determining the stages of segmentation.

Anthony et al[5] .'s use of a multiresolution version of the watershed segmentation algorithm implements a variant of the conventional watershed segmentation approach. To cut down on computing costs and get around the over-segmentation issue that the classic watershed technique has, this method creates a scale space representation using a watershed pyramid. High precision edges are produced using these pyramidal structures. This technique is used to get all of the edges contained in the picture with the goal of edge detection in an image, not necessarily a face.

In order to normalise the contrast of the picture and guarantee that there is neither over segmentation nor under segmentation, Malik et al. [6] recommend using the typical watershed transform. The picture is subjected to a probability-based pre-processing procedure called random walk. Using the image's threshold, a seed is created to serve as the random walk algorithm's starting point. The algorithm's objective is to increase picture quality such that the watershed approach may be used.

CONCLUSION

Every expert in the field of image processing is aware of how important computing efficiency is when choosing an algorithm. The current watershed implementations are all either wrong or take much too long to complete (even on specialised architecture). The over-segmentation that watershed algorithms result in is a further unresolved problem. Even though there are various ways to reduce this issue, there isn't yet a perfect answer. When establishing countours on photos that will be used for counting, over-segmentation might be a significant problem (such as blood cells).

REFERENCES

[1]. Fernand Meyer , “The watershed concept and its use in segmentation : a brief history” http://arxiv.org/abs/1202.0216v1.

[2]. R. Romero-Zaliz and J.F. Reinoso-Gordo, “An Updated Review on Watershed Algorithms”, Springer International Publishing AG 2018.

[3]. Mariela Azul Gonzalez and Virginia Laura Ballarin , “Comparing marker definition algorithms for Watershed segmentation in microscopy images ” https://www.researchgate.net/publication/255652748_Comparing_marker_definition_algorithms_for_Watershed_segmentation_in_microscopy_images?enrichId=rgreq-1a600ef3589aa5c682a35e159ca395cb-XXX&enrichSource=Y292ZXJQYWdlOzI1NTY1Mjc0ODtBUzoxMDQxNzI5MDUwNDE5MjNAMTQwMTg0ODE2NDEyMA%3D%3D&el=1_x_2&_esc=publicationCoverPdf .

[4]. S.Beucher and F Meyer, “The Morphological Approach to Segmentation: The Watershed_Transformation” https://people.cmm.minesparis.psl.eu/users/beucher/publi/SB_watershed.pdf

[5]. Anthony S Wright and Scott T. Acton, ”Watershed Pyramids For Edge Detection”, In the Proceedings of International Conference on Image Processing, 1997, Volume 2, pg 578 – 581, Oct, 1997.

[6]. J. Malik, R. Dahiya and G. Sainarayanan, “Harris Operator Corner Detection using Sliding Window Method”, International Journal of Computer Applications, vol. 22, pp. 28–37, May (2011).