Gushing and Immersion Alternative Watershed Algorithm*

Tianzhou CHEN, Jiaoying SHI

State Key Lab of CAD&CG, Zhejiang University
Hangzhou, Zhejiang 310027, P.R. CHINA

ABSTRACT

In this paper, a fast and flexible watershed algorithm in digital grayscale images is introduced, which imitating the gushing and immersion process. The algorithm is accuracy and less consuming in time and memory space. It can easily get the information of the regions, and can remove the noise during segmentation.

I INTRODUCTION

Watersheds are one of the classics problems in topography. Its definition in topography is the ridge or upland between two basins. Figuratively, a drop of water falling in one side of this line flows down until it reaches one lake, river or ocean, whereas a drop falling on the other side flows down to other ocean. The two regions which watershed line separates call the catchment basin, the two oceans are the two minima associated with these catchment basin.

In mathematical morphology, by considering gray levels as altitude information, an image \( I \) is very often regarded as a topographic relief. Morphological transformation extracts and describes topographical primitives. For example, the Minkowski opening operator removes some peaks and crest lines, whereas the Minkowski closing operator tends to fill in basins and valleys. As we shall see in this paper, the notions such as minima, catchment basin and watersheds can be well defined for grayscale image.

Quite naturally, the first algorithm for computing watersheds are found in the field of topography. Topographic surfaces are numerically handled through digital elevation models (DEM's). Automated watersheds extraction from DEM's has received increasing attention in the past few years[1]. The very local approach of the first step and the lack of objective rules to perform the second one usually lead to poor results[2]. The introduction of the watershed transformation as morphological in the field of image processing is due to H. Digabel and C. Lantuejoul. Later, a joint work of C. Lantuejoul and S. Beucher extend the watershed the more general framework of grayscale images[3]. In [3], a watershed algorithm based on immersion analogy were first proposed. According to the algorithm, the geodesic influence zones of a level inside the next one are determined via binary thickening until idempotence with structuring elements \( M \) and \( E \). The watersheds computed this way, in some special configurations, contain undesirable arcs. Beucher[4] proved that the watersheds of a function are nothing but the closed arcs of its skeleton. The nonclosed arcs of the skeleton can easily removed by thinning it until idempotence with the structuring element \( E \). This results in a highly time consuming algorithm which falls into undesirable arcs. The algorithm proposed by Friedlander in [5] is a sequential one. This procedure is relatively fast, since all steps are performed sequentially. In addition, since a labeling of the different catchment basin is used in the algorithm, undesired arcs are removed. However, the propagation step being based on the definition in terms of flow path, the resulting watershed lines may be improperly located, i.e., not even on crest lines of the image. An algorithm based on an immersion process analogy is presented by L. Vincent and P. Soille[6]. It includes two major steps: the first step consists in an initial sorting of the pixels in the increasing order of their values. In the second step, a fast computation of geodesic influence zones is enabled by a breadth-first scanning of each threshold level. This particular scanning is implemented via the use of the queue of pixels, i.e., a first-in-first-out data structure. This second step calls flooding step. The algorithm is an efficient and fast algorithm, and it is discussed in section III-A.

It is hard applying the watershed transformation in the complex image processing. As available algorithms for computing the watershed transformation are high consuming, inaccurate and inconvenient for get catchment basins’ information, the tremendous practical interest of this transformation is not as obvious as it should be. The difficulty lies in: first, watersheds transformation is very sensitive to noise. Second, the number of catchment basins are too large in medical image processing. Segmentation only using watershed transformation obviously is not enough. Removing the noise and merging the catchment basins are two effective ways. P. Soille[2] proposed filling the small catchment basins to get rid of the noise. The merge algorithm can use classic merge algorithm.

The purpose of this paper is to introduce an efficient implementation and algorithm of watershed. Roughly speaking, it is based on a sorting of the pixels in the increasing order of their gray values, and two procedures of immersion and gushing.

* This work is funded by NFS in Zhejiang Province of China under Grant 600019 and 699083
The modified immersion procedure is accurate. It can get the watershed pixels which have equal distance to three or more catchment basins without ambiguity, while the other existed algorithms will cause wrong result. The gushing procedure is introduced for fast recognizing the pixel of lakebed and getting the flood level of the catchment basin. These two procedures use fast breadth-first scanning of the plateaus enabled by a first-in-first-out type data structure. It is accurate and less memory consuming. Our algorithm is fast than available algorithms. It is general: its adaptation to any kind of underlying grid (4-, 6-, 8-connectivity...) is straightforward, and it can be fairly easily extended to n-dimensional image and even to graphs. In addition, algorithm can remove the effect brought by noise, get the information for next merge step.

II 

Definitions and Notions

Let us consider a two-dimension grayscale image \( I \) whose definition domain is denoted by \( D_I \subset \mathbb{Z}^2 \). \( I \) is supposed to take discrete(gray) values in a given range \([0,N]\), \( N \) being an arbitrary positive integer, \( p \) is the pixel in image \( I \):

\[
\begin{align*}
\left(D_I \subset \mathbb{Z}^2 \rightarrow \{0,\ldots, N\}\right) \\
p \mapsto I(p).
\end{align*}
\]

In the following, we equally consider gray scale image as numerical functions or as topographic relief.

Let \( G \) denote the underlying digital grid, which can be of any type: a square grid in four or eight connectivity, or a hexagonal grid in six connectivity. \( G \) is the subset of \( \mathbb{Z}^2 \times \mathbb{Z}^2 \).

**Definition 1**: A path \( P \) of length \( l \) between two pixel \( p \) and \( q \) in the image \( I \) is a \((l+1)\)-tuple of pixels \( \{p_0, p_1, \ldots, p_{l-1}, p_l\} \), such that \( p_0=p, p_l=q, \) and \( \forall i \in [1,l], (p_{i-1}, p_i) \in G \).

In the following, we denote \( l(P) \) the length of a given path \( P \) and also denote \( N_G(p) \) the set of the neighbors of a pixel \( p \), with respect to \( G \):

\[
N_G(p) = \left\{ p' \in \mathbb{Z}^2 : (p, p') \in G \right\}.
\]

Let us denote minima \( M \) of \( I \) at altitude \( h \) is a connected plateau of pixels with the value \( h \) from which it is impossible to reach a point of low altitude without have to climb. A minima is thus a connected and iso-intensive area where the gray level is strictly darker than on the neighboring pixels (See Fig. 1).

Naturally, we can approach the definition catchment basin and watershed using the topographic method.

**Definition 2** (Catchment basin, watershed, first definition): Let \( I \) be a grayscale image. The catchment basin \( CB(M) \) associated with the minima \( M \) is the set of pixels \( p \) of \( D_I \) such a water drop falling at \( p \) flows down along the relief, following a certain descending path called the downstream of \( p \) and eventually reaches \( M \). The lines which separate different catchment basins build what is called the watershed of \( I \).

We can proceed with the definition of catchment basins and watershed, using immersion and gushing.

By analogy, we can imagine that we have pierced holes in each regional minima of \( I \), the image \( I \) being regarded as a topographic surface. We slowly immerse our surface into a lake starting from the minima of lowest altitude. While the water level rising, at each pixel where the water coming from different minima would merge, we build a dam. When a minima \( M \) reaches the water level of lake \( h \), the water gush from the hole, until the water level reach the flood level (See Fig. 2). At that time, the water will flow to another minima through the overflow lip. After completing the gushing and building dam at water level \( h \), the water in the lake will progressively fill up. At the end of the immersion and gushing procedure, each minima is completely surrounded by dams, which delimits its associated catchment basin. The whole set of dam which has been built thus provides a tessellation of \( I \) in its different catchment basin. These dams correspond to the watersheds of \( I \).
Let us express this immersion and gushing process more formally. $I$ being the grayscale image under study, denote $h_{\text{min}}$ the smallest value taken by $I$ on its domain $D_I$. Similarly, denote $h_{\text{max}}$ the largest value taken by $I$ on $D_I$. In the following, $T_h(I)$ stands for the threshold of $I$ at level $h$: \[ T_h(I) = \{ p \in D_I, I(p) \leq h \} \] (2)

We denote $WS(C)$ the watershed associated with a catchment basin $C$, including the dams surrounded the catchment $C$. We also denote $CB(M)$ the catchment basin associated with a minima $M$. And we denote $CB_h(M)$ the subset of this catchment basin made of points having an altitude smaller than or equal to $h$:
\[ CB_h(M) = \{ p \in CB(M), I(p) \leq h \} = CB(M) \cap T_h(I) \] (3)

Concerning the minima of $I$, $\text{min}_h(I)$ refers to the set of points belonging to the minima at altitude $h$.

During the gushing process, there is an important notion flood level, which can indicates the elevation when the gushing process is finished. This means that when the water level is lower than the flood level, there is no watershed. By this notion, algorithm need not process the pixels which is lower than the flood level. Now let us define the notion visually.

**Definition 3 (Flood level, overflow level, first definition)** The flood level $FL(C)$ of a catchment basin $C$ is the infimum elevation of the pixels belonging to the $WS(C)$. The pixels set whose elevation equal to $FL(C)$ calls overflow lip $OL(C)$.

Usually, while we gush from a minima $M$, we do not know the watersheds $WS(M)$ beforehand. The purpose we led the notion flood level is to obtain the watershed. So we define the notion using the minima.

**Definition 4 (Flood level, second definition)**: The flood level $FL(M)$ associates with a minima $M$ is the elevation of the pixels, which have a downstream eventually reaches $M$, and have $p' \in N_o(p), I(p')<I(p)$, the path $P(p, M)$ have an increasing tuple:
\[ FL(M) = \inf\{ I(p), \exists P(p, M), \forall i \in [1, I(P)], I(p_{i-1}) < I(p_i), \& \exists p' \in N_o(p), \forall P(p', M), \exists i \in [1, I(P)], I(p_{i-1}) > I(p_i) \} \] (4)

Notice that the catchment basin consists of two parts: the elevation of pixels of one part is greater or equal to the $FL(M)$ and elevation of pixels of the other part is smaller than the $FL(M)$. The second pixels set lies under the water level. We call it lakebed, simulated the riverbed. The lakebed associated with a minima $M$, denotes as $LB(M)$. We call the first pixels set levee, denotes as $LE(M)$. The reason of dividing the catchment basin into two part, lies in decreasing the computation. The pixels in lakebed can be directory labeled. It need not consider the distance and the plateau.

**Definition 5**: The notion lakebed associates a minima $M$, $LB(M)$, is the pixels set whose elevation is smaller than $FL(M)$, and have a downstream eventually reaches $M$.
\[ LB(M) = \{ p, I(p) < FL(M), \exists P(p, M), \forall i \in [1, I(P)], I(p_{i-1}) \geq I(p_i) \} \]
\[ LB(I) = \bigcup_{h=h_{\text{min}}}^{h_{\text{max}}} LB(h_{\text{min}}(I)) \] (5)

**Definition 6 (Levee, first definition)**: The levee associates a minima $M$, $LE(M)$, is the pixels set whose elevation is greater than or equal to $FL(M)$, and have a downstream eventually reaches $M$.
\[ LE(M) = \bigcup_{h=FL(M)}^{h_{\text{max}}} LE(M) \]
\[ LE(I) = \bigcup_{h=h_{\text{min}}}^{h_{\text{max}}} LE(I) \] (6)
Let us suppose that \( X \) be a connected subset of the \( T_{h+1}(I) \), and \( Y = CB_h(I) \cap X \). In order to get the levee of a minima \( M \), we analyse the three possible relations of \( X \) and \( Y \) [Fig. 3]. Case A refers that \( X \) is the minimum at altitude \( h+1 \). We have introduce gushing analogy method corresponding it. For case B, It is obvious: if \( M \) is the minimum of \( Y \), then \( X = CB_{h+1}(M) \). The case B1 associates the lakebed and case B2 associates the levee. The difference between B1 and B2 is whether \( Y \) connected with \( \overline{X} \). Now, let us see the case C.

First, let us recall the definition of the geodesic distance and of the geodesic influence zones [7].

Let \( A \) be a set which is first supposed to be simply connected. The geodesic distance \( d_A(x,y) \) between two pixels \( x \) and \( y \) in \( A \) is the infimum of the length of paths which join \( x \) and \( y \) which is totally include in \( A \).

Suppose now that \( A \) contains a set \( B \) made of several connected components \( B_1, B_2, \ldots, B_k \). The geodesic influence zones \( IZ_A(B_j) \) of a connected component \( B_i \) of \( B \) in \( A \) is the locus of the points of \( A \) whose geodesic distance to \( B_i \) is smaller than their geodesic distance to any other component of \( B \).

\[
IZ_A(B) = \bigcup_{j=1}^{k} IZ_A(B_j)
\]

According the notion geodesic influence zone, we can separate the case C in Fig. 3 into two part. Supposed that \( Y_1 \) associates with the minima \( M_1 \) and \( Y_2 \) associates with the minima \( M_2 \), \( M_1 \neq M_2 \), then \( CB_{h+1}(M_1) = IZ_X(Y_1) \), \( CB_{h+1}(M_2) = IZ_X(Y_2) \). The watershed is \( WS \) [see Fig. 3].

We can rewrite the notion levee like:

**Definition 7 (Levee, second definition):** The levee \( LE(M) \) associating with the minima \( M \) is the set \( X_{h_{\text{max}}} \) obtained after following recursion:

\[
\begin{align*}
a) & \quad X_{FL(M)-1} = LB(M) \\
b) & \quad \forall h \in [FL(M), h_{\text{max}}], \quad X_h = X_{h-1} \cup IZ_{X_{h-1}}(\{X_h\})
\end{align*}
\]

Now, we obtain the following definition.

**Definition 8 (Catchment basin and watershed by gushing and immersion, second definition):** The set of catchment basins of the grayscale image \( I \) is the union of two parts:

\[
CB(I) = LB(I) \cup LE(I)
\]

The watershed of \( I \) correspond to the complement of this set in \( D_I \), i.e., to the set of point of \( D_I \), which do not belong to any catchment basin.

III Proposed Algorithm

**A. Vincent and Soille's algorithm**

Among the algorithms published, the Vincent and Soille's algorithm [6] is the fastest and efficient one. The algorithm first resort the pixels to an increasing order of their gray value. Then, algorithm uses a first-in-first-out data structure and a breadth-first scanning from \( h_{\text{min}} \) to \( h_{\text{max}} \). The principle of algorithm is:

\[
\begin{align*}
a) & \quad X_{h_{\text{min}}+1} = T_{h_{\text{min}}}(I) \\
b) & \quad \forall h \in [h_{\text{min}}, h_{\text{max}}], \quad X_h = X_{h-1} \cup IZ_{X_{h-1}}(\{X_h\})
\end{align*}
\]

Fig. 3 Three possible relations of the \( X \) and \( Y \). In case A, \( Y \) is null set. Case B refers the \( Y \) has one and only one connected component. It can separate into two case: case B1 refers \( Y \) connect with the outside, whereas case B2 refers that \( Y \) does not connect with the outside. In case C, there are two or more connected component of \( Y \), denoted as \( Y_1 \) and \( Y_2 \).
b) \( \forall h \in [h_{\min}, h_{\max}], \)
\[
X_{h+1} = \min_{h'\in(I)} I Z_{T_{h'},(I)}(X_{h})
\]

c) \( CB(I) = X_{h_{\max}} \) \( \quad (10) \)

Briefly, the flooding step is describe as:

Repeat \( h \) from \( h_{\min} \) to \( h_{\max} \):

0. Assign current distance \( I \).

1. Label the pixels at altitude \( h \) a special value \( MASK \). If there exist a pixel in its neighbor and have be success labeled, push the point to the queue, the distance from these point to the set previous labeled.

2. Pop a point \( p \) from queue.

   For every \( p' \) which is neighbor of \( p \):

   The \( p' \) is the neighborhood of \( p \) and assume the geodesic distance between \( p' \) and the set previous labeled is smaller than current distance, the relation of the label of \( p' \) and \( p \) like table 1:

3. If \( p' \) labeled as \( MASK \) and the geodesic distance between \( p' \) and the set previous labeled has not calculated, push \( p' \) to queue and assign current distance add 1 as its geodesic distance.

4. If have complete process at the distance current distance, increase the current distance.

5. If the queue is not empty, goto 3.

6. Those which have altitude \( h \) and have not processed during 2 to 4, then they are \( \min_{h}(I) \), give them a new label of catchment basin.

B. Discussion

Analyzing the algorithm in IIIA, we will find that there have some procedure need update.

First, some special cases have not been payed attention to in the algorithm. If there is a point \( p \) which is at same distance from the three or more adjacent catchment basins, the labeling of \( p \) would be incorrect. The accuracy labeling of \( p \) is watershed. But computing from the above algorithm, it will labeled as catchment basins. It is illustrated as Fig 4.

Second, there exist some unnecessary computation. The geodesic distance between those points which belong to the lakebed need no calculation.

Third, the memory consuming is great. Since the queue would store the points which have same elevation of the whole image, there would be a very large number of memory require.

In addition, the algorithm can not conveniently get the information about catchment basins during the processing, such as lakebed's volume, flooding level, etc.

C. Gushing and Immersion Alternative Algorithm

The main idea of the gushing and immersion alternative algorithm, is gushing during immersion. The main algorithm is immersion. When water touch a minima \( M \), the algorithm use the gushing method to get the lakebed of catchment basin faster. During this process, the information of a catchment basin can be easily got.

We can describe our algorithm as three step: first one is sorting. The algorithm we used is distribution algorithm[8] which resorts to address calculation. The algorithm first determined all the exact frequency distribution of each image gray level. The cumulative frequency distribution is then computed. This induce the direct assignment of each pixel to a unique cell in the sorted array.

<table>
<thead>
<tr>
<th>Old label of pixel ( p )</th>
<th>label of pixel ( p' )</th>
<th>label of pixel ( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASK</td>
<td>CB1</td>
<td>CB1</td>
</tr>
<tr>
<td>MASK</td>
<td>WSHED</td>
<td>WSHED</td>
</tr>
<tr>
<td>CB1</td>
<td>CB2</td>
<td>WSHED</td>
</tr>
<tr>
<td>CB1</td>
<td>WSHED</td>
<td>WSHED</td>
</tr>
<tr>
<td>WSHED</td>
<td>WSHED</td>
<td>WSHED</td>
</tr>
<tr>
<td>WSHED</td>
<td>CB1</td>
<td>CB1</td>
</tr>
</tbody>
</table>

Table 1. The relation of \( p \) and the neighbor \( p' \)

Fig. 4. The adjacent labeling of pixel \( p \) on a hexagonal grid is show in (a), where WS denotes watershed, and CB1, CB2, CB3 denote three catchment basins. When the scanning begin with pixel \( p2 \) and have clockwise direction, the labels of pixel \( p \) is CB1, show as (b). But while the scanning begin with \( p1 \), the labeling of \( p \) changes to CB2.
The second step is the immersion, which get the levee of the catchment basin. The algorithm uses a fast breadth-first scanning. The geodesic distance is got from the scanning. The labels of the pixels are carefully designed. Our modified immersion procedure is accurate. It can get the watershed pixels which have equal distance to three or more catchment basins without ambiguity. The pixels whose geodesic distance to two catchment basins are equal, are given a special value, hereafter denoted as WSHED1 while the pixels whose geodesic distance to three or more catchment basins are equal denoted as WSHED2.

The third step is the gushing step. It is introduced for fast recognizing the pixel of lakebed and getting the flood level of the catchment basin. The gushing procedure is different from the flood fill algorithm. Obviously, flood level can denote the end of the gushing. It uses a breadth-first scanning in the lakebed of the catchment basin. The gushing procedure will stop at the overflow lip.

The global analogy is immersion, when water level reach the minima, the gushing procedure is activated. These gushing scanning in the immersion scanning will get the accuracy result and the consuming decrease.

The previous two steps are based the first-in-first-out data structure—queue. When the program process the immersion, it uses as pure first-in-first-out structure. During the gushing process, the data structure has another function. It is used as buffer to store the contour pixels of the gushing region. These pixels are store indexed by the elevation. When algorithm processes the water level of gushing local h, it pops the pixels having the elevation local h. If the water reaches the flood level, the label of the contour pixels whose elevation is greater than the flood level need reset to initial value.

### IV Results and Discussion

Our algorithm can run in linear time with respect to the number N of pixels in the image which is proceeded. Indeed:

- In the sorting step, two and only two scanning of the whole image are necessary to construct the sorted array of pointers to pixels. An additional scanning of the frequency array is also required.

- In the next procedure, each pixel belonging to levee is scanned three times at the worst case: at each threshold level global h, all the connected pixels are first assigned value INIT. Then, they can be considered a second time during the breadth-first scanning of the plateaus at altitude h. Lastly, all the pixels at altitude h are scanned again in order to see if new minima have appeared. The pixel belonging to the lakebed, is scanned only one time to scanning in gushing processing. In the worst case, that means that there are many plateaus and little lakebed, the time consuming of algorithm is equal to which of Vincent and Soille's.

On HP720 workstation, the computation of the watersheds of twenty 150×200 NMRI image of 8 bits takes approximately 30 seconds. Compared with 34 second of the Vincent and Soille's algorithm, it is some faster.

Concerning the memory requirements, our algorithm is a little more restricting. Since the algorithm uses:

- An output image imo of the twice larger than the initial image im1. The output is coded on 2 bytes per pixel for the number of catchment basins will bigger than 255.

- A sorted array of pointers to pixels. Its size if N(number of pixels in im1) and a pointer is generally represented on 4 bytes.

- A distance image imd of the same size of im1.

- An array of pointers to pixels, which must be large enough to contain all the pixels in the queue, at each step. The large queue requirement has been distributed by gushing process. While the immersion process the elevation h, many pixels needn't push to queue since they belong to the lakebed and have been labeled before. Thus, the total memory consuming is decreased. For the twenty NMRI images, the minimum size of queue on our algorithm is 2190×4 bytes, comparing to 3130×4 bytes on Vincent and Soille's. And when the queue is using as contour buffer during the gushing process, the size of contour buffer is small. For the twenty NMRI images, the maximum contour number of one catchment basin is 250.

During the computation, algorithm can get the information of the catchment basin. Since the lakebed and levee are obtained from difference procedures, we can get the information about them. These information are well useful to remove the noise and merge the two adjacent catchment basins. For example, the information of the lakebed, such as flood level, area of lakebed, volume of lakebed, can indicate the noise. The elevation between flood level and the minima of a catchment basin which is caused by noise is small. There are some constraints can recognize the noise from the image. The constraints adopt IF-THEN rules like: IF the area of a lakebed of catchment basin is less than a threshold number, THEN the catchment basin will be noise. Using these process, the initial segmentation can be remove some noise. In addition, these information of the catchment basin, including the levee and lakebed, are important to decide the similarity of two adjacent catchment basins. If two connected catchment basins are similar enough, they are merged to one region.
V CONCLUSION

The algorithm introduced in this paper is extremely powerful compared with the existing ones. The algorithm using gushing and immersion are better than the algorithm only using the immersion. The time consuming of our algorithm is decreased because of the gushing process, and the memory consuming is reduced since the gushing distributes the memory require. In addition, the information of the catchment basins can easily got and the noise can be removed during the watershed algorithm.

REFERENCES


