Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

A Novel Implementation of Watershed Transform Using Multi-Degree Immersion Simulation

Shengcai. Peng[1], Lixu. Gu[1]
[1]Department of Computer Science, Shanghai Jiaotong University, Shanghai, China

***Abstract*—A novel implementation of watershed transform using a multi-degree immersion simulation is presented in this paper. The method is based on the improvement of traditional watershed by simulating immersion, which is originally proposed by Vincent & Soille in 1991. By changing the simulation procedure to multi-degree, which means the flood step is different on each degree of intensity, the proposed implementation resists the over segmentation problem effectively. After conducting a comparison of both the computing time and memory cost between the proposed method and a topographical distance based implementation, we draw the conclusion that the presented implementation achieves more accurate result with lower cost.**

Keywords—**watershed transform, segmentation, multi-degree, simulating immersion**

I. Introduction

Image segmentation is one of the most critical tasks in automatic image analysis. For example, segmentation is a prerequisite for quantification of morphological disease manifestations and for radiation treatment planning, for construction of anatomical models, for definitions of flight paths in virtual endoscope, for content-based retrieval by structure, and for volume visualization of individual objects. The application that provided the incentive for our work was manifestations of disease from CT/MRI data.

Many different algorithms have been proposed to address the segmentation, which may fall to two major streams: model based and region-based methods. The model based method such as Snake or Level set is relatively faster but sometimes can't achieve satisfactory accuracy, especially at the narrow sharp boundary area. The region based method like Watershed [1],[2] or Morphological operations may get more accurate results but more costly. However, Watershed can be parallelized easily to improve its performance, and has been widely used in medical image segmentation. Since S.Beucher and F.Mever firstly proposed Watershed algorithm [5], many scientists have addressed their efforts to improve it [3]. Some researchers employed diffusion algorithms before running the watershed transform because watershed is very sensitive to noises [7]. Some researchers use probability theory to compute the landscape of the image or use different strategy to compute the saliency of merge to merge the watershed images [6]. Most of these research focused on the over segmentation problem and it is widely agreed that the time and memory cost of watershed is hard to decrease because of its birth of region based. This paper focuses on both the over segmentation problem and the computing cost. As the implementation by simulating immersion is fast and flexible [4], the proposed method is based on it and improves it to have the ability to avoiding over segmentation problem.

The following parts of the paper will present the novel implementation of the watershed transform. In Section II, we firstly review the definition of the traditional watershed by simulating immersion and its implementation. Then by redefine *threshold set*, the multi-degree immersion simulation will be introduced in detail. The experimental results are presented in Section III and there is a comparison among three implementations of watershed to indicate their advantages and drawbacks. A brief discussion of the experimental results and the conclusion is given in Section IV and Section V, respectively.

II. Methodology

A. Watershed Definition

In this section we quickly review the definitions of the watershed transform, which may be viewed as a generalization of the skeleton by influence zones (SKIZ) to grey value images. Let $f: D \rightarrow N$ be a digital grey value image, with h_{min} and h_{max} the minimum and maximum value of f.

The *threshold set* of f at level h is

$$T_h=\{p \in D | f(p) \leqq h\} \tag{1}$$

The geodesic distance $d_A(a,b)$ between a and b within A is the minimum path length among all paths within A from a to b. If B is a subset of A, define:

$$d_A(a,B)=MIN_{b \in B}(d_A(a,b)) \tag{2}$$

Let $B \sqsubset A$ be partitioned in k connected components B_i, i = 1, ….., k. The geodesic influence zone [8] of the set B_i, within A is defined as:

$$IZ_A(B_i,B)=\{ p \in A | d_A(p,B_i)<d_A(p,B \setminus B_i)\} \tag{3}$$

The set $IZ_A(B)$ is the union of the geodesic influence zones of the connected components of B,

$$IZ_A(B)=\bigcup_{i=1}^{k} IZ_A(B_i, B) \qquad (4)$$

Now we give the definition of watershed by immersion. Define a recursion with the grey level h increasing from h_{min} to h_{max}, in which the basins associated with the minima of f are successively expanded. Let X_h denotes the union of the set of basins computed at level h. A connected component of the threshold set T_{h+1} at level $h + 1$ can be either a new minimum, or an extension of a basin in X_h: in the latter case one computes the geodesic influence zone of X_h within T_{h+1}, resulting in an update X_{h+1}. Let MIN_h denote the union of all regional minima at altitude h, and define

$$\begin{cases} X_{h_{\min}} = \{p \in D \mid f(p) = h_{\min}\} = T_{h_{\min}} \\ X_{h+1} = \mathrm{MIN}_{h+1} \cup \mathrm{IZ}_{T_{h+1}}(X_h), h \in [h_{\min}, h_{\max}) \end{cases} \qquad (5)$$

B. Simulating immersion

Vincent& Soille presented an implementation of the watershed transform of the definition above. In this implementation there are two steps: (i) sorting the pixels by increasing grey value for direct access to pixels at a certain grey value; (ii) a flooding step, proceeding level by level and starting from the minima. The Fig.1 indicates the flood step of this simulating procedure.

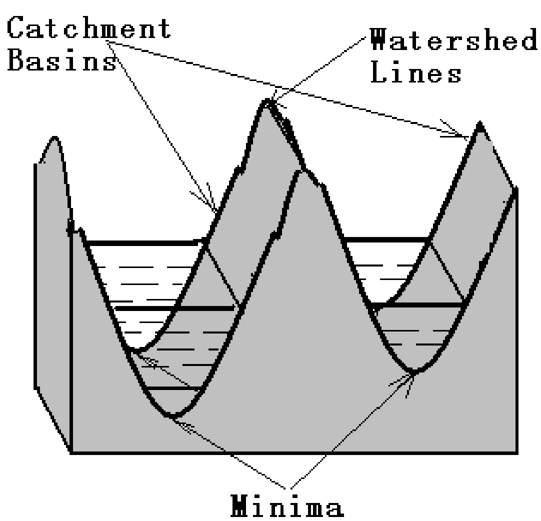


Fig. 1. Find watershed lines by simulating immersion.

The implementation uses a FIFO queue of pixels, that is, a first-in-first-out data structure. In the flooding step, all nodes with grey level h are first given the initial label. Then those nodes that have labeled neighbors from the previous iteration are inserted in the queue, and from these pixels geodesic influence zones are propagated inside the set of initial pixels.

C. Multi-degree immersion

From the introduction about the immersion simulation above, we can see that the level-by-level method during the flood procedure is not unchangeable. Actually, by contrast with the wide value range of the intensity of medical image this level-by-level method is excessively particular and helps to cause the over-segmentation problem.

To improve the level-by-level method to a multi-degree method, we redefine the threshold set of f at level h instead of (1) like this:

$$T_h=\{p \in D \mid f(p) - Diff(p) \leqq h\} \qquad (6)$$

However, other equations need not be changed. Here $Diff(p)$ is a function which presents the immersion level when the flood procedure reaches pixel p. The segmentation results are sensitive to the value of this function. Generally speaking, the greater value of $Diff(p)$ means immersing more points (Fig.2).

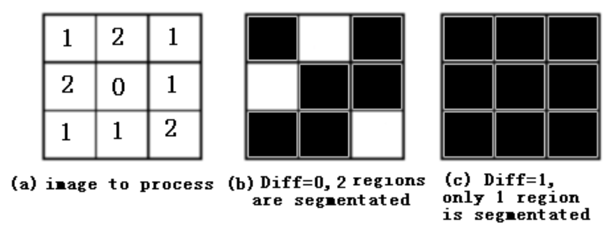


Fig. 2. Different segmentation results when *Diff* function is set.

In Fig.2, a processing image with a 3*3 matrix is shown. If we set the connectivity as four and define Diff as a constant function equal to 0, when the flood process goes to h=1, we get the points immersed like (b). The black area means the points immersed. We can find in 4-connectivity case, there are two minima in (b), but if we change *Diff* to a constant function equal to 1, we'll get all points of the image immersed and there is only one minimum as shown in (c).

In fact, if $Diff(p)=0$ is given, (6) can be regarded as a special case of (1). According to the user requirement, $Diff(p)$ can be even a constant function or a function computed according to the local information about p. In this paper, we define it as:

$$Diff(p)=\sum_{q \in Neighbor(p)} \frac{|f(p)-f(q)|}{conn} \qquad (7)$$

Where *Neighbor(p)* is the point set consists of the neighbor points of p and *conn* is the connectivity predefined.

III. Experimental Results

We implemented our experiment on a normal PC with Pentium 2.4GHz CPU and 2G DDR RAM, where the program was developed in Python + VTK environment and run on a Windows XP Platform. Two datasets were employd in this experiment: One is a brain MRI data downloaded from BrainWeb. It is a 181*217*181 voxel volume, with isotropic 1 mm^3 voxel. Another one is a real clinical canine cardiac CT data, which is a 512*512*86 volume with isotropic 0.35*0.35*1.25 mm^3 voxels.

Three watershed implementations are employed in our experiment for comparing purpose. A traditional immersion simulation, which is integrated into our system; a topographical distance based implementation, which is provided by Insight Toolkit[1] and the proposed multi-degree immersion method. We choose these three implementations just because that there are other definitions of watershed transform beside simulating immersion method, such as topographical distance based definition, and based on these definitions there are corresponding implementations. After making a comparison among them, we find that simulating immersion is a faster way to implement watershed transform.

A comparison among above three implementations is made in three aspects: time-elapsed, memory cost and the accuracy of the results. The following two tables indicate the result of this comparison:

TABLE I
COMPARISON AMONG THE THREE WATERSHED IMPLEMENTATION ON THE BRAIN DATA

	Traditional immersion simulation	Topographical distance based implementation	Multi-degree Immersion
Time	41s	180s	40s
Memory	448MB	408MB	454MB
watershed regions	10991	84	35

From Table I we can find that multi-degree immersion method resists the over-segmentation problem effectively. And the two simulating immersion implementations are both faster than the topographical distance based implementation.

Since the traditional immersion simulation method suffered from a very serious over-segmentation problem, it is hard to extract interested object from its resulting image. However the other two methods provide a complete segmentation results which are shown in Fig.4.

However, while we tried to implement the same experiment on the cardiac data, the topographical distance based implementation couldn't achieve a result because of not enough memory.

As shown in Table II, although multi-degree immersion method relieved the over-segmentation problem, the segmented areas are still too many to extract full object from them. Meanwhile, the topographical distance based method required huge memory which could not implemented in our PC environment. The proposed multi-degree method improved to resist the over-segmentation problem successfully and demanded a reasonable memory cost.

TABLE II
COMPARISON AMONG THE THREE WATERSHED IMPLEMENTATION ON THE CARDIAC DATA

	Traditional immersion simulation	Topographical distance based implementation	Multi-degree Immersion
Time	117s	N/A	115s
Memory	1450MB	>2400MB	1695MB
Watershed regions	48551	N/A	589

IV. Discussion

Immersion simulation (both the traditional and the multi-degree) is faster because of its FIFO data structure. After the initial sort procedure, it needs to iterate all points only once. And there is almost no difference in the running time between these two methods because multi-degree method only improves the sequence of the points to be processed. Although the reason also increases the maximum length of the FIFO queue slightly, the FIFO queue is usually not full during the flood process. This explains why the memory consumed is still almost equal.

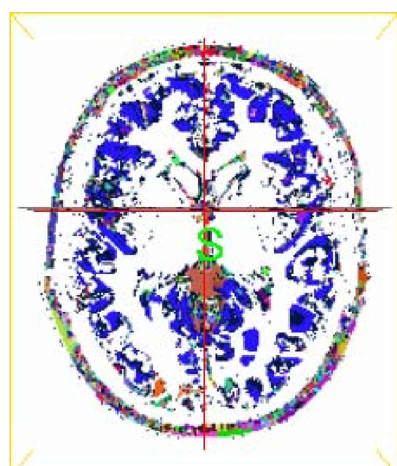
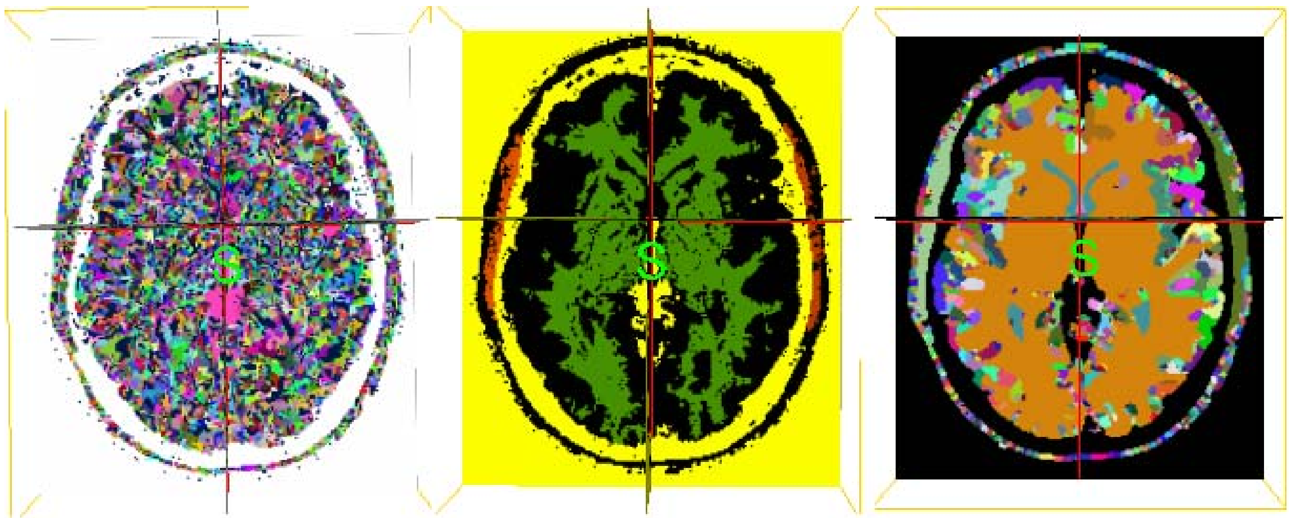


Fig. 3. The region of interest merges into background when *Diff* is too large

[1] http://www.itk.org



(a)Traditional immersion simulation (b)Multi-degree immersion (c)Topographical distance based implementation

Fig. 4. Segmentation results of the brain data

It must be emphasized that choosing a proper *Diff* function is very important in multi-degree immersion. If *Diff* is too large, the segmented areas will merge unexpectedly and even the object merges into background (Fig.3). If *Diff* is too small, the over-segmentation problem will not be resolved. Also it is not recommended that set *Diff* as a constant function, because in this case, some regions of the image may be merged excessively and some other regions may be still over segmented.

In our experiment we use (7) as our *Diff* function, when *Diff* > 0, the flood procedure will merge some steps of flooding. If the distribution of the intensities is uniformity, about half of steps will be merged.

V. Conclusion

By redefining the threshold set of an image, a novel implementation of watershed transform was presented. This implementation has the same time and memory complexity with the implementation by simulating immersion and improved to resist the over-segmentation problem effectively. It decreases the amount of segmented areas to 0.3%-1.2% without losing its accuracy.

But the function can be optimized because (7) depends on statistic information only and does not consider the global image information. This will be our future work.

Acknowledgment

This work is partially supported by the Shanghai municipal research fund, "3D Dynamic Cardiac Modeling and Visualization", 045107045. This work is also partially supported by NDI. We are also grateful to our team members in the lab for their valuable helps.

References

[1] J.B.T.M.Roerdink and A.Meijster (2000) The Watershed Transform:Definitions, Algorithms and Parallelization Strategies. Vol 41. *Fundamenta Informaticae*, pp.187-228.

[2] Kari Saarinen (1994) Color Image Segmentation By A Watershed Algorithm And Region Adjacency Graph Processing. Vol 3.*Image Processing*, 1994. Proceedings. ICIP-94., *IEEE International Conference,* pp 1021-1025.

[3] L.Shafarenko, M.Petrou and J.Kittler (1998) Histogram-Based Segmentation in a Perceptually Uniform Color Space. Vol 7. No.9 *IEEE Transactions On Image Processing*,1998.

[4] Luc Vincent and Pierre Soille (1991) Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. Vol 13. No.6 *IEEE Transactions On Pattern Analysis And Machine Intelligence*,1991.

[5] S.Beucher and F.Meyer (1993) The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*. New York, pp.443-481.

[6] V.Grau, A.U.J.Mewes, and M.Alcaniz (2004) Improved Watershed Transform for Medical Image Segmentation Using Prior Imformation. Vol 23. No.4 *IEEE Transactions On Medical Imaging*, 2004.

[7] Zhao Jianwei, Wang Peng and Liu Chongqing (2003) Watershed Image Segmentation Based on Wavelet Transform. *Acta Photonica Sinica* Vol 32. No.5. China.

[8] C.Lantuejoul, "Skeletonization in quantitative metallography," in *Issues of Digital Image Processing*, R.M. Haralick and J.C.Simon, Eds. Groningen, The Netherlands: Sijthoff and Noordhoff,1980