Annales UMCS Informatica AI 3 (2005) 169-178

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Watershed based region growing algorithm

Jakub Smołka[*]

Institute of Computer Science, Faculty of Electrical Engineering and Computer Science,
Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This paper presents a solution to a major drawback of watershed transformation: over segmentation. The solution utilizes one of its main advantages – very good edge extraction. It is a method that simulates pouring water onto a landscape created on a basis of a digital image. Unfortunately transformation produces a region for each local minimum so, usually, the number of watersheds (catchment basins) is too big. Watershed region growing is based on a minimum variance region growing algorithm [1-3]. It differs from the original in that it grows a homogenous region by adding and removing entire watersheds (catchment basins) and not separate pixels. The generalized watershed based region dilation and contraction are presented. Thanks to the use of watershed transformation, the region growing process is not able to grow a region easily outside the object boundaries. Test segmentations of two class images and a comparison between the minimum variance and the watershed based region growing are presented.

1. Introduction

Segmentation is a very important step in image processing. It determines the accuracy of subsequent processes [4]. The goal of segmentation is to extract objects shown in the image with the best possible accuracy. The region growing methods use information about pixel location and value. Watershed segmentation, on the other hand, uses information about the object boundaries. Watershed based region growing combines these methods in order to eliminate their drawbacks.

2. **Watershed** segmentation

Watershed segmentation was introduced by Beucher and Lantuejoul in [5]. It treats an image *I(x)* as a height function that describes a landscape. The assumption with this method is that higher pixel values indicate the presence of boundaries in the original image *f(x)*. That is why the gradient operator is often used in the watershed transformation for obtaining the height function

[*] *E-mail address*: jakub.smolka@pollub.pl

$I(x)=\left|\nabla f(x)\right|$. This is illustrated in figure 1. Since the gradient operator is very sensitive to noise, the original image is usually filtered with an edge preserving smoothing filter such as gradient diffusion [6].

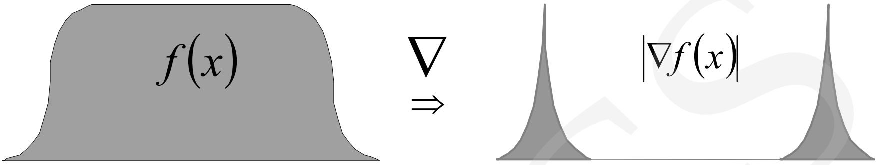

Fig. 1. Obtaining the height function *I*

The principle of watershed segmentation can be easily explained by analogy to rain pouring into a landscape. When water rains onto a landscape it flows with gravity to collect in low catchment basins. The size of those basins grows as the amount of precipitation they receive increases. At a certain point basins would start spilling into one another, causing small basins to merge together into larger basins. To prevent them from merging, the algorithm starts building dams between them (Figure 2). More formally, a catchment basin consists of all points whose paths of steepest descent on the graph of *I(x)* terminate at the same local minimum. Thus, there are as many catchment basins in an image as there are local minima in *I(x)* and each of them has a unique label [7-9].

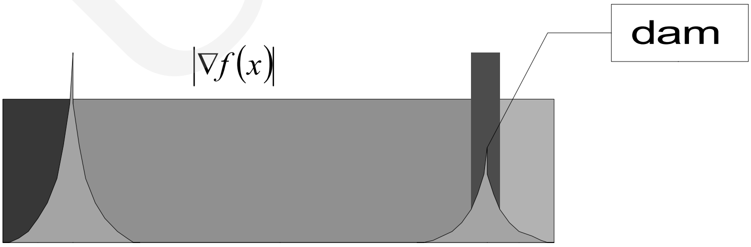

Fig. 2. Preventing basins from spilling into one another. As a result each basin has a unique label

This is the major drawback of watershed segmentation. In practice it produces too many regions and over segmentation results [7-9]. To alleviate this problem, one can establish a minimum watershed depth. It is the maximum depth of water a region could hold without flowing into any of its neighbors. A watershed segmentation algorithm can combine watersheds whose depths fall below the minimum until all of the watersheds have sufficient depth. Figure 3a depicts the minimum depth concept.

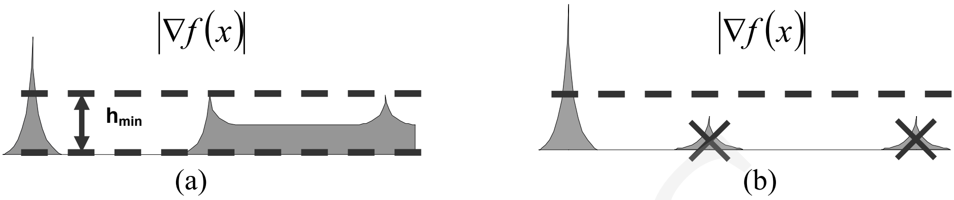

Fig. 3. The two ways of reducing over segmentation: (a) minimum watershed depth and (b) thresholding

Another way of reducing over segmentation is thresholding [7]. The gradient image *I(x)* prior to watershed segmentation is thresholded, using a small threshold (Figure 3b). This helps remove small values from the gradient image. Such values are usually the result of noise and create artificial catchment basins. Watershed based region growing utilizes minimum watershed depth and thresholding. Due to over segmentation the watershed transformation is often considered an intermediate step in a hybrid segmentation method. This is the case with the watershed based region growing where watersheds are joined to form a final segmentation of the image by the region growing process.

3. Watershed based dilation

Dilation is a very well known morphological operation widely used in image processing [10-12]. The watershed based region growing utilizes dilation for growing the region. Usually dilation is performed on the images whose elements are arranged in a regular (in most cases rectangular) mesh. In such a case it is possible to define a structuring element and its origin, which is then placed over every pixel of the image. When a structuring element moves over all pixels of binary image, an OFF pixel at the origin is set to ON if any part of the structuring element overlaps ON pixels of the image. The ON pixels are left unchanged. Watersheds may have very irregular shapes, and they are almost never arranged in any particular way. Thus, it is impossible to define the unchanging structuring element and apply the above definition. A new definition is necessary.

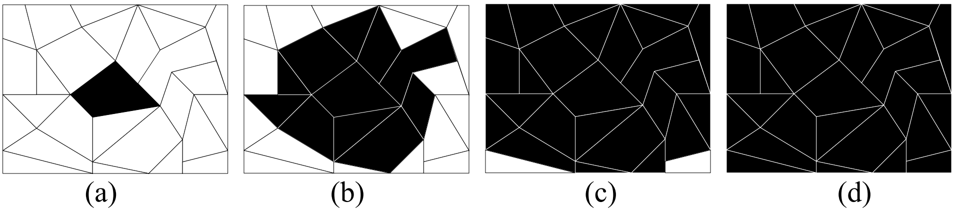

Fig. 4. Watershed based – dilation: (a) original region (1 basin), (b) region after the first dilation (+10 basins), (c) region after the second dilation (+14 basins), (d) region after the third dilation (+2 basins)

A region is described using a binary image as shown in figure 4. The watersheds are described by a different image. Each watershed has a unique label. When dilating a region, the watershed dilation algorithm first determines all watersheds neighboring the current region. Then it adds them to the region. If we define region r_i as a sum of watershed w_i and its watershed lines wl_j (each watershed line is a boundary between two watersheds)

$$r_i = \bigcup_{j=0}^{n_i} wl_j \cup w_i \tag{1}$$

and note that one watershed line may belong to two regions which implies

$$\exists_{i \neq j} r_i \cap r_j \neq \varnothing \tag{2}$$

then we can formally define the watershed dilated region R_{WD} as a sum of the original region R and all regions r_i that have at least one common point with region R.

$$R_{WD} = R \cup \{r_i \mid r_i \cap R \neq \varnothing\}. \tag{3}$$

4. Restoring region homogeneity

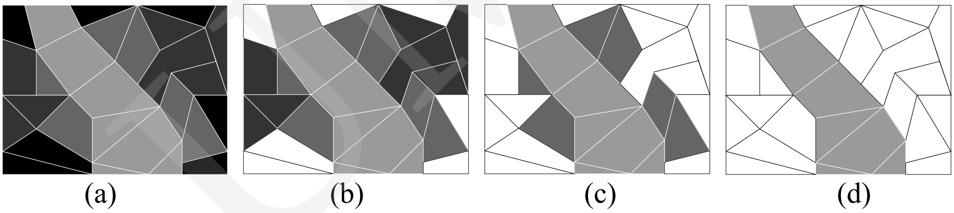

Fig. 5. Removing watersheds (basins) in region contraction (a) all basins belong to the region (b-d) basins are removed in the increasing order of their mean values (darker basins have lower mean values)

The watershed based region growing algorithm uses the standard deviation for assessing the region homogeneity. If the region σ is less or equal to a given threshold σ_t the region is homogeneous. During a watershed region growing process the region may become inhomogeneous, and a method of restoring its homogeneity is required. This method is region contraction. First the algorithm checks if the region is homogenous. If not, it looks for the lowest mean value among the watersheds that belong to the current region. After the lowest mean has been found, the algorithm goes through a list of watersheds and removes, from the region, those whose mean value is equal to the minimum found earlier. Then a new standard deviation is calculated in order to check if the region became homogeneous. If the region is not homogeneous, the entire sequence is repeated. Otherwise the algorithm stops. Watersheds are removed in the increasing order of mean values (Figure 5). The following presents a pseudo-code description of the region contraction:

(a) (b) (c) (d) are panel labels; included here:

```
$\sigma$:=standardDeviation(R);
while $\sigma>\sigma_t$ do //while region is not homogeneous

lowestMean:=findLowestMeanValueInRegion(Watersheds,R);
  //remove all watersheds whose mean value = lowestMean
  for i:=1 to numberOfWatershedInRegion do
    if (mean(Watersheds[i])=lowestMean) then
      R:=removeWatershedFromRegion(i,R);
    end;
  end;
  //calculate new standard deviation
  $\sigma$:=standardDeviation(R);
end;
```

Region contraction is similar to the histogram contraction introduced by Revol and Jourlin in [2]. The two main differences are: (1) region contraction uses watershed mean values rather than actual pixel values, (2) it works by removing entire watersheds and not single pixels. The presented version of region contraction starts by removing the darkest watersheds. This results in the brightest possible object being segmented. If segmenting the darkest object in the image is required, then the region contraction should remove watersheds in the order of decreasing mean values. Of course different strategies for restoring region homogeneity are possible. These strategies include: (1) starting with the most inhomogeneous watershed, (2) while removing watersheds, prefer the ones that were added in the last iteration. The mentioned methods will be tested in a future work.

5. Watershed based region growing process

The watershed based region growing algorithm, which has been implemented, is a generalized version of the algorithm proposed by Revol and Jourlin [1]. The generalization consists in making the algorithm grow regions by adding and removing irregularly shaped watersheds and not single pixels. However, the generalized version inherits the properties of the original. Watershed based region growing has the ability to grow non-connected regions. The watersheds, which belonged to a homogeneous region at a certain step, can be removed later. This property allows the algorithm to segment a complex region, not requiring a seed watershed in all of its parts. However the algorithm requires a start region consisting of selected watersheds. They will not necessarily be included in the final, grown region. At each step a homogenous region R_n is dilated using watershed dilation, yielding a region R_{n+1}. Then, if it is necessary, the algorithm restores the region R_{n+1} homogeneity using the region contraction procedure described above and proceeds to the next iteration. This sequence is repeated until the results of two consecutive iterations are identical. The following presents a pseudo-code description of the watershed region growing process.

```
//previous region and start region are different
$R_0$:=emptyRegion();
$R_1$:=startRegion;
n:=1;
while ($R_n \neq R_{n-1}$) do
  $R_{n+1}$:=watershedDilation($R_n$);  //grow the region
  $R_{n+1}$:=restoreHomogeneity($R_{n+1}$,$\sigma_t$);
  n:=n+1;  //next iteration
end;
```

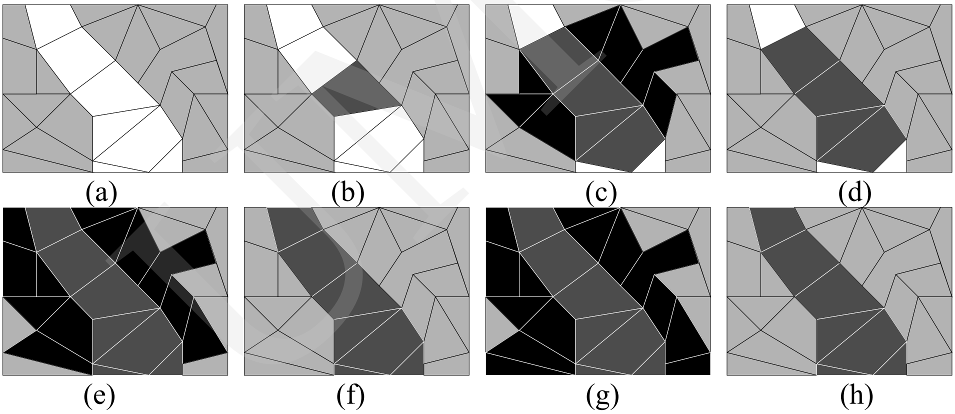

Fig. 6. Example of watershed based region growing sequence. An assumption is made that adding any of the gray watersheds to the region makes it inhomogeneous. (a) original image with watersheds, white watersheds belong to (b) start region (c) dilation (+10 basins) (d) contraction (-7 basins) (e) dilation (+15 basins) (f) contraction (-12 basins) (g) dilation (+14 basins) (h) contraction (-14 basins)

6. The results

The following shows the results obtained with watershed based region growing. The results come from a preliminary test. Since the presented algorithm is best suited for segmenting two class images, the test images have bimodal histograms. The first test image – shown in Figure 7a – was preprocessed with an edge preserving gradient diffusion smoothing filter [6] in order to remove noise. Before applying the watershed transformation, the gradient image was calculated. The gradient image was in turn thresholded to reduce over segmentation, and the watershed transformation was applied. A start region was selected from created watersheds, and a region was grown using a watershed based region growing algorithm with *$\sigma_t=3$*. The result is shown in Figure 8c. Figures 9 and 10 show another test image with a bimodal histogram. It was preprocessed in the same way as the first image. The final region was grown by a watershed based region growing algorithm with $\sigma_t=8$.

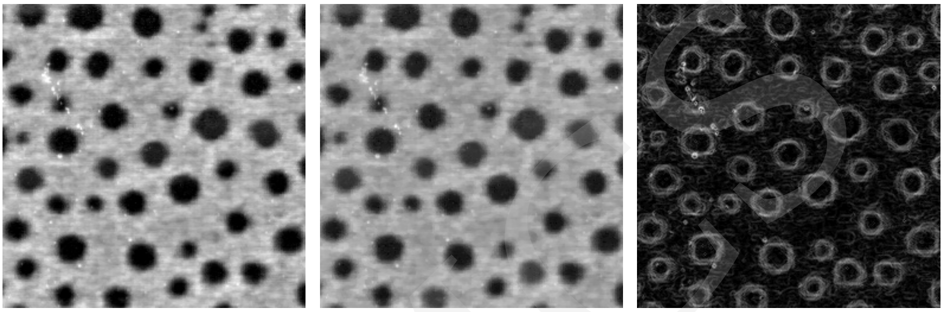
(a) (b) (c)

Fig. 7. Monomolecular film: (a) original image, (b) image blurred with gradient diffusion, (c) gradient image

(a) (b) (c)

Fig. 8. Watershed based region growing results: (a) watersheds ($h_{min}=0,\ t=0.05\left(I_{\max}-I_{\min}\right)$), (b) start region, (c) grown region ($\sigma_t=3,\ \sigma_{image}=20.79$)

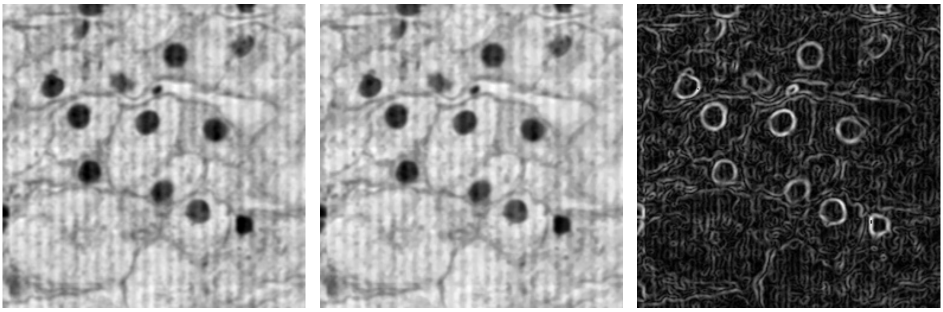
(a) (b) (c)

Fig. 9. Liver tissue: (a) original image, (b) image blurred with gradient diffusion, (c) gradient image

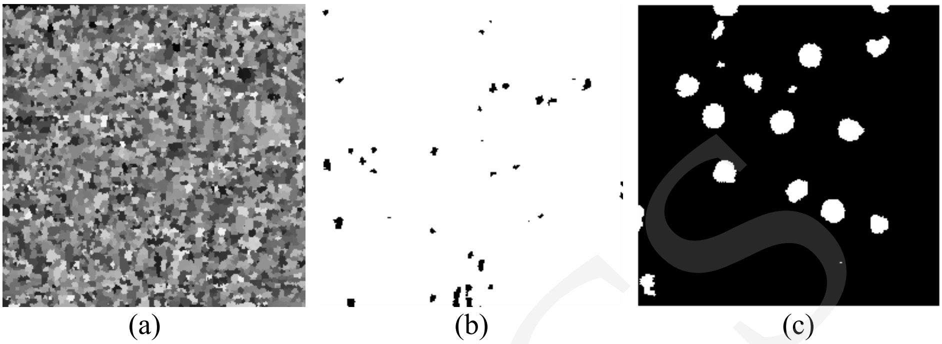

Fig. 10. Watershed based region growing results: (a) watersheds ($h_{min}=0$, $t = 0.05\left(I_{\max} - I_{\min}\right)$), (b) start region, (c) grown region ($\sigma_t=8$, $\sigma_{image}=25.05$)

7. Conclusions

The results presented above show that the concept of region growing algorithms can be used for joining the watersheds in an over segmented image. Moreover, the use of watersheds ensures that the boundaries of objects in the image overlap those of the grown region. A watershed based region growing algorithm, while performing a segmentation, takes into account not only pixel values and their location but also information about the boundaries of segmented objects. This property allows for obtaining accurate results. Figures 11, 12 and 13 show a comparison between the minimum variance region growing [1,2] and the watershed based region growing.

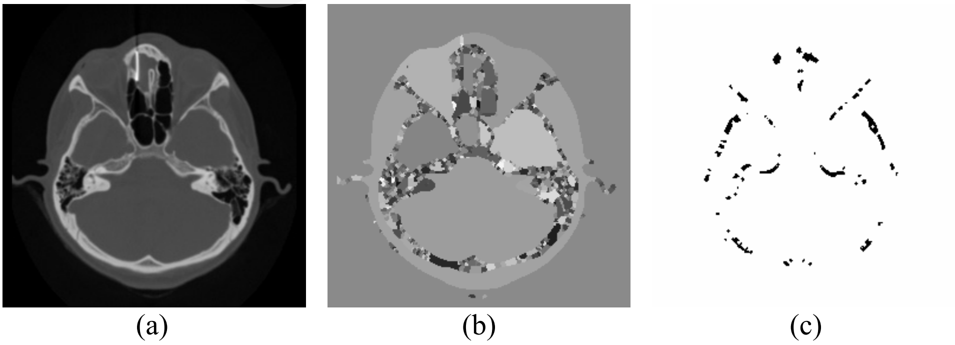

Fig. 11. CT scan: (a) original image, (b) watersheds ($h_{\min}=0$, $t = 0.05\left(I_{\max} - I_{\min}\right)$), (c) start region

The watershed transformation was applied to an original image. Selected watersheds were included in a start region, which was identical for both methods. The minimum variance region growing does not use the information about object boundaries. When the standard deviation threshold σ_t is too high, this property causes artifacts (Figure 13c) to appear. Use of watersheds, on the other hand, does not allow the watershed based region growing to include pixels belonging to another region without including the entire watershed. This makes adding parts of another object less likely. Figures 12b and 12c show that, when σ_t increases, the watershed based region growing simply segments another object.

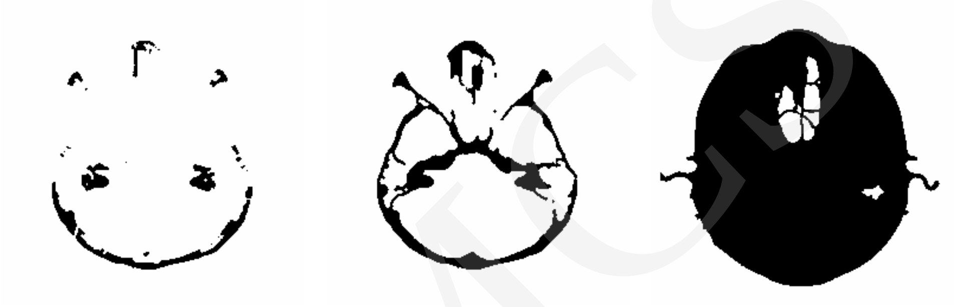

Fig. 12. Regions grown with the watershed based region growing for increasing σ_t.

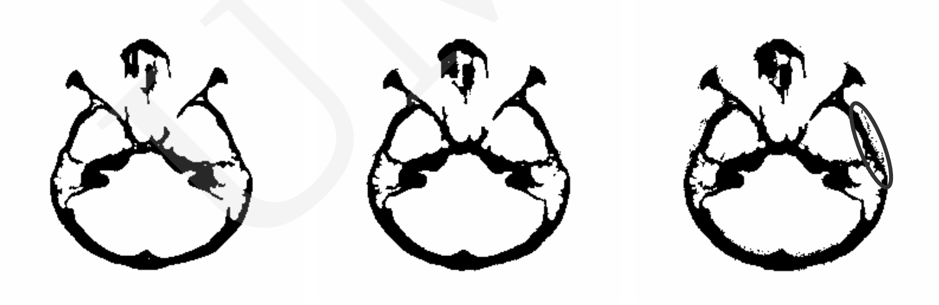

Fig. 13. Regions grown with the minimum variance region growing [1,2].

References

[1] Revol Ch., Jourlin M., *A new minimum variance region growing algorithm for image segmentation*, Pattern Recognition Letters 18 (1997) 249.

[2] Revol-Muller Ch., Peyrin F., Carrillon Y., Odet Ch., *Automated 3D region growing algorithm based on an assessment function*, Pattern Recognition Letters, 23 (2002) 137.

[3] Smołka J., *Multi-class region growing algorithm*, Annales Informatica UMCS, Lublin, 1 (2003) 165.

[4] Suri J., Setarehdan S., Singh S., *Advanced Algorithmic Approaches to Medical Image Segmentation*, Springer-Verlag London Limited (2002).

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

[6] Perona P., Malik J., *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis Machine Intelligence, 12 (1990) 629.

[7] Ibanez L., Schroeder W., Ng L., Cates J., *The ITK Software Guide*, Kitware Inc., (2003)

[8] Beucher S., *The watershed transformation applied to image segmentation*, Scanning Microscopy International, 6 (1992) 299.

[9] Bleau A., Leon L.J., *Watershed-based segmentation and region merging*, Computer Vision and Image Understanding, 77 (2000) 317.

[10] Seul M., O'Gorman L., Sammon M.J., *Practical Algorithms for Image Analysis: Description, Examples, and Code*, Cambridge Univerisity Press (2000).

[11] Gonzalez R.C., Woods R.E., *Digital Image Processing*, Addison-Wesley Publishing Company (1993).

[12] Myler H.R., Weeks A.R., *The Pocket Handbook of Image Processing Algorithms in C*, Prentice Hall PTR.