A New Algorithm for Image Segmentation via Watershed Transformation

Maria Frucci and Gabriella Sanniti di Baja

Institute of Cybernetics "E. Caianiello", CNR
Via Campi Flegrei 34, 80078 Pozzuoli (Naples), Italy
{m.frucci,g.sannitidibaja}@cib.na.cnr.it

Abstract. A new segmentation method is presented. The watershed transformation is initially computed starting from all seeds detected as regional minima in the gradient image and a digging cost is associated to each pair of adjacent regions. Digging is performed for each pair of adjacent regions for which the cost is under a threshold, whose value is computed automatically, so originating a reduced set of seeds. Watershed transformation and digging are repeatedly applied, until no more seeds are filtered out. Then, region merging is accomplished, based on the size of adjacent regions.

1 Introduction

Image segmentation is a key process for many image analysis and computer vision tasks. It aims at dividing the image into a number of disjoint regions, ideally corresponding to the regions perceived by a human observer as constituting the scene. While all the pixels in the same region are similar with respect to some property, e.g., color, intensity, or texture, adjacent regions significantly differ from each other with respect to the same property. The result of segmentation is either a collection of regions, constituting a partition of the image, or a collection of contours, each of which delimiting a region of the image.

Segmentation has received much attention in the literature (e.g., refer to [1-9] and to the references quoted therein). Histogram thresholding, edge detection, region growing, fuzzy clustering, probabilistic Bayesian methods, watershed transformation are among the most commonly followed approaches. Selection of the method to be used mostly depends on the specific image domain and application. In this work, we consider watershed-based segmentation.

As far as we know, watershed-based segmentation was introduced in [10]. Then, a number of papers suggesting solutions to the main problems affecting the watershed segmentation, or dealing with different applications of watershed transformation have been published (see, e.g., [11-16]).

Basically, watershed segmentation is based on the identification of suitable *seeds* in the gradient image of the input image, followed by a growing process that originates from the selected seeds. The seeds are generally detected as the sets of pixels with locally minimal value (called *regional minima*). In turn, the growing process associates to each seed the pixels that result to be closer, in terms of a given property, to that seed more than to any other seed.

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part II, LNCS 6979, pp. 168–177, 2011.

A positive feature of watershed segmentation is that the contours delimiting the regions into which the image is divided are mostly placed where human observers perceive them. In fact, the growing process is performed on the gradient image, where the edges are enhanced. A negative feature is that the image may result to be divided into a number of parts that is remarkably either larger (over-segmentation) or smaller (under-segmentation) than the expected number of parts. Over-segmentation is mainly caused by the fact that, by using all the regional minima in the gradient image, a too large number of seeds is obtained. Thus, suitable criteria to reduce the number of seeds are necessary. In turn, under-segmentation occurs when the criteria adopted for filtering the seeds are too selective.

The new method presented in this paper consists of two phases, both aimed at reducing the number of regions into which the image is partitioned to the most significant ones. During the first phase, a reduced set of seeds is computed by means of digging, while during the second phase adjacent regions are merged. The watershed transformation is initially computed starting from all seeds detected as regional minima in the gradient image. The cost for digging a canal to link regional minima of each pair of adjacent regions is computed. Then, digging is actually performed for each pair of adjacent regions for which the cost is under a threshold, whose value is computed automatically. Digging filters out a number of seeds so that a smaller number of regions are obtained when the watershed transformation is applied again. Watershed transformation and digging are repeatedly applied, until the number of seeds, and hence of regions, becomes stable. The second phase of the method is a merging process that is done in one inspection of the image resulting at the end of the first phase, and is based on the size of the regions.

The paper is organized as follows. Section 2 introduces the standard watershed transformation; Section 3 presents the new method; experimental results are discussed in Section 4; finally, concluding remarks are given in Section 5.

2 Watershed Transformation

An easy way to explain how the watershed transformation works is the *landscape* paradigm. A 2D digital image can be interpreted as a terrain elevation map, where the gray level g of a pixel p with coordinates (x, y) is the elevation at position (x, y) of the corresponding 3D landscape. The bottom of each valley of the landscape (called *pit*) is mapped into a connected set of pixels of the 2D image characterized by locally minimal gray level, and the top of each hill (called *peak*) is mapped into a connected set of pixels of the 2D image characterized by locally maximal gray level. Let us suppose that the landscape is immersed into water after its pits have been pierced. Immersion into water causes flooding of the landscape. Valleys with the lowest pits are reached first by the water and the corresponding catchment basins start to be transformed into lakes. When the water in a lake reaches the edge separating the catchment basin of that lake from an adjacent basin, a dam has to be built to prevent the water of the lake to overflow into the adjacent basin. Dam construction continues until the water reaches the highest peaks of the landscape. When this is the case, the top lines of the dams constitute the closed watershed lines, each of which surrounds a catchment basin, and the input image results to be partitioned into a number of catchment basins.

To implement the standard watershed transformation, two different strategies can be followed, known as watershed by topographical distances and watershed by immersion (see [12] for a detailed description of both strategies). In any case, to guarantee that the catchment basins are separated from each other by a leak-proof set of watershed lines, different metrics have to be adopted for basins and lines. We use 4-connectedness for the catchment basins and 8-connectedness for the watershed lines. Connected component labeling completes watershed transformation by assigning an identity label to each of the regions into which the image is partitioned. An example of watershed segmentation is given in Fig. 1. There, image # 42049 of the Berkeley segmentation dataset [17] is to the left and the standard watershed transform is in the middle. Since all 7569 regional minima detected in the gradient image have been used as seeds, the image is partitioned into 7569 regions. The resulting image is clearly over-segmented.

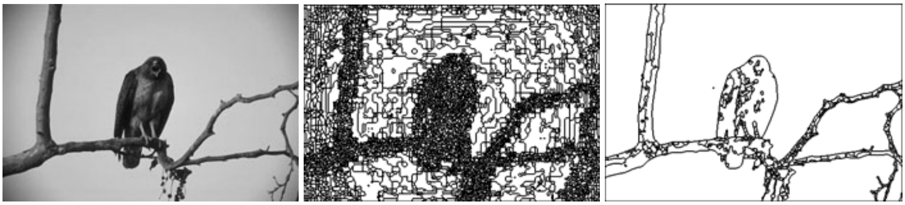

Fig. 1. The gray-level image #42049, left, the standard watershed segmentation in 7569 regions, middle, and segmentation in 645 regions by the algorithm [13], right

To reduce over-segmentation, a careful selection of the regional minima to be used as seeds for region growing is necessary, or reliable criteria for region merging have to be adopted. In [13], an effective but computationally expensive watershed segmentation method has been suggested that significantly reduces over-segmentation. After the watershed transform is computed starting from all seeds, the significance of each obtained region X is evaluated in terms of the interaction of X with every adjacent catchment basin Y. Then, flooding and digging are employed to cause disappearance of the regional minima corresponding to non-significant regions and the watershed transform starting from the reduced set of seeds is computed. Detection of non-significant regions, flooding and digging, and watershed transformation are repeated until all regions in the current watershed transform result to be significant. By applying the watershed segmentation algorithm [13] to the image in Fig. 1 left, the partition into only 645 regions shown in Fig. 1 right is obtained. Unfortunately, the computational cost of this algorithm is rather high, due to the large number of applications of the watershed transformation and, mainly, to the processes aimed at computing region significance.

Flooding and digging, though both aimed at filtering out non-significant seeds, act differently as concerns the resulting watershed transform and should be applied under different circumstances. To understand what happens if the seed associated to a given catchment basin X is removed by either flooding or digging, let us refer to the simple example shown in Fig. 2, where X is shown together with the adjacent catchment basin Y.

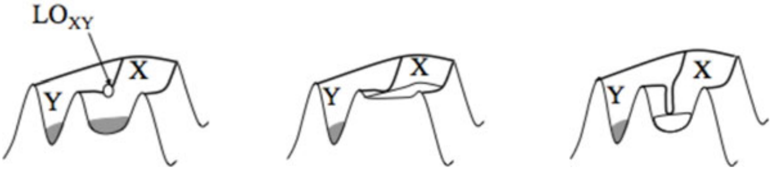

Fig. 2. Local overflow for the catchment basins X and Y. Effect of flooding, middle, and of digging, right.

Let LO_{XY} be the *local overflow pixel*, i.e., the pixel with minimal height along the watershed line separating X from Y and let l_{XY} denote the gray level of LO_{XY}. To filter out the seed associated to X, we should modify the gradient image in such a way that the set of pixels that constituted the seed for X is no longer a regional minimum. To reach the goal via flooding, we must increase to l_{XY} the gray level of all pixels of X with gray level lower than l_{XY}. In turn, to reach the goal by digging, we should dig a canal connecting the regional minima of X and Y. The canal is identified in the gradient image as the minimal length path linking the regional minima of X and Y, and passing through LO_{XY}. The gray level of all the pixels in the path is set to the lower value between those of the regional minima of X and Y.

Clearly, both flooding and digging cause the suppression of the regional minimum associated to X and, hence, X will result as merged to Y when the watershed transformation is applied again. However, digging is generally preferable since it allows to merge X and Y without altering the watershed lines separating X any another adjacent region W with which X should not be merged, even if the local overflow pixel LO_{XW} has gray level l_{XW} smaller that l_{XY}. For this reason, in our new segmentation method we use only digging for seed filtering in the gradient image.

3 The Suggested Method

Our segmentation method consists of two phases, both aimed at reducing the number of regions into which the image has been partitioned by an initial watershed transformation. During the first phase, seed reduction is obtained via digging, while during the second phase adjacent regions are merged if some conditions on their size are verified.

The watershed transformation is initially computed starting from all regional minima in the gradient image. Then, we compute the cost for digging a canal to link the regional minima of each pair of adjacent regions and perform digging for each pair of adjacent regions with cost under a threshold. The value of this threshold is computed automatically. The effect of digging is a reduction in the number of seeds, and hence in the number of regions, that are obtained by applying again the watershed transformation. Watershed transformation and digging are repeatedly applied, as far as seed removal is possible. Then, the second phase is performed. This is a region merging based on the size of the regions and accomplished in one inspection of the image.

3.1 First Phase

As mentioned in the previous section, we prefer to filter out seeds by digging rather than by flooding since digging does not alter the watershed lines separating a region whose minimum has to be suppressed from other regions that should remain in the watershed transform. To perform digging between two adjacent regions X and Y, we need to build a path linking the regional minima of X and Y. The larger is the height of the local overflow pixel LO_{XY} with respect to the regional minima of X and Y, the larger is the cost C_{XY} necessary to build the path and, hence, to perform digging between X and Y passing through LO_{XY}.

If we denote by l_{XY}, min_X and min_Y the gray levels of LO_{XY} and of the regional minima of X and Y respectively, the cost C_{XY} is computed as follows:

$$C_{XY} = (l_{XY} - min_X) + (l_{XY} - min_Y)$$

The contour of a region X consists of a number of parts, each of which is a watershed line separating X from one of the basins adjacent to X. For each region Y adjacent to X, we identify the local overflow pixel LO_{XY} and compute the corresponding cost C_{XY} necessary for digging a canal from X to Y through LO_{XY}.

In general, the watershed lines separating X from the adjacent regions are likely to be characterized by different digging costs. To reduce over-segmentation, we unavoidably need to filter out a number of seeds. Then, it seems safer to perform digging through the local overflow pixel with the smallest associated digging cost, while the remaining watershed lines of X remain unaltered.

However, if the above process is indiscriminately done for all regions in the initial watershed transform, the resulting image would be under-segmented and if digging is iterated the whole image would obviously merge into a unique region.

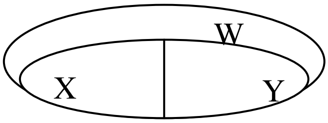

Fig. 3. Three adjacent basins. See text.

We observe that if we want to maintain untouched the watershed lines separating X from other adjacent regions while we dig out a canal linking the regional minimum of X to the regional minimum of Y, some specific conditions should be fulfilled as regards the digging costs along the contour of Y. More in detail, let us refer to Fig. 3 and suppose that we want to dig a canal between X and Y, leaving the remaining part of the contour of X untouched.

Let us consider the costs of digging the canals from X to its adjacent basins Y and W. By taking into account the above issues, we can suppose that for the basin X, C_{XW} is the maximal cost, while C_{XY} is the minimal cost. If we want to preserve the watershed line separating X from W while we merge X and Y via digging, then C_{YW} should be larger than C_{XY}. Otherwise, if C_{YW} is smaller than C_{XY}, besides merging X with Y we would also merge Y to W. As a result, X would be merged to W, while our intention was to keep unaltered the watershed lines separating X from W. Thus, a reasonable condition to prevent digging from X to Y is that C_{XY} is the minimal cost, C_{XW} is the maximal cost, and a basin W adjacent to both X and Y exists such that $C_{YW}<C_{XY}$.

For each maximal cost actually found in the initial watershed transform, we count the number of times that the above situation occurs. In this way, we build the histogram of the distribution of maximal digging costs, where we detect peaks and valleys, i.e., local maxima and local minima. Our aim is to identify a suitable valley whose value can be used as a threshold on digging cost, so as to reduce over-segmentation when the watershed transformation is newly applied. We have experimentally found that, in the average, the best threshold value is the digging cost corresponding to the deepest valley. The depth of a valley is obtained by computing the differences in height between each of the two peaks delimiting any valley and the valley itself, and by taking the largest of the two differences. If the same maximal depth characterizes more than one valley, we use as threshold the digging cost corresponding to the rightmost deepest valley, i.e., we take as threshold value the largest digging cost among those found in all valleys having the maximal depth.

Actually, both very small and rather large maximal cost values are not considered in the histogram to detect the deepest valley. In fact, small digging costs are very frequent in the initial watershed transform, where many noisy partition regions are due to regional minima found in the gradient image in correspondence with almost homogeneous areas. Considering all small digging costs to analyze the histogram would lead to rather high peaks in correspondence with these small costs. This, in turn, would lead to detect a too small threshold value that would result as not adequate to reduce over-segmentation. On the other hand, peak height along the histogram decreases when the digging cost increases, so that the peaks, if any, in the rightmost part of the histogram have rather small height. Considering them would only make histogram analysis more expensive. Moreover, merging between regions for which the digging cost is rather high would mean merging regions that are associated to two regional minima rather different in the gradient image. These regional minima most possibly originate regions in between which a human observer would perceive a separation.

Thus, we build the histogram only for maximal digging costs ranging from a minimum value 10 to a maximal value 40, where both limits have been experimentally found as adequate to provide good segmentation results in the majority of cases.

Once the value of the threshold θ on digging cost has been computed, digging should be done for each pair of adjacent regions X and Y such that $C_{XY}\leq\theta$ (of course, if this condition holds for X and more than just one adjacent region Y, more canals are built, which will produce a larger fusion in the newly computed watershed transform). To this purpose, a linking path passing through LO_{XY} should be built to connect the regional minima of each pair of regions X and Y for which the digging cost does not overcome the threshold. To avoid unwanted digging of X with any other adjacent region W for which a digging cost $C_{XW}>\theta$ is found, the linking path connecting the regional minima of X and Y is not built if the path necessarily passes through pixels that, besides belonging to the watershed line separating X and Y, also belong to the watershed line separating X and W.

The effect of digging is not only that of causing merging of suitable pairs of adjacent regions for which the digging cost is under the threshold, but also that of making possible, at a successive application of the watershed transformation, the construction of linking paths in correspondence with those regions that were characterized by digging cost under the threshold, but for which the paths could not be built. Thus, watershed transformation, computation of the digging costs, and digging are repeated until no more seeds are filtered out. A small number of iterations are generally sufficient to obtain a stable watershed transform. We have experimentally found that in the average the first phase of segmentation requires at most three iterations.

Fig. 4 left shows the result at the end of the first phase of segmentation for the running example. The automatically computed value of the threshold is θ=32, and 666 regions have been found.

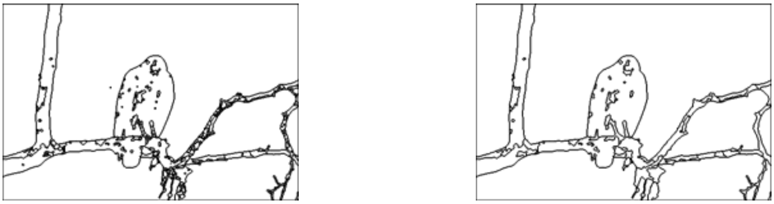

Fig. 4. Result at the end of the first segmentation phase in 666 regions, left, and at the end of the second segmentation phase in 82 regions, right

3.2 Second Phase

The area of each region of the watershed partitioned image is computed by counting the number of pixels it includes. Let A be the arithmetic mean of the areas of all regions. We divide the regions in two classes, respectively including regions with *large* area and regions with *small* area. A region R_i with area A_i is classified as small if it is $A_i/A < \tau$, where the value of the threshold τ has to be fixed depending on problem domain. In this work we have experimentally found that the value τ= 0.33 produces generally good results.

All regions with small area and adjacent to each other are merged into a unique region. For the running example, it results A=231. Due to the selected value for τ, this means that regions with area consisting of less than 77 pixels are regarded as small regions and are accordingly merged.

The segmentation of the running example at the end of the second phase is shown in Fig. 4 right. Only 82 regions characterize the segmented image, which is remarkably less than the 645 regions found by the algorithm [13].

4 Experimental Results

The segmentation algorithm has been tested on a number of images with different resolutions, taken from public databases. The three images #62096, #118035, and #41004 shown in Fig.5, all taken from [17], are used here to show the performance of the segmentation algorithm.

In all cases, the value of the threshold θ has been automatically computed by analyzing the histogram of maximal digging cost distribution in the range from 10 to 40. As for the threshold τ used during the second phase of the process, the value τ=0.33 has been used in all cases. The results obtained at the end of phase 1 and of phase 2 are shown in Fig. 6 top and Fig. 6 bottom, respectively.

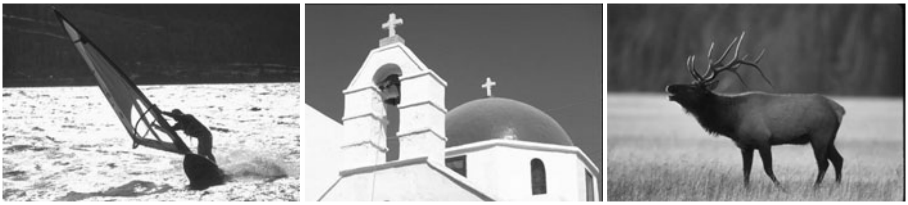

Fig. 5. Test images. From left to right: #62096, #118035 and #41004.

The automatically computed value for the threshold θ is 38, 21, and 20, for images #62096, #118035, and #41004, respectively. Starting from the initial watershed partitions into 14853, 7569 and 9851 regions, at the end of the first phase 2069, 769 and 1966 regions are obtained.

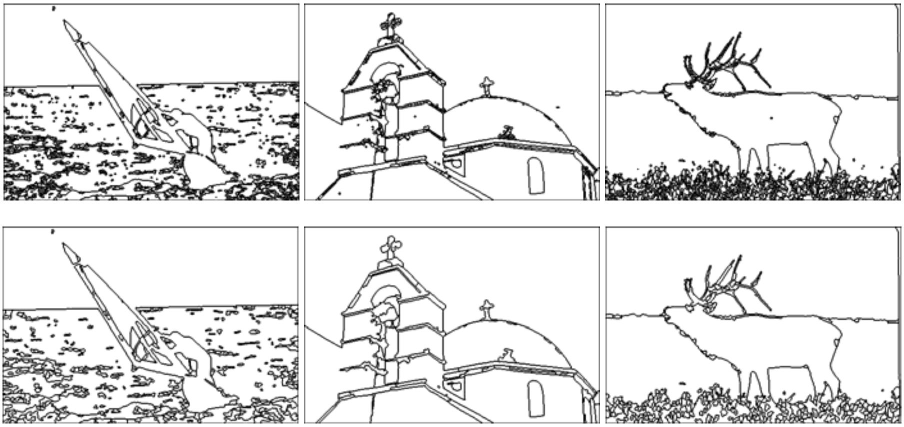

Fig. 6. Results at the end of the first segmentation phase, top, and final segmentations, bottom

In turn, the arithmetic mean of the area of the watershed regions resulting at the end of phase 1 is 74 for #62096, 200 for #118035 and 78 for #41004. As already pointed out, the default threshold value τ= 0.33 has been used for all examples. The final segmentations include 358, 78 and 260 regions, for images #62096, #118035, and #41004, respectively.

We point out that the values 10 and 40 as limits for the analysis of the histogram of the maximal digging costs and the value of the threshold τ=0.33 suggested in this paper have to be interpreted as default values producing, in the average, satisfactory results. Of course, it is not guaranteed that by using the default values the best segmentation is achieved whichever input image is handled. Moreover, we point out that the value of θ automatically computed in correspondence with the deepest valley of the histogram of the maximal digging costs, could generally be slightly increased without producing under-segmented results. As an example refer to Fig. 7, where for image #42049 other two different segmentations, obtained with different values for the segmentation parameters are given. As shown in Fig. 4 right, the value θ=32, automatically computed, produced 666 and 82 regions at the end of the first and of the second segmentation phase. By setting θ=36 (θ=39), 552 and 74 regions (486 and 64 regions) are respectively obtained.

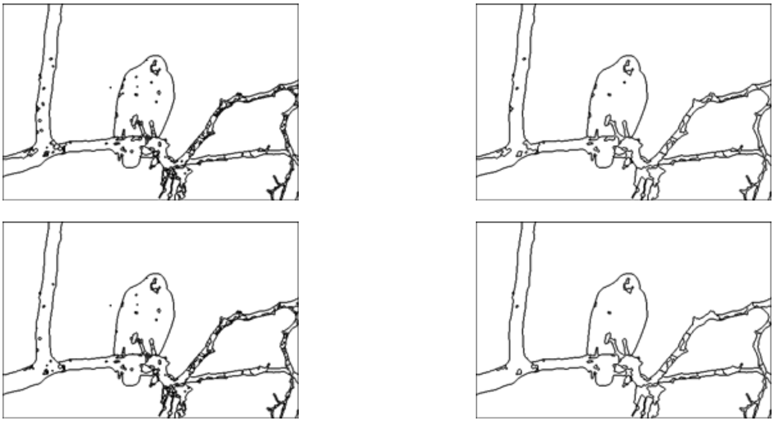

Fig. 7. Different segmentations obtained after the first segmentation phase, left, and after the second phase, right, for θ=36, top, and θ=39, bottom

Analogously, different results are also possible by changing the value of the merging threshold τ used during the second phase of segmentation.

5 Concluding Remarks

We have suggested a segmentation method based on the watershed transformation. To reduce the main drawback of watershed transformation, i.e., the generally too large number of partition regions with respect to the intuitively expected ones, we have introduced the notion of cost of digging. Then, we have established a criterion to compute automatically a threshold on digging cost. For each pair of adjacent basins characterized by a digging cost smaller than the threshold, a path linking the corresponding regional minima has been built so as to merge the two distinct regional minima in one regional minimum. In this way, by applying again the watershed transformation, all regions such that the corresponding regional minima have been connected to each other by means of linking paths result to be merged into a unique region. The whole process consisting of watershed transformation and digging is repeated as far as linking paths can be built to merge adjacent regions. Once the digging cost is above the threshold for all regions, a second phase of segmentation is accomplished to merge adjacent regions based on their relative size. The algorithm has been tested on a large variety of images, producing generally satisfactory results.

References

[1] Lucchese, L., Mitra, S.K.: Color image segmentation: A State-of-the-Art Survey. Proc. of the Indian National Science Academy (INSA-A) 67A(2), 207–221 (2001)

[2] Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition 34, 2259–2281 (2001)

[3] Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet Another Survey on Image Segmentation: Region and Boundary Information Integration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 408–422. Springer, Heidelberg (2002)

[4] Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Electronic Imaging 13(1), 146–165 (2004)

[5] Wirjadi, O.: Image and video matting: a survey. Fraunhofer Institut für Techno und Wirtschaftsmathematik ITWM 2007, ISSN 1434-9973 Bericht 123 (2007)

[6] Zhang, H., Fritts, J.E., Goldman, S.A.: Image segmentation evaluation: A survey of unsupervised methods. In: CVIU, vol. 110, pp. 260–280 (2008)

[7] Shamir, A.: A survey on mesh segmentation techniques. Computer Graphics Forum 27(6), 1539–1556 (2008)

[8] Senthilkumaran, N., Rajesh, R.: Edge detection techniques for image segmentation: a survey of Soft Computing Approaches. Int. J. Recent Trends in Engineering 1(2), 250–254 (2009)

[9] Yang, Z., Chung, F.-L., Shitong, W.: Robust fuzzy clustering-based image segmentation. Applied Soft Computing 9, 80–84 (2009)

[10] Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection. In: Proc. Int. Workshop on Image Processing, Real-time Edge and Motion Detection/Estimation, Rennes, France, pp. 12–21 (1979)

[11] Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. PAMI 13(6), 583–598 (1991)

[12] Roerdink, J.B.T.M., Meijster, A.: The watershed transform: definitions, algorithms and parallelization strategies. Fundamenta Informaticae 41, 187–228 (2001)

[13] Frucci, M.: Oversegmentation reduction by flooding regions and digging watershed lines. IJPRAI 20(1), 15–38 (2006)

[14] Frucci, M., Perner, P., Sanniti di Baja, G.: Case-based reasoning for image segmentation by watershed transformation. In: Case-based Reasoning on Images and Signals, vol. 73, pp. 319–352. Springer, Berlin (2007)

[15] Soille, P., Vogt, P.: Morphological segmentation of binary patterns. Pattern Recognition Letters 30(4), 456–459 (2009)

[16] Maulik, U.: Medical image segmentation using genetic algorithms. IEEE Trans. Information Technology in Biomedicine 13(2), 166–173 (2009)

[17] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/