

ROBUST WATERSHED SEGMENTATION USING THE WAVELET TRANSFORM

C.R. JUNG¹, J. SCHARCANSKI²

¹ UNISINOS - Universidade do Vale do Rio dos Sinos
Centro de Ciências Exatas e Tecnológicas
Av. UNISINOS, 950. São Leopoldo, RS, Brasil, 93022-000
crjung@exatas.unisinos.br

² UFRGS - Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500. Porto Alegre, RS, Brasil 91501-970
jacobs@inf.ufrgs.br

Abstract. The watershed transform has been used for image segmentation relying mostly on image gradients. However, background noise tends to produce spurious gradients, that cause over-segmentation and degrade the output of the watershed transform. Also, low-contrast edges produce gradients with small magnitudes, which may cause different regions to be erroneously merged. In this paper, a new technique is presented to improve the robustness of watershed segmentation, by reducing the undesirable over-segmentation. A redundant wavelet transform is used to denoise the image and enhance the edges in multiple resolutions, and the image gradient is estimated with the wavelet transform. The watershed transform is then applied to the obtained gradient image, and segmented regions that do not satisfy specific criteria are removed.

Keywords: denoising, edge enhancement, watersheds, segmentation, wavelets.

1 Introduction

Image segmentation is a challenging problem in image analysis. Segmentation consists of isolating the different regions, which generally correspond to meaningful objects that compose a given scene.

In particular, the *watershed transform* is one of the most well known image segmentation approaches based on morphological principles [1, 2, 3]. If we regard a grayscale image as a topographic relief, the gray value at a given location represents the elevation at that point. If this relief is to be flooded, the water will fill up lower elevation points first, and then the water level will increase. When water coming for two different regions meet, a *watershed* is created. The different regions that were flooded are called *catchment basins*.

If this process is applied to a gradient image (in which each pixel corresponds to the modulus of the gradient at a particular point), the watersheds correspond exactly to the crest lines of the gradient, which are associated with the edges of the image. Therefore, the catchment basins are the segmented objects in the image.

To use watersheds for image segmentation, a robust image gradient computation technique is required. Images are inherently noisy, and noisy graylevel fluctuations tend to cause spurious gradients. These gradients generate un-

desired watersheds, causing over-segmentation.

Usually, a simple threshold is not sufficient to eliminate the gradients associated with noise, specially in images where the amount of noise is large and/or low contrast edges occur. There are methods proposed in the literature to simplify an image, by removing small details from it, before applying the watershed transform. Meyer [4] introduced the *levelings* approach, which consists of applying morphological filters to reduce small details in the image. This approach works well for images with small amounts of noise, but has limitations when the amount of noise is greater and/or when low contrast edges are involved. Haris *et al.* [5] proposed an edge-preserving statistical noise reduction approach as a pre-processing for the watershed transform, and a hierarchical merging process as a post-processing stage. The results obtained with this approach are satisfactory, but edge enhancement is not explored, causing regions with weak borders to be erroneously merged. Weickert [6] proposed to pre-process the image using partial differential equations in order to reduce noise and enhance edges. However, the edge enhancement procedure appears to have not been sufficiently explored in [6], and consequently the performance of this technique in images containing weak edges (e.g. blurred images) was not reported.

We propose to pre-process the image before applying the watershed transform, so noise is removed and image

edges are kept and/or enhanced. In this work, we discuss a wavelet-based method for image denoising and edge enhancement in multiple resolutions, which improves the robustness of the watershed transform. In this approach, the gradients of the denoised and enhanced image are estimated using the wavelet transform, and then the watershed transform is applied to the obtained gradient image. The watershed implementation uses the immersion simulations approach [2]. A post-processing stage is finally utilized to remove over-segmented regions with small areas, and to merge erroneously segmented regions, that are separated by weak borders after denoising and enhancement.

2 Pre-Processing

To improve the quality of the segmentation based on watersheds, we need to have an image with well defined edges (preventing different regions from being merged because of gaps in their common borders), and the image should contain small amounts of noise (preventing the occurrence of over-segmentation caused by false edges related to noise).

In a previous work [7], the authors proposed a multi-resolution denoising technique based on the wavelet transform. In the present work, this approach was modified to include edge enhancement, so edges and background noise are discriminated more easily. Our proposed image denoising and enhancement approach is detailed next.

In the pre-processing stage, the discrete wavelet decomposition using only two detail images (horizontal and vertical details) is applied to the image [8]. As a result, the detail images $W_{2^j}^1 f[n, m]$ and $W_{2^j}^2 f[n, m]$ are obtained at each scale 2^j , as well as the smoothed image $S_{2^j}^f[n, m]$. The detail images $W_{2^j}^1 f$ and $W_{2^j}^2 f$ may be considered as local differences along the x and y directions, and may be used to obtain a good approximation of the image gradient. Therefore, the gradient (i.e. edge) magnitudes can be calculated in terms of $W_{2^j}^1 f$ and $W_{2^j}^2 f$ as follows [8]:

$$M_{2^j} f = \sqrt{(W_{2^j}^1 f)^2 + (W_{2^j}^2 f)^2}, \quad (1)$$

and the edge orientation is given by the gradient direction, which is expressed by:

$$\theta_{2^j} f = \arctan\left(\frac{W_{2^j}^2 f}{W_{2^j}^1 f}\right). \quad (2)$$

For image denoising, we want to find a non-negative non-decreasing shrinkage function $g_j(x)$, $0 \leq g_j(x) \leq 1$, such that the wavelet coefficients $W_{2^j}^1 f$ and $W_{2^j}^2 f$ are updated according to the following rule:

$$NW_{2^j}^i f[n, m] = W_{2^j}^i f[n, m]g_j[n, m], \quad i = 1, 2, \quad (3)$$

where $g_j[n, m] = g_j(M_{2^j} f[n, m])$ is called *shrinkage factor*. To preserve edges during the noise removal process,

the shrinkage factors should be close to 1 near edges, and close to 0 in homogeneous regions. Thus, differentiation between noise and edge-related coefficients is crucial. This discrimination is accomplished after three steps: first, image gradients are analyzed (since gradients related to noise are typically smaller than gradients related to edges); second, adjacent scales are compared (because responses due to noise tend to vanish faster than responses due to edges as the scale increases); third, the spatial distribution of the gradients is analyzed, since edges appear in contours, and no isolated.

2.1 Wavelet Shrinkage

As proposed in [7], the distribution of the magnitudes $M_{2^j} f = \sqrt{(W_{2^j}^1 f)^2 + (W_{2^j}^2 f)^2}$ at each resolution 2^j , are approximated by a combination of Rayleigh probability density function:

$$p_j(r) = w_{\text{noise}}^j p_j(r|\text{noise}) + (1 - w_{\text{noise}}^j) p_j(r|\text{edge}), \quad (4)$$

where $p_j(r|\text{noise})$ and $p_j(r|\text{edge})$ are Rayleigh distributions of noise-related and edge-related coefficients, respectively, w_{noise}^j is the *a priori* probability for the noise-related gradient magnitude distribution (and, consequently, $1 - w_{\text{noise}}^j$ is the *a priori* probability for edge-related gradient magnitudes). To simplify the notation, we remove the index j , and equation (4) can thus be written as:

$$p(r) = w_{\text{noise}} p(r|\text{noise}) + (1 - w_{\text{noise}}) p(r|\text{edge}). \quad (5)$$

The parameters of the Rayleigh function, as well as the *a priori* probabilities are obtained by fitting the theoretical model (equation (5)) to the actual data, using a maximum likelihood approach. The shrinkage function $g(r)$ is given by the posterior probability function $p(\text{edge}|r)$, which is calculated using Bayes theorem as follows:

$$g(r) = \frac{(1 - w_{\text{noise}}) p(r|\text{edge})}{(1 - w_{\text{noise}}) p(r|\text{edge}) + w_{\text{noise}} p(r|\text{noise})} \quad (6)$$

Further discrimination between edge and noise can be achieved by analyzing the evolution of the shrinkage functions along consecutive scales and applying spatial constraints, as discussed next.

2.1.1 Consistency Along Scales

For each scale 2^j , the value $g_j[n, m]$ may be interpreted as a confidence measure that the coefficient $M_{2^j}[n, m]$ is in fact associated to an edge. If the value $g_j[n, m]$ is close to 1 for several consecutive levels 2^j , it is more likely that $M_{2^j}[n, m]$ is associated with an edge. On the other hand, if $g_j[n, m]$ decreases as j increases, it is more likely that $M_{2^j}[n, m]$ is actually associated with noise.

For each scale 2^j , we use the information provided by the function g_j , and also by the functions g_{j+k} , for $k = 1, 2, \dots, K$, where $K + 1$ is the number of consecutive resolutions that will be taken into consideration for the consistency along scales. The harmonic mean is used to combine the shrinkage factor in consecutive scales:

$$g_j^{\text{scale}}[n, m] = \frac{K + 1}{\frac{1}{g_j[n, m]} + \frac{1}{g_{j+1}[n, m]} + \dots + \frac{1}{g_{j+K}[n, m]}}. \quad (7)$$

This updating rule is applied from coarser to finer resolution. The shrinkage factor $g_j^{\text{scale}}[n, m]$, corresponding to the coarsest resolution 2^J , is equal to $g_J[n, m]$. However, for other resolutions 2^j , $j = 1 \dots J - 1$, the shrinkage factors $g_j^{\text{scale}}[n, m]$ depend on scales $2^j, 2^{j+1}, \dots, 2^\kappa$, where $\kappa = \min\{J, j + K\}$.

2.1.2 Geometric Consistency

Even better discrimination between noise and edges may be achieved by imposing geometrical constraints. Usually, edges do not appear isolated in an image. They form contour lines, which we assume to be polygonal (i.e. piecewise linear). In our approach, a coefficient $M_{2^j} f[n, m]$ should have a higher shrinkage factor if its neighbors along the local contour direction also have large shrinkage factors. To detect this kind of behavior, we first quantize the gradient directions $\theta_{2^j} f$ into $0^\circ, 45^\circ, 90^\circ$ or 135° . The contour lines are orthogonal to the gradient direction at each edge element, so we can estimate the contour direction from $\theta_{2^j} f$. We then add up the shrinkage factors $g_j^{\text{scale}}[n, m]$ along the contour direction, obtaining the shrinkage factors $g_j^{\text{geom}}[n, m]$, as follows:

$$(8) \quad \begin{cases} \sum_{i=-N}^N \alpha[i] g_j^{\text{scale}}[n + i, m], & \text{if } C_{2^j}[n, m] = 0^\circ, \\ \sum_{i=-N}^N \alpha[i] g_j^{\text{scale}}[n + i, m + i], & \text{if } C_{2^j}[n, m] = 45^\circ, \\ \sum_{i=-N}^N \alpha[i] g_j^{\text{scale}}[n, m + i], & \text{if } C_{2^j}[n, m] = 90^\circ, \\ \sum_{i=-N}^N \alpha[i] g_j^{\text{scale}}[n + i, m - i], & \text{if } C_{2^j}[n, m] = 135^\circ, \end{cases}$$

where $C_{2^j}[n, m]$ is the local contour direction at the pixel $[n, m]$, $2N + 1$ is the number of adjacent pixels that should be aligned for geometric continuity, and $\alpha[i]$ is a Gaussian window that allows neighboring pixels to be weighted differently, according to their distance from the pixel $[n, m]$ under consideration.

In the presence of noise, randomly aligned coefficients occur, and could also be strengthened. To overcome this potential difficulty, we compare the contour direction in two

consecutive levels. It is expected that contours would be aligned along the same direction in two consecutive levels (it is the same contour at different resolutions), but responses due to noise should not be aligned (gradients associated to noise will not be oriented consistently in consecutive resolutions). Therefore, a second updating rule is applied to the shrinkage factors $g_j^{\text{geom}}[n, m]$. The second updating rule takes into account the normalized inner product of corresponding vectors in consecutive resolutions:

$$g_j^{\text{geom}}[n, m] = g_j^{\text{geom}}[n, m] \cdot |\cos(\theta_{2^j} f[n, m] - \theta_{2^{j+1}} f[n, m])|. \quad (9)$$

Notice that the factor $|\cos(\theta_{2^j} f[n, m] - \theta_{2^{j+1}} f[n, m])|$ provides a measure for direction continuity. It has value one if the same direction occurs in two consecutive levels, and value zero if the orientations differ by 90° (i.e., are orthogonal).

2.2 Edge Enhancement

The approach described above can be extended to enhance edges in noisy images. The shrinkage factors $g_j^{\text{geom}}[n, m]$ assume values between 0 and 1, which is useful for the selective shrinkage of the wavelet coefficients in image denoising. However, for edge enhancement purposes, we allow these shrinkage factors to be greater than 1, so the corresponding wavelet coefficients will be enhanced, as well as some relevant image edges, when the inverse transform is applied.

Therefore, we introduce an edge enhancement function $h_j : [0, 1] \rightarrow [0, +\infty)$, which is used for updating the shrinkage factors $g_j^{\text{geom}}[n, m]$ according to:

$$g_j^{\text{enh}}[n, m] = h_j(g_j^{\text{geom}}[n, m]). \quad (10)$$

A simple choice for $h_j(v)$ is a linear function $h_j(v) = \beta_j v$, for $\beta_j \geq 1$. The values β_j are chosen to control the amount of edge enhancement at each scale 2^j , allowing the enhancement of structures with different sizes. To enhance small objects, we should choose high values for β_j with j small. To enhance only large structures, high values should be chosen for β_j with j large. The parameters β_j should be chosen based on the average size of the objects to be segmented in the image. The updated wavelet coefficients $NW_{2^j}^i f[n, m]$ are computed using equation (3) with $g_j^{\text{enh}}[n, m]$ instead of $g_j[n, m]$, and the denoised/enhanced image is obtained.

For example, consider the *bacteria* image shown Figure 1. It can be noticed that background noise is intense, and some edges are fuzzy. Figure 2 shows the denoising and edge enhancement of the *bacteria* image, using three levels in the wavelet decomposition, and $\beta_1 = 1, \beta_2 = 4.5, \beta_3 = 2$ were the enhancement factors.

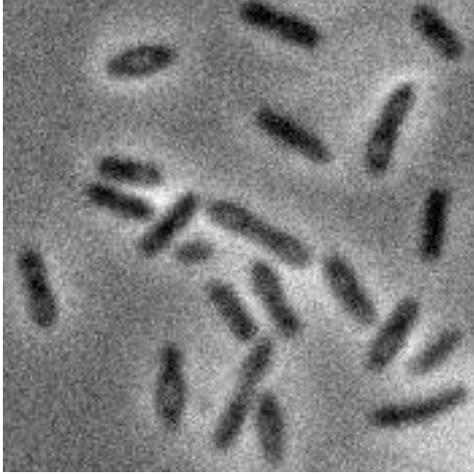


Figure 1: *Bacteria* image.

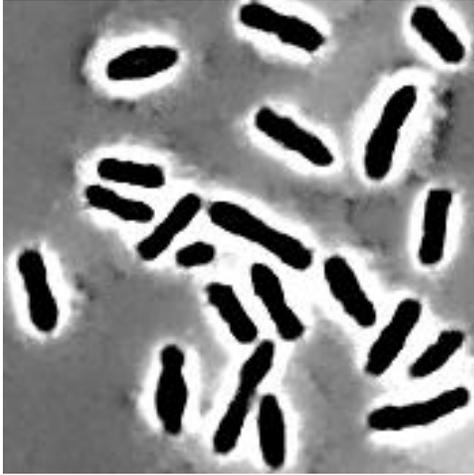


Figure 2: Enhanced *bacteria* image.

3 Computing the Watershed Transform

After applying the technique described above, we obtain a denoised image with enhanced edges. The wavelet transform is recomputed for the enhanced image, and the detail images $WW_{2^j}^1 f[n, m]$ and $WW_{2^j}^2 f[n, m]$ are obtained at each scale 2^j . The enhanced edge magnitudes $M_{2^j}^{\text{enh}} f$ are then obtained as follows:

$$M_{2^j}^{\text{enh}} f = \sqrt{(WW_{2^j}^1 f)^2 + (WW_{2^j}^2 f)^2}, \quad (11)$$

An appropriate scale 2^j should be chosen depending on the size of the objects that should be segmented in the image. Let 2^J be this scale, and let $M^{\text{enh}} = M_{2^J}^{\text{enh}} f$ be the gradient image.

Even after the pre-processing stage, some spurious

gradients still remain in the image. To remove these undesired small gradients, a threshold T is applied to the gradient image, and values of M^{enh} smaller than T are set to zero. The threshold T is selected as a standard value, such as $T = \kappa \max(M^{\text{enh}})$, where $\max(M^{\text{enh}})$ is the maximum of the gradient magnitudes, and κ is a constant ($0 < \kappa < 1$).

The enhanced magnitudes M^{enh} for the *bacteria* image (using $\kappa = 0.2$) is shown in Figure 3.

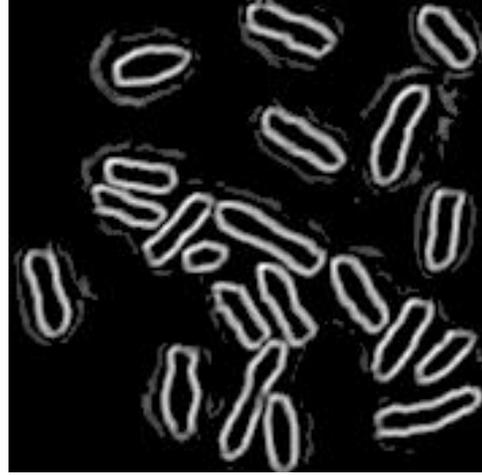


Figure 3: Enhanced gradient magnitudes for the *bacteria* image.

The watersheds of M^{enh} are then computed, and the segmented image is obtained. In this work, we used the watershed implementation based on immersion simulations [2]. With this approach, it is assumed that holes are pierced in each regional minimum of the gradient image (which is viewed as a topographic surface). This surface is slowly immersed into a lake, and the water will progressively fill up the catchment basins of the surface, starting from the lowest altitude points. At each pixel where the water coming from different minima would merge, a “dam” is built. After the hole surface is flooded, each minimum is completely surrounded by dams (the watersheds), which delimit its catchment basin (the segmented region).

The watersheds for the *bacteria* image are shown in Figure 4. The bacteria were segmented from the background, but some small undesired regions also appeared in the segmentation.

A post-processing stage is then applied to merge regions separated by low-contrast borders, and to remove erroneously segmented regions that are small in size and/or do not satisfy the problem specific criteria. In this case, we have chosen simple problem constraints, such regions smaller than a particular size are not considered to be *valid segmented regions*.

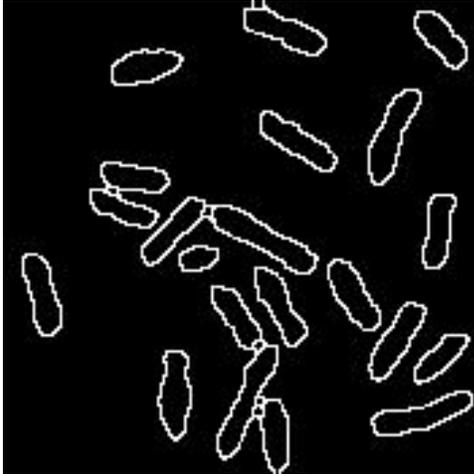


Figure 4: Watersheds computed for the *bacteria* image.

4 Post-Processing

Often, the realistic objects exist within a range of sizes in an image. Under these circumstances, it is possible to set a minimum region area to segment valid regions. We have adopted the following criteria for merging adjacent regions, using the area constraint. Let T_A be the smallest area allowed. If a certain region has an area smaller than T_A , then its borders with the neighboring regions are searched, and this region is merged with the neighboring region which has the widest border.

Also, if the border between two adjacent regions is weak (i.e. has low contrast), the regions are merged. To decide if a border is strong enough to keep two regions apart, a hysteresis-like procedure is used, similar to Canny's [9].

Two thresholds $T_1 < T_2$ are chosen, and a particular border is kept if all the gradients along this border are greater than T_1 , and at least a fraction p of the gradients are greater than T_2 . Otherwise, the regions separated by this border are merged.

As an illustration of the efficiency of our post-processing technique, the final segmentation of the *bacteria* image is displayed in Figure 5. Areas smaller than 9 pixels were merged ($T_A = 9$), and the thresholds T_1 and T_2 in the hysteresis-like step are, respectively, 25% and 40% of the maximum gradient magnitude obtained for this image.

5 Experimental Results

We compared our proposed technique with the watersheds obtained after a smoothing process (denoising), and edge detection (to obtain the gradient magnitudes). More specifically, we used a Gaussian kernel for denoising [10], and the Prewitt operator [11] to extract the digital gradient (a threshold was used to remove gradients with a magnitude

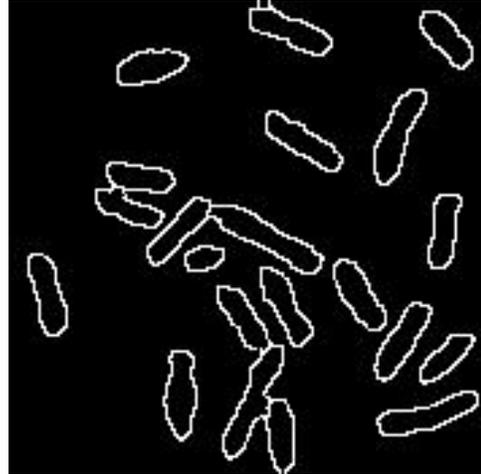


Figure 5: Post-processed watersheds computed for the *bacteria* image.

smaller than 20% of the maximum magnitude).

The watersheds obtained using the Prewitt operator to estimate the gradients of the *bacteria* image are shown in Figure 6. Comparing with Figure 4, it is clear that the proposed technique produces a more accurate segmented image.

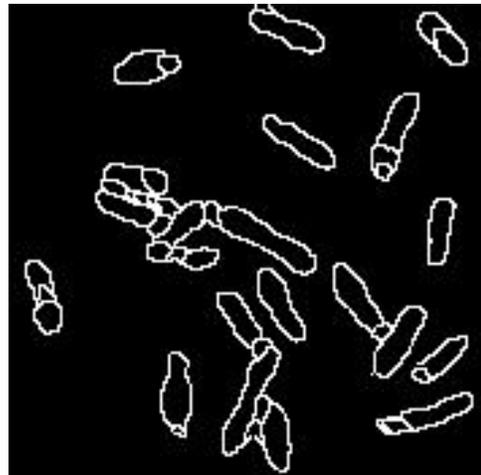


Figure 6: Watersheds computed for the *bacteria* image using the Prewitt operator to estimate the gradient image.

Another example is shown in Figure 7. This image displays grains of rice in a slowly varying background intensity (in the vertical direction). This image is noisy, and presents low contrast when the background is lighter. Figures 8(a) and 8(b) show the obtained watersheds using the Prewitt operator (pre-processed with a Gaussian filter), and

our denoising and enhancement technique, respectively.

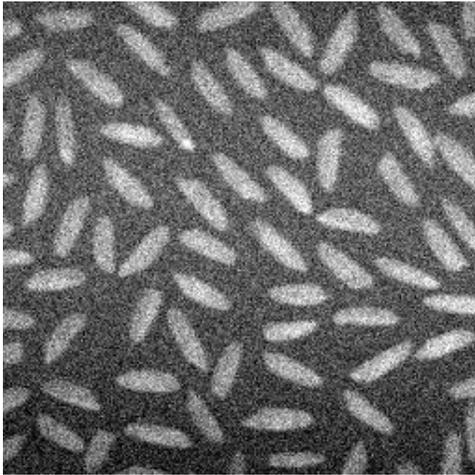


Figure 7: *Rice* Image.

Figure 9 shows four slices of a tomographic image of a soil sample. Although these images are not very noisy, the borders between the objects are weak and blurry. Figures 10 and 11 show, respectively, the watersheds obtained using the Prewitt operator (with a Gaussian filter), and the proposed denoising and enhancement technique. It is noticeable that without the enhancement provided by our technique, the regions in the interior of the slice are missed completely (this is clear in the watersheds obtained using with Prewitt operator).

6 Conclusions

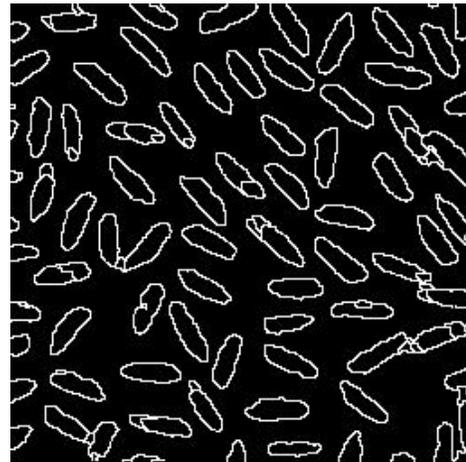
In this paper, a new method for improving the robustness of the segmentation based on watersheds was proposed. It relies on a pre-processing stage, for image denoising and edge enhancement. The watershed transform is then applied to the gradient image, obtained in the wavelet domain, at a given resolution 2^J . Finally, a post-processing procedure is applied to remove regions with small areas, and to merge regions with low-contrast boundaries.

Our preliminary results indicate that the over-segmentation problem, which is characteristic of the watersheds technique, can be reduced. Also, broken contours due to weak edges are significantly eliminated with our approach.

The proposed technique is robust when applied to noisy and/or blurred images, and in general performs better than using conventional filtering and edge detection techniques as a pre-processing stage.

Future work will concentrate on extending this approach to color image segmentation and analysis.

(a)



(b)

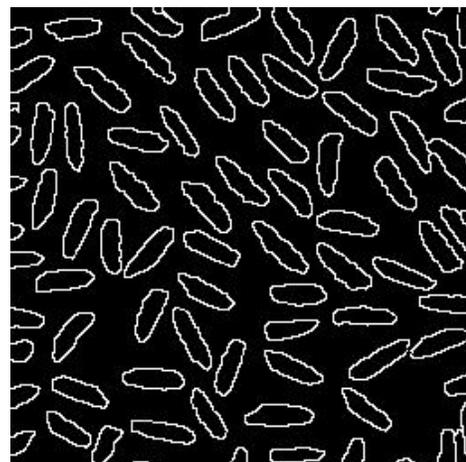


Figure 8: Watersheds for *rice* image. (a) Using the Prewitt operator (with Gaussian filtering). (b) Using the proposed denoising/enhancement procedure with post-processing.

7 Acknowledgements

This work was supported by FAPERGS and CNPq.

References

- [1] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," in *Proc. IEEE International Workshop on Image Processing, Real-Time Edge and Motion Detection / Estimation*, (Rennes, France), 1979.
- [2] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion sim-

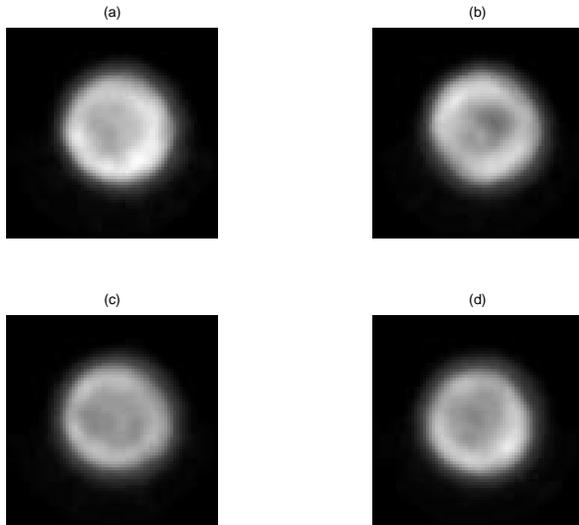


Figure 9: Four slices of a porous solid tomography.

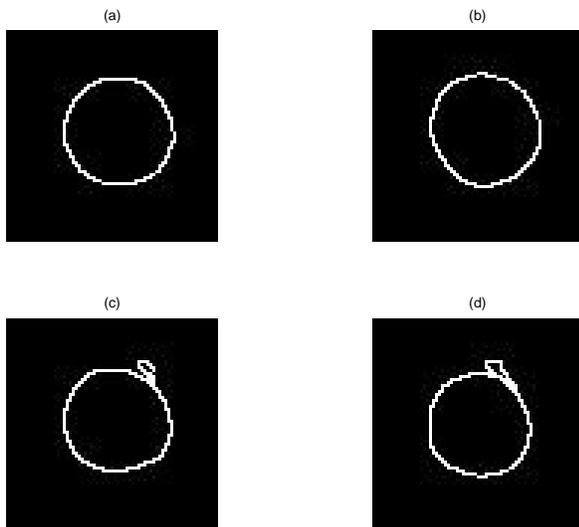


Figure 10: Watersheds computed for the tomographic image using the Prewitt operator to estimate the gradient image.

ulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 583–598, Jun 1991.

- [3] A. Bieniek and A. Moga, “An efficient watershed algorithm based on connected components,” *Pattern Recognition*, vol. 33, no. 6, pp. 907–916, 2000.
- [4] F. Meyer, “Levelings and morphological segmentation,” in *Proceedings of SIBGRAPI’98*, (Rio de Janeiro, Brazil), pp. 28–35, 1998.

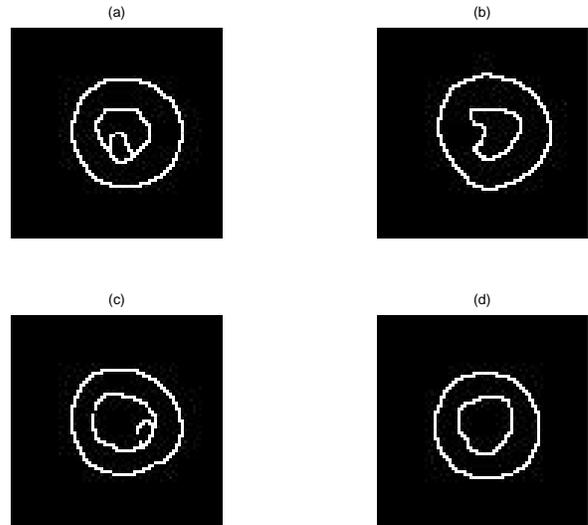


Figure 11: Watersheds computed for the tomographic image using the proposed denoising/enhancement technique with post-processing.

- [5] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Transactions on Image Processing*, vol. 7, pp. 1684–1699, December 1998.
- [6] J. Weickert, “Efficient image segmentation using partial differential equations and morphology,” 2000.
- [7] J. Scharcanski, C. R. Jung, and R. T. Clarke, “Adaptive image denoising using scale and space consistency,” *IEEE Transactions on Image Processing*, 2001. To appear in the September 2002 issue.
- [8] S. G. Mallat and S. Zhong, “Characterization of signals from multiscale edges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, 1992.
- [9] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.
- [10] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [11] W. K. Pratt, *Digital Image Processing*. New York: John Wiley and Sons, 1991.