Improving the Stochastic Watershed[☆]

Karl B. Bernander[1], Kenneth Gustavsson[1], Bettina Selig, Ida-Maria Sintorn, Cris L. Luengo Hendriks[*]

Centre for Image Analysis,
Uppsala University and Swedish University of Agricultural Sciences,
Box 337, 75105 Uppsala, Sweden

Abstract

The stochastic watershed is an unsupervised segmentation tool recently proposed by Angulo and Jeulin. By repeated application of the seeded watershed with randomly placed markers, a probability density function for object boundaries is created. In a second step, the algorithm then generates a meaningful segmentation of the image using this probability density function. The method performs best when the image contains regions of similar size, since it tends to break up larger regions and merge smaller ones. We propose two simple modifications that greatly improve the properties of the stochastic watershed: (1) add noise to the input image at every iteration, and (2) distribute the markers using a randomly placed grid. The noise strength is a new parameter to be set, but the output of the algorithm is not very sensitive to this value. In return, the output becomes less sensitive to the two parameters of the standard algorithm. The improved algorithm does not break up larger regions, effectively making the algorithm useful for a larger class of segmentation problems.

Keywords: Mathematical morphology, Image segmentation, Random process, Stochastic watershed, Seeded watershed, Uniform grid

1. Introduction

The watershed is a well-known tool for image segmentation (Beucher and Lantuejoul, 1979; Vincent and Soille, 1991). It tessellates the image such that each region has precisely one local minimum, and the region boundaries run along the crest lines. It is typically applied to the gradient magnitude image,

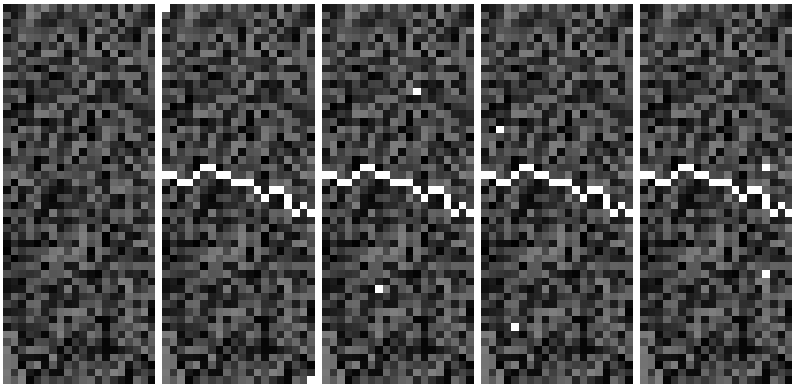
[☆]Published in: Pattern Recognition Letters 34(9):993-1000, July 2013
[*]Corresponding author. Tel: +46 18 471 3471. Fax: +46 18 55 34 47
Email address: cris@cb.uu.se (Cris L. Luengo Hendriks)
[1]These authors contributed equally to this paper.

yielding regions of approximately constant intensity. However, due to noise, any non-synthetic image has too many local minima. This can be solved with either a method to remove local minima as a preprocessing step, or a region merging method as a postprocessing step. Alternatively, if a marker can be created for each region to be delineated, the image can be transformed such that it only has local minima at these markers (Meyer and Beucher, 1990); this is usually referred to as *seeded watershed*. This does, however, not solve the segmentation problem; it just transforms it from a region merging problem to a object detection problem.

Angulo and Jeulin (2007) recently proposed the *stochastic watershed*. This algorithm works in two steps. First, through repeated application of the seeded watershed with random markers, the stochastic watershed constructs a probability density function (PDF) for the boundaries in the image. Second, a final segmentation is obtained from this PDF by suppressing unimportant local minima and applying the classical watershed algorithm. Minima are suppressed using a volume criterion (Angulo and Jeulin, 2007) or preferably a depth criterion (Angulo and Velasco-Forero, 2010). This latter criterion can be imposed using the h-minima transform (Salembier and Serra, 1995). The PDF computed by the stochastic watershed in n iterations converges as $n \to \infty$ to the probability of markers falling on both sides of each boundary. This probability is given by the relative sizes of the catchment basins on either side of the boundary (Meyer and Stawiaski, 2010). However, the PDF typically becomes stable after about $n = 50$ iterations (Angulo and Jeulin, 2007).

The seeded watershed can be seen as a path minimization problem (see Cousty et al. (2010) for a recent description unifying the frameworks) where each point of the image is assigned to the marker with the shortest distance. The distance measure in this case is given by the largest gray value along the optimal path between the point and the marker. This is an incredibly stable measure, very insensitive to the location of the markers. For example, the image in Figure 1(a), when segmented using markers in the top left and bottom right corners, produces the segmentation line shown in Figure 1(b). These two markers can be moved anywhere within their respective regions without changing the segmentation result. Figure 1(c-e) show the result of the segmentation with markers in other random places; the resulting lines are identical.

This stability is what allows the stochastic watershed to find ridges in the input image. However, this stability also consistently places segmentation lines in places where there are no actual ridges, as in the example in Figure 1. Consequently, larger regions tend to be split by this method. Furthermore, as it is less likely that a small region receives a marker in any one iteration, the PDF at borders between smaller regions is lower than that between larger regions. Both effects can be seen in Figure 2, which demonstrates that the stochastic watershed prefers segmenting an image into equally sized regions.



(a) (b) (c) (d) (e)

Figure 1: The seeded watershed is very insensitive to marker placement. (a) Input image. (b-e) Segmentation result (white line) as a result of the two markers shown as white pixels.

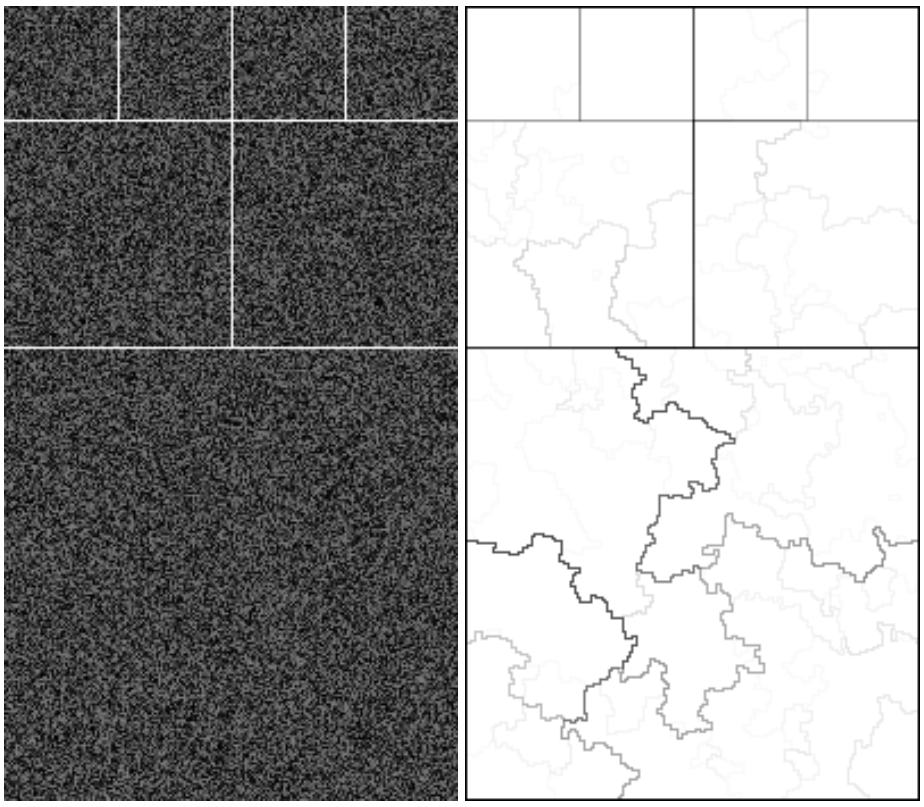
In this paper we propose two simple modifications to the stochastic watershed that will strongly improve its properties: we randomize the way in which the watershed grows the regions by adding noise to the input image at every iteration, and we distribute the markers using a randomly placed grid.

2. Methods

2.1. Adding noise

The very strong stability of the watershed segmentation discussed in the introduction, as seen by its insensitivity to the placement of markers, is due to the order in which pixels are added to the regions. The markers are the seeds, the initial regions. These grow by adding the neighbors with lowest grey value first. Two regions will therefore always meet at pixels with a high grey value. Ridges in the input image, which indicate the object boundaries, are therefore consistently found. However, noise will also naturally form ridges at which these regions meet. A uniform area in the image, which has no ridge, will consistently produce the exact same watershed line if there are markers on both sides of this line. This is because pixels in this area are processed in the same order during every iteration, yielding the same region boundary. This order can be broken by adding a small amount of noise to the image. The noise must be strong enough to hide the noise in the image, but not so strong that it hides the true ridges we are looking for.

We suggest the addition of uniform noise to the input image before each iteration of the seeded watershed. Normally distributed noise has the exact same effect, but takes more time to compute and is not bounded, potentially producing extremely large or small values.



(a) (b)

Figure 2: The stochastic watershed finds regions of similar size. (a) Input image. (b) PDF computed by 50 iterations of the stochastic watershed with 7 markers. White indicates a probability value of 0, black a probability value of 50 (which is the maximum obtainable with 50 iterations). Note how lines across the largest region have a very large value (up to 35), whereas the boundaries of the small regions at the top of the image have a much lower value (as low as 17).

2.2. Randomly placed grid

Given a set of markers produced by a Poisson process (equivalent to taking a set of coordinates from a uniform distribution), it is expected that some points will be close together whereas other points will be far away from any other point. This is a sub-optimal situation, as points that are close together are more likely to be within the same region, causing the seeded watershed to split that region in two. It has been suggested to direct the markers using a spectral classification (Noyel et al., 2008) or a rough foreground/background segmentation (Faessel and Jeulin, 2010). These methods require additional computing and somewhat tailor the stochastic watershed to a specific problem. Instead, we propose to use points on a regular grid, randomly placed over the image. While trying out different marker distributions (results not shown), we found that a more even distribution improved the segmentation results, whereas clustering the markers made the segmentation worse. A uniform grid is the easiest way to obtain an even distribution of makers.

In stereology, points thrown randomly over an image are used to quantify the surface area of an object in that image (Mouton, 2002). Instead of independent random sampling (i.e. using Poisson distributed points), uniform random sampling is used exclusively. In uniform random sampling, a grid is placed randomly over the image, and grid points are used as sample points. It has been shown that uniform random sampling produces the same unbiased estimate for area that independent random sampling does, but more efficiently in that fewer samples are needed to obtain the same precision (Mouton, 2002). This lead us to investigate uniform grids for the purpose of producing markers for the stochastic watershed.

We looked at the square and hexagonal grids, both well known in the image analysis community, as a means to generate markers. We introduced a random offset and rotation to the grids, so that each iteration of the stochastic watershed produced a different result. Grids were generated with a density equivalent to the density of the Poisson process we were replacing, and a marker was placed at each grid point. Every instance of the grid was translated randomly over one grid period, and rotated randomly according to the rotational symmetry of the grid (i.e. the square grid rotated up to 90°, the hexagonal grid rotated up to 60°). This way of randomizing the grid position in the image caused each pixel in the image to be equally likely to receive a marker. The random offset and rotation caused each instance of the grid to produce a slightly different number of markers within the image. As the stochastic watershed is not too sensitive to the number of markers, as can be seen by the experiment in Section 3.3, this does not affect the result in any way.

In geostatistics, a method called spatially stratified sampling (Cressie, 1993) is often used. One implementation of stratified sampling is dividing the image into squares, and randomly selecting one pixel from each square. If we put a marker

on pixels selected this way, we obtain yet another way of randomizing markers for the stochastic watershed. In our experiments we did not find an advantage for this method over the random grids described above (data not shown).

2.3. Quantitative evaluation of PDFs

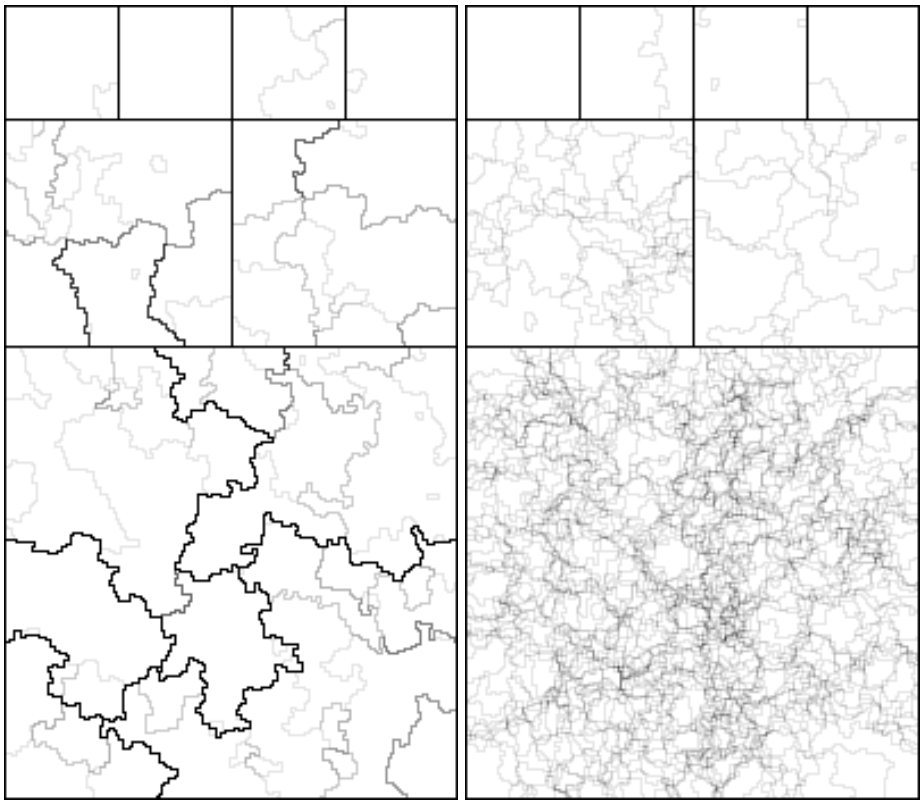
The PDF produced in the first step of the stochastic watershed is the input to a final watershed, which suppresses some of the local minima based on the volume or depth of their catchment basins (Angulo and Jeulin, 2007; Angulo and Velasco-Forero, 2010). This step thus involves yet another parameter. To quantify the behavior of the stochastic watershed while ignoring this additional parameter, we decided to obtain a measure on the PDF directly. Given the known, ideal segmentation, we determined the smallest PDF value on the segmentation lines (t), and the largest PDF value where there are no lines (f). In principle, a threshold at a value in between these two would yield a correctly segmented image. The ratio of these two values, t/f, is a measure of how easy it is to find that threshold value, and, by extension, the parameter to the minima suppression. A ratio of one or less indicates that it is not possible to obtain a correct segmentation given the PDF.

3. Experiments and results

3.1. Adding noise

To demonstrate the effect of adding noise at each iteration of the stochastic watershed, we did so for the image of Figure 2(a). Figure 3(b) shows the result. For comparison, Figure 3(a) shows the result without the noise (this is the same result as shown in Figure 2(b), but with the contrast modified). Here, we plotted the PDFs such that values of 10 or more are shown in black; white indicates a value of 0. It can be seen that the standard stochastic watershed produced some false lines with a very large value of the PDF, especially in large regions. By adding noise, which randomized the order in which pixels were added to the various regions, each iteration of the seeded watershed produced such false lines in different locations within these large regions. Therefore, they did not add up to significant values.

We quantified the improvement obtained through this modification in the following manner. We generated a test image of 480 by 480 pixels (see Figure 4), containing lines that divided the image into 16 large and 12 small regions. The large regions had an area five times that of the small regions. The lines had a grey value of 255. The background contained normally distributed noise with a mean of 64 and a standard deviation of 32. We applied the stochastic watershed using $n = 50$ iterations and $s = 28$ markers (equal to the number of regions). The markers were randomly distributed using a Poisson process of density yielding s markers on average, thus the actual number of markers changed from instance



(a) (b)

Figure 3: Adding noise at each iteration effectively avoids repeatedly finding the same watershed line where there is no actual ridge in the input image. (a) Contrast-stretched version of the image in Figure 2(b). Pixels with an intensity of 10 or more are shown as black. Many pixels have a value of 0, meaning that they are never seen as part of a possible segmentation boundary ($t = 17$, $f = 35$). (b) PDF computed in the same way, but adding noise at each iteration. False contours appear at a different location at each iteration, and only at true ridges does the PDF receive a large value ($t = 15$, $f = 7$).

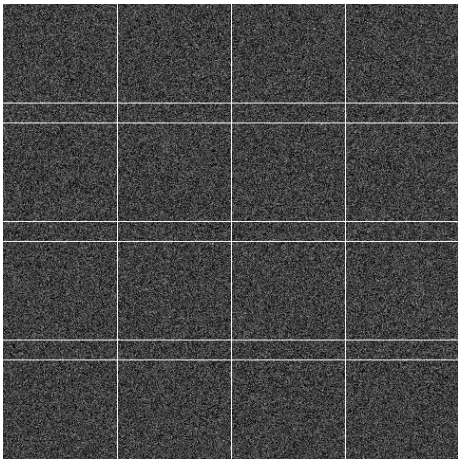


Figure 4: Test image used for Figures 5(a) and 5(b).

to instance. From the resulting PDF we obtained the ratio t/f, as described in Section 2.3. We repeated the procedure 40 times for each of 5 different strengths of noise added, including 0 to correspond with the standard stochastic watershed. Noise was added to the input image before each application of the seeded watershed. The results are shown as box plots in Figure 5(a). As can be seen, any sufficiently large amount of noise produces satisfactory results, with the best results for noise stronger than that in the input image, but not strong enough to hide the signal.

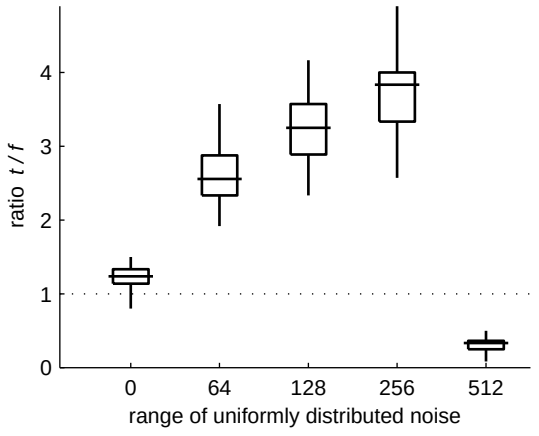
3.2. Randomly placed grid

We used the exact same procedure to quantify the improvement due to the new method of marker selection. Figure 5(b) shows statistics for 40 repetitions with each of the methods: the standard Poisson process distribution, a square grid, and a hexagonal grid. In each of these cases, we added uniformly distributed noise with a range of 256 at every iteration of the algorithm (this was the optimal strength found in the previous experiment). Either grid type improved the result of the random distribution.

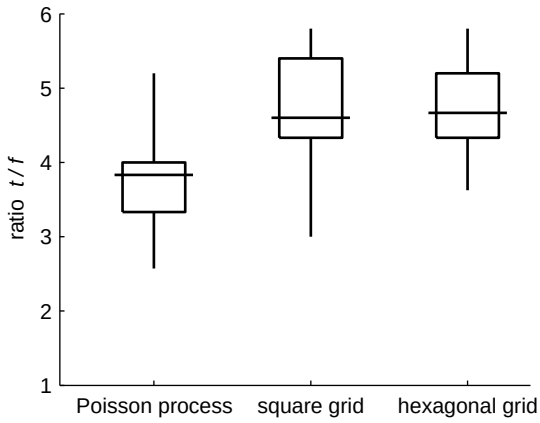
We performed the same experiment but without adding noise. In this case the difference between the various methods of distributing markers was insignificant (data not shown). This demonstrates that, to allow the stochastic watershed to correctly segment an image with a strong variation in region sizes, it is required to add noise to the image at every iteration.

3.3. Number of markers

The standard stochastic watershed performs optimally when using as many markers as there are regions in the image (Angulo and Jeulin, 2007; Selig and



(a)



(b)

Figure 5: (a) Performance of the stochastic watershed with different noise strengths. Even small amounts of noise show a large improvement over the zero noise case (equivalent to the standard stochastic watershed). Too much noise destroys the algorithm's ability to detect ridges. (b) Performance of the stochastic watershed with different marker distributions. Using regular grids, placed randomly over the image, significantly improved the result when compared to the standard random distribution. In both graphs, the box indicates the second and third quartiles, the horizontal line the median, and the vertical lines the minimum and maximum datum.

Luengo Hendriks, 2012). As the number of markers is increased, the probability of two markers hitting the same region grows, and thus the value of the PDF along false lines increases as well. But because the improvements proposed in this paper strongly reduce the influence of these false lines, it is reasonable to assume that the improved method degrades less quickly with increasing number of markers.

Likewise, for fewer markers than number of regions, the value of the PDF on region boundaries decreases, as it is less likely that a marker is placed on both sides of a boundary at every iteration. However, the value of the PDF at false lines also decreases. It is reasonable to assume that the ratio t/f we have been using to characterize the PDF converges to a similar value, but at a reduced pace. That is, more iterations are needed for convergence when using fewer markers. This behavior should apply to both the standard and the improved algorithms.

We tested these assumptions by looking at the t/f ratio after every iteration, for 40 repetitions and during 100 iterations. We did this for both the standard method and the improved method (hexagonal grid and uniform noise with a range of 256), varying only the number of markers s. The test image was constructed as that in Figure 4, but the lines created 36 square regions of equal size, so that the standard method could produce good results. The number of markers used was varied between one fourth and four times the ideal value of $s = 36$. The results are plotted in Figure 6.

When increasing s above its ideal value, the standard method converged to a lower ratio, and became useless with $s = 144$. The improved method, however, converged to high ratios over the full range of selected values of s. For both methods, the value to which the ratio converged changed when reducing s below its ideal value, but not very strongly.

By noting at what point the lines crossed $t/f = 1$, one can see that, for the improved algorithm, fewest iterations were needed to reach $t/f > 1$ when $s = 36$ (the ideal value). This number of iterations approximately doubled every time s was halved. This indicates that the convergence rate is linearly dependent on the number of markers, for fewer than the ideal number. For larger values of s this dependency no longer holds.

One clear feature of these graphs is that, for larger number of seeds, the results of the various independent runs of the algorithm (grey lines) are closer together. This is due to there being less possible variation as the number of seeds increases. Another interesting feature is the regular zig-zag behavior of the improved algorithm with $s = 36$ (and also, but less strongly, for other values of s). This can be attributed to the interaction of the grid used to place the markers and the grid in the image.

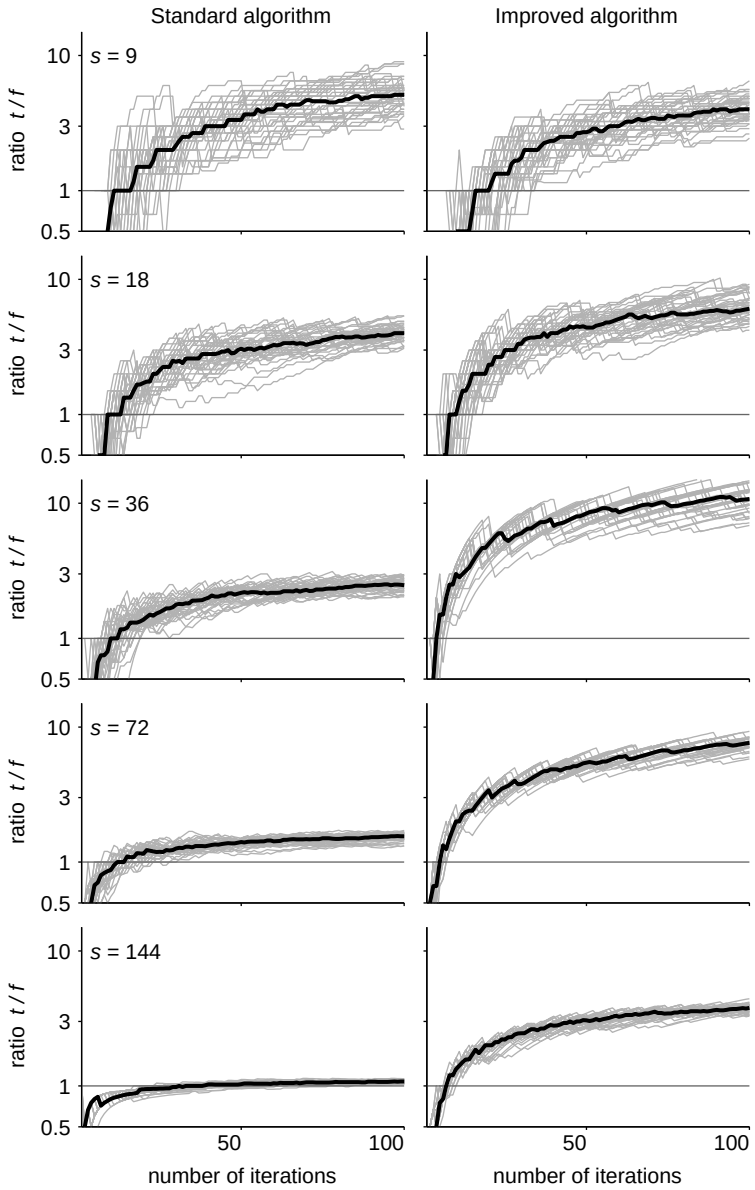


Figure 6: Comparison of standard vs improved stochastic watershed, convergence behavior with respect to the number of markers s (ideally, $s = 36$). The standard stochastic watershed converged to a high t/f ratio as long as the number of markers was smaller than the number of regions in the image, albeit at a smaller rate for fewer markers. For a larger number of markers, the ratio dropped quickly, making the operation useless. In contrast, the improved stochastic watershed still converged to high ratios even for marker densities four times larger than optimal. The gray lines show 40 repetitions, the black line follows the median value at each iteration.

3.4. Example application: fluorescent microscope images of nuclei

To show the improvement over the standard stochastic watershed provided by the ideas presented in this paper, we apply the method to a database of images with hand-drawn ground-truth segmentation (Coelho et al., 2009). Of the two different collections in this database, we picked the first one, containing 48 images of U2OS cells (one of which was unreadable, leaving us with 47 images), as in Figure 7. We divided the set into a training set of 20 images and a test set of 27 images. We computed the gradient magnitude (Gaussian gradients with $\sigma = 1$ px) of each image, normalized them to a maximum value of 1, and sub-sampled them by a factor 2 in each dimension to reduce computation time. The resulting images were used as input to the standard and the improved stochastic watershed.

We first applied the two algorithms to the training set, with a large range of values for the two parameters s (number of markers) and h (depth threshold). We fixed the number of iterations to 50. The improved method used uniform noise in the range [0,0.1] and a random hexagonal grid to place the markers. Each resulting segmentation was compared with the hand-drawn ground truth using the same methodology as Arbeláez et al. (2011). In short, a routine found the best assignment between pixels on the computed boundaries and those in the ground truth. Assignments further than 4 pixels are considered a mismatch. The fraction of matched pixels in the segmentation is considered the true positive fraction or precision; the fraction of matched pixels in the ground truth is considered the recall. The reported number is the F-measure, the geometric average of precision and recall. For the standard stochastic watershed, the maximum average F-measure ($F = 0.65$) was obtained for $s = 25$ and $h = 4$; for the improved method this was $F = 0.86$, $s = 200$ and $h = 6$.

Subsequently we applied the two algorithms, with the optimal parameters as determined on the training set, to the remaining 27 images. We obtained average F-measures of $F = 0.63$ (min: 0.53, max: 0.78) and $F = 0.86$ (min: 0.63, max: 0.93) for the standard and improved stochastic watershed, respectively.

In Figure 7 we show the image from the test set for which the two F-measures were closest to the averages for the set. As expected given the previous examples, the PDF from the standard method contains high-valued lines crossing the background, where no boundaries exist in the image. Consequently, the segmentation result has the background split into many regions. In contrast, the PDF for the improved method has low values everywhere in the background region, showing that lines dividing the background appear at a different place every iteration. The resulting segmentation, though not perfect by any means, has a single region corresponding to the background of the image. Note that this image is, as most of the images in the set, not particularly difficult to segment for a dedicated algorithm. And in fact, even a simple morphological smoothing before computing the gradient magnitude could have greatly improved the segmentation result. What we

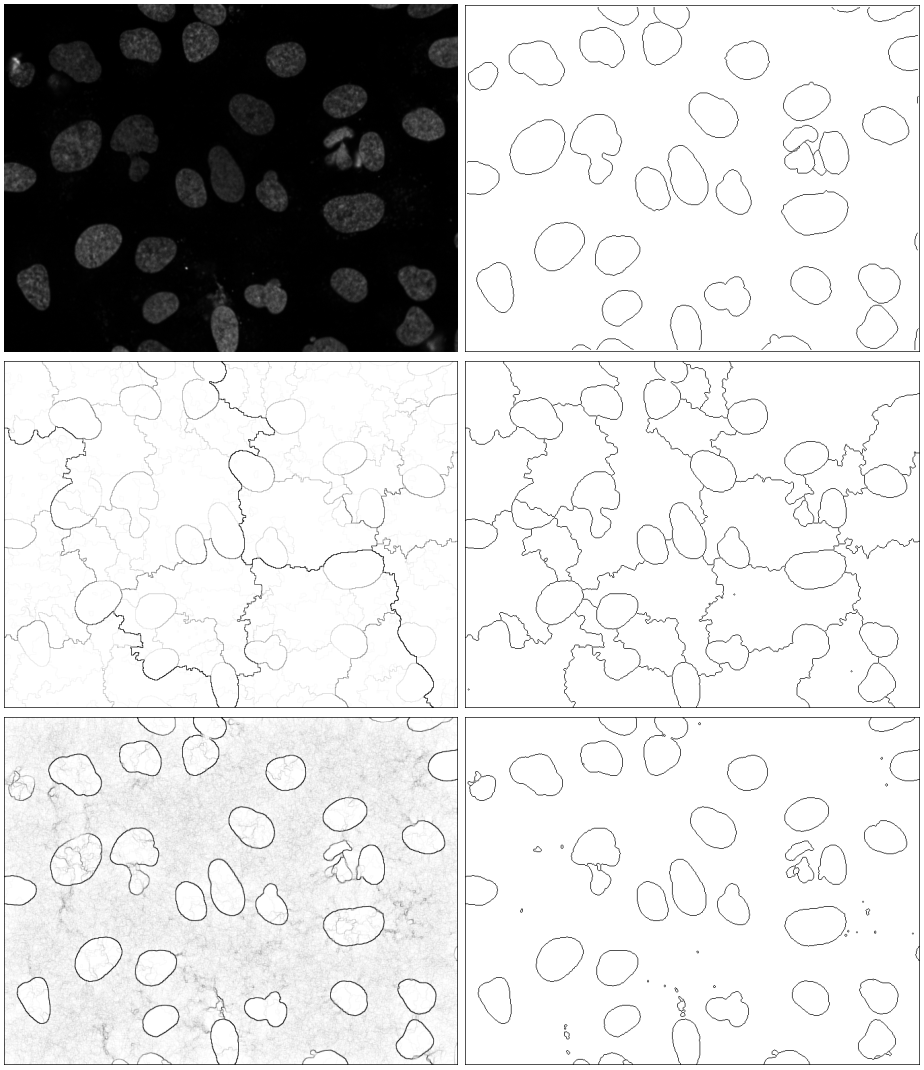


Figure 7: One image from the test set (top left) and its ground-truth segmentation (top right). The middle row is the result of the standard stochastic watershed on the gradient magnitude ($F = 0.65$), and the bottom row is the result of the improved method ($F = 0.86$); for both rows, left is the PDF and right is the final segmentation. The parameters for the algorithms are those that optimized the average F-measure on the training set.

intend to show with this experiment, though, is the large improvement obtained with the simple suggestion in this paper.

4. Discussion and Conclusions

The stochastic watershed is a powerful tool for unsupervised segmentation. In essence, it has only two parameters that need to be adjusted to the problem at hand: the number of markers s, and the depth threshold h for the final watershed on the PDF (the parameter to the h-minima transform). The number of markers should be close to the number of regions that are expected in the image. Obviously, it is often not known a priori how many regions there are in the image, making it difficult to choose the optimal value for parameter s. The parameter h does not change as much from application to application. A limitation of the stochastic watershed is that it is applicable only when all the regions in the image are of similar size.

The two simple modifications proposed in this paper improve the stochastic watershed in three ways: (1) the algorithm does a better job of segmenting images with large differences in region size, broadening the class of segmentation problems that the method is suitable for; (2) the output is less sensitive to the number of markers, making it easier to set the parameter s; and (3) the output has an increased ratio t/f, meaning that a correct segmentation is obtained for a wider range of values of h, which makes it easier to set that second parameter. To obtain these benefits, one new parameter is introduced: the range of the uniformly distributed noise that is added to the image at each iteration. This noise should ideally be a little stronger than the noise in the input image. It should be possible to guess the best value for this parameter automatically by estimating the noise in the image, for example using the method by Immerkær (1996). Nonetheless, the output is quite insensitive to changes in this parameter, as any small amount of noise produces acceptable results.

The first modification, adding noise at every iteration, is by far the most important of the two. It is this change that provides the three improvements enumerated above. Placing the markers using a random grid further increases the ratio t/f, but by itself does not make large changes to the PDF of the standard stochastic watershed.

The computation time for the stochastic watershed is increased when adding noise to the input image at every iteration. In essence, the program must compute a random number for each pixel in the image, n times (with n the number of iterations of the stochastic watershed). Given that the seeded watershed is an $O(N \log q)$ operation,[2] the complexity is not changed (here N is the number of

[2]If the input to the seeded watershed is an interger-valued image, it is possible to use a $O(1)$

pixels in the image and $q \approx \log N$ is the number of pixels in the propagating boundary). Placing the markers using a random grid is computationally more efficient, in theory, than placing them using a Poisson process simulation. However, because our implementation for the translated and rotated grid was not optimized, this difference was hardly visible in our tests. For the test image of Figure 4 (480 pixels square), and 50 iterations, the computation time increased from 7.8 s to 8.8 s.

In the original stochastic watershed paper, Angulo and Jeulin (2007) suggested the option of regionalizing the random markers by placing them where the gradient magnitude is large. That is, they suggested placing markers only close to expected region boundaries. We have not considered this option in this paper, but it could of course be combined with the improvements suggested here. This leads to yet another possibility, that of changing the noise intensity within the image based on the local image content. Given that the noise is used to avoid the formation of high PDF values in flat areas of the input, one could consider using noise that is stronger in these areas, and less strong close to expected region boundaries, so as to avoid disrupting real edges. The noise strength could thus, for example, be inversely proportional to the gradient magnitude.

Acknowledgments

We thank our colleagues Carolina Wählby and Gunilla Borgefors, at the Centre for Image Analysis (Uppsala, Sweden), for discussion and ideas. Many thanks also to the anonymous reviewers for their very good suggestions and ideas.

References

Angulo, J., Jeulin, D., 2007. Stochastic watershed segmentation, in: Banon, G.J.F., Barrera, J., Braga-Neto, U.d.M., Hirata, N.S.T. (Eds.), Proceedings of the 8th International Symposium on Mathematical Morphology (ISMM 2007), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos. pp. 265–276.

Angulo, J., Velasco-Forero, S., 2010. Semi-supervised hyperspectral image segmentation using regionalized stochastic watershed, in: Proceedings of SPIE symposium on Defense, Security, and Sensing: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI, SPIE, Bellingham. p. 76951F.

priority queue. However, for the purposes of the tests in this paper, we preferred to use floating-point pixel values, and hence we needed an $O(\log q)$ priority queue (Luengo Hendriks, 2010).

Arbeláez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 898–916.

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

Coelho, L.P., Shariff, A., Murphy, R.F., 2009. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, in: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009., IEEE. pp. 518–521.

Cousty, J., Bertrand, G., Najman, L., Couprie, M., 2010. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 925–939.

Cressie, N.A.C., 1993. Statistics for spatial data. John Wiley & Sons, New York. revised edition.

Faessel, M., Jeulin, D., 2010. Segmentation of 3D microtomographic images of granular materials with the stochastic watershed. Journal of Microscopy 239, 17–31.

Immerkær, J., 1996. Fast noise variance estimation. Computer Vision and Image Understanding 64, 300–302.

Luengo Hendriks, C.L., 2010. Revisiting priority queues for image analysis. Pattern Recognition 43, 3003–3012.

Meyer, F., Beucher, S., 1990. Morphological segmentation. Journal of Visual Communication and Image Representation 1, 21–46.

Meyer, F., Stawiaski, J., 2010. A stochastic evaluation of the contour strength, in: Proceedings of the 32nd DAGM conference on Pattern recognition, Springer-Verlag, Berlin Heidelberg. pp. 513–522.

Mouton, P.R., 2002. Principles and practices of unbiased stereology: an introduction for bioscientists. Johns Hopkins University Press, Baltimore.

Noyel, G., Angulo, J., Jeulin, D., 2008. Classification-driven stochastic watershed: application to multispectral segmentation, in: IS&T's Fourth European Conference on Color in Graphics Imaging, and Vision (CGIV 2008), pp. 471–476.

Salembier, P., Serra, J., 1995. Flat zones filtering, connected operators, and filters by reconstruction. Image Processing, IEEE Transactions on 4, 1153–1160.

Selig, B., Luengo Hendriks, C.L., 2012. Stochastic watershed – an analysis, in: Kjellström, H. (Ed.), Proceedings SSBA 2012, KTH Royal Institute of Technology, Stockholm.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 583–598.