Improving the Fast Evaluation of the Robust Stochastic Watershed

Abdul Rahman Khankan
Abstract

Improving the Fast Evaluation of the Robust Stochastic Watershed

Abdul Rahman Khankan

The stochastic watersheds algorithm was first proposed by Angulo and Jeulin (2007) as a marker-controlled watershed-based stochastic segmentation method using Monte Carlo simulation. This project is based on the work of Selig et al. (2015), Fast Evaluation of the Robust Stochastic Watershed, which was the extension of Malmberg and Luengo Hendriks (2014) and Malmberg et al. (2014) who introduced an exact and efficient evaluation method of the stochastic watershed. The algorithm proposed running the exact evaluation method three times after adding noise to the input image then averaging the three edge probabilities together. Their method was identical in terms of average F-measure, but it was an order of magnitude shorter. This project aimed to propose an improved version of Selig et al.’s algorithm which is better in terms of accuracy and faster in terms of processing time. The final result was an algorithm that is matching in accuracy but about 25% faster.
To my family, my loving parents, my sisters, and my brother. You never left my side and I could not have done this without your support.

To my friends, who have walked this path with me and were my cheerleaders.

To my teachers, you always were the inspiration that pushed me to work harder.
# Contents

Part I: Introduction .............................................................................................................. 7
  1  Image Analysis and Digital Image Processing .................................................. 9
  2  Image Segmentation ............................................................................................... 10

Part II: Preliminaries ........................................................................................................ 11
  3  Morphological Watersheds and Watershed Transformation ..................... 13
     3.1  Pixel Intensity ............................................................................................... 13
     3.2  Watersheds ...................................................................................................... 13
     3.3  Seeded watersheds ........................................................................................ 16
  4  Pixel Adjacency Graphs and Edge Weighted Graphs .................................. 17

Part III: Stochastic Watersheds ..................................................................................... 19
  5  H-minima Transform ............................................................................................. 21
  6  Method ..................................................................................................................... 22
  7  Experiment .............................................................................................................. 26
  8  Evaluation and Results .......................................................................................... 28

Part IV: Conclusion and Future Work ........................................................................... 31

Acknowledgement ........................................................................................................... 34

References ..................................................................................................................... 35
Part I:
Introduction
1. Image Analysis and Digital Image Processing

Interest in image analysis and digital image processing started in the early 1920s [10] and today it has grown to be one of the biggest research areas in the field of computer science.

Image analysis is the science of extracting meaningful information from an image using image processing methods and algorithms. The modern applications of digital image processing and image analysis became limitless. Applications such as object detection and recognition, object tracking, pattern recognition such as car’s license plate for example, anomaly detection in medical images, security applications such as fingerprint and iris recognition, weather forecasting, image segmentation, and many more applications. The term “digital image processing” refers to processing digital images by a digital computer [9, 10].

This project tackles one application of digital image processing in particular, image segmentation.
2. Image Segmentation

Image segmentation is dividing an image to a set of regions or objects in order to extract useful information or to make the image easier to analyse. The magnitude of segmentation depends on the problem in hands, thus, segmentation should stop when the desired outcome is reached to avoid “over-segmenting” which is having too many redundant outlier regions or objects [10, 17]. It is useless to carry out segmentation beyond the level of detail needed to identify the wanted elements [10, 19].

Segmentation of complex images is considered amongst the most difficult tasks in image processing [10].

There are several segmentation methods, Edge Detection, Thresholding, Region-Based Segmentation, and the method this project focuses on, Segmentation using Morphological Watersheds -which will be covered in a separate section. Each of method has its advantages and disadvantages.
Part II: Preliminaries

Before starting, some key concepts must be cleared and defined.
3. Morphological Watersheds and Watershed Transformation

3.1 Pixel Intensity

The pixel (picture element) is the smallest element in an image, its value \( f(x, y) \) of each pair of coordinates \( x \) and \( y \) is called the intensity of that certain point [10]. The value (colour) of a pixel can be represented by various components, most famous ones are the components of Red, Green and Blue (RGB), in many image processing applications and in this project, another component is used to represent the image’s pixel values, which is grayscale, it represents the image using the different levels of grey between black and white [12].

3.2 Watersheds

The notion of watershed is to picture the image in 3-dimensions as in figures 3.1 and 3.2, the two pixel coordinates dimensions plus the intensity as the third dimension [10].

![Figure 3.1. 3D image space](image)
As described by Gonzalez [10] there are three types of points:

a) Points that belong to a local minimum 

b) Points at which a drop of water would descent to a particular minimum if dropped at the location of such points 

c) And points at which a drop of water would equally drop to more than one minimum.

The set of points that fulfils condition (b) are called the “watershed” of that minimum and those that fulfil condition (c) are denoted as “watershed lines”.

This segmentation method was first proposed by S. Beucher and C. Lantuejoul [4] in 1979, the main goal of this method is to identify those watershed lines. To explain it in a simpler way, let us say of have certain points which are the local minimums, the surface is flooded by water that starts to rise. When the rising water in two different regions are about to merge a dam is built. The process continues until only dams are visible. Those dams are the watershed boundaries of the image.

Figure 3.2. Image as a topographical landscape\textsuperscript{1}

Figures 3.3 and 3.4 show an example of segmenting an image using watersheds.

Figure 3.3. Example of watershed segmentation, original image

Figure 3.4. Example of watershed segmentation, segmented image

3.3 Seeded watersheds

In practice direct application of watershed segmentation could produce “over-segmentation” due to noise in an image that causes the presence of many local minima. To solve over-segmentation, markers (seeds) are used so that they are the only local minimum, by starting the flooding from those pre-defined set of markers then all other extra local minima are removed [10, 17]. Figure 3.5 shows an example of over-segmentation and how it can be solved using seeds (markers).

![Original image](https://se.mathworks.com/help/releases/R2015b/examples/images_product/WatershedSegmentationExample_14.png)

![Over-segmentation](https://se.mathworks.com/help/releases/R2015b/examples/images_product/WatershedSegmentationExample_14.png)

![Marker-controlled](https://se.mathworks.com/help/releases/R2015b/examples/images_product/WatershedSegmentationExample_14.png)

![Final segmentation](https://se.mathworks.com/help/releases/R2015b/examples/images_product/WatershedSegmentationExample_14.png)

Figure 3.5. Example of marker-controlled segmentation

---


16
4. Pixel Adjacency Graphs and Edge Weighted Graphs

One way to represent a digital image is *pixel adjacency graphs* where each pixel is a vertex in the graph, and edges connect adjacent elements, while the difference of intensities between two adjacent vertices is the weight of their connecting edge [14]. Figure 4.1 shows an example of pixel adjacency graph.

*Figure 4.1. Pixel adjacency graph*

\[ w_{m,n} = |P_n - P_m| \]
Part III:
Stochastic Watersheds

The concept of unsupervised watershed-based stochastic segmentation was introduced by Angulo and Jeulin [1]. It is based on the concept of marker-controlled watershed (seeded watershed). Their method first builds a probability density function (PDF), it is estimated by running calculations multiple times for randomly selected markers, which is also known as the Monte Carlo simulation. Then they assign a probability of significance for each region of the contour to determine the probability of being part of the segmentation boundary. The higher the probability of the contour, the more important it is. Afterwards, they remove insignificant local minima and apply more processing before getting the final segmentation.

Their work was interesting and inspiring for many researchers who continued trying to improve it, Meyer and Stawiaski [16] introduced a way to calculate the PDF exactly without any Monte Carlo simulations, however, their proposed method comes with high computational cost which makes it unusable in practice. Plus, their method did not directly work with images, rather on a tree representation of the image.

Extending their work were Malmberg and Luengo Hendriks [13]. In their work, they presented a quasi-linear algorithm for exact computation of the PDF, their method was faster than all the previous methods by more than two orders of magnitude. That work was later continued by Malmberg et al. [14], they introduced an efficient method for computing the PDF exactly without any seeded watershed computations, and they extended the result to be represented back as an image.
The latest work by Selig et al. [18] -Fast Evaluation of the Robust Stochastic Watershed- extended the previous algorithms and combined them to produce a method that is as accurate but with a considerable reduced computation time. They proposed adding noise to the input image, and then running the exact evaluation method three times, with different noise on each run, then to combine the three edge probabilities together by averaging. Their method was matching in terms of average F-measure, however, it was an order of magnitude shorter. A thorough examination of this algorithm was conducted and this project was made in a try to improve this algorithm further in terms of computation time and accuracy. The study will be explained in chapter 7.

Please refer to the mentioned works for more detailed information.
5. H-minima Transform

One parameter that was used in the algorithm was \( h \), it determines the depth parameter for the H-minima transform. When calculating stochastic watershed, it usually starts by calculating a PDF. The PDF gives lines of different intensity to indicate edges of different level of importance. Lines with low intensity are not relevant, but there are tons of these, all over the image. If regular watershed is applied to the PDF without doing an H-minima transform first, it will produce tons of edges. By selecting the right value for “\( h \)”, it sets a threshold indicating which edges are important enough. Thus, it will cut down unimportant edges [11]. Figure 5.1 shows an example of segmentation after applying H-minima transform (right image) to an oversegmented image (left image).

![Image of oversegmentation and segmentation](https://se.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html)

*Figure 5.1. Example of applying h-minima transform to remove oversegmentation of microscope image of steel grains*

---

6. Method

Starting from the previous work conducted by Selig et al. [18], Fast evaluation of the robust stochastic watershed, and after studying the algorithm that was used, started an experiment to implement an improved design of their work that will be faster to execute but yet, still has the same accurate results.

For the rest of the study, the method that was designed by Selig et al. will be referred to as the Old Stochastic Watershed (OSW); And to the enhanced version as the New Stochastic Watershed (NSW).

To avoid a complex comparison, I will try to simplify the (OSW) algorithm, while the mentioned paper [18] can be referred to for further reading.

The algorithm as shown in figure 6.1 starts with an image to segment and the number of markers (seeds) for the expected regions.

The method then initializes by calculating the image gradient with an additional normalization step to avoid out of range values. Starting from the image gradient, two copies with added noise are created. The next steps are applied to all three images, the gradient and the two copies.

An image adjacency graph is built using 4-connectivity. The weight of each edges of the graph is the average between the intensities of each connected pixels’ pair. From the adjacency graph a minimum spanning tree (MST) is calculated. There exist multiple MSTs if two or more edges in the graph have the same weight. To avoid this ambiguity, edges are sorted and given new weights according to this order. The new weights are all distinct, so only one MST exists.

And finally, PDF is calculated for all the edges in the MST followed by calculating boundary probability of all the pixels, which is stored to the image that was applied to.

After that process, the three images are blurred with a small Gaussian filter. The purpose of the Gaussian is to allow for edges that are slightly shifted in one or two of the three images to still be found. The final step is combining the three images using geometric mean.

The resulting image is the stochastic watershed PDF (probability density function). A segmented image (binary image) is obtained by applying a watershed to the PDF.

After analysing the OSW algorithm, I concluded that the most time demanding processing was mainly in five places
• Adding noise (two times)
• Constructing pixel adjacency graph (three times)
• Calculating the MST (three times)
• Sorting the MST (three times)
• Calculating the PDF (three times)

And thus, an investigation started looking for a way to replace adding noise before creating the graph and calculating the MST and designing a method to calculate one MST and then adding noise to it by rearranging the edges, eliminating the heavy computations needed three times to make it faster.

However, I could not implement an efficient approach to accomplish that nor could I find another efficient way to span the graph.

After that focus switched toward something different, which was altering the way the MST is being created.

The new proposed design -shown in figure 6.2- changed the old method by adding the noise on the edges of the MST as it was being calculated, and by adding a random noise to every single edge, the chances of having two consecutive edges with the same weight were practically nil, so there was no need for sorting as there was no ambiguity anymore.

By eliminating the computations of adding noise two times and calculating three adjacency graphs, and then sorting their respective MSTs, the new algorithm was expected to be faster.

To sum it up, the abstract of the full segmentation algorithm:
– Compute gradient.
– Compute different MSTs from adjacency graph.
– Apply stochastic watershed to MSTs (result = PDF).
– Apply H-minima transform to PDF.
– Apply regular watershed to result.
– Get a binary image containing the edges of objects in the image.
Figure 6.1. OSW
Figure 6.2. NSW
# 7. Experiment

For the experiments, the dataset consisted of 46 microscopic images 1390 x 1030 and their ground truths. The images were in grayscale, with levels varying from 0 to 255.

The parameters that needed to be determined were the optimal number of seeds $n$, the amount of the noise $s$ and finally the threshold $h$ for H-minima transform which a step for the final segmentation.

Parameters ranges:
- $n$ had five different values between 20 and 200
- $s$ had fourteen different values between 0.006 and 0.1
- $h$ had seventeen different values between 0.004 and 0.11

Following the testing method conducted by Selig et al. [18], which is based on the work of Arbelaez et al. [2], F-measure were used to compare the segmentation results with the ground truth.

“F-measure is the harmonic mean of precision and recall and determines the quality of the segmentation results” [18]. The values range between 0 and 1, the higher the F-score, the closest the segmentation to the ground truth. For a pixel to be a mismatch, it has to be more than three pixels away from the corresponding boundary [18].

To obtain the optimal parameter, also inspired by the work of Selig et al. [18], the optimal parameters were trained to the segmentation algorithm. To validate the algorithm, the training was done on a group of images (45 images) and the testing on another images (1 image), according to Leave-one-out cross-validation (LOOCV).

The F-measures for each image and each triple of parameters (4-D matrix: 46x5x14x17) were computed.

Then,
1. Take one image as the test image, and average the F-measures for the other 45 images, this gives a 5x14x17 matrix with average F-measures. It follows by finding the highest F-measure and figuring out which parameters correspond to that specific F-measure. Those parameters are the “optimal parameters” for those specific 45 images.
2. Calculate the F-measure for those parameters for the test image. This is the F-measure the algorithm obtains on the one test image.
Then repeat steps 1 & 2 for each of the 46 images.

In the end, the average F-measure of the algorithm is the average of all 46 test F-measures.

All experiments were run under Linux Ubuntu 14.04 (x64) on an Intel Core i7-4700HQ CPU at 2.40GHz with 12GB of RAM.

All code was implemented in C++, and compiled with gcc v4.8.2. MATLAB 2014b was used to evaluate the old and new implementation. DIPimage library v2.7 for image processing was also used. It is a platform independent library, written in C.

For the benchmarking The Berkeley Segmentation Dataset and Benchmark (BSD500) was used [15].
8. Evaluation and Results

After repeating the experiment for few times and conducting the analysis over the 46 images dataset, results were obtained as the following:

The optimal parameters from the test images are:

– The optimal value for nseeds is 150
– The optimal value for noise is 0.015
– The optimal value for h is 12

![Figure 8.1](image) Improved algorithm’s F-measures for segmentation results using OSW and NSW with optimal parameters

Figure 8.1 compares the F-measure results between OSW and NSW, the average F-measure of the new algorithm (NSW) over all 46 test images is 0.87. While the average F-measure of the old algorithm (OSW) over all 46 test images is 0.85.

1The values in the report are rounded to two decimal places
The 1.93% increase in F-measure is not that significant improvement nor was it the main goal of the project. However, the real payoff in the algorithm was the shrinkage in segmentation time.

Figure 8.2 illustrates the time difference between OSW and NSW. The new method delivered a significant time boost, with a 24.44% faster running time with the new design. This time was determined by computing the average time of all images using the old algorithm, then computing the average time of all images using the new algorithm and then calculating the difference percentage between the two averages.

![Figure 8.2. Time analysis](image-url)
Part IV:
Conclusion and Future Work

This project has followed closely the work done by Selig et al. [18], *Fast evaluation of the robust stochastic watershed*.

Their work was improved by presenting a faster method that provided the same high accuracy but it was about 25% faster. That was done by removing some of the heavy computations that were being used, the algorithm was modified to generate one edge graph instead of three, the edge sorting loops that were used to remove ambiguity were also removed, and replacing that with random noise added to the edge weights while the MST was being created. Doing that the new algorithm guaranteed -in less steps- having different MSTs that will always be unambiguous in terms of edges weights.

The timeframe of this project was only enough to conduct the few experiments mentioned in the study. However, the algorithm still has potential for improvement, and thus, I propose two future experiments to explore. One is to find a way to create two MSTs by rearranging the edges of the original MST, such solution is expected to make things much faster because a big chunk of time is being used to create two MSTs from scratch, and that time is directly proportional to the size of the image and the size of its edge weighted graph. This solution needs more insight on graph theory and a dedicated study. The second experiment is testing different algorithms to calculate the MST, the method used now is Prim’s algorithm, I suggest trying and testing how different algorithms behave taking in consideration the time that it will take
for medium and large images. This experiment could be merged with the first one, they both require more knowledge and research on graph theory.
List of Figures

Figure 3.1: 3D image space .................................................................................. 13
Figure 3.2: Image as a topographical landscape .................................................. 14
Figure 3.3: Example of watershed segmentation, original image ...................... 15
Figure 3.4: Example of watershed segmentation, segmented image ................. 15
Figure 3.5: Example of marker-controlled segmentation ................................. 16
Figure 4.1: Pixel adjacency graph ....................................................................... 17
Figure 5.1: Example of applying h-minima transform to remove over segmentation of microscope image of steel grains ................................................. 21
Figure 6.1: OSW .................................................................................................. 24
Figure 6.2: NSW .................................................................................................. 25
Figure 8.1: Improved algorithm’s F-measures for segmentation results using OSW and NSW with optimal parameters ...................................................... 28
Figure 8.2: Time analysis .................................................................................... 29
Acknowledgement

I wish to express my sincere gratitude to Cris L. Luengo Hendriks, my supervisor, who believed in me, guided me throughout this project and shared his knowledge and feedback.

I sincerely thank Filip Malmberg, my reviewer, who shared his valuable experience and feedback with me to make this report.

I also thank Olle Gällmo, my examiner, for his valuable instructions and feedback.
References


