



Technical University of Crete
Department of Electronic
and Computer Engineering

Diploma Thesis:

“Biomedical Image Segmentation using the Watershed
Transform to initialize Region Growing”

Maria Zafeiri

Examination Comitee:

Zervakis Michael, Prof. (supervisor)

Balas Costas, Ass.Prof.

Petrakis Euripides, Ass.Prof.

Chania, 2008

ABSTRACT

In the present work the goal was to create a fully automatic, but effective method to segment gray scale biomedical images. The literature offered several suggestions, many of which were considered and further investigated. Two commonly used algorithms are the Watershed algorithm and the Seeded Region Growing algorithm, as well as their variations. Although they are based on promising ideas, they both lack in certain areas. The Watershed algorithm is automatic, but it is too sensitive in gray value fluctuations, causing undesired oversegmentation of the image. The ISRG performs quality segmentation, but it requires a manually selected seed set to determine the regions of interest.

We propose in this work a method that combines the two algorithms, performing fully automated, reliable image segmentation. The Watershed algorithm is first used to define roughly the regions of interest, and support the seed selection. Then the ISRG algorithm is inputted these seeds and performs the actual segmentation of the image. A final post processing step to refine the segmentation is required.

The proposed method was applied in gray scale biomedical images and produced satisfactory results. It outperformed both algorithms and its output is comparable to manual segmentation.

Table of Contents

ABSTRACT.....	1
Chapter 1 - INTRODUCTION.....	3
1.1. Image Segmentation.....	3
1.2. Image Segmentation in biomedical applications.....	6
Chapter 2 - COMMON APPROACHES.....	8
2.1. The Watershed Algorithm.....	12
2.1.1. Method description.....	13
2.1.2. Overview.....	18
2.2. The Seeded Region Growing Algorithm.....	18
2.2.1. Method description.....	18
2.2.2 Overview.....	20
2.2.3. The Improved Region Growing Algorithm.....	21
2.3. Necessity- Motivation.....	22
Chapter 3 - PROPOSED METHOD.....	24
3.1. Description.....	24
3.1.1. Overview.....	29
3.2. Validation.....	30
Chapter 4 - RESULTS.....	33
4.1 Comparison with other methods.....	33
4.2 Results for Biomedical Images – Proposed Method.....	37
Chapter 5 - CONCLUSIONS & FUTURE WORK.....	54
5.1. Conclusions.....	54
5.2. Future Work.....	55
REFERENCES.....	57

Chapter 1 - INTRODUCTION

The term digital image processing refers to tasks, such as analysis, interpretation and manipulation, which are performed using computer algorithms, and for which the input is a digital image. The output of the processing is either an image or a set of characteristics or parameters related to the image, and it consists the primary or one of the auxiliary functions of various applications, from photo manipulation and computer graphics to feature extraction and remote sensing.

In image processing a common goal is to partition a given image into regions with respect to various criteria. This procedure is known as image segmentation. The desired outcome of image segmentation is to create a content based separation of the image, such that every region has a meaningful disassociation from its surroundings, as regards the purpose of each application.

The task considered in the present work is image segmentation, and more specifically an automatic, but effective method to segment gray scale biomedical images.

1.1. Image Segmentation

Although it may seem as an easy and straightforward procedure for human

perception to identify objects, recognize shape, spot mutation, track movement and decide contours, when it comes to machine vision these are complicated tasks that demand a lot of effort in order to be achieved to the minimum. Computer vision scientists are faced with the challenge to interpret the human vision function and reproduce it using algorithms. An important step towards this endeavor is to recognize homogeneous regions within an image as distinct, and successively as belonging to different objects. Image segmentation deals with these issues in a primary level.

To address the problem in a more practical manner let us consider a typical gray scale image that portrays a face in front of a homogeneous background, the face covering about half of the image. It would certainly be easy for a human to distinguish the face from the background. Then, given a medium quality of the image, we could easily detect the hair, the mouth and the eyes, and given a better quality almost every detail such as small wrinkles and eyelashes. When it comes to computer algorithms, especially the fully automated ones, the case is quite different. In terms of algorithmic input, all these meaningful features are just groups of different gray level pixels spread around the image arbitrarily.

The first step to give an image, a vague set of pixels, some meaning, when a computerized process is involved, is image segmentation. Image segmentation is the procedure of deciding which pixels are to be grouped together and which ought to be separated. The way in which this is done may vary, depending on either the approach followed or the application involving the segmentation step, or both. Nevertheless, regardless the approach or the application, when image segmentation is performed in an image the goal is to create partitions of it and to try to make each partition represent something that, in a non strict sense, consists an entity clearly separated from the other entities in the image.

Before we continue with a rough explanation of how image segmentation algorithms work in general, it would be helpful to note that we refer to non intelligent systems; this applies for the algorithms using artificial neural networks as well, since they too require a training stage prior to performing the final segmentation. So the only useful information that the algorithms can actually exploit, with the exception of the ones that use an extra manually constructed

input, are the gray value of each pixel along with its position in the image, or, to put it differently, its distance from the other pixels.

All the above having been mentioned, the main principle that the image segmentation algorithms follow is somewhat easy to imagine. As solid entities, in the sense that they are described above, do not usually demonstrate intense fluctuations of their gray value, the main principle image segmentation algorithms are based on is to group neighboring pixels together as long as they have the same or similar gray values. They perform this grouping action, each based on its working principle and specific parameters that may relate to the image or be set by a user, until every pixel within an image has been grouped with neighboring ones. The result is the whole image being partitioned into regions and contours being set to separate them.

Although the main principle behind image segmentation algorithms has become clear, one last point should be made. What defines the separation of a given image – the image segmentation- is the actual application it is intended for. There are two important implications to this: i) one single image segmentation algorithm cannot be used in all cases that image segmentation is required, at least not without alterations or variations, despite its good performance, and ii) human intervention cannot be totally omitted, even in the case of fully automated image segmentation algorithms, as the choice of the appropriate tool, according to the task that must be performed, is very important.

Disjoining every small particle in an image could be the desired outcome for a certain application, while it could be a thorn for another. Thus, the first step when performing image segmentation is to define what it will be used for, or in other words, how much detail is useful. However we must keep in mind that all these issues need to be predefined only roughly, as it is common for image segmentation algorithms to include a post processing step that resolves various problems, handling specific occurrences as exceptions subject to particular conditions.

1.2. Image Segmentation in biomedical applications

Image processing has greatly advanced over the last years partly due to its wide use in biomedical applications. Imaging is a powerful tool for both diagnostic and researching purposes in a variety of medically relevant procedures. The strong relation between biomedical applications and image processing is manifested by the fact that there exist several digital image protocols dedicated to biomedical use exclusively. To offer a simple and everyday example of how closely biology and medicine are connected to the imaging science let us remind the irreplaceable contribution of CT scans, MRI s, digital microscopes and ultrasounds to modern medicine.

Medical imaging is often perceived as the means to non invasively produce images of the internal aspect of the body, that is as a purely visualization instrumentality. However, there is a lot of work done recently to engage the advancements of image processing in biomedical applications. Image segmentation has already made its breakthrough in the medical science, providing numerous benefits to the doctors, patients and researchers.

As pointed out by Vannier and Haller in [1], biomedical image segmentation is the parcellation of scenes into its component regions, a prerequisite for labeling of organs, organelles, and anatomic substructures found in images. Segmentation is the signal and image processing equivalent of anatomic or surgical dissection that results in separate components. The purpose of biomedical image segmentation is to define subregions that correspond to biological entities, typically by delineation of substructure boundaries, and assignment of class membership to each of the original image pixels or voxels. Class assignment may be dichotomous or probabilistic. The number of classes present in a scene is typically orders of magnitude less than the number of image elements.

The most significant biomedical applications of image segmentation are morphometry and change detection. The segmented regions of medical images are used for comparison within or between individuals. Reference data that facilitates interpretation of geometric and mass property measurements from

segmented image regions are assembled into anatomic atlases, including recently developed electronic versions. Many disciplines, including anthropometry, physical anthropology, neuroscience, oncology, orthotics and prosthetics use these data compilations in atlases.

Furthermore, image guided treatment planning and delivery in surgery and radiotherapy use segmentation to define the target lesion and critical normal structures. Surgical navigation uses segmented results to guide instruments to a target while minimizing the trauma to normal structures. Radiotherapy plans treatments and delivers therapy with the goal of maximizing the tumour and its penumbra dose while avoiding complications due to exposure of adjacent structures, some of which may be especially radiosensitive, by a variety of strategies. Recent advancements in radiotherapy including inverse treatment, stereotactic methods, and 3D conformal dynamic therapy require image segmentation, usually of CT or MRI scans or both. In fact, the segmentation step is usually considered the most time consuming and costly step in the entire process of planning and administering modern therapies.

The importance of image segmentation in biomedical applications as well as the the need for application related segmentation provide us with a strong motivation to examine the performance of established segmentation methods, to attempt an upgrade by alligation and post processing and try to improve their effectiveness for biomedical image segmentation by parameter manipulation.

Chapter 2 - COMMON APPROACHES

Various techniques have been employed for image segmentation, most of which follow one or a combination of approaches based on : i) histogram thresholding, ii) edge detection, iii) tree/graph analysis, iv) clustering, v) probabilistic/Bayesian models, vi) neural networks and vii) region growing [2].

Histogram thresholding is based on constructing color and hue histograms [2]. Ohlander [3] proposed a thresholding technique that is very useful on segmenting outdoor color images. The picture is thresholded at its most clearly separated peak. The process iterates for each segmented part of the image until no separate peaks are found in any of the histograms. The criteria to separate peaks was based on the ratio of peak maximum to peak minimum to be greater than or equal to two. Textured areas were separated from uniform regions by using a Sobel operator marking regions that contain large edge activity.

Another thresholding approach has been implemented by Cheriet et al. [4] in the area of document images, specifically for segmenting bank cheques. This approach segments the brightest homogeneous object from a given image at each recursion, leaving the darkest homogeneous object. This method is developed without any constraints on the number of objects in the digital image. The method is based on discriminant analysis. The thresholding operation is regarded as the partitioning of pixels of an image into two classes: object and background. For each iteration, the histogram of the image is drawn and the largest peak is separated from the rest of the image. The process is continued till there are no more peaks left in the histogram.

In a number of applications, histogram thresholding is not possible simply because the histogram may be unimodal [2]. In some cases the images may be of such quality that any preprocessing may not improve the contrast between objects sufficiently and hence one may not achieve two or more peaks in the histogram for selecting thresholds for segmentation. Unimodal distributions are typically obtained when the image consists of mostly of a large background area with small, but significant regions. This often happens in medical imaging applications, a fact that makes histogram thresholding unsuitable for biomedical image segmentation, at least not without resolving these issues first.

Edge detection methods attempt to perform image segmentation through boundary analysis. Prager [5] proposed a set of algorithms the goal of which is to locate the boundaries of an object correctly in a scene. First, pre-processing of the images is done to clean up the raw data by smoothing and noise-removal. Second, the edge representation is generated. Differentiation is done to find the edge-strength at each point in the image. Suppression is then done to remove multiple edges formed by spatial differentiation of boundaries. Third, the edges are joined into line segments and features are computed. The features include: length, contrast, frequency, mean, variance and location of each line segment. Fourth, post-processing is done to remove unwanted line segments and to build confidence for each of the remaining segments. The output of the system is a set of line segments with a list of attributes, such as length and confidence.

It is acknowledged [2] that edge based segmentation has not been very successful because of small gaps that allow merging of dissimilar regions. In order to avoid these problems Perkins [6] proposes an expansion-contraction technique in which edge regions are expanded to close gaps and then contracted after the separate regions have been labeled. The size of expansion is controlled such that small regions are not engulfed by this process. The process involves the use of Sobel filter for producing edge strengths and directions at every point. The edges are thinned and the result is automatically thresholded leaving only ridges. The ridges separate regions of different intensity but there may be small gaps. Segmentation is performed by expanding active edge regions, labeling the segmented uniform intensity regions, and then contracting edge regions.

Tree/graph analysis was proposed by Cho and Meer [7] as a new approach for segmentation, which is derived from the consensus of a set of different segmentation outputs on one input image. Instead of statistics characterizing the spatial structure of the local neighborhood of a pixel, for every pair of adjacent pixels their collected statistics are used for determining local homogeneity. Several initial segmentations are derived from the same input image by changing the probabilistic component of the hierarchical Region Adjacency Graph (RAG) pyramid based technique. From the ensemble of these initial segmentations, for every adjacent pixel pair a co-occurrence probability is derived, which captures global information (about the image) at the local level (pixel level). The final segmentation of the input image is obtained by processing the co-occurrence probability field with the same RAG pyramid technique. The pixel pairs with high co-occurrence probability are then grouped together based on the consensus about local homogeneity. This technique can also be used to extract the high confidence homogeneous regions from the co-occurrence probability field. Bayesian networks were then used to extract features from images. The features extracted were variance of the width of the region, ratio of average width to length and the average grey level. Then post-processing of over-segmented images is done based upon a priori information about the sought features. The RAG of the final segmentation provides the spatial relationship between regions and can be used for further interactive analysis of the image. This segmentation method is completely unsupervised.

Clustering of pixels has been used to perform image segmentation effectively [2]. Cluster analysis allows the partitioning of data into meaningful subgroups and it can be applied for image segmentation or classification purposes. Clustering analysis either requires the user to provide the seeds for the regions to be segmented or uses non-parametric methods for finding the salient regions without the need for seed points. Clustering is commonly used in a range of applications such as image segmentation and unsupervised learning [8]. A number of issues related to clustering are worth studying including how many clusters are the best and how to determine the validity of clusters. In a number of segmentation techniques, such as fuzzy c-means clustering, the number of

clusters present in the image have to be specified in advance. Several techniques that do not require such initialization have been proposed in literature .

The validity of clusters is also important to study. Yarman-Vural and Ataman [9] critique several areas of clustering methodology including the definition of clusters, determination of the number of clusters, heuristic partitional clustering algorithms and the effect of noise on determining accurate clusters. Cluster validity criteria including maximum likelihood information criteria and sum of squared errors is discussed. The first criteria is found to be better when the number of clusters changes.

Probabilistic algorithms use co-occurrence based approaches for image segmentation making use of region and boundary information in parallel for improved performance on a sequence of images. Haddon and Boyce [10] examined image segmentation by unifying region and boundary information using co-occurrence matrices. The co-occurrence matrices were used to generate the feature space. The analysis was performed in the context of an ensemble of images. Based on the location of the intensities of each pixel and its neighbours in the co-occurrence matrix, initial segmentation is done. Each pixel is then associated with a tuple which specifies whether it belongs to a given region or if it is a boundary pixel. This tentative segmentation was then refined by relaxation labelling that ensures local consistency of pixel labelling during segmentation by minimising the entropy of local neighbourhoods. If a pixel does not belong to the boundary, then it is assigned to one of the regions. This classification is entirely uni-dimensional in the co-occurrence direction and contains no explicit local consistency. The consistency for regions and boundary was obtained assuming that boundaries are not wider than one pixel.

Neural networks have also been proposed as an automatic segmentation and classification method. Campbell et al. [11] and Papamarkos et al.[12] have developed the procedure of image segmentation using Self-Organising Feature Maps (SOFM). The use of these neural network paradigms is considered equivalent to multithresholding where the output of the network defines a number of homogeneous clusters [2].

Region growing algorithms take one or more pixels, called seeds, and grow

the regions around them based upon a certain homogeneity criteria. If the adjoining pixels are similar to the seed, they are merged with them within a single region. The process continues until all the pixels in the image are assigned to one or more regions. For region growing, seeds can be automatically or manually selected. Their automated selection can be based on finding pixels that are of interest, e.g. the brightest pixel in an infra-red image can serve as a seed pixel. They can also be determined from the peaks found in an image histogram. On the other hand, seeds can also be selected manually for every object present in the image.

Two of the most representative algorithms of this approach are the Seeded Region Growing (SRG) [2] and the Watershed Transformation based algorithms [13], along with their variations. As these algorithms were utilized for the present work, they will be presented in detail next.

2.1. The Watershed Algorithm

In gray scale mathematical morphology the watershed transform, originally proposed by Digabel and Lantuejoul [14] and later improved by Beucher and Lantuejoul [15], is the method of choice for image segmentation [16]. The intuitive idea underlying this method comes from geography: it is that of a landscape or topographic relief which is flooded by water, watersheds being the divide lines of the domains of attraction of rain falling over the region [16]. An alternative approach is to imagine the landscape being immersed in a lake, with holes pierced in local minima. Basins (also called 'catchment basins') will fill up with water starting at these local minima, and, at points where water coming from different basins would meet, dams are built. When the water level has reached the highest peak in the landscape, the process is stopped. As a result, the landscape is partitioned into regions or basins separated by dams, called watershed lines or simply watersheds [13].

One of the difficulties with this intuitive concept is that it leaves room for various formalizations. Many sequential algorithms have been developed to compute watershed transforms. Here, for reasons of simplicity and consistency,

we follow the presentation in [17], which is based on distance functions.

2.1.1. Method description

A formal definition of a catchment basin is given by Meyer [17] and [18] in the following way: Let $f(p)$ be a function of gray values representing a digital image with the domain $\Omega \subset \mathbb{Z}$. Each pixel $p \in \Omega$ has a gray value $f(p)$ and a set of neighboring pixels $p' \in N(p)$ with a distance function $dist(p, p')$ to each neighbor. In most cases a 4- or 8- connectivity neighborhood is used with a constant distance of 1 to all 4 or 8 neighboring pixels. From functions on continuous space, Meyer derived a formal definition of catchment basins for the digital space through the definitions below :

Definition 1 (Cost function based on Lower slope): The cost for walking on the topographical surface from position p_{i-1} to $p_i \in N(p_{i-1})$ is:

$$cost(p_{i-1}, p_i) = \begin{cases} LS(p_{i-1}) \cdot dist(p_{i-1}, p_i) & : f(p_{i-1}) > f(p_i) \\ LS(p_i) \cdot dist(p_{i-1}, p_i) & : f(p_{i-1}) < f(p_i) \\ \frac{1}{2} (LS(p_{i-1}) + LS(p_i)) \cdot dist(p_{i-1}, p_i) & : f(p_{i-1}) = f(p_i) \end{cases}$$

The lower slope $LS(p) = \max_{\forall p' \in N(p)} \left\{ \frac{f(p) - f(p')}{dist(p, p')} \mid f(p') < f(p) \right\}$ and

is not defined if no such p' exists.

Definition 2 (Topographical Distance): The topographical distance between the pixels p and q of the image is the minimal π - topographical distance among all paths between p and q inside Ω :

$$TD_f(p, q) = \inf_{\forall \pi \in \Omega} TD_f^\pi(p, q)$$

where $TD_f^\pi(p, q) = \sum_{i=2}^n cost(p_{i-1}, p_i)$ is the π - topographical distance of a path $\pi = p_1, p_2, \dots, p_n$ with $p_i \in \Omega$, $p_1 = p$ and $p_n = q$.

Definition 3 (Catchment Basin based on topographical distance): A catchment basin $CB_{TD}(m_i)$ of a regional minimum m_i is the set of pixels $p_i \in \Omega$ where

the topographical distance is closer to m_i than to any other regional minimum m_j into account :

$$CB_{TD}(m_i) = \left\{ p \mid f(m_i) + TD_f(p, m_i) < f(m_j) + TD_f(p, m_j) \quad \forall j \neq i \right\}$$

With these definitions Meyer proposed the following theorem:

Theorem 1: The topographical distance between a pixel p and the regional minimum m_i in the depth of its catchment basin is minimal and equal to $f(p) - f(m_i)$ and the geodesic line between them is a line of steepest descent.

The reversal of *Theorem 1* states that a path of steepest descent causes minimal costs. The construction of the catchment basins is reduced to a problem of finding one of the shortest paths between each pixel and a local minimum.

Segmentation based on local condition: A local condition for a correct watershed segmentation is presented and the relation to the above formalism based on the cost function is shown. The formalism is developed for images without plateaus. The extension to include images with plateaus is presented later. We define the watershed segmentation and catchment basins based on the sets of those neighboring pixels that can be part of the path of steepest descent.

Definition 4 (Neighbors on a path of steepest descent): $NLS(p)$ is the set of pixels $p' \in \mathcal{N}(p), p, p' \in \Omega$ such that:

$$NLS(p) = \left\{ p' \mid \frac{f(p) - f(p')}{\text{dist}(p, p')} = LS(p), f(p') < f(p) \right\}$$

For the special case of neighborhoods with $\text{dist}(p', p) = 1$ for all $p' \in \mathcal{N}(p)$ we can simplify the set to:

$$NLS(p) = \left\{ p' \mid f(p') = \min_{\forall p'' \in \mathcal{N}(p)} f(p''), f(p') < f(p) \right\}$$

The path of steepest descent from a pixel p down to the local minimum m_i will pass only pixels of the set $\cup_{p \in \Omega} NLS(p)$ Sp NLS_p This leads to a definition of a watershed segmentation and catchment basins based on $NLS(p)$:

Definition 5 (Watershed segmentation for images without plateaus): For any

image without plateaus a segmentation is called watershed segmentation if every local minimum m_i has a unique label $L(m_i)$ and for every pixel $p, p' \in \Omega$ and $NLS(p) \neq \emptyset$ the following condition holds:

$$\exists p' \in NLS(p) \text{ with } L(p) = L(p')$$

Definition 6 (Catchment Basin): For a watershed segmentation as defined above, a catchment basin $CB_{LC}(m_i)$ of the local minimum m_i is the set of pixels with the label $L(m_i)$:

$$CB_{LC}(m_i) = \{p \mid L(p) = L(m_i)\}$$

$CB_{LC}(p \rightarrow m_i)$ denotes the catchment basin of m_i containing pixel p . It is clear that an image may have many valid watershed segmentations The following theorem shows the relationship between the denition of a catchment basin based on the topographic distance and the local condition:

Theorem 2: A catcment basin based on the topographical distance, as in *Definition 3*, is a subset of the catchment basin based on the local condition of

Definition 5.

The formal construction of the catchment basin, according to *Definition 5*, can be described with a recursion. The recursion starts with the set of pixels belonging to the local minima m_i . All these pixels are labeled with $L(m_i)$. In each step pixels are added to the previous set. The recursion ends if no pixels can be added.

$$CB_{LC}^0(m_i) = m_i$$

$$CB_{LC}^{k+1}(m_i) = CB_{LC}^k(m_i) \cup \Delta CB_{LC}^k(m_i)$$

$$\Delta CB_{LC}^k(m_i) = \left\{ p \mid \exists p' \in NLS(p) \text{ and } p \notin CB_{LC}^k(m_j) \forall j \text{ and } p' \in CB_{LC}^k(m_{(i)}) \right\}$$

Each added pixel p has a neighboring pixel p' being part of the catchment basin $CB_{LC}^k(m_i)$. Thus the local condition of *Definition 5* is valid for each p . It is concluded that:

$$\begin{aligned}
p' &\in \text{NLS}(p) \\
\text{LS}(p) &= \frac{f(p) - f(p')}{\text{dist}(p, p')} \\
\text{LS}(p) \cdot \text{dist}(p, p') &= f(p) - f(p') \\
\text{cost}(p, p') &= f(p) - f(p')
\end{aligned}$$

According to *Theorem 1*, one recursion step adds only those pixels p building paths of steepest descent down to $\text{CB}_{\text{LC}}^k(m_i)$ with the minimal costs $f(p) - f'(p)$, $p \in \text{CB}_{\text{LC}}^k(m_i)$. After the recursion is finished, all paths between the pixels of the catchment basin and its minimum are paths of steepest descent. Therefore it is not possible to construct a steeper path to a different minimum m_j . However, there might exist another steepest path to a different local minimum m_j . In this case the pixel is a watershed pixel according to *Definition 3*. This proves that $\text{CB}_{\text{TD}}(m_i)$ is a subset of $\text{CB}_{\text{LC}}(m_i)$.

The difference between *Definition 3* and *Definition 6* is the treatment of the pixels that have steepest paths to more than one minimum. According to *Definition 3* these pixels are watershed pixels. Following *Definition 6* based on the local condition, such a pixel is assigned to one of the minima m_i , which is connected by a steepest path and where the condition $p' \in \text{NLS}(p)$, $L(p') = L(m_i)$ holds.

Treatment of plateaus: The topographic distance has the same value for all pixels on a plateau. Therefore we have to use the geodesic distance in addition to ensure that a pixel on a plateau gets labeled from the nearest border pixel with a lower neighbor. The geodesic distance between two pixels p and p' on a plateau is equal to the length of the shortest path between p and p' within the considered area. We will show how to extend the set of lower neighbors on a path of steepest descent NLS (*Definition 4*) to include images with plateaus.

A plateau PL is a connected set of pixels that have the same altitude. For minima plateaus, which are distended over more than one subdomain, no

reflooding is necessary. They are linked by selecting a globally unique identifier for each minimum plateau.

Let $\partial_{PL} = \{p' \mid \text{NLS}(p') \neq \emptyset, p' \in PL\}$ denote the set of pixels on the border of plateau PL which have a lower neighbor, and let further $\text{dist}_{PL}(p', p)$ be the geodesic distance between p and p' within the plateau. The minimal geodesic distance between a pixel p on the plateau PL and all border pixels $p' \in \partial_{PL}$ is $\text{dist}_{min}(p, \partial_{PL}) = \min_{p' \in \partial_{PL}} \text{dist}_{PL}(p, p')$. Extending *Definition 4* we can define a watershed segmentation for images with plateaus:

Definition 7 (Extended neighbors on a path of steepest descent): The set $\text{NLS}'(p)$ contains the pixels of the set $\text{NLS}(p')$ of all border pixels $p' \in \partial_{PL}$ that have the minimal geodesic distance to p .

$$\text{NLS}'(p) = \left\{ \bigcup_{p' \in \partial_{PL}} \text{NLS}'(p) : \text{dist}_{PL}(p, p') = \text{dist}_{min}(p, \partial_{PL}) \right\}$$

Definition 8 (Watershed segmentation for images with plateaus): For any image with plateaus a segmentation is called watershed segmentation if every local minimum m_i has an unique label $L(m_i)$ and for every pixel $p \in \Omega$ and $\text{NLS}(p') \neq \emptyset$ the following condition holds:

$$\exists p' \in \text{NLS}(p) \text{ with } L(p) = L(p')$$

The watershed algorithm is implemented following *Definitions 3, 6 and 8* and *Theorem 2*, in a recursive manner. The first step to this implementation is to define all local minima. That is, for every pixel in the image to examine its neighbors' gray values, and if all have greater (or equal in the case of a plateau) gray value, to label it as a local minimum. After all local minima have been detected and labeled, the recursive construction of each catchment basin follows. In each recursion of the algorithm the neighboring pixels that build a path of steepest descent down to the catchment basin with the minimum cost are accumulated (appended the respective label). At the beginning of each iteration the pixels of the previous step are added to the set of pixels with the same label (the ones that belong to the same catchment basin). The recursion

stops when no more pixels can be added.

2.1.2. Overview

The watershed transform, being a completely unsupervised method, exhibits extreme sensitivity to gray value fluctuations. A catchment basin is created every time a local minimum is detected. Often these minima do not represent any meaningful transition of the image content and are mainly due to noise. This causes the watershed method in its initial form to produce severe oversegmentation of the image, disqualifying it from being used autonomously.

2.2. The Seeded Region Growing Algorithm

Adams and Bischof [19] studied the effectiveness of the seeded region growing approach for image segmentation of greyscale images, for which the seeds are manually selected. The method is employed to segment an image into different regions using a set of seeds. Each seeded region is a connected component comprising of one or more points and is represented by a set A_i . The set of immediate neighbours bordering the pixel is calculated. The neighbours are then examined and if they intersect any region from set A_i , then a measure δ (difference between a pixel and the intersected region) is computed. If the neighbours intersect more than one region, then the set is taken as that region for which difference measure δ is maximum. The new state of regions for the set then constitutes input to the next iteration. This process continues until all of the image pixels have been assimilated into regions. Hence, for each iteration the pixel that is most similar to a region that it borders is appended to that region.

2.2.1. Method description

Seeded region growing, as presented by Adams and Bischof [19] performs a segmentation of an image with respect to a set of points, known as seeds, starting with a number of seeds which have been grouped into n sets, A_1, A_2, \dots, A_n . Sometimes, individual sets may consist of single points. It is in the choice of seeds that the decision of what is a feature of interest and what

is irrelevant or noise is embedded. Given the seeds, SRG then finds a tessellation of the image into regions with the property that each connected component of a region meets (nonempty intersection with) exactly one of the A_i , and, subject to this constraint, the regions are chosen to be as homogeneous as possible. We present here a description of the method as applied to gray scale images. The method can be implemented on any shape grid (or graph) in any number of dimensions.

The process evolves inductively from the seeds, namely, the initial state of the sets A_1, A_2, \dots, A_n . Each step of the algorithm involves the addition of one pixel to one of the above sets. We now consider the state of the sets A_i , after m steps. Let T be the set of all as-yet unallocated pixels which border at least one of the regions

$$T = \left\{ x \notin \bigcup_{i=1}^n A_i \mid N(x) \cap \bigcup_{i=1}^n A_i \neq \emptyset \right\}$$

where $N(x)$ is the set of immediate neighbors of the pixel x . In this work we will use a rectangular grid with immediate neighbors being those which are 8 connected to the pixel x . If, for $x \in T$ we have that $N(x)$ meets just one of the A_i then we define $i(x) \in \{1, 2, \dots, n\}$ to be that index such that

$N(x) \cap A_{i(x)} \neq \emptyset$ and define $\delta(x)$ to be a measure of how different x

is from the region it adjoins. The simplest definition for $\delta(x)$ is

$\delta(x) = |g(x) - \text{mean}[g(y)]|$, $y \in A_{i(x)}$, where $g(x)$ is the gray value

of the image point x . If $N(x)$ meets two or more of the A_i , we take

$i(x)$ to be a value of i such that $N(x)$ meets A_i and $\delta(x)$ is

minimized. Alternatively, in this circumstance, we may wish to classify x as a

boundary pixel and append it to the set B of already-found boundary pixels.

Flagging such boundary pixels is useful for display purposes or for use with a semiinteractive corrective procedure. We then take a $z \in T$ such that

$\delta(z) = \min \{ \delta(x) \}, x \in T$ and append z to $A_{i(z)}$. This completes step

$m+1$. The process is repeated until all pixels have been allocated. The

process commences with each A , being just one of the seed sets.

In programming SRG, we make use of a data structure which is called a sequentially sorted list (SSL). This is nothing new, although it has not often been used in image processing applications. A SSL is just a linked list of objects, in this case pixel addresses, which are ordered according to some attribute. When considering a new pixel, for example, at the beginning of each step of SRG, we take that one at the beginning of the list. When adding a pixel to the list, we must place it according to its value of the ordering attribute. In our case, the SSL stores the data of A which is ordered according to δ .

The algorithm for implementing SRG (boundary flagging case) is as follows:

```
Label seed points according their initial grouping.
Put neighbors of seed points (the initial  $T$ ) in the SSL.
While the SSL is not empty:
    Remove first point  $y$  from SSL.
    Test the neighbors of this point:
    If all neighbors of  $y$  which are already labeled (other than with the
    boundary label) have the same label:
        Set  $y$  to this label.
        Update running mean of corresponding region.
        Add neighbors of  $y$  which are neither already set nor
        already in the SSL to the SSL according to their value of  $\delta$ .
    Otherwise:
        Flag  $y$  with the boundary label.
```

Note that previous entries in the SSL are not updated to reflect their differences from the new region mean. This leads to negligible difference in the results, but greatly enhanced speed. This stepwise description shows that, in executing the algorithm, each pixel is visited just once, although at each visit we also view each of the neighbors. Hence, it makes for a very rapid program.

2.2.2 Overview

The SRG algorithm has the advantage of being fairly robust, quick, and parameter free. However there are issues to be resolved. As the starting point is a set of manually selected seeds, the algorithm cannot be categorized as automatic. The areas of interest need to be specified, a fact that limits the algorithm's usability in a number of applications. Moreover, as pointed out by

Mehnert and Jackway [20] the SRG has both inherent pixel order dependencies and implementation order dependencies. The first manifests itself whenever, during an iteration, several $x \in T$ determine the same, minimum, δ value. Several possible choices for z are then offered. The particular z chosen influences the running mean of the region that it is assigned to. This in turn influences the δ values calculated for the $x \in T$ in the next iteration, and ultimately affects the final segmentation. The second order dependency manifests itself whenever the chosen z has the same δ value for several regions that it borders. Once again resolution of the deadlock ultimately influences the final segmentation. In implementing the SRG algorithm, Adams and Bischof utilise the sequentially sorted list (SSL). In their implementation the SSL is a linked list of pixel addresses, ordered with respect to δ . A pixel can be arbitrarily inserted into the list in the position prescribed by its δ value. However, only the pixel with the smallest δ value can be removed from the SSL. Effectively, the SSL stores the points of the set T ordered according to δ . Also, the implementation does not update previous entries in the SSL to reflect new differences from a region whose mean has been updated. As a consequence, in addition to the pixel order dependencies induced by the SRG algorithm, two other implementation pixel order dependencies exist. The first order dependency manifests itself during the initial process of adding the neighbours of the seed regions to the SSL. In particular, if a pixel borders two or more seed regions it is given a δ value based on its similarity to that seed region which happens to be first in terms of the order of processing of the image pixels. Once inserted into the SSL the pixel position is never updated. The second order dependency manifests itself whenever the neighbours of a newly labelled pixel are added to the SSL. The order in which the neighbours are scanned can affect the δ value assigned to each and hence their ordering within the SSL.

2.2.3. The Improved Region Growing Algorithm

Mehnert and Jackway [20] improved the Seeded Region Growing algorithm by making it independent of the pixel order of processing and making it more parallel. Their study presents a novel technique for Improved Seeded Region

Growing (ISRG). ISRG algorithm retains the advantages of SRG such as fast execution, robust segmentation and no parameters to tune. The algorithm is also pixel order independent. If more than one pixel in the neighbourhood have same minimum similarity measure value, then all of them are processed in parallel. No pixel can be labelled and no region can be updated until all other pixels with the same priority have been examined. If a pixel cannot be labelled, because it is equally likely to belong to two or more adjacent regions, then it is labelled as 'tied' and takes no part in the region growing process. After all of the pixels in the image have been labelled, the pixels labelled 'tied' are independently re-examined to see whether or not the ties can be resolved. To resolve the ties an additional assignment criterion is imposed, such as assigning a tied pixel to the largest neighbouring region or to the neighbouring region with the largest mean. This is a post-processing step. The algorithm in this study was tested on the image of man made objects such as a car, an aeroplane, and buildings. The authors concluded that the ISRG algorithm produces consistent segmentation because it is not dependent on the order of pixel processing. Parallel processing ensures that the pixels with the same priority are processed in the same manner simultaneously.

2.3. Necessity- Motivation

The watershed algorithm, though fully automatic, exhibits absolutely no robustness since it is designed to detect all local gray level fluctuations. Image oversegmentation is a standard problem when the watershed transformation is implemented independently in its original form. While accuracy can be guaranteed, its actual practicability in existing applications is far from being achieved. The oversegmented output image can provide some information about the critical points or areas of an image, but the decision criteria should be relaxed, before reliable results can be obtained.

The SRG algorithm's drawbacks relating to pixel order dependencies have been effectively compensated by the ISRG algorithm, generating a powerful tool for image segmentation. If the choice of seeds is pertinent, we can expect an

output image in which the separation of the regions can be described as borderline intelligent. However, the major disadvantage of the two seeded region growing algorithms is that they demand initialization by a user. They cannot work independently, since they are designed to expect more than the image as input; they need the initial seeds to be selected manually or another generating process to provide them.

Nevertheless, despite their apparent disadvantages, both algorithms have potential of contributing to the advancement of image segmentation techniques, were their weaknesses overcome. The SRG algorithm provides quite efficient image segmentation. It has been successfully tested for natural (outdoors, portrays etc) and artificial (usually biomedical) images. To make it fully functional the obvious extension is simply to automate the method of seed selection. The watershed algorithm on the other hand is fully automated. Various filtering methods could resolve the oversegmentation problem and a lot of solutions have been proposed about it in the literature [13], [21], [22]. But an alternative use for it could be, instead of improving it, to employ it as a first stage in a combination of image segmentation algorithms, where a problematic automated method can provide initialization for a superior non automated method.

Chapter 3 - PROPOSED METHOD

In the present work the goal was to create an automated, yet effective technique for biomedical image segmentation. The literature was reviewed thoroughly and various methodologies were studied. Several issues were taken into consideration before starting the implementation, all pointed out throughout the published literature. Many questions were raised concerning every aspect of the problem; from just segmentation efficiency and application suitability to speed and memory. Combining two successful, well established approaches of the field seemed as the most promising course of action.

The methods combined in this work are the watershed transform, as described by Meyer in [17], and the Improved Seeded Region Growing algorithm, as described by Mehnert and Jackway in [20]. Although they are both long existing and popular algorithms, besides their implementation, significant effort and experimentation were required before the way they would be connected was decided. Apart from that, post processing was also required to surpass the limitations imposed by the algorithms' design principles.

3.1. Description

The course of work for the proposed scheme was planned as follows. First the watershed algorithm would be used to provide a rough initial segmentation of the image. This output would be refined later by the Improved Seeded Region

Growing algorithm. A transitional step would be required so that the initial rough segmentation could be utilized as additional input to the Improved Seeded Region Growing algorithm, that is to generate a replacement for the manual seed selection. So, the seed selection was handled as a post processing step of the watershed algorithm. Next the post processed output needed to be transformed in an appropriate structure to function as input for the ISRG algorithm. After the ISRG was applied, any corrections and special cases would be handled by post processing. Last the output needed to be demonstrated on the initial image and a solid validation method to evaluate the results to be decided. Prior to a more detailed description, the block diagram that follows illustrates this process:

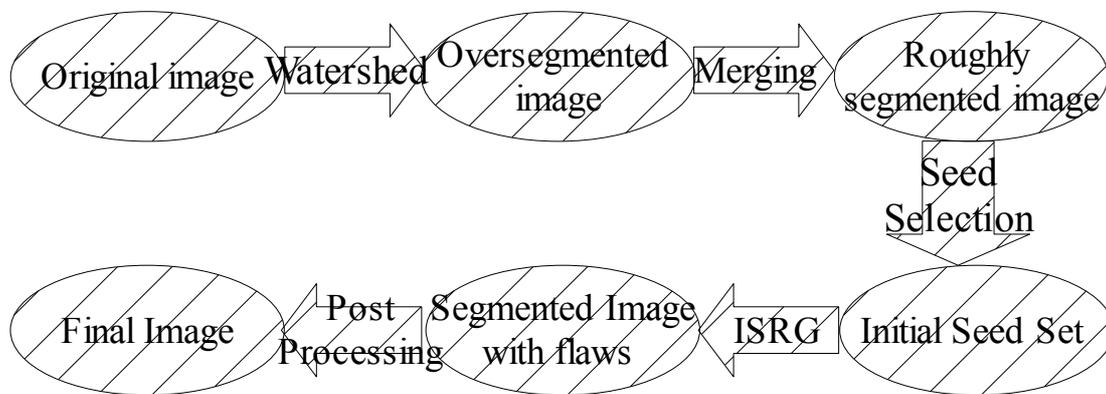


Figure 1: The proposed method – Block Diagram

Step 1 – The Watershed algorithm: The initial stage of this implementation, the first segmentation with the watershed algorithm, was performed with respect to the image pixels' value 'topographical' distance, as described by Meyer in [17]. This operation's output is a label matrix. A label matrix has the size of its corresponding image, each cell representing the respective pixel, but the pixels, instead of being assigned a gray value, are assigned a label indicating the region they were allocated to; in our case the label was numerical, starting from 1 and reaching the total number of regions (with step equal to 1). Among pixels with different labels a line representing the watershed line was created. Then this matrix was superimposed on the image. As expected, the output image was

severely oversegmented providing absolutely no useful results. The pixels being divided in numerous small groups is almost as good as not being grouped at all. But, even though we realized that the image was partitioned every time the slightest variations of gray value occurred, and that this amount of detail was undesirable, it was a fact that the image was also segmented when bigger changes were present. This is the contribution of the first level of segmentation that we were seeking.

Step 2 – Merging of oversegmented regions: At this point the next task is apparent. We needed to employ a technique that would allow a form of filtering, to help us distinguish the significant gray value fluctuations from the insignificant ones. Common thresholding methods require the use of histograms, and since, having already incorporated quite a few data type transitions (labels to gray values and vice versa) in our scheme, we were trying to confine the complexity of the structure when it was feasible, an alternative thresholding technique was implemented. In addition to preserving simplicity of the algorithm, another issue that needed alternative handling was the possible presence of noise that was responsible for the oversegmentation to start with. So a statistic (mean gray value in our case) related approach seemed more appealing.

The tactic actually followed to decide the significance of some watershed lines over others was merging of the segments produced by the watershed segmentation. More specifically the mean gray value of each segment was calculated. Then, starting from the first region, the absolute difference between the mean gray values of each region and its adjacent ones was calculated, and, if it was found to be smaller than a certain threshold, these regions were grouped together. This was done iteratively until no more adjacent regions qualified for merging with the group under examination. The same was repeated for the first region that had not been merged (the regions being labeled numerically, this is the first consecutive label number that was listed as non-merged, an actually random process, since there is no consistency in the order that labels are assigned), until no further merging was possible. This raises questions about order dependencies induced by the implementation choices. They will be discussed more thoroughly later.

The matter that needs to be attended forthwith is how the threshold that determines whether two or more regions should be merged is selected. It is clear that, as it is a similarity criterion for mean gray values, it should be a gray value itself. A constant value cannot be expected to exhibit good behavior without regard to the input image. But the option of adjusting it exclusively for every image is also not available, as our intention is to create a fully automated method. Thus, we decided to use a threshold that is related to the statistics of the image. Mean gray value of the image is an indicative parameter of the image, and so it is reasonable the threshold to be expressed as a fraction of the mean gray value. The exact percentage has to be determined through experimentation, taking into account that different classes of images may behave in a totally different manner. Although this will be presented in more detail in the result chapter, let us note that, for both biomedical and natural gray scale images, the threshold value that ensures the desired outcome of image segmentation, when the proposed scheme is implemented, was found to be 10–20% of the image's mean gray value, causing neither over- nor undersegmentation.

So far we have overcome the oversegmentation problem and eliminated any unnecessary responsiveness of the watershed algorithm to minor changes of gray level. The image is now segmented into larger regions that have been constructed by merging smaller ones with similar mean gray values. This process would be sufficient for the watershed's oversegmentation problem to be fixed and for us to have our automatic non-oversegmenting algorithm working if it had not been so roughly approached. The reason why more refinement was not pursued is that we are yet only half way through our implementation and detailed region separation will be achieved by the following stage.

Step 3 – Seed Acquisition: The next step was to connect the as far processing with the second major algorithm we intended to utilize. The Improved Seeded Region Growing algorithm can supply the refined segmentation we are trying to achieve, as long as the seed selection is done correctly. Through the watershed algorithm and its post processing we have obtained a coarse segmentation of the image, that is not sufficient for our application's demands. It

is, however, relatively close to the desired output, having already identified the homogeneous regions. This coarse segmentation was at this point used to provide the Improved Seeded Region Growing algorithm with its initial seed set.

The Improved Seeded Region Growing algorithm was implemented as described by Jackway and Mehnert in [20]. According to the authors each seed is a single point or a set of points (pixels) in the image. To improve the odds that our seeds are selected correctly, we used one point seeds to start the region growing process. Each seed resulted from one of the partitions created during the previous step. The Improved Seeded Region Growing algorithm works in such a way that the regions do not grow evenly around the seed. This gives us the trust that, if we succeed to elect the seeds so that they belong anywhere within each expected region, the algorithm will manage to produce a reliable partition of the image. Once more, in order to improve our odds of selecting the appropriate seed, we selected one of the middle points of each existing partition to serve as a seed for the Improved Seeded Region Growing algorithm, when it was possible (for convex -in a relaxed sense- partitions), or just a non boundary one for non convex partitions.

Step 4 – The Improved Seeded Region Growing Algorithm: After selecting the seeds, the ISRG algorithm performs the final segmentation. Let us once more note that the number of the final partitions equals the number of the initial seed set. This means that the ISRG has to grow a region for every single seed that it is provided with, although it is expected to restrain the size of any partition that seems to be falsely initialized. Since the seed selection process was fully automatized, and the ISRG algorithm exhibits a global behaviour that does not allow problematic areas to be expanded, it would be both wise and feasible to perform one last processing step to ensure the best output possible. This post processing step aims to minimize the possibility that unnecessary partition has been done to the image.

Step 5 – Elimination of small, insignificant regions: What we expect to obtain after the ISRG is a fully partitioned image, that will not present oversegmentation. So, small areas will either have some meaning, or they should not exist at all. This was handled with the last step of the algorithm. A process

similar to the merging of the watershed's partitions merging followed. The difference is that the size of the partition to be merged was taken into account and that the mean gray value similarity constraint was relaxed. More specifically, the regions that had dimensions less than 3% of the image's respective dimension will be the new candidates for merging. The reason that each dimension separately and not the total size (we did not use e.g. the total pixel number) is taken into account, is that if a region has grown along one dimension, it will probably be significant despite its small size. The new merge-candidate regions are next examined one by one, and the absolute difference between their mean gray value and the mean gray values of all their neighboring regions is calculated. If this difference is less than 10% of the neighbor's mean gray value, then the candidate-region is merged with that neighbor. If more neighbors qualify for merging, the larger partition will be selected.

This step concludes the proposed scheme. One last point that should be made is that throughout implementation many different percentages were tested before we concluded to the ones mentioned above. Although these numbers did not always give the best results, their behaviour was always quite satisfying, and wanting to make the algorithm free of parameter tuning, they were selected as optimum for the general case.

3.1.1. Overview

As it will become obvious in the Results chapter, the proposed scheme provides us with trustworthy image segmentation. In particular it performs well for segmenting biomedical images, it is fully automatic and parameter free and it combines sensitivity to gray value fluctuations and robustness. It seems to demonstrate weaknesses because of the dependencies caused by implementation choices during the grouping process after the first segmentation with the watershed algorithm. Although they could have been avoided, of course causing increased complexity of the scheme, we find that they were overcome by the fact that they were used merely to provide the seed set. Moreover, they were never intended to give exact segmentation, so their goal to act indicatively was achieved.

3.2. Validation

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure [23]. In that sense image segmentation consists a clustering method. In many biomedical applications cluster validation provides solutions for the systematic evaluation of the proposed clustering methodologies. Several cluster validation methods have been proposed throughout the literature. To evaluate the present work the Davies-Bouldin Validation Index was chosen to be used, since it has shown to be a robust strategy for the prediction of optimal clustering partitions [24].

The *Davies-Bouldin Validation Index* aims to identify sets of clusters that are compact and well separated. For any partition $U \rightarrow X : X_1 \cup \dots X_i \cup \dots X_c$, where X_i represents the i th cluster of such partition U , the Davies – Bouldin validation index, DB, is defined as:

$$DB(U) = \frac{1}{c} \sum_{i=1}^c \max_{i \neq j} \left\{ \frac{\Delta(x_i) + \Delta(x_j)}{\delta(x_i, x_j)} \right\}$$

where $\delta(x_i, x_j)$ defines the intercluster distance between clusters X_i and X_j , $\Delta(x_i)$ represents the intracluster distance ('diameter') of cluster X_i , and c is the number of clusters of partition U . Small index values correspond to good clusters, that is to say, the clusters are compact and their centres are far away.

In the present work two criteria were used to calculate the distances; one based on euclidean distance and one based on absolute gray value distance. The *intercluster euclidean distance* was calculated as the euclidean distance between the clusters' centres of mass, the *intercluster gray value distance* as the absolute difference between the clusters' mean gray values, the *intracluster euclidean*

distance as the euclidean distance between each pixel and its corresponding cluster's centre of mass and the *intracluster gray value distance* as the absolute difference between a pixel's gray value and its corresponding cluster's mean gray value.

If each pixel represents a point and the image a two dimensional plane, then all the pixels are described by their coordinates, e.g. $P(x, y)$. The digital images of course can only represent a digital plane, so the pixel's coordinates are discrete values. In our implementation, in specific, the images are stored in matrices with dimensions the dimensions of the image and each element holding the gray value of the respective pixel located at this position. So the coordinates of each pixel are now described by the matrix' row and column it is stored in, giving: $P(col, row)$. So, even though it is not defined in continuous space, the Euclidean Distance criterion $Dist_{EU}$ between pixels P and P' described here is defined as $Dist_{EU} = |P(col, row) - P'(col', row')|$, or following the Euclidean Distance definition, $Dist_{EU}^2 = (col - col')^2 + (row - row')^2$. The following figure illustrates the aforementioned distances in a clearer way.

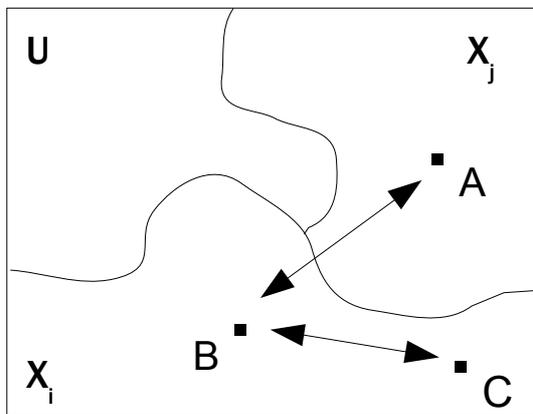


Figure 2: The euclidean distance –
Intercluster and Intracluster

We follow the formalization used to describe the Davies – Bauldin Validity Index. Let us consider a partition $U \rightarrow X : X$, where X_i represents the i th cluster of such partition U . Figure 2 illustrates an instance of $U : X_i$ and X_j are two neighboring clusters, points A and B are their centres (without

defining the distance criteria in specific, we have to use such a vague term) respectively, and point C belongs to cluster X_i . The Intracluster Distance $\delta(x_i, x_j)$ for our case is the distance between A and B. Now let us consider for cluster X_i the distance $\Delta_m(x_i)$ between its centre and a point m , that belongs in cluster X_i . For our example, for point C this would be the distance between B and C. The Intercluster Distance $\Delta(x_i)$ for cluster X_i is the maximum distance that can be defined that way, for all the pixels that belong to X_i .

Chapter 4 - RESULTS

As we discussed earlier, the method comprises five steps. For the *first step* the input is the *original image* and the output is the *watershed result*, an oversegmented image. In the *second step* this *oversegmented image* is processed and a *roughly segmented image* is acquired. The *roughly segmented image* is used in the *third step* to provide the *seed set*. Then, in the *fourth step* the *seed set* consists the input for the ISRG, and the *ISRG result* is produced. Finally, in the *fifth step* some post processing offers refinement of the *ISRG result*, providing the *final segmentation*.

The input and output of each step, as well as the ISRG result for random seed selection are illustrated for an outdoor gray scale image next. Their evaluation using the Davies - Bauldin Validity Index follows.

4.1 Comparison with other methods

To evaluate the proposed method against the Watershed and the ISRG algorithms we used a sample outdoor gray scale image [25]. We segmented the sample image using the Proposed Method (with threshold=0.15). The same image was also segmented using the ISRG algorithm for seed sets of various sizes. The seeds that initialize the ISRG algorithm were selected randomly.

Next we present the segmentation result for i) the watershed algorithm, ii) the watershed grouping (step 2 of the Proposed Method), iii) the Proposed

Method, iv) the ISRG algorithm with a few (5) random seeds, v) the ISRG algorithm with many (40) random seeds and vi) the ISRG algorithm with random seeds that are as many as our implementation's resulting number of regions. The seed set is also presented. After we demonstrate the images for the aforementioned cases, they will be compared using the Davies – Bauldin Validity Index.



Image 2: Original Image



Image 1: Watershed Result



*Image 3: Merging
(thr=0.15)*

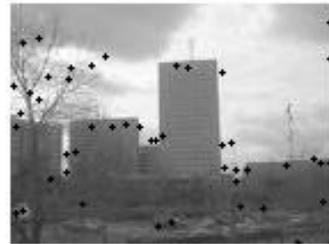
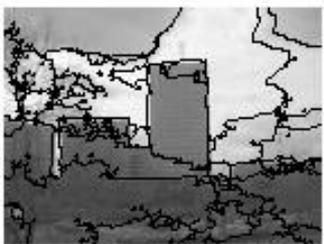


Image 4: Initial Seed Set



*Image 5: ISRG Result
with automatic seed
selection(before post
processing)*



*Image 6: Final Result
(after post processing)*



Image 7: Random seed selection-5 seeds



Image 8: ISRG Result with random seed selection (5 seeds)



Image 9: Random seed selection-36 seeds

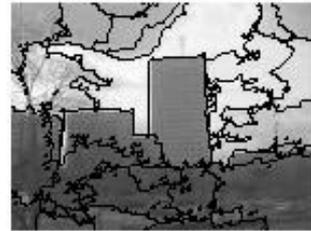


Image 10: ISRG Result with random seed selection(36 seeds)



Image 11: Random seed selection-40 seeds



Image 12: ISRG Result with random seed selection (40 seeds)

The Davies – Bauldin Validity Index was used for the methods' evaluation, using two criteria; the Euclidean Distance and the Gray Value Distance, as described in section 3.2.

Method	Watershed	Watershed Grouping	ISRG 5 seeds	ISRG 40 seeds	ISRG 36 seeds	Proposed Method
Number of Regions	602	47	5	40	36	36
Validity Index (Euclidean distance)	3.4049	6.8452	1.9283	7.7498	6.2347	6.1494
Validity Index (Gray value)	22.8321	15.9769	4.3188	30.3468	16.1436	14.4711

Table 1: Comparative Results of the Methods

From the results presented above, we observed that the Davies – Bauldin Validity Index seems to be mainly dependent on the number of clusters, and successively on the clusters' average size. Let us remind that smaller Davies – Bauldin Validity Index value implies better segmentation. According to this, the Watershed (oversegmented) result – for the euclidean distance criterion- and the randomly initiated ISRG (undersegmented) result are superior to the other results. This, however, is not supported by the images observation. The Davies – Bauldin Validity Index is defined as:

$$DB(U) = \frac{1}{c} \sum_{i=1}^c \max_{i \neq j} \left\{ \frac{\Delta(x_i) + \Delta(x_j)}{\delta(x_i, x_j)} \right\}$$

So, for the case of the Watershed algorithm, its good performance is due to small intercluster distances $\Delta(x_i)$, because of the clusters' small size. For the case of the ISRG algorithm with few seeds, the intracluster distance $\delta(x_i, x_j)$ among the clusters is large because of the clusters' big size. This causes the Davies – Bauldin Validity Index to decrease implying better performance.

Given these observations, the safest way to evaluate the proposed method's performance are visual observation and comparison with the ISRG algorithm

when the number clusters is equal. The proposed method performs better than the ISRG in this case. All meaningful entities on the image are detected and their contours are formed with precision.

4.2 Results for Biomedical Images – Proposed Method

The proposed scheme was implemented for four biomedical images, a chest CT, a chest CR and a knee and a head MRIs and it was programmed with Matlab. The experiments were conducted for seven thresholds for each image, from 10% up to 22% of the original image's mean gray value, with a step of 2%. The sample images were taken by S. Barre's online repository [26]. The images were in DICOM format, a format used in biomedical applications, and prior to the processing they were converted to JPEG and they were subsampled. Following the segmentation procedure, the results were validated using the Davies – Bauldin Validity Index. The algorithm's results will be presented and discussed for every image separately, while a total evaluation will be presented in the Conclusions chapter.

I. Chest CR

The first image we processed was a Computer Radiography (an electronic X-ray) image of a human chest. First, we present the original image, next the watershed algorithm result, and then the output of our implementation for various thresholds.

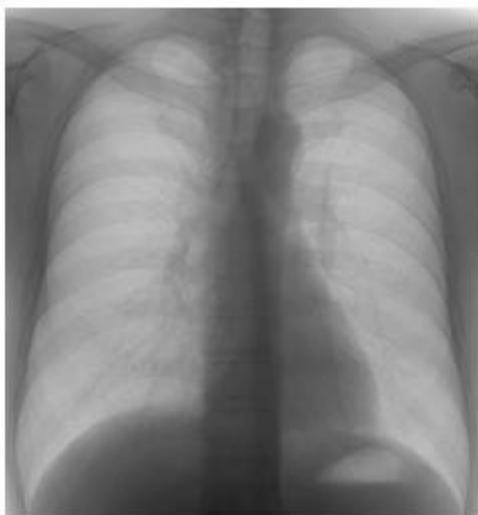


Image 13: Chest CR (original image)



Image 14: Chest CR - Watershed

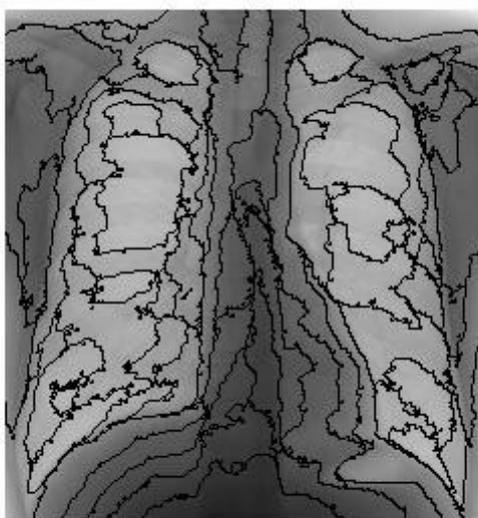


Image 15: Chest CR - Proposed scheme (threshold = 0.1)

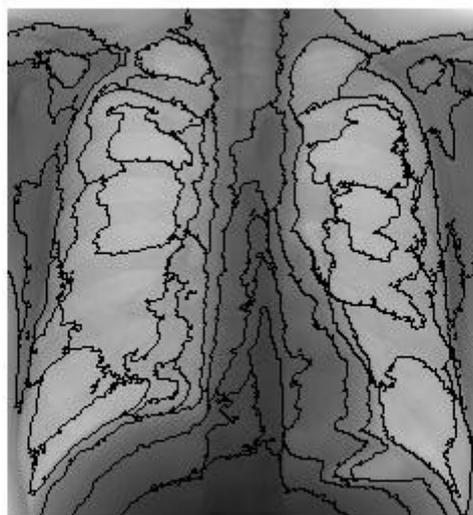


Image 16: Chest CR - Proposed scheme (threshold = 0.12)

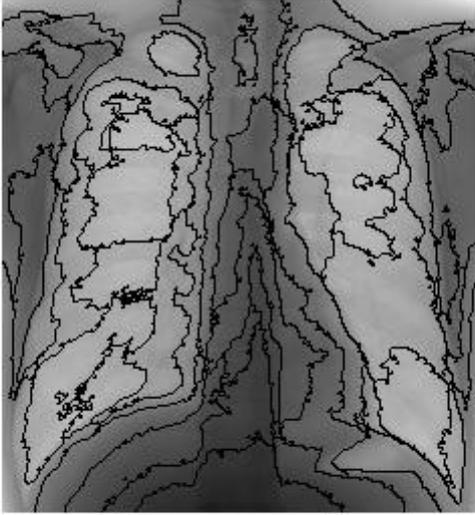


Image 18: Chest CR - Proposed scheme (threshold = 0.14)

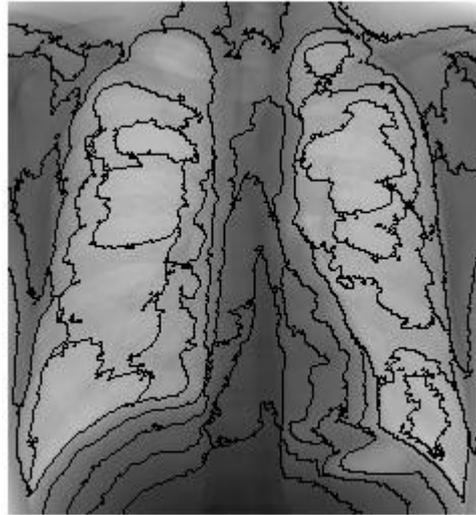


Image 17: Chest CR - Proposed scheme (threshold = 0.16)

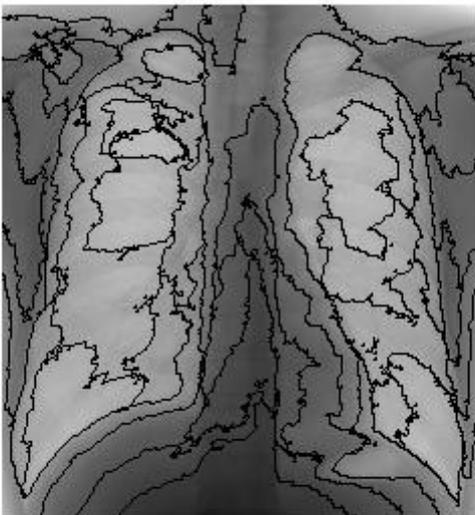


Image 19: Chest CR - Proposed scheme (threshold = 0.18)

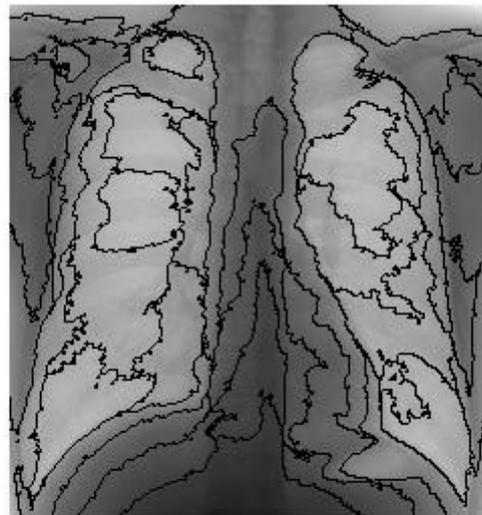


Image 20: Chest CR - Proposed scheme (threshold = 0.2)

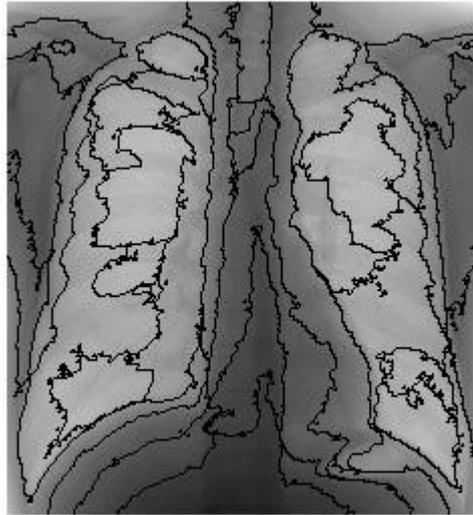


Image 21: Chest CR - Proposed scheme (threshold = 0.22)

Thresholds/ (mean Gray value)	0.1	0.12	0.14	0.16	0.18	0.2	0.22
Number of Regions	95	95	62	57	53	53	50
Validity Index (Euclidean distance)	3.3896	3.4324	4.9275	3.2464	3.0897	3.2227	11.6964
Validity Index (Gray value)	22.0743	18.9827	8.9058	11.2936	19.7293	17.6986	18.6787

Table 2: Validity Indexes – Chest CR

First we observe that the number of regions decreases as the threshold increases. This was expected; the threshold determines the merging process of step 2 of the Proposed Method. Higher thresholds allow more regions to be grouped together. So, fewer seeds are selected to initialize step 4 of the method, resulting in fewer final partitions in the image.

The Validity Index behave differently when different criteria are used. The

Validity Index according to the Euclidean Distance criterion is minimized for threshold equal to 18% of the mean gray value of the image, while according to the Gray Value Distance it is minimized for threshold equal to 14% of the mean gray value of the image. Visual observation supports the conclusion according to the Euclidean Distance, so it is found to be more reliable. The Validation Index, as it is defined, when calculated using the Gray Value criterion is sensitive to fluctuations that we tried to overcome during the method's designing stage. One single pixel's gray value can affect the intercluster distance, thus affecting the Validity Index. This is not acceptable, so, although the Validity Index according to the Gray Value Distance offers useful conclusions when no errors are present, when it contradicts with the conclusions exported by the Validity Index according to the Euclidean Distance, it is treated with skepticism.

II. Chest CT

The second image we processed was a Computerized Tomography image of a human chest. First, we present the original image, next the watershed algorithm result, and then the output of our implementation for various thresholds.



Image 22: Chest CT (original image)

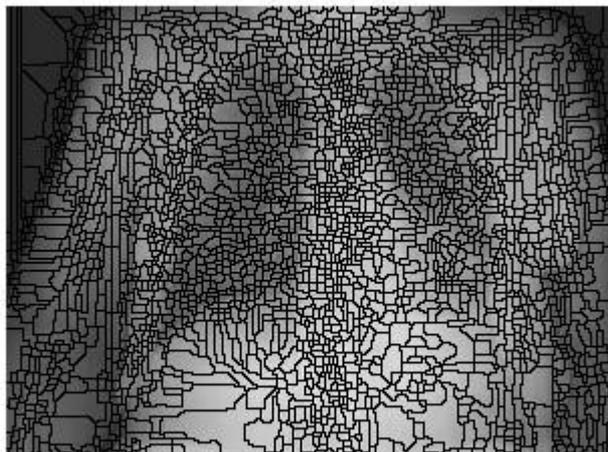


Image 23: Chest CT - Watershed

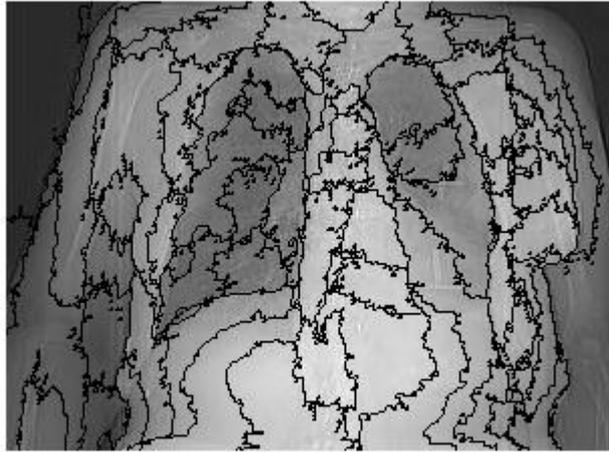


Image 24: Chest CT - Proposed scheme (threshold = 0.12)

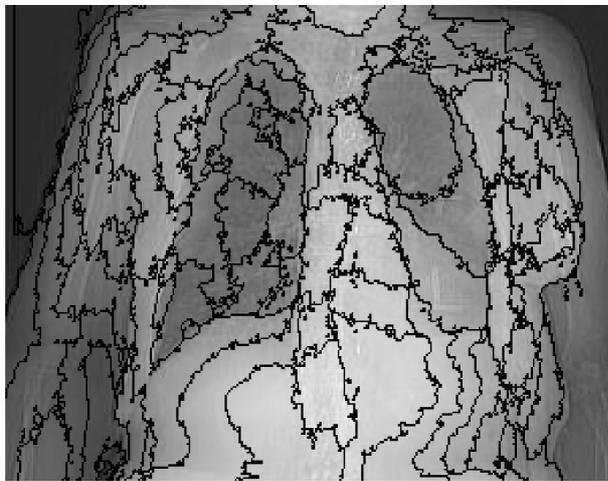


Image 25: Chest CT - Proposed scheme (threshold = 0.1)

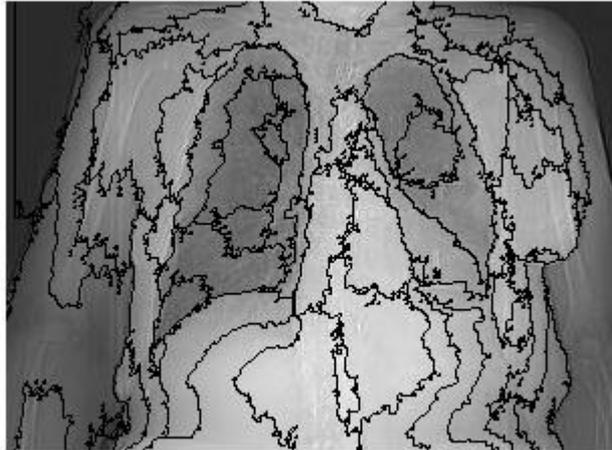


Image 26: Chest CT - Proposed scheme (threshold = 0.14)

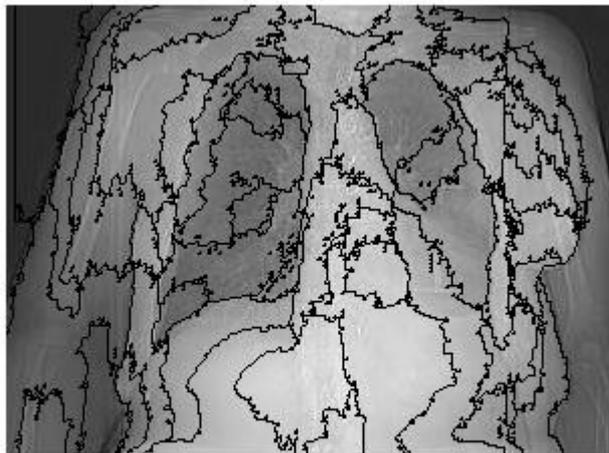


Image 27: Chest CT - Proposed scheme (threshold = 0.16)

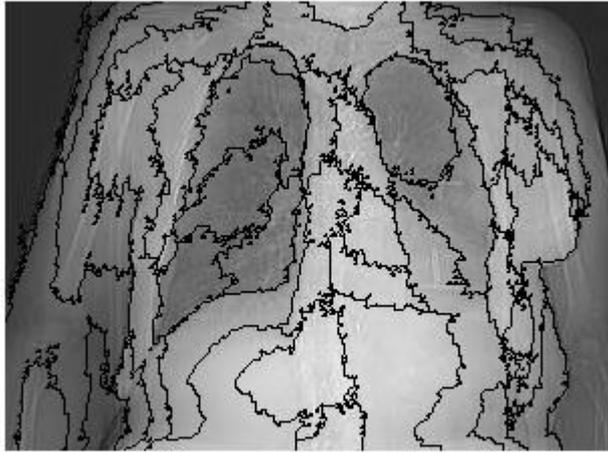


Image 28: Chest CT - Proposed scheme (threshold = 0.18)

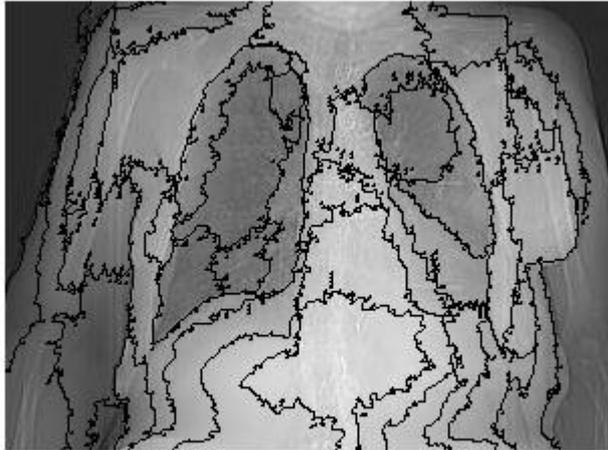


Image 29: Chest CT - Proposed scheme (threshold = 0.2)

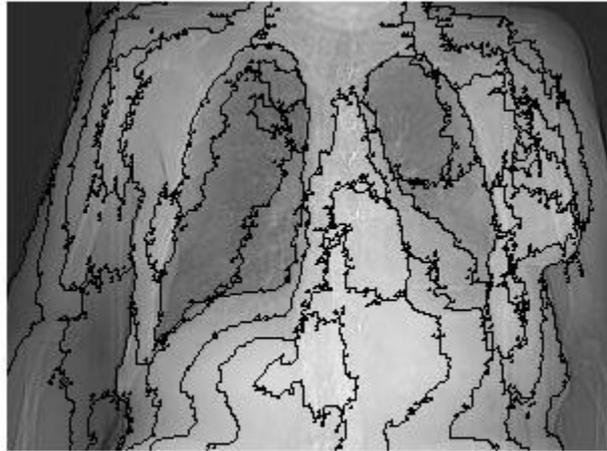


Image 30: Chest CT - Proposed scheme (threshold = 0.22)

Thresholds/ (mean Gray value)	0.1	0.12	0.14	0.16	0.18	0.2	0.22
Number of Regions	69	57	56	46	46	41	36
Validation (Euclidean distance)	7.3628	6.1001	5.8016	5.4734	3.6560	7.6478	14.1355
Validation (Gray value)	28.745	26.1708	15.866	11.309	4.3058	28.5602	16.852

Table 3: Validity Indexes - Chest CT

Once more we observe that the number of regions decreases as the threshold increases. The Validity Index according to both the Euclidean Distance and the Gray Value Distance criterion shows that the best segmentation is achieved when the threshold equals 18% of the image's mean gray value. This is supported by visual observation as well.

III. Head MRI

The third image we processed was a Magnetic Resonance Image of a head. First, we present the original image, next the watershed algorithm result, and then the output of our implementation for various thresholds.

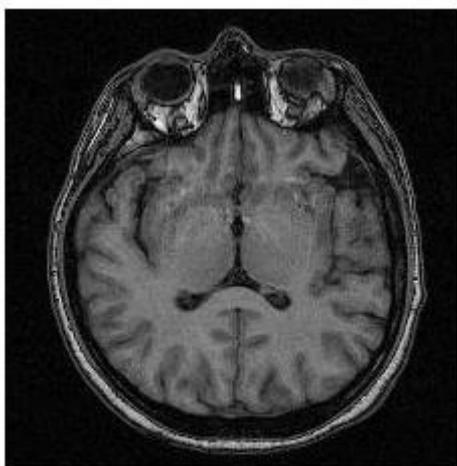


Image 31: Head MRI (original image)

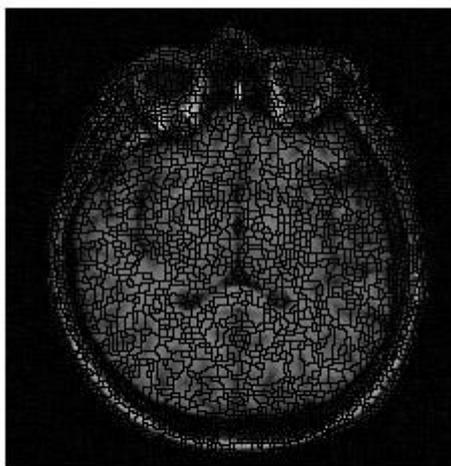


Image 32: Head MRI - Watershed

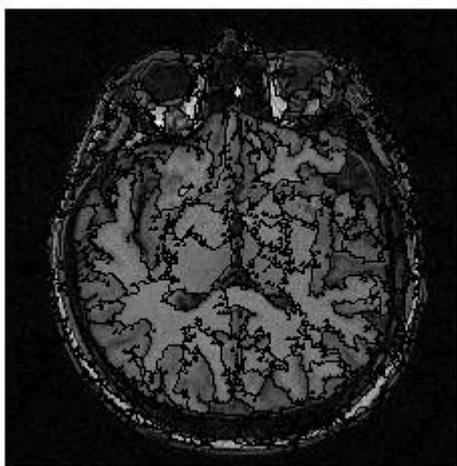


Image 34: Head MRI - Proposed scheme (threshold = 0.1)

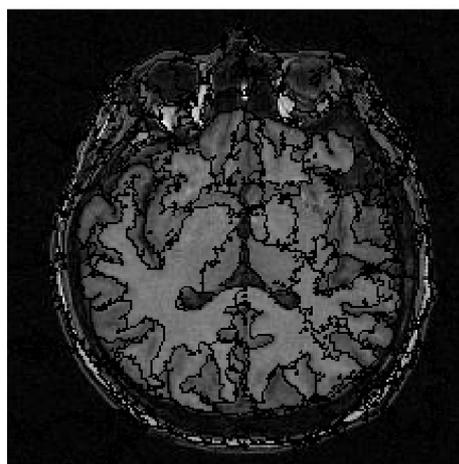


Image 33: Head MRI - Proposed scheme (threshold = 0.12)

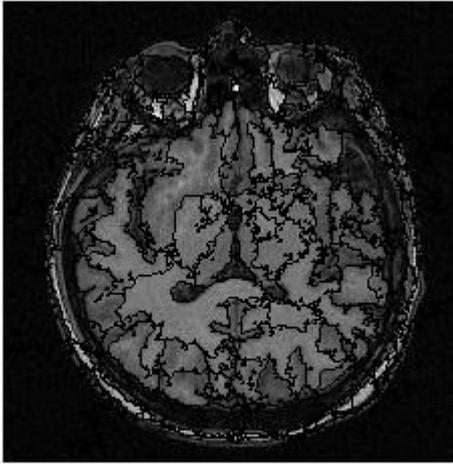


Image 35: Head MRI - Proposed scheme (threshold = 0.14)

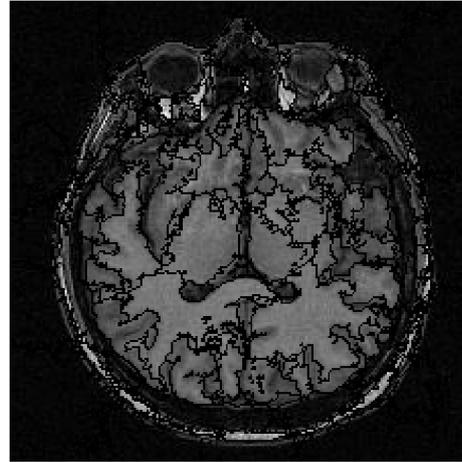


Image 36: Head MRI - Proposed scheme (threshold = 0.16)

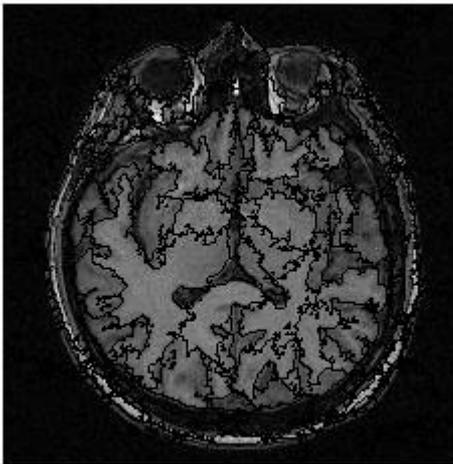


Image 37: Head MRI - Proposed scheme (threshold = 0.18)

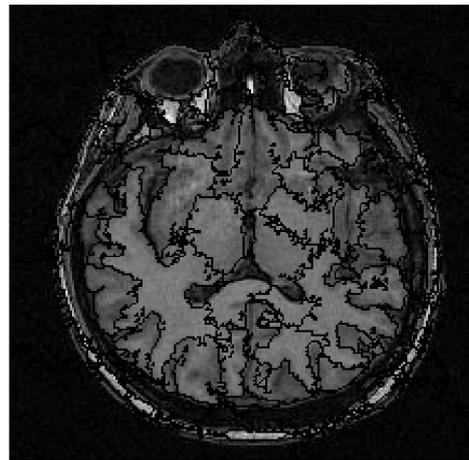


Image 38: Head MRI - Proposed scheme (threshold = 0.20)

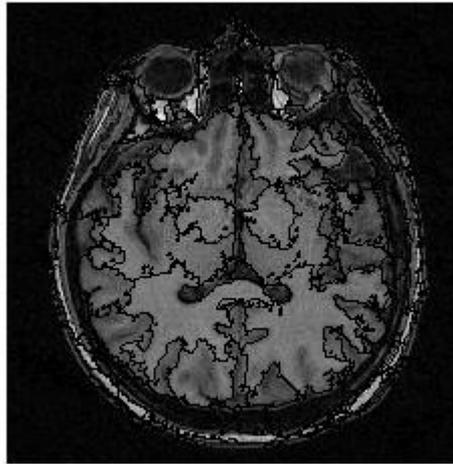


Image 39: Head MRI - Proposed scheme (threshold = 0.22)

Thresholds/ (mean Gray value)	0.1	0.12	0.14	0.16	0.18	0.2	0.22
Number of Regions	192	172	158	140	134	120	113
Validation (Euclidean distance)	5.6393	6.1465	4.3254	6.2936	5.1273	5.3437	4.06
Validation (Gray value)	23.5667	25.3853	16.5654	9.4693	12.7493	13.6033	18.5434

Table 4: Validity Indexes - Head MRI

For the third image we observe that the Validity Index according to the Euclidean Distance is minimized for threshold equal to 16% of the original image's mean gray value. Although the Validity Index according to the Gray Value Distance is not minimized for the same threshold, it behaves in similar way, supporting the first conclusions.

IV. Knee MRI

The fourth image we processed was a Magnetic Resonance Image of a human knee. First, we present the original image, next the watershed algorithm result, and then the output of our implementation for various thresholds.



Image 40: Knee MRI (original image)



Image 41: Knee MRI - Watershed

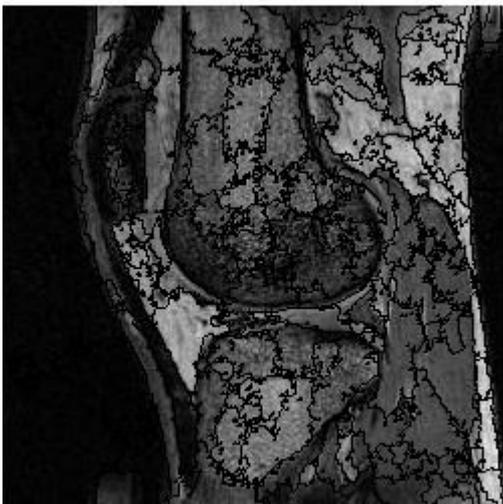


Image 43: Knee MRI - Proposed scheme (threshold = 0.1)

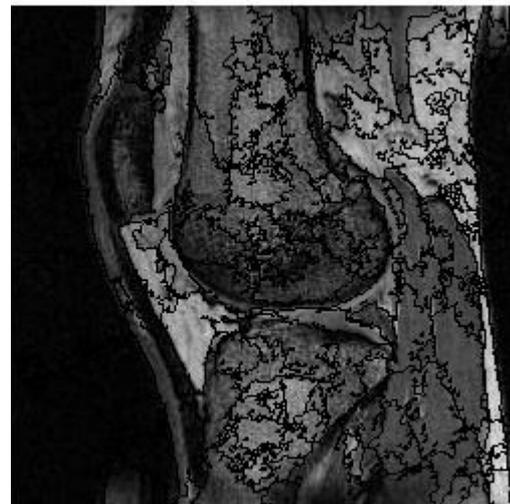


Image 42: Knee MRI - Proposed scheme (threshold = 0.12)

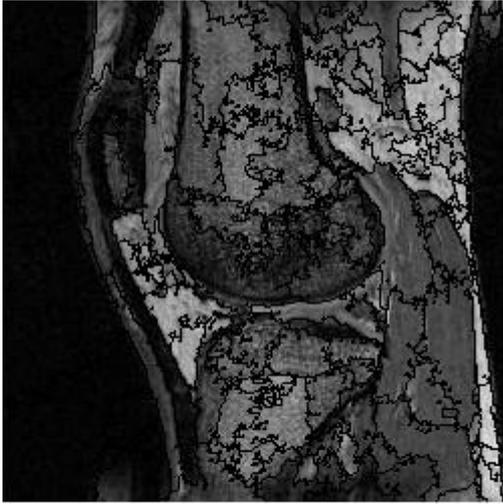


Image 44: Knee MRI - Proposed scheme (threshold = 0.14)

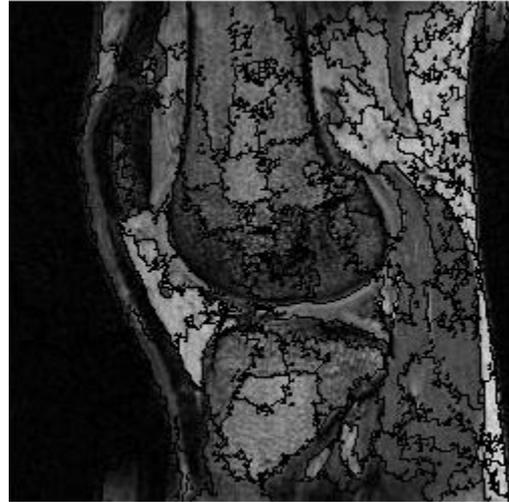


Image 45: Knee MRI - Proposed scheme (threshold = 0.16)

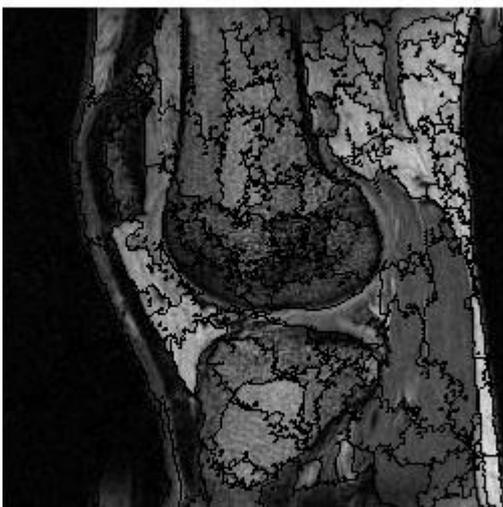


Image 47: Knee MRI - Proposed scheme (threshold = 0.18)

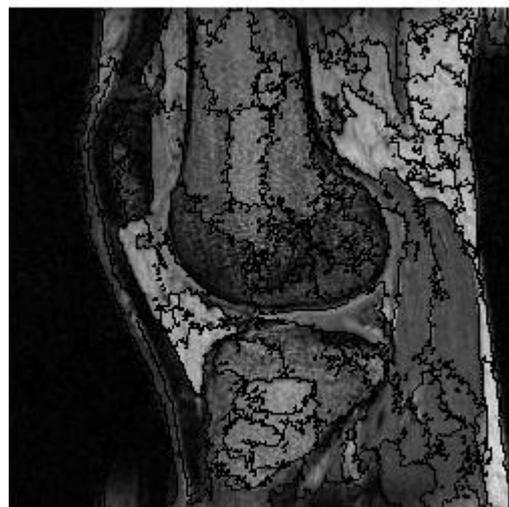


Image 46: Knee MRI - Proposed scheme (threshold = 0.2)

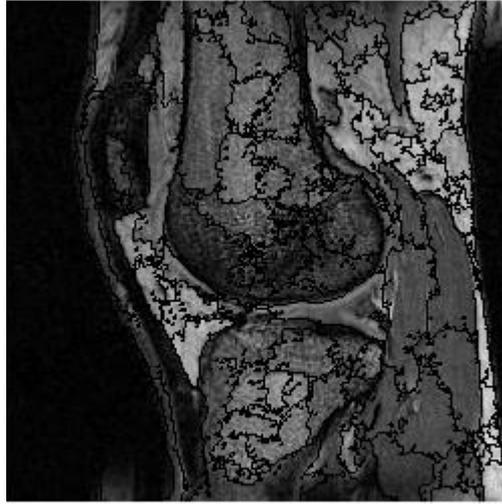


Image 48: Knee MRI - Proposed scheme (threshold = 0.22)

Thresholds/ (mean Gray value)	0.1	0.12	0.14	0.16	0.18	0.2	0.22
Number of Regions	205	182	168	154	136	127	116
Validation (Euclidean distance)	5.7231	7.1284	6.4595	6.1866	8.9479	8.7083	8.1595
Validation (Gray value)	25.7061	24.65	23.3185	14.3082	23.08	25.2644	24.0073

Table 5: Validity Indexes - Knee MRI

Again, we observe that the number of regions decreases as the threshold increases. The Validity Index according to both the Euclidean Distance and the Gray Value Distance criterion shows that the best segmentation is achieved when the threshold equals 16% of the image's mean gray value. This is supported by visual observation as well.

The grouping process, that provides a first estimation about the regions of interest in the image, depends on the similarity criterion according to which the

watershed regions are merged. So the final segmentation is greatly influenced by the choice of the threshold in step 2 of the proposed method. For all the images, both observation and cluster evaluation with the Davies – Bauldin Validation Index, conclude that the Proposed Method offers optimum segmentation when the threshold is about 16% of the mean gray value.

Chapter 5 - CONCLUSIONS & FUTURE WORK

5.1. Conclusions

As imaging becomes a major component of most biomedical applications, methods that warrant reliable image segmentation have become a necessity during the last years. Various algorithms have been presented throughout the literature, following different approaches, requiring different input and resources and offering different levels of performance.

In this work two popular methods for image segmentation were implemented and combined to improve segmentation performance on biomedical gray scale images. First the watershed transform was used to produce an automatic but inaccurate segmentation, and after post processing the appropriate seed set for the Improved Seeded Region Growing algorithm was created. The ISRG was then utilized to produce a more refined segmentation, and after one more post processing step the final segmentation was obtained. The choice of the similarity threshold in the seed selection step was found to affect the output of the proposed scheme and its optimum value was determined through experimentation.

The proposed method, although fully automatic, offered reliable image segmentation. It proved to clearly outperform the watershed algorithm,

providing meaningful separation of the image. It was accurate recognizing the gray level transitions, without presenting the undesired watershed's sensitivity to noise and small gray value fluctuations.

The method's final output was largely dependent on the ISRG algorithm's performance. However, it managed to overcome the ISRG's biggest weakness, manual seed selection. Although the method was not expected to outperform the ISRG when content based initialization is provided manually, the results were very satisfactory compared to ISRG with random seed selection.

The ISRG automatic initialization relied on the suitable selection of a merging criterion for the watershed result. The tests proved that the appropriate merging technique was to group together the neighboring regions, the mean gray value different of which was about 14 to 16 percent of the image's mean gray value. This threshold, along with the final deletion of the insignificant areas caused this method's output to be comparable with manual image segmentation.

The proposed method ensures robust, accurate segmentation of gray scale images. It offers fully automatic and reliable segmentation of biomedical images, contributing to several biomedical tasks, such as morphometry and change detection that require precision, objectiveness and reproducibility.

5.2. Future Work

The proposed scheme proved to have met its design expectations. There are, however, some extensions that could improve its outcome and broaden its applicability.

The most obvious disadvantage of this method is the contour display. The line that divides the regions is irregular, and though it follows the region growing process precisely, its appearance is not natural, especially when mild transitions of the gray level occur. For the present work the contours were outlined just to illustrate the final region formation, but for a more complete application they need to become smoother and more regular. This can be performed as a post processing step, affecting solely the line appearance, not actually altering the segmentation.

There are also a few limitations to the proposed method. First, its functionality was only tested on gray scale images. We can extend this method to colored images, by treating each subband individually and combining the results. More elaborate techniques that take into account inter-subband correlation can also be applied. Second, the images were uncoded (bmp, JPEG). Medical images are usually provided in application specific formats (Dicom, analyze75), that include an abundance of additional information. This information can be extracted and further support the segmentation (e.g. by taking into account exposure time or other parameters and preprocess the image).

REFERENCES

- [1] Michael W. Vannier , John W. Haller, Biomedical Image Segmentation, Proceedings International Conference on Image Processing (ICIP), 1998
- [2] Maneesha Singh, <http://www-staff.lboro.ac.uk/~coms>
- [3] R.B. Ohlander, Analysis of natural scenes, PhD Thesis, Carnegie Institute of Technology, Carnegie Mellon University, Pittsburgh, PA 1975
- [4] M. Cheriet, J. N. Said and C. Y. Suen, A recursive thresholding technique for image segmentation, IEEE Transactions on Image Processing , Vol., No, pp., 1998
- [5] J.M. Prager, Extracting and labeling boundary segments in natural scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol.2, No1, pp.16-27, 1980
- [6] W.A. Perkins, Area segmentation of images using edge points, IEEE Transactions on Pattern Recognition and Machine Intelligence , Vol.2, No1, pp.8-15, 1980
- [7] F.H.Y. Chan, F. K. Lam and H. Zhu, Adaptive thresholding by variational method, IEEE Transactions on Image Processing , Vol.2, No3, pp.168-174, 1998
- [8] A.K. Jain and R.C. Dubes, Algorithms for clustering data, Prentice Hall, Englewood Cliffs, N.J., 1988
- [9] F. Yarman-Vural and E. Ataman, Noise, histogram and cluster validity for Gaussian-mixed data, Pattern Recognition , Vol.20, No4, pp.385-401, 1987
- [10] J.F. Haddon and J.F. Boyce, Image segmentation by unifying region and boundary information, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol.12, No10, pp.929-948, 1990
- [11] N.W. Campbell, B.T. Thomas and T. Troscianko, Automatic segmentation and classification of outdoor images using neural networks, International Journal of Neural Systems , Vol.8, No1, pp.137-144, 1997
- [12] N. Papamarkos, C. Strouthopoulos and I. Andreadis, Multithresholding of colour and gray-level images through a neural network technique, Image and Vision Computing , Vol.18, No, pp.213-222, 2000
- [13] J. B. T. M. Roerdink and A. Meijster, The Watershed Transform: Definitions, Algorithms and Parallelization Strategies, Fundamenta Informaticae , Vol.41, No1-2, pp.187-228, 2001
- [14] H. Digabel and C. Lantuejoul, Iterative algorithms, Proceedings 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, 1977
- [15] S. Beucher and C. Lantuejoul, Use of watersheds in contour detection, Proceedings International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation, 1979
- [16] J. Serra, Image Analysis and Mathematical Morphology, Academic Press , 1982
- [17] F. Meyer, Topographic distance and watershed lines, Signal Processing , Vol.38, No1, pp.113-125, 1994
- [18] Andreas Bieniek, Hans Burkhardt, Heiko Marschner, Gerald Schreiber and Michael Nolle, A parallel watershed algorithm, Proceedings 10th Scandinavian Conference on Image Analysis (SCIA'97) , 1997
- [19] R. Adams and L. Bischof, Seeded region growing, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol.16, No6, pp.641-647, 1994
- [20] A. Mehnert and P. Jackway, An improved seeded region growing algorithm, Pattern Recognition Letters , Vol.18, No1, pp.1065-1071, 1997
- [21] F. Meyer and S. Beucher, Morphological Segmentation, Journal of Visual

Communication and Image Representation , Vol.1, No1, pp.21-46, 1990

[22] Luc Vincent, New trends in morphological algorithms, Proceedings SPIE/SPSE, 1991

[23] www.en.wikipedia.org

[24] N. Bolshakova and F.Azuaje, Improving Expression Data Mining through Cluster Validation, Proceedings 4th Annual IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003

[25] <http://visionlab.ece.uiuc.edu/datasets.html>

[26] S. Barré, <http://www.barre.nom.fr/medical/samples>