

TECHNICAL UNIVERSITY OF CRETE, GREECE  
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

# Automatic coastline extraction from aerial images using edge detection and optimization procedures



Vasilis Paravolidakis

Thesis Committee  
Professor Michael Zervakis  
Professor Katerina Mania  
Dr. Lemonia Ragia

Chania, April 2018



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτόματη Εξαγωγή Ακτογραμμής από  
Αεροφωτογραφίες Χρησιμοποιώντας  
Τεχνικές Εύρεσης ακμών και Διαδικασιών  
Βελτιστοποίησης



Βασίλειος Παραβολιδάκης

Εξεταστική Επιτροπή  
Καθηγητής Μιχαήλ Ζερβάκης  
Καθηγήτρια Αικατερίνη Μανιά  
Δρ. Λεμονιά Ράγια

Χανιά, Απρίλιος 2018





# Abstract

Nowadays, coastline detection and formation monitoring has become of high importance due to the increasing effects of climate change, global warming and rapid growth of human population on the morphological characteristics of coasts. Coastal areas play a significant role on the economy of an entire region especially in major tourist destinations, where tourist infrastructure has been developed in the last decades. This thesis addresses the problem of coastline extraction and morphological change monitoring/tracking in the course of decades. The proposed methodology presents an automatic coastline extraction algorithm, which utilizes grayscale aerial images. Our method is summarized in a four-step algorithm that enables robust extraction of the coastline.

Specifically, the first step of our approach is a noise distortion reduction stage with the goal of ameliorating its impact on the input data. The subsequent step involves the application of a local threshold in the input image to segment it into a binary class image, land and sea. The result is then passed to the third stage of our approach which applies morphological operators on the binary image, with the aim of removing small objects and preserving only the objects of interest. Finally in the last stage of our method, we perform edge detection and active contours fitting, in order to extract and model the coastline. The aforementioned algorithmic steps are illustrated through examples, which demonstrate the efficacy of the proposed methodology. The algorithm is also applicable to one band satellite imagery due to the fact they have same characteristics with grayscale aerial images.



## Περίληψη

Στις μέρες μας η εύρεση των ακτογραμμών και ο εντοπισμός των αλλαγών τους γίνεται ολοένα και πιο αναγκαίο λόγω των κλιματικών αλλαγών, το φαινόμενο του θερμοκηπίου και τις ραγδαίες αύξησης του πληθυσμού. Οι περιοχές τις ακτής διαδραματίζουν καθοριστικό ρόλο στην οικονομίας μίας ολόκληρης περιοχής, ειδικά σε μεγάλους τουριστικούς προορισμούς, όπου νέα τουριστικά κτήρια και υποδομές έχουν δημιουργηθεί τα τελευταία χρόνια. Η παρούσα διπλωματική εργασία ασχολείται με το πρόβλημα της εύρεσης ακτογραμμών καθώς και την εύρεση των αλλαγών τους στο πέρασμα των χρόνων. Η προτεινόμενη μεθοδολογία για μία αυτόματη εξαγωγή της ακτογραμμής, χρησιμοποιώντας αεροφωτογραφίες σε κλίμακα του γκρι, αποτελείται από τέσσερα βήματα έτσι ώστε η εξαγωγή της να είναι εφικτή ανεξάρτητα από την περιοχή ενδιαφέροντος και την ποιότητα της εικόνας. Αρχικά, μειώνεται η παραμόρφωση λόγω θορύβου έτσι ώστε να βελτιωθεί η ποιότητα στα επόμενα βήματα επεξεργασίας. Στη συνέχεια, η εικόνα τμηματοποιείται σε 2 μέρη, τη θάλασσα και την περιοχή της ακτής, με τη χρήση της εφαρμογής τοπικών κατωφλίων, έτσι ώστε τελικά να δημιουργηθεί μία binary (ασπρόμαυρη) εικόνα. Μια περαιτέρω επεξεργασία στα αποτελέσματα με τη βοήθεια μορφολογικών τελεστών, μας παράγει μία εικόνα χωρίς μικρά αντικείμενα. Δηλαδή μία εικόνα που περιέχει μόνο περιοχές ενδιαφέροντος (θάλασσα και ακτή). Τέλος, εφαρμόζονται τεχνικές ευρέσεως ακμής και 'ενεργών' περιγραμμάτων έτσι ώστε να εξαχθεί και να μοντελοποιηθεί η ακτογραμμή. Τα βήματα του αλγορίθμου απεικονίζονται στα παραδείγματα τα οποία απεικονίζουν την αποτελεσματικότητα της προτεινόμενης μεθοδολογίας. Ο αλγόριθμος είναι εφαρμόσιμος επίσης σε φωτογραφίες σε κλίμακα του γκρι από δορυφόρο λόγω των κοινών χαρακτηριστικών που έχουν με τις αεροφωτογραφίες.



## Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Michail Zervakis for his trust and continuous support during the course of this thesis.

Next, the Display Lab Supervisor Konstantia Moirogiorgou for all the help she provided me. Also Dr. Lemonia Ragia for her guidance and motivation.

My friends from Chania and Irakleio, Nikos (MM) Eleni (Bouklou), Eleni (Paoki), Katerina (tondo), Kostas (Bouclas), Marinio, Nikolas, Katerina (Ksanthia), Christos (Xristaras), Nikos (Androu) and Maria ('inserts heart emoticon'). I will always remember all the great memories we created.

Last, but not least, I would like to thank my parents and my sisters for their support and constant encouragement.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Thesis Contribution . . . . .	2
1.2	Thesis Outline . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Coastline definition . . . . .	3
2.2	Imagery Data Sources . . . . .	3
2.3	Noise reduction . . . . .	6
2.3.1	Gaussian Noise . . . . .	6
2.3.2	Salt and Pepper Noise . . . . .	7
2.3.3	Speckle Noise . . . . .	7
2.3.4	Filtering Techniques . . . . .	7
2.4	Image Segmentation . . . . .	11
2.4.1	Edge Detection . . . . .	11
2.4.2	Segmentation using Clustering Based Thresholding . . . . .	13
2.4.3	Active Contours . . . . .	15
<b>3</b>	<b>Problem Statement and Related Work</b>	<b>17</b>
3.1	Problem Statement . . . . .	17
3.2	Related Work . . . . .	17
3.2.1	SAR Imagery Based Approaches . . . . .	18
3.2.2	LIDAR imagery Approaches . . . . .	20
3.2.3	Aerial imagery Approaches . . . . .	21
<b>4</b>	<b>Our Approach</b>	<b>23</b>
4.1	Pre-processing . . . . .	25

## CONTENTS

---

4.2	Region Segmentation . . . . .	28
4.2.1	K-means Clustering . . . . .	28
4.2.2	Block Division and Local Thresholding . . . . .	29
4.2.3	Local thresholds by fitting a bimodal Gaussian curve . . . . .	29
4.3	Post-Processing . . . . .	33
4.4	Coastline modeling . . . . .	35
<b>5</b>	<b>Results</b>	<b>39</b>
5.1	Algorithm Accuracy . . . . .	39
5.1.1	Coastline extraction using Aerial images . . . . .	40
5.1.2	Coastline extraction using Google Earth images . . . . .	43
5.1.3	Coastline extraction using Satellite images . . . . .	44
5.2	Algorithm Performance . . . . .	44
5.3	Framework Description . . . . .	45
<b>6</b>	<b>Conclusion</b>	<b>51</b>
6.1	Conclusion . . . . .	51
6.2	Future Work . . . . .	52
	<b>References</b>	<b>55</b>



# List of Figures

2.1	Aerial Image. . . . .	4
2.2	Sar Image. <a href="#">source</a> . . . . .	4
2.3	Lidar Image. <a href="#">source</a> . . . . .	5
2.4	Noisy images: (a) original image, (b) salt and pepper noise, (c) Gaussian noise. <a href="#">source</a> . . . . .	6
2.5	Median Square Kernel . . . . .	7
2.6	Results of $3 \times 3$ , $5 \times 5$ and $7 \times 7$ mean filters. <a href="#">source</a> . . . . .	8
2.7	Results of $3 \times 3$ , $5 \times 5$ and $7 \times 7$ median filters. <a href="#">source</a> . . . . .	8
2.8	Smoothing the (a)affected by Gaussian noise image with (b) a $7 \times 7$ Gaussian mask. <a href="#">source</a> . . . . .	9
2.9	Sobel Kernels . . . . .	12
2.10	Robert Kernes . . . . .	12
4.1	Block Diagram of the proposed methodology . . . . .	24
4.2	Filtering effect: (a) initial image, (b), (c) and (d) image after applying Gaussian, median and mean filters respectively . . . . .	26
4.3	Close-up view of images after applying the filtering approaches: (a) initial image, (b), (c) and (d) image after applying Gaussian, median and mean filters respectively . . . . .	27
4.4	Application of anisotropic diffusion into filtered image . . . . .	28
4.5	Representation of block division . . . . .	30

## LIST OF FIGURES

---

4.6	Blocks reflecting various cases of the bimodality procedure: (a) a block that contains mostly sea area, (d) a block that contains coastline area, (b), (e) presents the two Gaussian distributions modelling the distribution of the first and second image, respectively, (c) and (f) reflects the extracted mixture of two Gaussian distributions for the (a) and (d) image, respectively	32
4.7	Concatenation procedure.	34
5.1	Automated coastline extraction from aerial imagery (a) initial image, (b) filtered and enhanced image, (c) k-means clustering, (d) segmented image (e) region grouping and removing small objects	41
5.2	Estimation of the coastline and a close-up of the coastline area showing the results of the two methods applied on aerial image	42
5.3	Estimation of the coastline and a close up of the coastline area applied on another aerial image	43
5.4	Estimation of the coastline and a close-up of the coastline area applied on a Google Earth image	44
5.5	Estimation of the coastline using a Satellite image	45
5.6	The main process panel of the Framework. On the left is the original image. On the right, the two estimated coastlines are shown, blue refers to the first method (region segmentation) and red refers to the second method (active contours)	46
5.7	The green box shows the selected area (tennis court) from ktimatologio.gr, the Blue box shows the same area from aerial image. We can also see the text box where we import the real units and the distance.	47
5.8	The computed length of the estimated coastline	48
5.9	Computed perimeter.	49
5.10	Computed distance.	49
5.11	The coastline differences between 2003 and 2010. The dashed line refers to 2003 image, while the straight refers to 2010 image. Images was taken from Google Earth.	50

# List of Algorithms

1	Our Approach . . . . .	38
---	------------------------	----

## LIST OF ALGORITHMS

---

# Chapter 1

## Introduction

Coastal zones are among the most productive areas in the world. They offer a wide variety of valuable habitats and ecosystems which have always attracted humans and human activities. Currently, more than 200 million European citizens live near coastlines, stretching from the North-East Atlantic and the Baltic to the Mediterranean and Black Sea.

This intensive concentration of population and excessive exploitation of natural resources puts enormous pressure on our coastal ecosystems. This leads to biodiversity loss, habitat destruction, pollution, as well as conflicts between potential uses, and space congestion problems. The rapid economic growth of the coastal areas has driven the public and private sector to make investments in infrastructure. The urbanization and development of these areas have led to the construction of new roads, small villages, entertainment facilities and infrastructure for sports activities. The intense touristic development has led to the environmental downgrading of the areas near the waterline.

A noteworthy fact is that coastal areas are among the most vulnerable to climate change and natural hazards, marking them as fragile landscapes. Flooding, erosion, sea level rise, as well as extreme weather events, are some of the most common natural hazards we can encounter. As negative factors, we have human interventions and natural phenomena due to the climate change. A disappointing example is that around 30 percent of the Greek coastline is affected by erosion [1].

As a result coastline mapping and change detection are essential for safe navigation, resource management, environmental protection, and sustainable coastal development and planning.

### 1.1 Thesis Contribution

Computer Science in the fields of Image Processing and Computer Vision is being connected with a number of Scientific fields, including geophysical hazards. This thesis describes a new approach developed for the purpose of an automatic extraction of the coastline using edge detection and optimization procedures.

We present a new approach of a pre-existing estimation algorithm for coastline extraction based on Canny edge detection and locally adaptive thresholding methods introduced by Liu and Jezek [2]. Their background work is enhanced and altered in order to produce smoother and more reliable results. We have also added another aspect to this approach, the active contours fitting in order to optimize the results from the previous step and extract the final model of the coastline.

Furthermore, we developed an automated graphical tool suitable for identifying and extracting the coastline. Also, given a relative value to pixels, we are able to extract metric information about the area, size for objects of interest and provide spatial information for coastal management.

### 1.2 Thesis Outline

In Chapter 2 we provide all the background information needed for this thesis. We present the coastline definition as well as the major filtering and image segmentation techniques developed for addressing problem like coastline extraction. Furthermore, we provide basic information about the differences between images based on the source (aerial photography, SAR, LiDar). In Chapter 3 we state the coastline extraction and characterization problem and we refer to different approaches that have been developed so far categorized by the input data. In Chapter 4 we describe our robust method which is based on edge detection and active contours (snake method). In Chapter 5 the performance of the proposed approach is semi evaluated. Also, there is an extended representation of the features available on the developed graphical tool. Finally, Chapter 6 acts as an epilogue for this thesis, presenting our conclusions along with future improvements. Also, some ideas for upgrading the graphical tool in the future are being mentioned.

# Chapter 2

## Background

### 2.1 Coastline definition

The coastline is defined as the line of contact between water and land. It is easy to define but difficult to capture, due to the fact that the water level is always changing (tidal flooding, sea level rise, land subsidence). However, different entities in the mapping community have compiled and published coastline delineations that are based on different coastline definitions. In order to avoid this confusion, we presume that the coastline is the exact line that separates the land from water, which is the most popular definition from previous works.

### 2.2 Imagery Data Sources

There are a lot of practices we can deploy in order to get images containing coastal areas. The oldest way is aerial photography, by using an aircraft or other flying objects, where a mounted camera may be triggered remotely or automatically. The main advantage of this application is that it is readily obtained. Additionally, in the past it was the only way to capture landscapes, making them the only data source available in order to track coastal changes between decades. Nevertheless, the task of coastline extraction using aerial images can be a challenging one, because they usually suffer from noise distortion, deformations and lack of color.

## 2. BACKGROUND

---



Figure 2.1: Aerial Image.

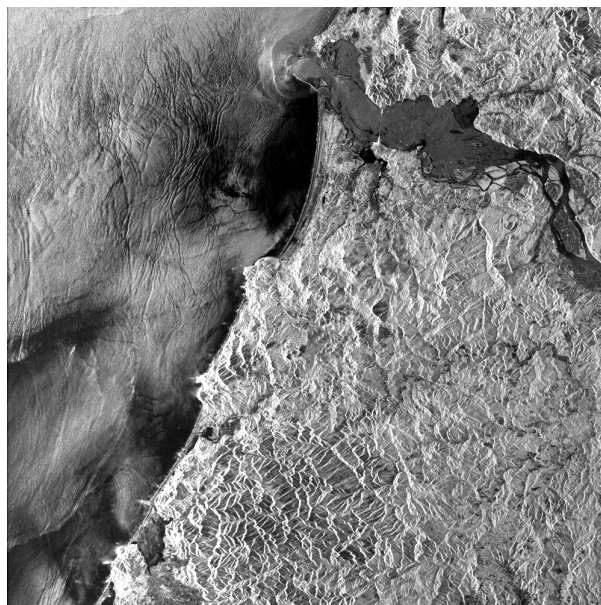


Figure 2.2: Sar Image. [source](#)

Another way to capture landscapes and furthermore coastal areas is using SAR (Synthetic-aperture radar) technology, where a radar deployed on a spacecraft is used



to create two or three-dimensional high-resolution images. SAR images have wide applications in remote sensing and mapping of the surfaces such as topography, oceanography, and geology. Hence, they still remain the main source of application dealing with coastal changes. This technology is widely used for several years and is relatively easy to find image data. On the other hand, it's quite expensive and difficult to find a specific area of study.

After the invention of the laser, a new way was developed based on them. Light Detection And Ranging (LiDar) or laser scanning is a surveying method that measures the distance to a target. This works by illuminating that target with a pulsed laser light and measuring the reflected pulses with a sensor. Differences in laser return times and wavelengths can then be used to make digital 3D-representations of the target. This method has application in geodesy, archaeology, geography, geology, and geomorphology because it can operate in the infrared spectrum, making it the easiest way of tracking coastline. But still, the technology at its current state is not targeted for the common household and it is only used for large-scale projects and analytic purposes.

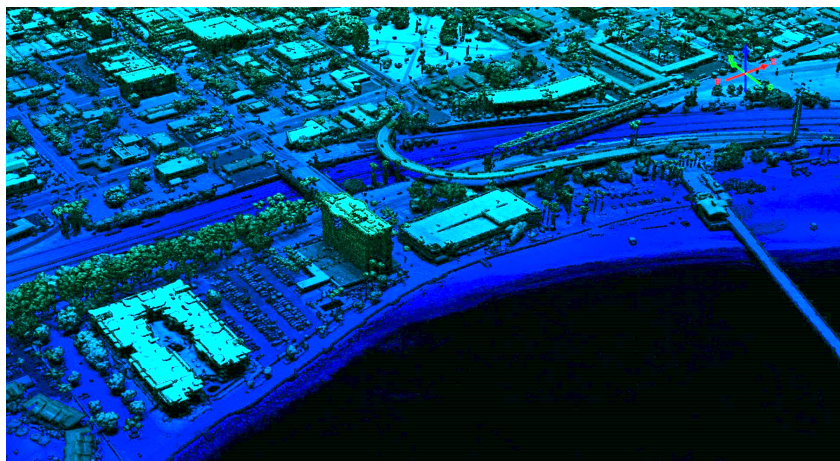


Figure 2.3: Lidar Image. [source](#)

## 2. BACKGROUND

---

### 2.3 Noise reduction

The aerial images captured are usually bound to suffer from Gaussian noise, salt and pepper noise and speckle noise. Therefore, obtaining an aerial image with high accuracy is a very difficult task. Speckle noise and non-uniform signal characteristics of the signals returning from the sea surface, can be mainly a problem for a SAR data and also can increase the difficulty of the image interpretation. An example of these noises affection in an image it is shown in Figure 2.4. There are a number of filtering techniques to detach the noise to preserve the integrity of captured image with the intention of providing more reliable results. The behavior of each of those noises and filtering is described below.

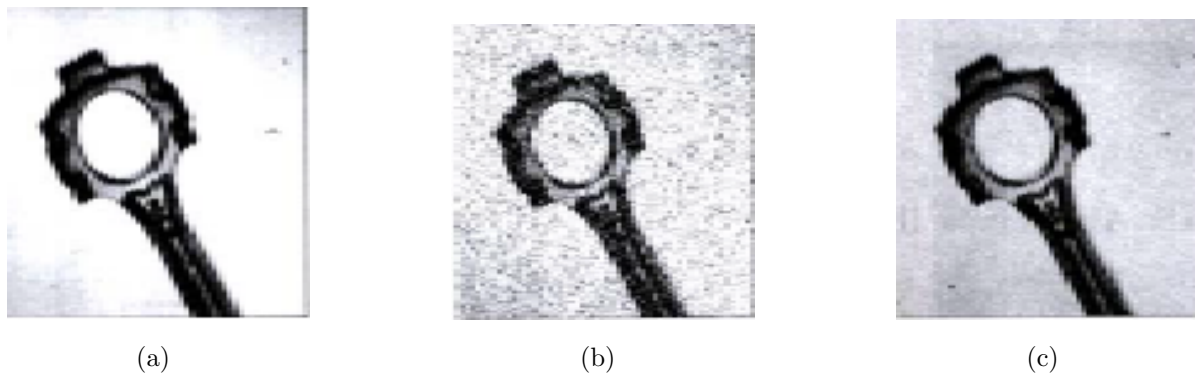


Figure 2.4: Noisy images: (a) original image, (b) salt and pepper noise, (c) Gaussian noise. [source](#)

#### 2.3.1 Gaussian Noise

The main feature of Gaussian noise is that it is evenly distributed over the signal. In particular, the noisy image pixel is the sum of the actual pixel value and a random Gaussian distributed noise value. This type of noise has a Gaussian distribution. The distribution function given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z - \mu)^2}{2\sigma^2}} \quad (2.1)$$

where  $z$ =gray level,  $\mu$ =mean/average of function,  $\sigma$ =standard deviation of noise and  $p(z)$ =probability density function.

### 2.3.2 Salt and Pepper Noise

Salt and pepper noise is caused by sharp, sudden disturbances in the image signal and its appearance is randomly scattered white or black pixels over the image. That means dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by dead pixels or on analog to digital conversion.

### 2.3.3 Speckle Noise

Generally, speckle noise is commonly found in synthetic aperture radar images and medical images. It is caused by coherent processing of backscattered signals from multiple distributed targets. Speckle noise in images have a multiplicative error and must be removed before the data can be used otherwise, the noise is merged into and degrades the image quality.

### 2.3.4 Filtering Techniques

A filter can be applied to a particular image with the aim to reduce noise and improve the visual and processing quality of the image. There is a variety of procedures able to handle the noise. In this section we are focused on the most known filtering techniques, that can deal with the noise affecting our image data.

- Mean Filter

Mean filtering is a simple and easy to implement method to reduce noise in images by reducing the amount of intensity variation between one pixel and the next. The idea of mean filtering is simply to replace each pixel value in an image with the mean or average value of its neighbors, including itself, which leads to the elimination of the pixel values that are unrepresentative of their surroundings.

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Figure 2.5: Median Square Kernel

## 2. BACKGROUND

---

A convolution is applied that is based around a kernel which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a  $3 \times 3$  square kernel is used, as shown in Figure 2.5, although larger kernels, for example,  $5 \times 5$  squares can be used for more severe smoothing. Figure 2.6 shows the results of noise reduction by applying the mean filter using  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  kernels on the image affected by salt and pepper noise.

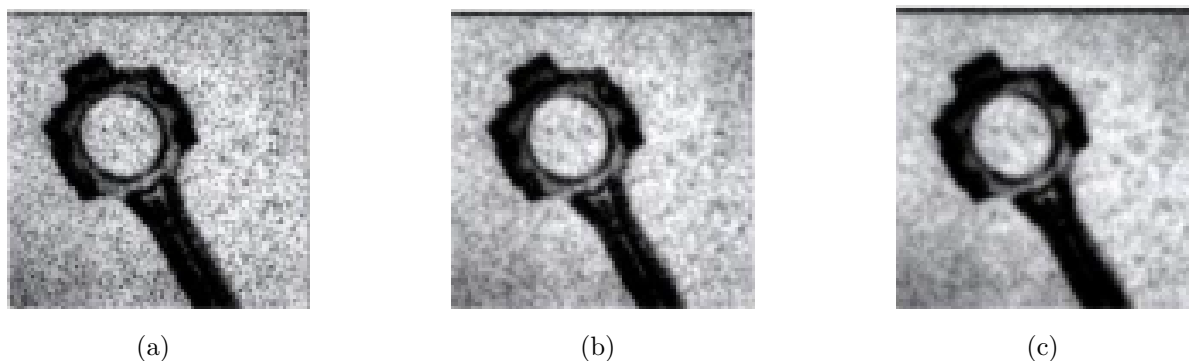


Figure 2.6: Results of  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$  mean filters. [source](#)



Figure 2.7: Results of  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$  median filters. [source](#)

- **Median Filter**

Median filter is a non-linear filter, that often does a better job than the mean filter of preserving useful detail in the image. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether it is representative of its surroundings. Instead of simply replacing

the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. Figure 2.7 shows the results of noise reduction by applying the median filter using  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  kernels on the image affected by salt and pepper noise.

- **Gaussian filter**

Gaussian filter is used to blur images, remove noise and detail and also it is a smoothing operator. It is a linear filter and highly effective when the image consists of Gaussian noise. When working with images we need to use the two-dimensional Gaussian function.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \quad (2.2)$$

where  $x$ =distance from the origin in the horizontal axis,  $y$ = the distance from the origin in the vertical axis and  $\sigma$ =standard deviation of the noise. Gaussian filter is relatively fast and a common first step in edge detection. Figure 2.8 shows the results of noise reduction by applying the Gaussian filter using a  $7 \times 7$  kernel on the image affected by Gaussian noise.

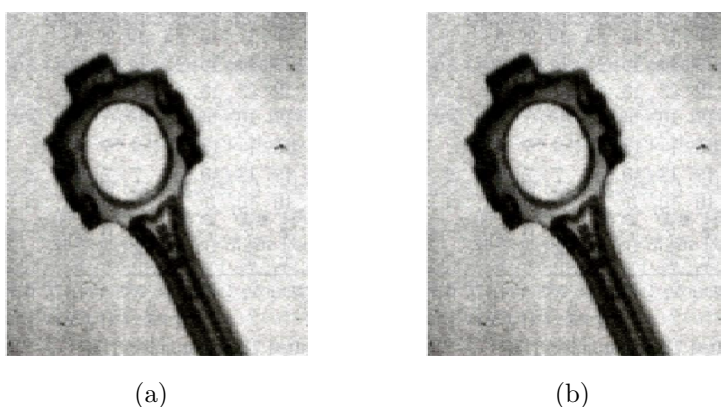


Figure 2.8: Smoothing the (a)affected by Gaussian noise image with (b) a  $7 \times 7$  Gaussian mask. [source](#)

## 2. BACKGROUND

---

- **Anisotropic diffusion**

Anisotropic diffusion, also called Perona and Malik diffusion [3], is a technique aiming at reducing image noise without removing significant parts of the image content, typically edges lines or other details that are important for the interpretation of the image. Anisotropic diffusion is a generalization of this diffusion process. More specifically, it produces a family of parameterized images, but each resulting image is a combination between the original image and a filter that depends on the local content of the original image. As a consequence, anisotropic diffusion is a non-linear and space-variant transformation of the original image. As a result, this technique's aim is to enhance strong edges, specifically in aerial coastal images, like coast area and land generally, or to suppress weak edges like the area of sea. The idea of this method is simple: let  $\Omega \in \mathbb{R}^2$  denote a subset of the plane and  $I(x, y, t) : \Omega \rightarrow \mathbb{R}^2$  be a family of grayscale image obtained by convolving the initial image with a Gaussian kernel. The anisotropic diffusion equation is defined as:

$$\frac{\partial I}{\partial t} = c(x, y, t)\Delta I + \nabla c \nabla I \quad (2.3)$$

where we indicate with  $\Delta$  the Laplacian operator, with  $\nabla$  the gradient operator and with  $c(x, y, t)$  the flux function which controls the rate of diffusion at any point in the image. Perona and Malik suggested the following flux functions:

$$c(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2} \quad (2.4)$$

$$c(\|\nabla I\|) = \frac{1}{1 + (\|\nabla I\|/K)^2} \quad (2.5)$$

where  $K$  controls the sensitivity to edges. The first function privileges high-contrast edges over low-contrast ones, while second function privileges wide regions over smaller ones. This means that they offer a trade-off between edge-preservation and blurring homogeneous regions. At this point, we highlight that in order to choose which of two functions will be applied, it is needed to run some experiments on the image data.

## 2.4 Image Segmentation

Image segmentation is the technique of dividing or partitioning an image into parts, called segments. Mostly, this technique is useful for applications like image compression or object recognition, because for these types of applications, it is inefficient to process the whole image. Ergo, image segmentation is used to divide the image into parts, for further processing. There exist several image segmentation techniques, which partition the image into several parts based on certain image features like pixel intensity value, color, and texture. We are focusing on edge detection and clustering based thresholding techniques as well as active contour techniques in order reduce the amount of data and filter out useless information while preserving the important structural properties in an image.

### 2.4.1 Edge Detection

Generally, edges are local changes in the image intensity. Edges typically occur on the boundary between two regions. The main features can be extracted from the edges of an image. Edge detection is a fundamental tool for image segmentation. In general, detection methods transform original images into edge images benefits from the changes of grey tones in the image. More specifically, detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. There are many edge detection techniques in the literature for image segmentation. The most commonly used edge detection techniques are reviewed in this section.

- **Sobel Edge Detection Operator**

The Sobel edge detection method is introduced by Sobel in 1970. The Sobel technique performs a 2-D spatial gradient quantity on an image and emphasizes regions of high spatial frequency that correspond to edges. Commonly, it is used to find the estimated absolute gradient magnitude at each point in an input grayscale image. The operator consists of a pair of  $3 \times 3$  convolution kernels as shown in the figure below.

## 2. BACKGROUND

---

-1	0	1	G <sub>x</sub>	1	2	1	G <sub>y</sub>
-2	0	2		0	0	0	
-1	0	1		-1	-2	-1	

Figure 2.9: Sobel Kernels

One kernel is simply the other rotated by 90°. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation ( $G_x$  and  $G_y$ ). The gradient magnitude  $G$  is given by:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

$$|G| = |G_x| + |G_y| \quad (2.7)$$

but using an approximation of the magnitude shown in equation 2.7 results to faster computations. The Sobel operator is generally slower to compute than the other operator, but its larger convolution kernel smooths the input image to a greater extent.

### • Roberts Edge Detection Operator

The Roberts edge detection method performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. Pixel values at each point in the output, represent the estimated absolute magnitude of the spatial gradient of the input image at that point and it is calculated in the same way as equation 2.7. The operator consists of a pair of  $2 \times 2$  convolution kernels as shown in figure 2.10. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator, but it is faster. On the other spectrum, it produces lower output values for similar edges.

1	0	G <sub>x</sub>	0	1	G <sub>y</sub>
0	-1		-1	0	

Figure 2.10: Robert Kernes



- **Canny Edge Detection**

The Canny edge detector is widely considered to be the standard edge detection method in the industry. It was first created by John Canny in 1983 [4], and still outperforms many of the newer algorithms that have been developed. The algorithmic steps for canny edge detection technique as follows:

1. Smooth the image with a two dimensional Gaussian.
2. Take the gradient of the image. This step help us to track the changes in intensity, which indicates the presence of edges.
3. Apply non-maximal or critical suppression to the gradient magnitude. Edges will occur at points where the gradient is at the maximum. Therefore, all points not at the maximum should be suppressed.
4. Apply threshold to the non-maximal suppression image, which is referred to as "hysteresis".

The algorithm is adaptable to various environments. It is an optimal edge detection technique as it provides good detection, clear response, and good localization. It is widely used in current image processing techniques with further improvements.

### 2.4.2 Segmentation using Clustering Based Thresholding

Thresholding is a special type of image segmentation. An image is segmented into its background and foreground by choosing the optimum threshold value,  $T$ . If the gray level or brightness of the pixel in an image  $I(x, y)$  is greater than the threshold value  $T$ , then it is a foreground pixel or otherwise it is a background pixel as shown below:

$$f(x, y) = \begin{cases} 0, & \text{for } I(x, y) \leq T \\ 1, & \text{for } I(x, y) > T \end{cases} \quad (2.8)$$

where  $f(x, y)$  is the binarized image and  $(x, y)$  is the pixel coordinate.

- **Otsu Method**

Otsu method is used to automatically perform cluster-based thresholding. It is a kind of image segmentation that involves iteration through all the possible values

## 2. BACKGROUND

---

of threshold and calculating a measure of spread for the pixel levels on each side of the threshold, i.e. the pixel either fall in background or foreground. This method is based on minimizing the variances within each of the two clusters. A threshold is selected that minimizes the within-class variances of two clusters. The within-class variance can be described as the weighted sum of the variances of each cluster and it is formulated as:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (2.9)$$

where  $q_1$  and  $q_2$  are the the probability of background and foreground class occurrences respectively separated by a threshold  $t$ ,  $\sigma_1^2$  and  $\sigma_2^2$  are the variances of these classes. The class probabilities are given as:

$$q_1(t) = \sum_{i=1}^t p(i) \quad (2.10)$$

$$q_2(t) = \sum_{i=t+1}^I p(i) \quad (2.11)$$

while the class means are given by:

$$\mu_1(t) = \sum_{i=1}^t \frac{p(i) * x(i)}{q_1(t)} \quad (2.12)$$

$$\mu_2(t) = \sum_{i=t+1}^I \frac{p(i) * x(i)}{q_2(t)} \quad (2.13)$$

where  $x(i)$  is the value at the center of the  $i$ th histogram bin. Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^t [x(i) - \mu_1(t)]^2 \frac{p(i)}{q_1(t)} \quad (2.14)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [x(i) - \mu_2(t)]^2 \frac{p(i)}{q_2(t)} \quad (2.15)$$

Now we can iteratively compute the threshold  $T$ ,  $1 \leq T \leq 256$  that minimizes the variance  $\sigma_w^2(t)$ .

- **K-means Clustering**

The most popular method regarding image clustering is K-means clustering algorithm. K-means clustering is an unsupervised algorithm and it is used to segment the interest area from the background. It classifies a given set of data into  $k$  number of disjoint cluster and consists of two separate phases. In the first, phase it calculates the  $k$  centroid and in the second phase, it takes each point to the cluster which has the nearest centroid from the respective data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done, it recalculates the new centroid of each cluster. Based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster, which have minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. So, K-means is an iterative algorithm in which the sum of distances from each object to its cluster centroid is minimized over all clusters. The threshold value is defined as the average of the  $K$ th class center and the  $(K - 1)$ th class center.

Although K-means has the great advantage of being easy to implement, it has some drawbacks. The quality of the final clustering results depends on the arbitrary selection of initial centroid.

### 2.4.3 Active Contours

Segmentation using active contours model (Snakes) was introduced by Kass et al. [5]. The idea behind active contours for image segmentation is quite simple. The user or a procedure specifies an initial guess for the contour, which is then moved by image driven

## 2. BACKGROUND

---

forces to the boundaries of the desired objects. In such models, two types of forces are considered; the internal forces and the external forces. The internal forces are defined within the curve and are designed to keep the model smooth during the deformation process. The external forces are computed from the underlying image data and are defined to move the model toward an object boundary or other desired features within the image. In order to obtain the contour deformations we need to minimize the energy functional which is formed by the internal and external forces and it is formulated as:

$$E_{snake} = E_{int} + E_{ext} \quad (2.16)$$

where  $E_{int}$  refers to internal forces and  $E_{ext}$  refers to external forces. A further analysis of these forces will form the final energy (cost) function.

$$E_{int} = \int_0^1 (\alpha |C'(s)|^2 + \beta |C''(s)|^2) ds \quad (2.17)$$

$$E_{ext} = \lambda \int_0^1 |\nabla I(C(s))| ds \quad (2.18)$$

And finally, the energy equation (2.16) can be expressed by:

$$E_{snake} = \int_0^1 (\alpha |C'(s)|^2 + \beta |C''(s)|^2 - \lambda |\nabla I(C(s))|) ds \quad (2.19)$$

where  $C(s) = (x(s), y(s))$ ,  $s \in [0, 1]$  is the initial curve in the original Image  $I$ , parameters  $\alpha$ ,  $\beta$  impose the elasticity and rigidity of the curve, parameter  $\lambda$  determines the level of the gradient regions that the curve will attract to and  $\nabla I$  is the gradient of the image intensity. As we said before, this active contour method relies on an energy minimization technique. So now, using an iterative procedure we are able to minimize the energy function and export the final curve.

# Chapter 3

## Problem Statement and Related Work

### 3.1 Problem Statement

Methods of detecting the coastline regardless of the image data have been one of the intensive topics in image processing. Coastline extraction can be especially difficult due to fact that the coastal area is a highly dynamic environment with many physical processes. Additionally, images can be inevitably distorted by noise due to weather conditions such as clouds, winds, sun reflection and shadows. As a result, a noise reduction is usually applied using noise removal filters. Afterwards, a water/land separation needs to be adapted in order to form the coastline. This is achieved either with region-based segmentation or edge-based segmentation. The development of coastline detection and extraction procedures can be semi-automated, meaning changing parameters depending on the image, or fully automated.

### 3.2 Related Work

To begin with, a lot of coastline detection algorithms have been applied based on the SAR imagery, because as already said they are easily found and there is a variety of landscape areas. In contrast, aerial images have been mainly used in earlier works, before the satellite imagery development. Although LIDAR images provide a more sufficient

### 3. PROBLEM STATEMENT AND RELATED WORK

---

information comparing to the other technologies, it is not widely used considering the high cost. Also, a semi-automated workflow is followed by many of these approaches in order to handle potential failures caused from the image quality. Those potential failures are the underlying reason the automated procedures are not preferable. Below, we present an overview of the most notable coastline detection methods categorized by the image data used.

#### 3.2.1 SAR Imagery Based Approaches

Over the recent years, satellite imagery has been used in semi-automatic or automated coastline extraction and mapping, especially in Europe, since the European Space Agency (ESA) is recognized as a world leader in SAR missions.

Erteza [6] published an automated approach which comprises three main algorithms. Firstly, a pre-processing algorithm that accentuates the water/land boundary in order to make a robust contour. This is accomplished by applying a median filter, histogram equalization, and dilation techniques on a grayscale image. Thereupon, the second algorithm automatically follows the water/land boundary and produces a single-pixel-wide coastline using the clockwise direction, following the algorithm developed by Duda and Hart [7]. Eventually, the algorithm ends up identifying the islands and then marking them. However, this particular step of the approach does not interest us in this project. To sum up, this procedure is quite fast, yet it lacks of robustness and it is only been tested in the Alaska coast.

Moreover, using multispectral images made the final estimation of the coastline more reliable. Braud and Feng [8] firstly and Frazier and Page [9] subsequently, evaluated threshold level slicing and multispectral (especially the Band 5) image classification techniques for detection of the coastline. The multispectral images provide reliable coastline estimations, but they also need different, sometimes faster approaches compared with panchromatic or grayscale images(one band). The main disadvantage of this type of images is the cost comparing to one band or panchromatic images.

Dellepiane et al. [10] introduced an innovative algorithm able to semi-automatically extract the coastline, using an InSAR (Interferometric Synthetic Aperture Radar) couple. This method has three main steps. At first, a coherence image is extracted using two

InSAR images  $I_1$  and  $I_2$  by applying the classical formula that involves the use of the statistical expectation operator  $E$ :

$$\Gamma = \frac{E[I_1 * I_2]}{\sqrt{E|I_1|^2 * E(|I_2|^2)}} \quad (3.1)$$

A second pre-processing step is followed in order to face the presence of texture, that generates the homogeneous average image. This step consists of the extraction of the average image, by applying a simple moving average filter. Then a segmentation process applied in parallel to the coherence image and the average image, resulting in a weighted connectivity map that is thresholded to extract the coastline using the following equation:

$$\begin{aligned} conn_{weighted}(p) &= W_\Gamma * conn(\Gamma, \alpha, p) \\ &\quad + W_T * conn_T(T, \alpha, p) \\ W_\Gamma, W_T &= (1 - W_\Gamma) \in [0, 1] \end{aligned} \quad (3.2)$$

where  $\Gamma$  and  $T$  means respectively grey level and textural average. Finally, a simple interactive thresholding step allows the identification of the most suitable threshold to extract the coastline from the image.

Modava and Akbarizadeh [11] developed a combination of two algorithms, the image classification and the active contours. Firstly, a fuzzy c-means clustering [12] [13] is applied to the input SAR image partitioning the image into land and water areas. Subsequently, a binarization of the resulted image is followed in order to initialize the level set function [14], a geometric model of active contours. The extracted coastline is one-pixel in thickness and is a continuous line. The advantages of this study include the fact that the contours represented by the level set function, can be split and merged naturally during evolution. But because the sea does not usually have a uniform grey level, this approach is inefficient in greyscale images. Also, although the final contours will become more precise as the iterations increase, there is a trade-off between speed and accuracy.

### 3. PROBLEM STATEMENT AND RELATED WORK

---

#### 3.2.2 LIDAR imagery Approaches

LIDAR (Light Detection and Ranging) is the laser scanning sensing system used for receiving high precision spatial data. The working principle of LIDAR is to measure the time of the laser signals, send from the laser scanning system and reflected from the object's surface, to go back to the receiver. Compared to terrestrial measurement methods and aerial photogrammetry, LIDAR requires less field work and has a lower evaluation cost. But the high cost makes LIDAR image only available on large-scale projects. Thus, only a few works have been developed based on this technology in order to estimate the coastline position.

Stockdon et al. [15] developed the cross-shore profile method, which consists of bands of LIDAR data 2–4 meters wide in the along-shore direction and spaced every 20 meters along the coast. Along each profile, a least squares linear regression line is passed through the cluster of data that encompasses the operational mean high water (MHW) datum and is limited to the seaward-sloping beach foreshore. The regression equation is then used to derive the horizontal intersection of the operational MHW datum with the profile, giving the coastline position for that profile. The shoreline points from individual profiles are then linked to represent the final coastline.

Using the contouring procedure, Robertson et al. [16] developed a method that subtracts the desired tidal datum from a LIDAR digital elevation model (DEM) and values of zero are obtained for the water line. These are contoured to derive the tidally referenced coastline. Manual editing and redigitizing are usually required to eliminate erroneous coastline and to keep and refine true coastlines.

Liu et al. [17] proposed a segmentation based image processing method to automate coastline extraction from airborne LIDAR data. It starts with an intersection of the LIDAR DEM and tidal datum surface in order to segment the binary version of the image into water and land pixels. A set of image processing algorithms including region grouping and labeling, two passes of image region scanning, a mathematical morphology operation, line tracing and vectorization is sequentially applied in order to extract the coastline. The accuracy assessment based on Monte Carlo simulation determines the confidence interval and summary error statistics for the horizontal position of each coastline section.



### 3.2.3 Aerial imagery Approaches

Aerial imagery was the only source of environmental management until the SAR and LIDAR technologies were developed. Thus, aerial photographs were the most commonly used data source in coastline mapping and also they give us access to historical coastline estimations. Distinguishing land and water in grayscale or black and white aerial photographs, especially where the waters are blurry and muddy, requires extensive terrestrial efforts. Also, because of the fact that aerial photographs were not digital, the extraction of information is difficult and causes mistakes. The main works that deal with the mentioned problems are presented below.

To start off with, automatic extraction of the shoreline features from aerial photos has been investigated using neural networks and image processing techniques. Ryan et al. [18] used an image segmentation method to approach the coastline extraction problem and tested their method on several small portions ( $256 \times 256$ ) of scanned US Geological Survey (USGS) aerial photographs. They applied a neural network method on a texture measure of the images to separate land and water regions. The use of texture influenced the accuracy and the need of training neural nets set this application as limited to a large volume of full-scene image data.

Chalabi et al. [19] developed a coastline extraction method based on segmentation techniques (Pixel-based-segmentation or grey-level thresholding) applied on satellite and aerial images. This method is known as a process that segments an image based on digital number (DN) threshold [20]. A classification method was then conducted, which was used to classify similar pixel values in a group and give them a name using unsupervised classification. Recode was set up by choosing two attributes, 1 for the classes with a value of 1 and all other classes to 0. The conversion from raster to vector was successful after recoding. The shoreline then was exported to a GIS environment and vectorized using ArcInfo software. As a result, it is possible to map coastline erosion through years, comparing not only data coming from aerial photographs, but also data exported from satellite imagery.

Lue and Jezek [2] introduced a comprehensive approach for an automated extraction of the coastline from satellite imagery, but is also applied in aerial image data. It consists of a sequence of image processing algorithms, in order to reduce noise and enhance major edges. Subsequently, a segmentation step is applied based on a locally adaptive

### 3. PROBLEM STATEMENT AND RELATED WORK

---

thresholding technique, where the image is divided into blocks with the aim of finding a local threshold for each block according to the local characteristics. This results into a binary class image consisting of homogeneous land (with a pixel value of 1) and water (with a pixel value of 0) regions. Finally, post-processing includes removal of small objects in order to separate true coastline edges from other object boundaries, such as ships or icebergs in the sea and morphological operation of dilation followed by erosion, to make the coastline morphologically smoother. Now, it is possible to create a raster image of the coastline by scanning the image objects and finding the boundary pixels. This method has reliable results but the overall success depends from the quality of the image sources.

Yousef and Iftekharuddin [21] applied a fusion of aerial images and LIDAR digital elevation models (DEM) data using mutual information and genetic algorithms for the purpose of coastline extraction. The LiDAR data and the aerial image fused by maximizing their mutual information through the genetic algorithm [22] assuming both images have the same spatial coverage, the same spatial resolution, and georeferenced to a common grid. Additionally, support vector machines (SVM) classify the fused data and segment it into land and water pixels. This approach can work without reference to a tidal datum, as described on LIDAR imagery Approaches and can extract the MHW shoreline. It was shown that this approach yields excellent results comparing them to a ground truth, but we should mind the cost of the LIDAR image data.

# Chapter 4

## Our Approach

In this chapter, we will present our approach for the topic of automatic coastline extraction, by introducing a new version of the method presented by Lue and Jezek [2] and it relies on edge detection and optimization procedures. The benefit of our approach, besides being computationally more efficient, is that it provides a more reliable and robust outcome.

Our methodology follows two main stages. The first one aims to provide an estimation of the coastline based on region segmentation techniques using a local thresholding method, while the second stage comes to optimize the results of the first approach and model the coastline based on active contours. The proposed method is implemented as a 4-step algorithm: (1) pre-processing, (2) region segmentation, (3) post-processing and (4) coastline modeling. More specifically, the pre-processing step is used to achieve noise reduction and edge enhancement in the image. The region segmentation divides the image into blocks of different content in order to provide a threshold based on local characteristics. At the post-processing step, we apply morphological operations and edge detection to extract the coastline. Finally, at the coastline modeling step we implement an active contour method to optimize the coastline result from the previous steps.

The two detection schemes are used in a complementary way, where the first (initial) estimate is optimized through the use of active contours fitted to the structure of the image under consideration. Alternatively, the two detection schemes can be viewed as independent approaches to derive the meaningful edges, so as to compare the results and the potential of different processing philosophies.

## 4. OUR APPROACH

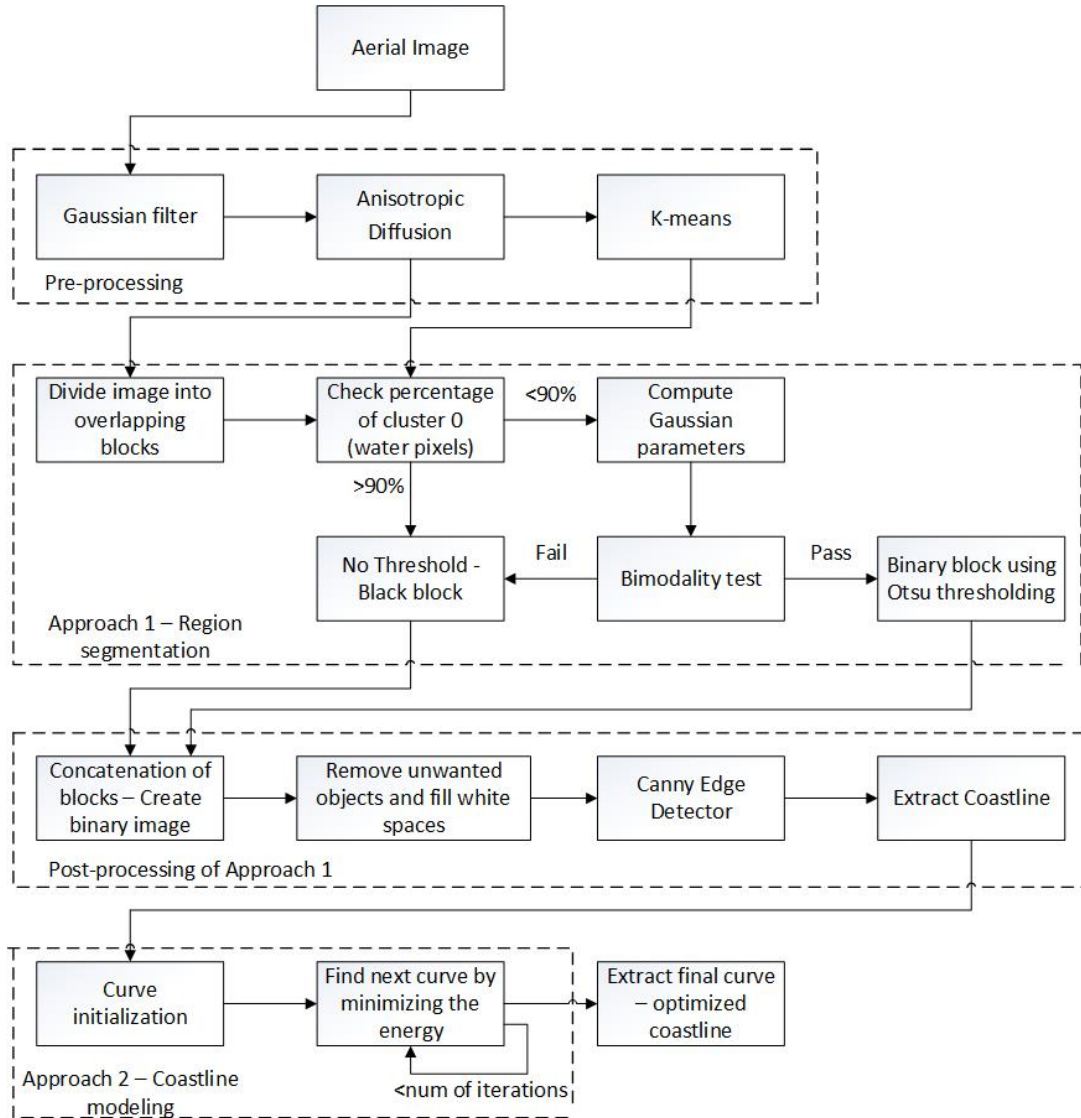


Figure 4.1: Block Diagram of the proposed methodology

A general scheme of the proposed methodology is presented at the block diagram in figure 4.1. The proposed method uses input image data that are provided by the Hellenic Military of Geographical Service (HMGS) [23] or even SAR images taken from Google Earth [24].

## 4.1 Pre-processing

Two primary procedures are applied at the pre-processing step so that it is possible to perform the image segmentation. The first procedure developed to filter the image noise to reduce noisy edges in the subsequent image segmentation. The second procedure developed to enhance the edges along the coastline while suppressing other unimportant edges inside land or sea area.

- **Noise Reduction**

Aerial photos can provide a good spatial coverage of the coast but they also are inevitably distorted by noise. A Gaussian filter is applied to reduce noisy parts without blurring the major edges. Other filters, such as Median, Mean and Sobel filter may be used to replace the Gaussian filter, but it will be proved that Gaussian filter provides the best results for our image data.

- **Edge enhancement**

Another issue we should carry out is the deformation of the image by unexpected factors, like sunlight, shadows or clouds, which can lead us to wrong results. For these reasons, we apply the anisotropic diffusion algorithm as presented by Perona and Malik [3]. This algorithm's aim is to enhance strong edges along the coastline or to suppress weak edges inside the sea or land area.

Figure 4.2 shows the effect of the filtering step using Gaussian, Median and Mean filters. Applying Gaussian filter is the most efficient filtering method in our images. It removes the noise, as we can see more clearly in figure 4.3, while keeping the image smooth. On the contrary, although the median and mean filters remove the noise, they blur the image edges, especially in the coastline area. Visually it is not very clear, but in practice, blurring the edges will lead us to wrong estimations. Also, anisotropic diffusion reduced weak edges, while strong edges along the coastline are enhanced as shown in figure 4.4.

## 4. OUR APPROACH

---



(a) Initial image



(b) Gaussian filtering



(c) Median filtering



(d) Mean filtering

Figure 4.2: Filtering effect: (a) initial image, (b), (c) and (d) image after applying Gaussian, median and mean filters respectively



(a) Initial image



(b) Gaussian filtering



(c) Median filtering



(d) Mean filtering

Figure 4.3: Close-up view of images after applying the filtering approaches: (a) initial image, (b), (c) and (d) image after applying Gaussian, median and mean filters respectively



## 4. OUR APPROACH

---



Figure 4.4: Application of anisotropic diffusion into filtered image

### 4.2 Region Segmentation

The second step of our methodology is the region segmentation. We start with the image segmentation into two groups-regions; land and sea. The pixels that belong to the borders of these two regions compose the coastline. We use local thresholding in order to separate the land objects from the water background, because a global threshold would lead to false estimations as a result of the image intensity heterogeneity. Therefore, we developed a region segmentation algorithm using local thresholding in order to find thresholds based on local characteristics. Below we present analytically each phase of this step.

#### 4.2.1 K-means Clustering

The first phase includes a clustering of the image using K-means algorithm. K-means clustering treats each object of the image as having a location in space. It finds partitions



such that objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. K-means clustering requires that you specify the number of clusters to be partitioned and a distance metric to quantify how close two objects are to each other. In our work we choose to have 3 clusters and use the Euclidean distance metric. More specifically, cluster 0 corresponds to sea pixels, while cluster 1 and cluster 2 corresponds to land pixels. We ended up choosing 3 clusters by taking into consideration experimental results on a variety of image data. Less or more clusters will eventually lead into producing inhomogeneous regions.

### 4.2.2 Block Division and Local Thresholding

The purpose of this phase is to divide the image into small blocks (regions) and find a threshold adjusted to each of them. The motivation of this strategy is that some of the blocks will contain entirely sea or land pixels, so there is no need for any processing of such blocks. In particular, we divide the image into square overlapping blocks of width 'w', large enough for each block to cover the necessary spatial information to reflect the appropriate edge structure for the targeted application. Based on many experiments on large-sized images ( $4000 \times 4000$  pixels) we decided that the best value of width 'w' is 200–300. In general, we should choose 5-8% of the total image size. Assuming the image has  $r$  rows and  $c$  columns, the total number of blocks will be  $2 \cdot r/w$  and  $2 \cdot c/w$  in the vertical and horizontal direction respectively, taking into consideration the 50% overlapping blocks. In figure 4.5 it is shown a sample of how the overlapping blocks applied to an image. Each rectangular refers to a pixel of the image while colored rectangulars refers to blocks.

### 4.2.3 Local thresholds by fitting a bimodal Gaussian curve

For each block, we examine the bimodality and determine a local threshold value to separate the land pixels from the water (background) pixels. In order to reduce processing time, for each region, we first examine the percentage of zero values after clustering based on K-means procedure. We found that if a block has a total number of zero pixels above of 90 percent of the whole image, it does not need to be processed. This is due to the fact that these blocks contain mostly water area and maybe waves or other masses on the sea surface.

#### 4. OUR APPROACH

---

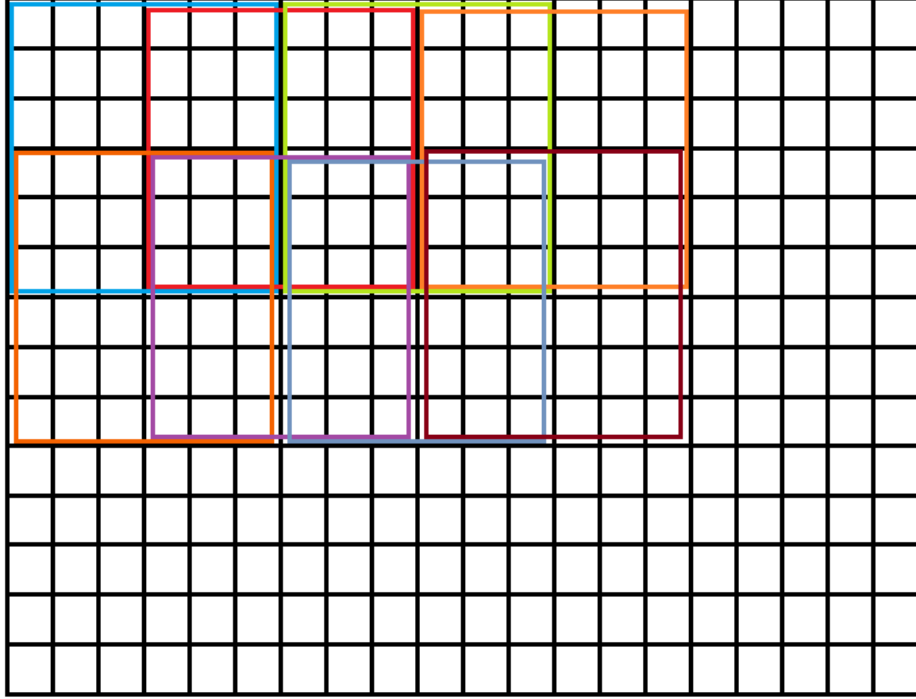


Figure 4.5: Representation of block division

For the remaining blocks, we examine the bimodality. More specifically, we assume that there are two Gaussian distributions of the intensity values, one for the description of the water and the other for the land. Their combination generates a Gaussian mixture model describing the observed intensity values of the image. If a block contains mostly water or land area pixels, the mixture of the distribution will only reflect one peak (unimodal). On the other hand, if a block is part of the coastline area, the mixture of distributions will consist of one valley and two peaks (bimodal). To be able to identify whether we have a bimodal or unimodal distribution, we follow the procedure as described below.

The probability density of a Gaussian distribution  $F(x)$  is given by:

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \mu)^2}{2\sigma^2}} \quad (4.1)$$

where  $x$  is the gray level,  $\mu$  mean of the distribution and  $\sigma$  stands for standard deviation. The probability density function  $p(x)$  of a mixture of two Gaussian distributions  $F_1(x)$  and  $F_2(x)$  is:

$$p(z) = p_1 F_1 + p_2 F_2 \quad (4.2)$$

and

$$p_1 + p_2 = 1 \quad (4.3)$$

Combining equations 4.1 and 4.3 we result in the following equation:

$$p(x) = \frac{p_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + \frac{1-p_1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \quad (4.4)$$

where  $\mu_1$  and  $\mu_2$  is the mean of the two distribution,  $\sigma_1$  and  $\sigma_2$  are the standard deviations and  $p_1$  is the coefficient of the mixture, which shows the ratio between the total number of sea area pixels and total size of blocks in pixels. Then we used a simple Expectation Maximization (EM) of Gaussian Mixture Model (GMM) algorithm [25] in order to compute the above 5 parameters of the mixture of the Gaussian distributions. At the first step of this algorithm, the Expectation obtains expected values for the missing data using an initial parameter estimate based on K-means and then follows the Maximization step where it is obtained the maximum likelihood estimation of the parameters using the estimated data. This procedure will repeat until the estimate converges.

Having Gaussian distributions parameters computed, we are able to test the bimodality by using 3 criteria. Firstly we need to compute the valley-to-peak ratio of the total histogram of the block:

$$\delta = \frac{\text{Min}\{\check{p}(i)\}}{\text{Min}\{\check{p}(\mu_1), \check{p}(\mu_2)\}} \quad (4.5)$$

where  $\check{p}$  is the Gaussian curve fitted in the previous steps and  $\mu_1 < i < \mu_2$ . In our applications we want the valley-to-peak ratio  $\delta < 0.9$ . Secondly, the means of the two component distribution must differ by more than 10 grey levels.

## 4. OUR APPROACH

---

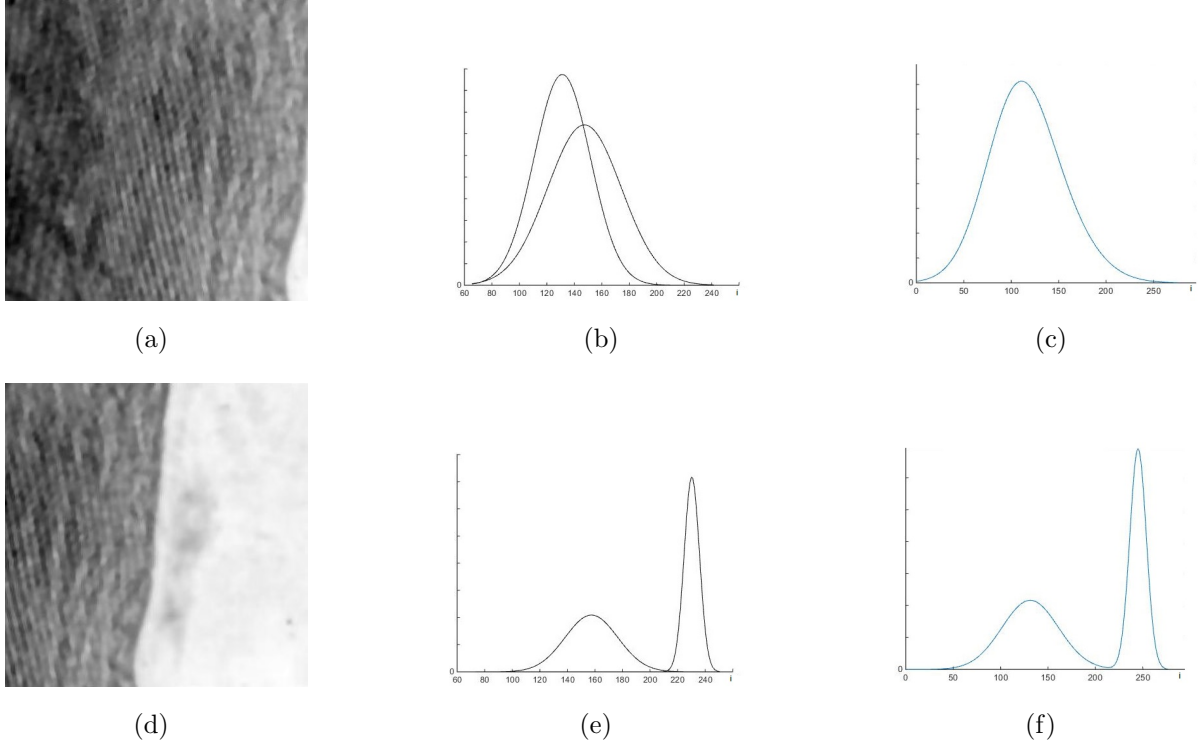


Figure 4.6: Blocks reflecting various cases of the bimodality procedure: (a) a block that contains mostly sea area, (d) a block that contains coastline area, (b), (e) presents the two Gaussian distributions modelling the distribution of the first and second image, respectively, (c) and (f) reflects the extracted mixture of two Gaussian distributions for the (a) and (d) image, respectively

Lastly, the Ashman's D [26] parameter is required to be greater than 2 for a clean separation of two distributions. In equation 4.6 it is shown that Ashman's parameter is a combination of means and standard deviations.

$$D = (2^{\frac{1}{2}}) \frac{|\mu_1 - \mu_2|}{\sqrt{(\sigma_1^2 + \sigma_2^2)}} \quad (4.6)$$

A block that would pass all three criteria is shown in figure 4.6, image (d) where as we can see it consist of a coastline area, while a block that would fail at the first criteria is also shown in figure 4.6, image (a), an image that mostly consist of sea area.

For every block that passes the bimodality tests, a threshold T is calculated by using the Otsu's method [27]. This method automatically performs clustering-based image

thresholding of a grayscale image by maximizing the weighted between-class variance. The algorithm follows the assumption that the block contains two classes of pixels (land, sea) through bimodal histogram (already established from the previous step) and then calculates the optimum threshold that classifies the two classes so that their combined spread is minimal. The weighted between-class variance is expressed, as we already showed in chapter 2, as:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (4.7)$$

where weights  $q_i$  are the probabilities of the two classes separated by a threshold  $t$ , and  $\mu_i$  denote the means of these classes. Using an iterative process for every  $t$  in  $1 \leq T \leq 256$ , we can compute the threshold that minimizes the between class variance. We apply the threshold to each block of the image and create a binary one following the procedure:

$$b(x, y) = \begin{cases} 0, & \text{for } I(x, y) \leq T \\ 1, & \text{for } I(x, y) > T \end{cases} \quad (4.8)$$

where  $T$  is the threshold of the block and  $I(x, y)$  is the intensity value of image pixel at points  $x, y$ . For every block that fails to pass the bimodality tests, a binary image is created with zero values at every point.

### 4.3 Post-Processing

At the last step of our first approach, we concatenate the blocks in order to recreate the image in a binary class form. The final value of each pixel is calculated by following the logic 'OR' condition between overlapping areas, which assigns the value 1 for the final pixel if and only if one of the overlapping blocks is 1 at this point. To explain why we use this procedure, we remind that blocks with over 90% of pixels in the sea area do not pass the bimodality test. However, in extreme cases, a small part of the coastline may be involved in such a block. It is expected that a neighboring block will identify the coastline because it will contain both parts of the water and land area. The use of the proposed conjunction condition aims to protect such edge pixels across the coastline that appears only at a portion of overlapping blocks. In Figure 4.7 we can see an example of

## 4. OUR APPROACH

---

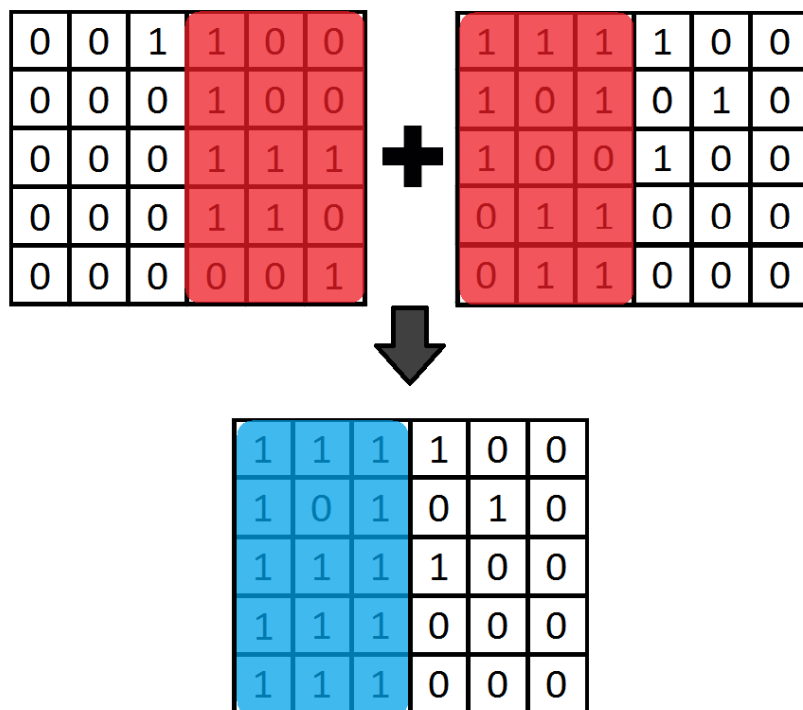


Figure 4.7: Concatenation procedure.

the concatenation procedure between two blocks and the produced block. This procedure continues until the input image is fully recreated in the binary form.

The binary form of the image is followed by the removal of false positives. Waves, ships or other objects on sea surface can lead us to false edge recognition in this area. To eliminate these effects, we apply the morphological operations of erosion and dilation. An erosion procedure followed by a dilation one removes small objects, while preserving the original abstract shape of the image. The structure element  $H$  that we use for the morphological operations is specified in a way that stray foreground structures larger than  $H$  must be preserved. Furthermore, we apply a closing operation, which is a dilation followed by an erosion. This operator fills black holes smaller than the structure element  $H$ , but also keeps the original shape of the image. Also, the binary form of the image is followed by a region grouping, in order to code as 0s the water pixel and 1s the land pixels. This will produce two large continuous land and water objects.

Finally, we use a simple edge detection operator to detect edges in the image. Notice

that at this stage of processing, the binary image involves a white area that belongs to land and a black area that belongs to sea, so that the boundary line refers to the coastline. We used the Canny edge detection method, which operates in 3 steps: (a) find the intensity gradient of the image, (b) apply non-maximum suppression, where pixels that are not considered to be part of an edge are removed and (c) hysteresis is applied to eliminate gaps. The result constitutes the first estimation of the coastline.

## 4.4 Coastline modeling

The estimated coastline in most of the cases needs improvement because of some random factors across the coastline, such as people, waves or buildings, which cannot be captured correctly by the previous steps and may produce wrong estimations. In order to deal with these problems, we adopt an open active contour method, which is based on snakes and extends the classical active contour model [5] with the condition of free boundary conditions. We initialize this curve using the already extracted coastline, obtained from the first approach. More specifically, a snake is an energy-minimizing spline that consists of an initial closed contour  $C_0$  near to a contour in the image and searches for deformations of  $C_0$  which move it towards the actual image contour. To obtain these deformations we need to minimize the energy functional formed by two energy terms regarding the intensity distribution of the image around the edge and reflect the internal and external regions of the edge itself. This function can be expressed as has already been shown in chapter 2 by:

$$E_{snake} = \int_0^1 (\alpha |C'(s)|^2 + \beta |C''(s)|^2 - \lambda |\nabla I(C(t))|) dt \quad (4.9)$$

where  $C(s) = (x(s), y(s))$ ,  $s \in [0, 1]$  is the initial curve in the original Image  $I$ , parameters  $\alpha$ ,  $\beta$  impose the elasticity and rigidity of the curve, parameter  $\lambda$  determines the level of the gradient regions that the curve will attract to and  $\nabla I$  is the gradient of the image intensity.

In most images of our case study, the land is separated from the sea by a vertical or horizontal curve positioned from the top to the bottom or from left to right respectively of the image, so that the endpoints of coastline are parts of the image margins. Thus, our approach needs to handle the case of an open curve. There are three types of open

## 4. OUR APPROACH

---

active contour characterized by their boundary conditions, namely 1) fixed boundaries, where the curve has two end points that cannot change through the curve evolution, 2) no boundaries, where there are no endpoint and the final curve may be deflected and 3) free boundaries, where the two endpoints will belong to two curves (boundary curves). We adopt Shemesh and Ben-Shahar's method [28] to develop a process using active contours with free boundary conditions. By default, the boundary curves will lie on the top and the bottom of the image, based on our image data, but they can manually be changed in order to fit every possible circumstance.

To be more specific, firstly, we need to initialize the curve  $C(s)$ . This will happen automatically by extracting the coastline based on region segmentation. Then, assuming the energy functional of the curve  $C(s)$  is limited to first order derivatives, it can be generalized as:

$$E_{snake}(C(s)) = \int_0^1 H(x, y, x', y') ds \quad (4.10)$$

where  $H$  refers to both internal and external term and can bring on the below Euler-Lagrange equations:

$$H_x - \frac{d}{ds} H'_x = 0 \quad H_y - \frac{d}{ds} H'_y = 0 \quad (4.11)$$

Now going back to equation 4.4, for simplicity we assume that  $\beta = 0$  and applying equation 4.11 on it will produce these two Euler equations:

$$-2\alpha x'' + \frac{\partial}{\partial x}(-\lambda|\nabla I(C(s))|) = 0 \quad (4.12)$$

$$-2\alpha y'' + \frac{\partial}{\partial y}(-\lambda|\nabla I(C(s))|) = 0 \quad (4.13)$$

which can provide us a complete equation of calculating the curve  $C(s)$  over the time:

$$C(s, t) = -2\alpha C'''(s, t) + \nabla g(s, t) \quad (4.14)$$



where  $C(s, t)$  represents the curve at  $t$ -th iteration and  $g(s, t) = -\lambda|\nabla I(C(s))|$ . We also know that the two end points of curve are restricted to lie on two boundary curves. Let  $B_0 : [0, 1]$  be the first boundary curve and  $q$  be the parameter value where  $C(s)$  crosses the boundary curve. The expressions that show us the restrictions for the first endpoint ( $q_0$ ) are :

$$(H_{x'}, H_{y'})|_{s=0} * (X'_0, Y'_0) = (H_{x'}, H_{y'})|_{s=0} * B'_0|_{q_0} \quad (4.15)$$

where  $H_{x'} = H_{x'}(x, y, x', y')$ ,  $H_{y'} = H_{y'}(x, y, x', y')$ ,  $s = 0$  is the first point of the curve  $C(s)$  and  $X_0, Y_0$  are the coordinates of the curve  $B_0$  at point  $q_0$ . Respectively, for the second endpoint

$$(H_{x'}, H_{y'})|_{s=1} * (X'_1, Y'_1) = (H_{x'}, H_{y'})|_{s=1} * B'_1|_{q_1} \quad (4.16)$$

Now, by applying equations 4.15 and 4.16 to  $H$  functions, give us the below restrictions which are necessary to be verified in order to keep the endpoints at boundary curves.

$$2\alpha(x', y')|_{s=0} * (X'_0, Y'_0) = 0 \quad (4.17)$$

$$2\alpha(x', y')|_{s=1} * (X'_1, Y'_1) = 0 \quad (4.18)$$

As we said previously, in our method we use one boundary curve on the top and one on the bottom of the image. We followed this procedure because we wanted an automated method of coastline extraction and the image dataset provides us this facility. We can now estimate iteratively the next possible position of the curve (coastline) by minimizing the energy of the snake, for 150 iterations. The number of iteration is based on experiments for our image dataset.

## 4. OUR APPROACH

---

---

**Algorithm 1** Our Approach

---

```
1: Read image I
2: if I == RGB then
3:   I = rgb2gray(I)
4: end if
5: Apply Gaussian filter to I
6: Apply anisotropic diffusion to I
7: Apply K-means algorithm to I
8: Divide image I into overlapping blocks (blk)
9: for  $blk = 1$  to  $blk = N$  do
10:  if percentage of cluster in the blk 0 > 90 then
11:    Create a binary block blkbinary with 0 value to all pixels
12:  else
13:    Compute Gaussian parameters
14:    Compute valley-to-peak ratio
15:    Compute deference between the two means of the distributions
16:    if Bimodality test = pass then
17:      Create a binary block blkbinary using otsu thresholding
18:    else
19:      Create a binary block blkbinary with 0 value to all pixels
20:    end if
21:  end if
22: end for
23: Concatenate blkbinary1 to blkbinaryN using 'OR' condition to create binaryI
24: Remove small objects from binaryI
25: Apply morphological operators to binaryI
26: Region grouping in from binaryI
27: Apply Canny edge detector to binaryI to extract the coastline  $C_0$ 
28: Initialize active contour curve C using  $C_0$ 
29: for  $iteration = 1$  to  $iteration = 150$  do
30:   Find next possible position of C by minimizing the energy
31: end for
32: Visualize the optimized coastline C
```

---

# Chapter 5

## Results

In this chapter, we present the results of our work. First, we need to clarify that our approach gives accurate estimates about the coastline estimation. The standard way of showing that an estimation method works well is to compare the estimated coastline to the true position of the coastline obtained by a “ground truth” source. This way an objective evaluation of estimation accuracy is obtained. Unfortunately, in lack of a ground truth source, this comparison cannot be performed.

Instead, we perform our method to a plenty of data sets, in order to prove, at least visually, the accuracy and robustness of the algorithm. To achieve this, we used input image data, especially in the coastal area of the region Georgioupoli, located at North West Crete, Greece. The image data are provided by the Hellenic Military of Geographical Service, Google Earth images, and Satellite Multispectral images.

Finally, we present information concerning the performance of the algorithm introduced, although we focused our research more on the topic of implementing an accurate algorithm and less on the optimization step. Even though the algorithms show satisfying performance results.

### 5.1 Algorithm Accuracy

In this section, the accuracy of the introduced algorithm is tested. We can consider an algorithm as accurate when the output data corresponds to an expected accurate value. As mentioned previously, the accuracy of the algorithm presented on the topic of the

## 5. RESULTS

---

coastline estimation, will depend on how accurately represents the estimation, compared to other algorithms and a user observations.

### 5.1.1 Coastline extraction using Aerial images

The aerial images used in this example were acquired by the HMGS during the period from 1990 to 2005. All images are approximately  $5200 \times 5400$  pixels and were taken to a height of 1550 meters. By applying our method to the aerial images, we have successfully extracted a complete, accurate coastline across the region of Georgioupoli.

Figure 5.1 illustrates the processing sequence applied to the first aerial image. The Gaussian filter and then the anisotropic diffusion operator is applied to the image, which is the pre-processing step. At this phase the image noise and is considerably reduced, intensity variations and weak edges inside the land or the ocean are effectively suppressed, while the coastline edges are enhanced (Figure 5.1 b).

At the next step, we apply k-means algorithm to cluster the image into two regions, land and sea 5.1 (c). Subsequently, the entire image is divided into overlapping image regions, each with  $400 \times 400$  pixels. Fifteen percent of image regions that have a high variance are selected for analytically determining optimal local thresholds. The five bimodal Gaussian parameters are computed and then a threshold is computed for every region that passes the 3 criteria of bimodality. The concatenated image is shown in 5.1 (d).

Afterwards, at the post-segmentation processing stage, region grouping is applied with the aim of creating a homogeneous area, where land pixels value is 1 while sea pixels value is 0. Also morphological operators remove noisy image objects (Figure 5.1 (e)).

Finally, we extract the coastline by using canny edge detection and present the results of the first approach of our method (Figure 5.1 (f)).

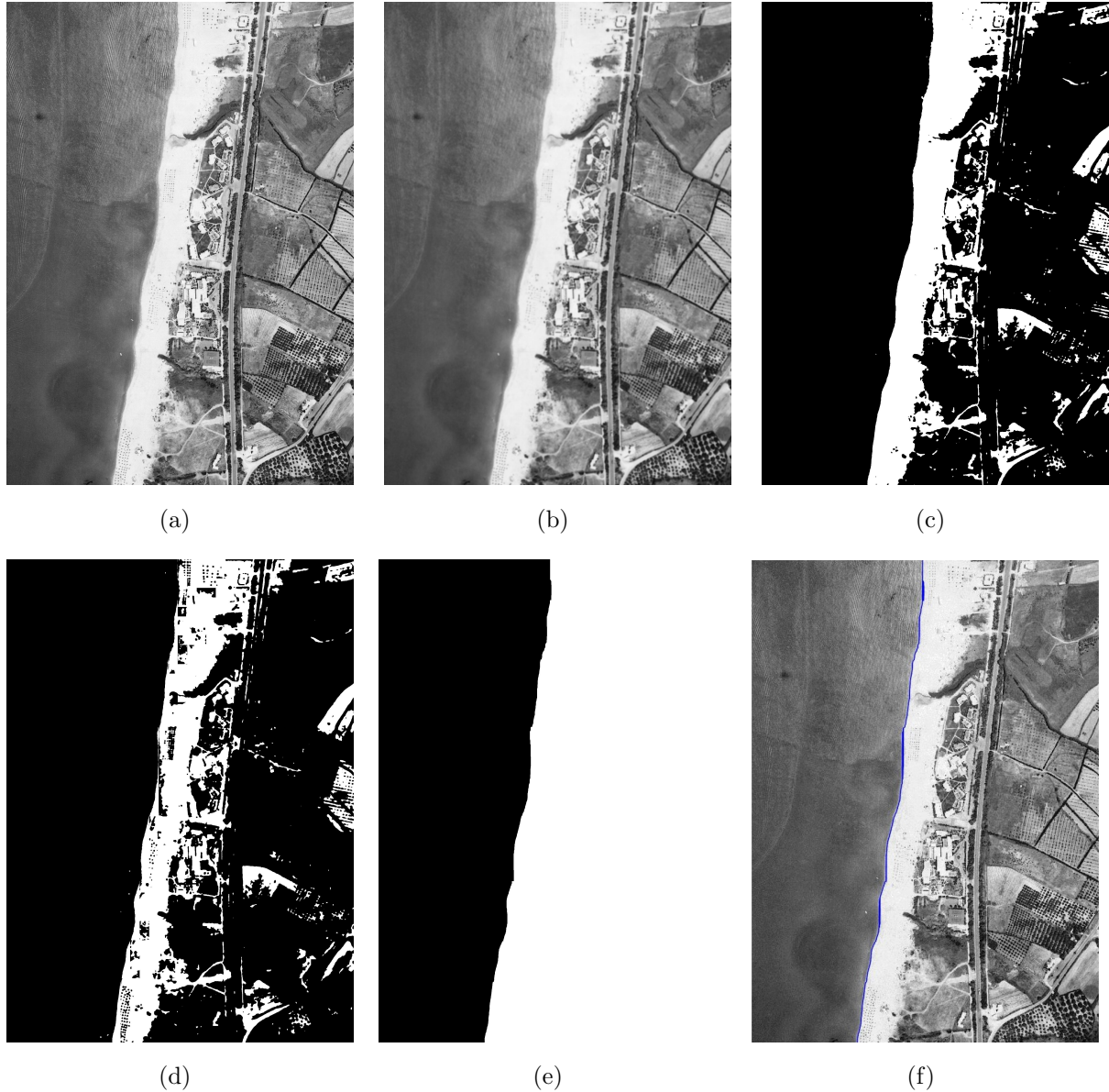


Figure 5.1: Automated coastline extraction from aerial imagery (a) initial image, (b) filtered and enhanced image, (c) k-means clustering, (d) segmented image (e) region grouping and removing small objects

Now we apply the coastline modeling. An automatic initialization of active contours is taking place using the outcome of the previous step. In Figure 5.2 it is shown the final estimated coastline with a close-up for a better identification of the changes. Blue line

## 5. RESULTS

---



Figure 5.2: Estimation of the coastline and a close-up of the coastline area showing the results of the two methods applied on aerial image

refers to the estimated coastline extracted from the first approach while red line refers to the optimized coastline obtained from the second approach. Comparing the two results we can see the added benefits of curve fitting in resulting in smoother estimates that are less affected by other physical formations close to the coastline.

In Figure 5.3 we provide the final estimation of the coastline using another aerial image, again in the region of Georgioupoli. At this case, we needed to manually change the boundary curves of the active contour. Now they will lie on the left to right instead of the default position of the top to bottom of the image. Once again, it is easily understood that the second approach improved the initial estimation of the region segmentation, offering visually valid results.



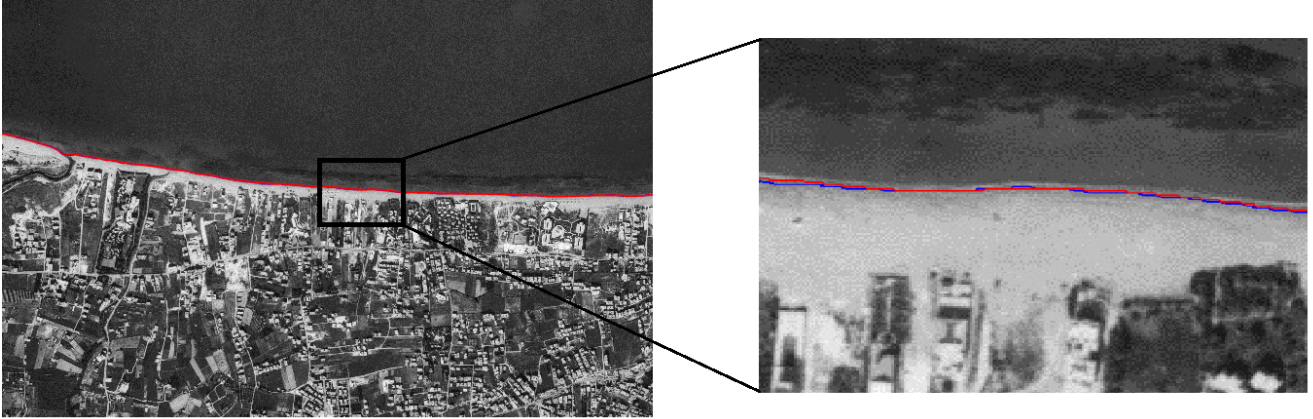


Figure 5.3: Estimation of the coastline and a close up of the coastline area applied on another aerial image

### 5.1.2 Coastline extraction using Google Earth images

Another example demonstrates the applicability of our coastline extraction method to Google Earth images. The input data is taken from the Google Earth, by using the save option of the program at the resolution of  $1920 \times 1080$ . We needed to convert the image from RGB to Grayscale in order to be processed from our algorithm. This image has the same 'behavior' as a digitized aerial image, as a result, we expected good estimations. In Figure 5.4, it is shown that our method leads to great coastline estimations. In this case, again blue line refers to the estimated coastline extracted from the first approach while red line refers to the optimized coastline obtained from the second approach. Having good results using Google images will be useful for two reasons. Easy access to a variety of coasts for further examination of our proposed method. Secondly, Google earth has an option to view the same area in different time periods, hence we can observe coastline changes through years.

## 5. RESULTS

---

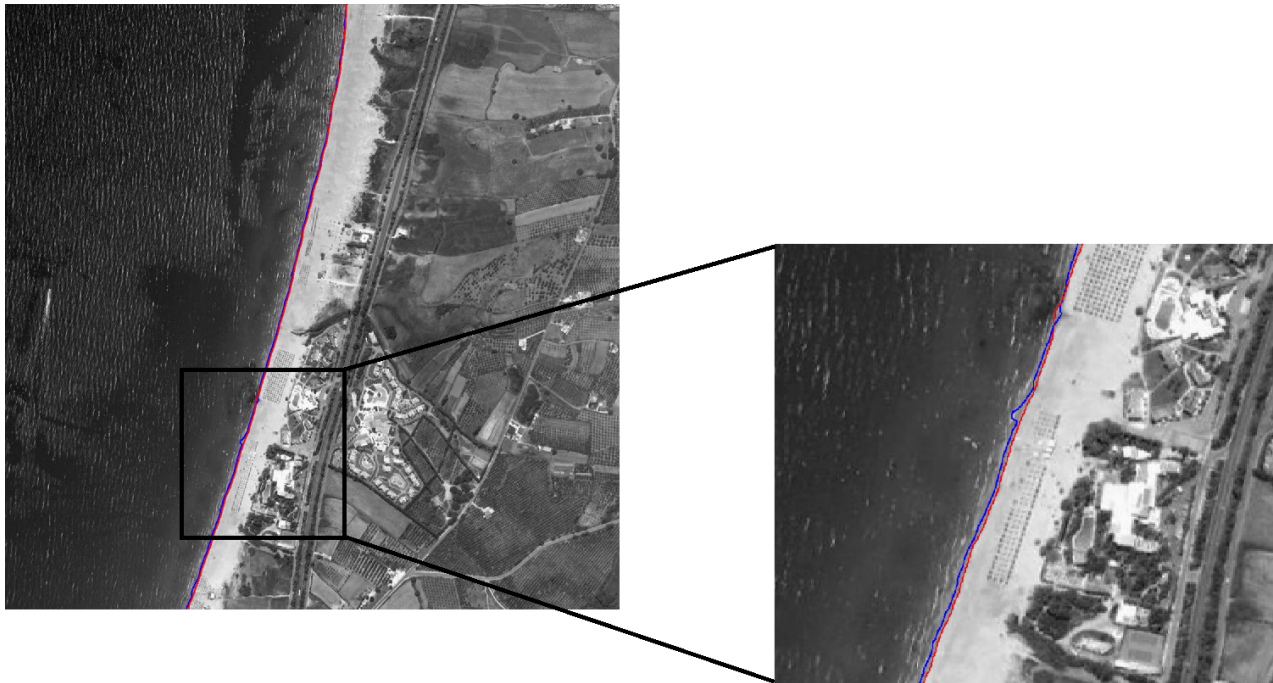


Figure 5.4: Estimation of the coastline and a close-up of the coastline area applied on a Google Earth image

### 5.1.3 Coastline extraction using Satellite images

The most common data used these days for coastline extraction is Satellite Images. We examined, if our method applied to this type image, is capable to provide a good estimation. As we see in Figure 5.5 our algorithm provides a reliable estimation, something that will help us with an easy evaluation of our method. This is because SAR images can provide geo-referenced information, as a result, the estimated coastline can be compared to ground truth measurements.

## 5.2 Algorithm Performance

The aim of this thesis was to provide a coastline estimation algorithm, and thus, no optimization attempts were made. Our approach takes approximately 40 seconds for the first stage (approach 1) and up to 1.65 minutes for the optimization step. This performance



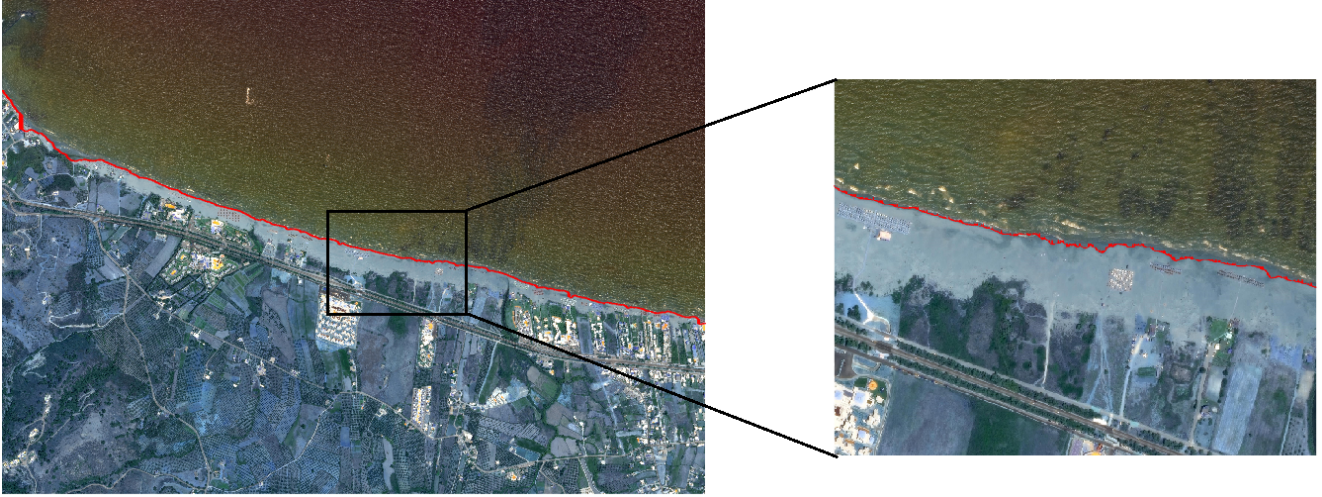


Figure 5.5: Estimation of the coastline using a Satellite image

speed is the average computation speed for various image sizes. The most time-consuming component is the computation of the next curve, which consumes approximately the 69 % of the computational time. The computational speed can be largely improved, by optimizing the algorithm of coastline modeling and effectively reducing the resolution of the images in order to reduce the computational procedure.

## 5.3 Framework Description

Our aforementioned approaches can be combined into a graphical user interface which will allow the visualization, extraction, and processing of coastline estimates. Also, Greek cadastral data are integrated into the framework and spatial data analysis can be processed with the coastal data and cadastral data simultaneously. Figure 5.6 presents the screen of the application where two central windows are devoted to the visualization of the original (left) and processed (right) image, respectively. The left section of the application depicts the parameters of the coastline extraction algorithm and provides the space for adjusting the algorithmic process. The right section presents the functionality of the software in association with the coastline analysis.

## 5. RESULTS

---

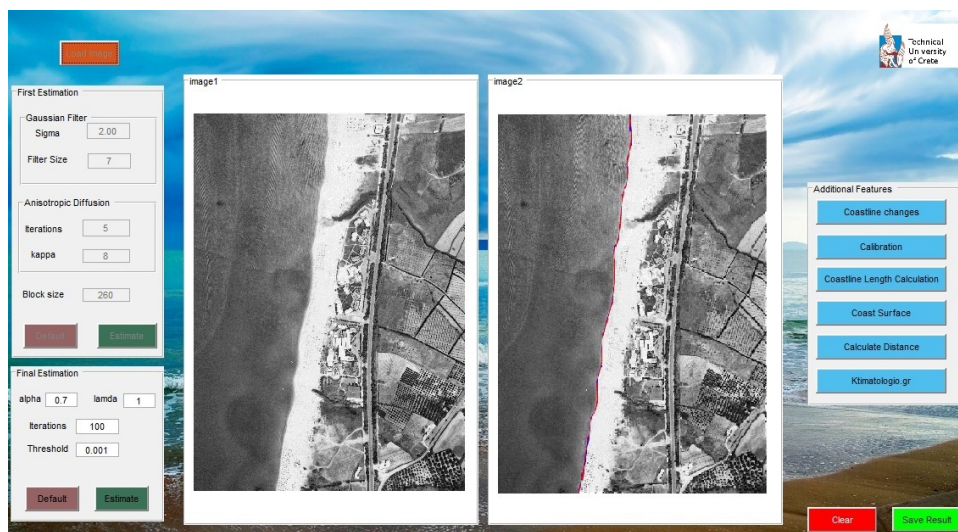


Figure 5.6: The main process panel of the Framework. On the left is the original image. On the right, the two estimated coastlines are shown, blue refers to the first method (region segmentation) and red refers to the second method (active contours)

The top left box of the framework refers to the first method's parameters. Sigma and filters size affect the Gaussian filter. Sigma refers to the standard deviation of the Gaussian distribution. Increasing the standard deviation the intensity of the noise is reduced, but also appears high-frequency detail attenuation. We have set the number of 2 as a default value. A larger sized filter corresponds to a larger convolution mask but also affects the details quality of the image. We have set 7 as a default value. It is optimized for the filter size to be about  $3 \cdot \sigma + 1$ , because, in this way, almost the whole Gaussian bell is taken into account.

Then, the user chooses the number of iterations and kappa value, which refers to anisotropic diffusion. Kappa controls the sensitivity to edges and it is usually chosen experimentally (the default value is 8), while the number of iterations must be 5-15, since a higher number may result in blurring the true edges (the default value is 5). Next, the block size is defined. This size depends on the initial image size because every block needs to contain the necessary information for our method. As we already said, we choose 5-8% of total image size, with default value 300, based on our image data. Next, to the setting of the parameters, the user can proceed with the first coastline estimation using region segmentation.

## 5.3 Framework Description

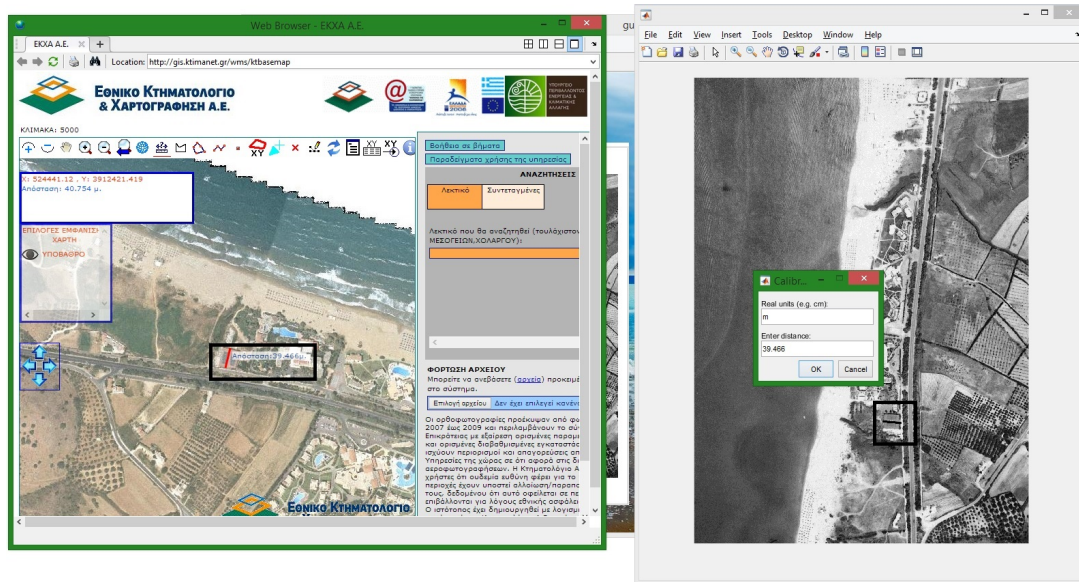


Figure 5.7: The green box shows the selected area (tennis court) from ktimatologio.gr, the Blue box shows the same area from aerial image. We can also see the text box where we import the real units and the distance.

When the first approach is completed, the user chooses the parameters for the second approach. The alpha parameter controls the internal energy function's sensitivity to the amount of stretch in the snake (elasticity). A large value for alpha decreases the possibility the snake to change and so the method's efficiency, so normally alpha value should be less than 1. In our case, we have set it to 0.7. Iterations must be defined wisely because there will be a point that no significant energy differences are detected. A proper number of iteration is 100-300. We, also provide an optional threshold to control the energy differences. If the difference between two consecutive energy values is below the threshold, the process stops and the final result is shown in the images section.

As far for the cadastral data associated with the extracted coastline results, in our application there is a link, called 'ktimatologio.gr' [29] (which means cadaster), that opens the map of Greece in the internet browser and the user can measure real-world distances, such as buildings. Then by pressing the 'calibration' button, the user can choose two points from the initial image which contains the same building or any other chosen area. A text box then opens to enter the known distance and the measurement



## 5. RESULTS

---



Figure 5.8: The computed length of the estimated coastline

unit (Figure 5.7). Now, we are able to compute the distance per pixel (pixel factor) by dividing the inserted distance by the total number of pixels contained between two chosen points.

Moreover, the user is able to perform coastline length calculation that is calculated by multiplying the total number of the pixels that belongs to the coastline by the pixel factor computed above. The measurement units are the same as the one user inserted at the previous step (Figure 5.8).

In addition, the option to compute the area and perimeter of a surface of interest, like the area of buildings, distance from the coastline or beach surface area, is provided by pressing the coast surface button. A new window opens where the user can select a polygonal area at the image. The task performed is the computation of the total number of the pixels which belong to the perimeter, then the multiplication by the pixel factor in order to show the perimeter in real-world units.

Finally, the computation of the total number of the pixels which belong inside this polygonal area is multiplied by the square of pixel factor in order to show the surface. Figure 5.9 displays an example of this an operation.

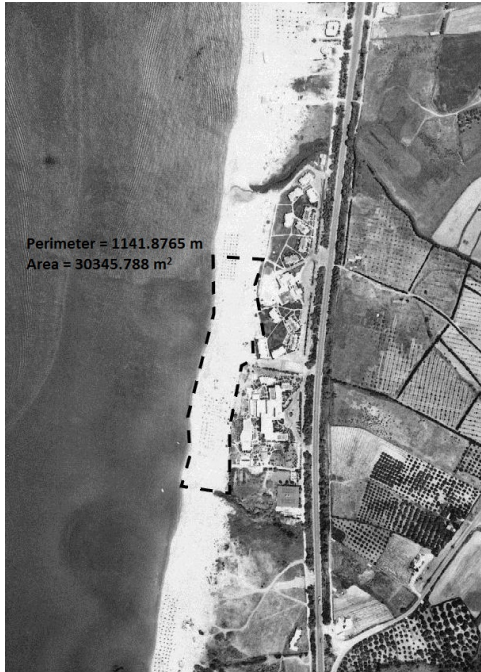


Figure 5.9: Computed perimeter.



Figure 5.10: Computed distance.

Another similar function provided is the calculation of the distance between two points. By pressing calculate distance, a new window opens where the user can choose two points of the image. As previously, we compute the total number of the pixels that belong to the chosen line, multiply by pixel factor and show the results (Figure 5.10). This can be useful if we have images of the same area but from different time periods and want to track the erosion of the coastline.

As a final feature of the application, the user can load two images from different time periods of the same area by pressing the 'Coastline changes' button. When the images load finishes, the two estimated coastline are shown in the same figure, thus the user is able to identify any possible changes (Figure 5.11). Without georeferenced images, it is difficult to work with aerial images. This feature works better with Google earth images.

## 5. RESULTS

---



Figure 5.11: The coastline differences between 2003 and 2010. The dashed line refers to 2003 image, while the straight refers to 2010 image. Images was taken from Google Earth.

Overall, we propose a framework to optimize the coastal development management. We aim to provide additional information through an automated, well-defined framework, to be taken into account to the definition of the spatial limits up to which constructions will be allowed and the kinds of allowed land uses in the Georgioupoli coastal zone.

To summarize, our approach provides faster, more reliable, and robust result comparing to the approach of Lue and Jezek. By using EM algorithm, the first part of our method produces faster an accurate estimation of the mixture of Gaussian parameters. Also, excluding blocks containing only water pixels saved us processing time. Finally, the most important addition we provide in relation to Lue and Jezek's approach is the coastline modeling by using active contours method. This procedure is responsible to fit the estimated coastline across the edges that belong to the coastline area, something that helps us eliminate the margin of error. This improvement can lead us to more reliable and robust results since we have revised the possible wrong estimations.

# Chapter 6

## Conclusion

### 6.1 Conclusion

The present thesis provides an automated coastline extraction methodology using aerial images through image processing techniques aiming at region segmentation and edge detection. Following the elimination of possible noise distortion in the input data and the enhancement of the edges of interest through an anisotropic diffusion algorithm, we apply region segmentation in order to split the image into two groups of regions, land, and sea. The processes of thresholding, edge detection, and active contour fitting improve the accuracy of the extracted coastline.

Also, we developed a framework, through which a user can load the images of interest, perform coastline extraction, view the results and, finally, calculate useful parameters with distance or area metrics by taking advantage of the connection of the results with cadastral data.

In the final analysis, our approach helps in the definition of the legislated allowed land uses according to the existing National Regional Plan and the Sectoral Regional Plan for Tourism that encourages the creation of infrastructure at a distance of 30 meters from the coastal line. In addition, our application can provide useful information towards the investigation of the legislated uses according to the Local Urban Plan of the Georgioupoli municipality, and to the restrictions and specifications which will ensure the sustainability and environmental health of the various uses in the examined coastal areas. This approach enables the assessment of plans and strategies in accordance with the provisions of Directive 2001/42/EC 2015 on strategic environmental assessment and

## 6. CONCLUSION

---

ensures that economic activities factor into the protection of natural resources at an early stage as well as identifying risks related to climate change and natural hazards.

### 6.2 Future Work

As for future work, we intend to improve the accuracy of the coastline estimation by implementing the transform for moving from image coordinates to real-world coordinates and view the image as a projective mapping from the 3D world coordinate system. This can be implemented by using geo-referenced images and provide a pixel-to-real-distance conversion, i.e., pixels/cm conversion scheme. Furthermore, this conversion may provide a robust basis for post-processing and computations of real-world measures, such as coastline length, coast surface, and erosion levels. Also increasing the performance is in our plan, by taking into consideration two fields; Firstly, computation optimization by optimizing the way of the data are accessed and the way their associations are calculated and secondly, by using additional constraints prior to image splitting, that will exclude blocks that will fail to satisfy the three criteria, as mentioned in [4](#).

Also, we intend to further explore the framework's functionality regarding the cadastral data and how this kind of data can help us calibrate the application's parameters and so improve the accuracy of the proposed methodologies.



# References

- [1] Valavanidis, A., Vlachogianni, T.: Agricultural pesticides: ecotoxicological studies and environmental risk assessment. Science advances on Environment, Toxicology and Ecotoxicology issues. [Link](#) (2011) 1
- [2] Liu, H., Jezek, K.: Automated extraction of coastline from satellite imagery by integrating canny edge detection and locally adaptive thresholding methods. International Journal of remote sensing **25**(5) (2004) 937–958 [2](#), [21](#), [23](#)
- [3] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence **12**(7) (1990) 629–639 [10](#), [25](#)
- [4] Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision. Elsevier (1987) 184–203 [13](#)
- [5] Kass, M., Witkin, A., Snakes, D.: Active contour models» international journal of computer vision. In: Computer Vision and Pattern Recognition, IEEE Computer Society Conference on. (1988) 1 [15](#), [35](#)
- [6] Erteza, I.A.: An automatic coastline detector for use with sar images. Technical report, Sandia National Laboratories (SNL-NM), Albuquerque, NM (1998) [18](#)
- [7] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification and scene analysis 2nd ed. ed: Wiley Interscience (1995) [18](#)
- [8] Braud, D., Feng, W.: Semi-automated construction of the louisiana coastline digital land/water boundary using landsat thematic mapper satellite imagery. Louisiana Applied Oil Spill Research and Development Program, OS2 RAPD Technical Report Series **97**(002) (1998) [18](#)

## REFERENCES

---

- [9] Frazier, P.S., Page, K.J., et al.: Water body detection and delineation with landsat tm data. *Photogrammetric engineering and remote sensing* **66**(12) (2000) 1461–1468 [18](#)
- [10] Dellepiane, S., De Laurentiis, R., Giordano, F.: Coastline extraction from sar images and a method for the evaluation of the coastline precision. *Pattern Recognition Letters* **25**(13) (2004) 1461–1470 [18](#)
- [11] Modava, M., Akbarizadeh, G.: Coastline extraction from sar images using spatial fuzzy clustering and the active contour method. *International journal of remote sensing* **38**(2) (2017) 355–370 [19](#)
- [12] Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences* **10**(2-3) (1984) 191–203 [19](#)
- [13] Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J.: Fuzzy c-means clustering with spatial information for image segmentation. *computerized medical imaging and graphics* **30**(1) (2006) 9–15 [19](#)
- [14] Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics* **79**(1) (1988) 12–49 [19](#)
- [15] Stockdonf, H.F., Sallenger Jr, A.H., List, J.H., Holman, R.A.: Estimation of shoreline position and change using airborne topographic lidar data. *Journal of Coastal Research* (2002) 502–513 [20](#)
- [16] Robertson, W., Whitman, D., Zhang, K., Leatherman, S.P.: Mapping shoreline position using airborne laser altimetry. *Journal of Coastal Research* (2004) 884–892 [20](#)
- [17] Liu, H., Sherman, D., Gu, S.: Automated extraction of shorelines from airborne light detection and ranging data and accuracy assessment based on monte carlo simulation. *Journal of Coastal Research* (2007) 1359–1369 [20](#)
- [18] Ryan, T., Sementilli, P., Yuen, P., Hunt, B.: Extraction of shoreline features by neural nets and image processing. *Photogrammetric Engineering and Remote Sensing;(United States)* **57** (1991) [21](#)

- [19] Chalabi, A., Mohd-Lokman, H., Mohd-Suffian, I., Karamali, K., Karthigeyan, V., Masita, M.: Monitoring shoreline change using ikonos image and aerial photographs: a case study of kuala terengganu area, malaysia. In: ISPRS Commission VII Mid-term Symposium b••Remote Sensing: From Pixels to Processesb••, Enschede, the Netherlands. (2006) 8–11 [21](#)
- [20] Schowengerdt, R.A.: Remote sensing: models and methods for image processing. Elsevier (2006) [21](#)
- [21] Yousef, A., Iftekharuddin, K.: Shoreline extraction from the fusion of lidar dem data and aerial images using mutual information and genetic algorithms. In: Neural Networks (IJCNN), 2014 International Joint Conference on, IEEE (2014) 1007–1014 [22](#)
- [22] Haupt, R.L., Haupt, S.E., Haupt, S.E.: Practical genetic algorithms. Volume 2. Wiley New York (1998) [22](#)
- [23] HMGS: Hellenic military of geographical service. [Link 24](#)
- [24] Google: Google earth, 2014. google earth virtual globe. [Link 24](#)
- [25] Aquaphoenix. [Link 31](#)
- [26] Ashman, K.M., Bird, C.M., Zepf, S.E.: Detecting bimodality in astronomical datasets. arXiv preprint astro-ph/9408030 (1994) [32](#)
- [27] Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1) (1979) 62–66 [32](#)
- [28] Shemesh, M., Ben-Shahar, O.: Free boundary conditions active contours with applications for vision. In: International Symposium on Visual Computing, Springer (2011) 180–191 [36](#)
- [29] Ktimatologio. [Link 47](#)