CORRELATION-BASED VARIATIONAL CHANGE DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GİZEM AKTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2018

Approval of the thesis:

CORRELATION-BASED VARIATIONAL CHANGE DETECTION

submitted by **GİZEM AKTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering**

Prof. Dr. Fatoş Tunay Yarman Vural
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Assist. Prof. Dr. Emre Akbaş
Computer Engineering Department, METU

Prof. Dr. Fatoş Tunay Yarman Vural
Computer Engineering Department, METU

Assist. Prof. Dr. Gülşah Tümüklü Özyer
Computer Engineering Department, Atatürk University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GİZEM AKTAŞ

Signature :

ABSTRACT

CORRELATION-BASED VARIATIONAL CHANGE DETECTION

Aktaş, Gizem
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. Fatoş Tunay Yarman Vural

June 2018, 140 pages

Change detection is an important research topic for observing the earth but there are various challenges to obtain an accurate change map such as image noise, subtle differences and image acquisition alteration. Various studies handled these problems as separate steps. In the first step, images are registered, then the image noise are eliminated. After images are normalized to eliminate image acquisition differences, actual change detection routine can be applied as a final step. However, this step-by-step approach leads to accumulation of errors in each step which leads to decrease in accuracy of final change detection map. Step-by-step approach is widely used because these problems are interpenetrated to each other and researchers use this to divide problem into sub-problems. In this study, a correlation based variational change detection (CVCD) method for elevation models is proposed. In essence, CVCD aims to produce smooth change maps while preserving the details of the terrain by minimizing a variational cost function. In this variational cost function a novel correlation-based data fidelity term is used with an ℓ_1-norm regularization term which imposes smoothness on obtained change map. In addition, ℓ_1-norm TV regularization term is preferred because it can preserve details such as point-changes, edges and corners of changes. In order to minimize the proposed cost function using simple approximations in an iterative manner, a simple and efficient algorithm is suggested. Quantitative experiments on synthetic noisy data show that CVCD can provide a detection rate of 95% while staying in the low false alarm regime, i.e. less than 10^{-2}. Also, qualitative experiments on real-world data show the success of the CVCD for the changes with different characteristics.

Keywords: change detection, correlation, total variation, point cloud, LIDAR, digital surface model

ÖZ

İLİNTİ TABANLI DEĞİŞİKLİK TESPİTİ

Aktaş, Gizem
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tunay Yarman Vural

Haziran 2018 , 140 sayfa

Değişiklik tespiti, dünya yüzeyini gözlemlemek için kullanılan oldukça önemli bir araştırma konusudur. Fakat doğru bir değişiklik haritası elde etmenin görüntünün gürültülü olması, gözle fark edilemeyen değişimlerin olması ve görüntü alma işlemindeki farklılıklar gibi çeşitli zorlukları vardır. Çeşitli çalışmalar bu problemleri, görüntü alma farklarını ortadan kaldırmak için gürültüyü azaltma, görüntü örtüştürme ve görüntü normalleştirme gibi ayrı adımlar olarak ele almıştır. Bu birbirinin içine geçmiş büyük bir problemi, alt problemlere ayırarak adım adım çözme yöntemi oldukça yaygın bir şekilde kullanılmaktadır. Fakat, bu yaklaşımda her adımda meydana gelebilecek hatalar bir sonraki adımın da sonucunu etkilemektedir ve dolayısıyla her adımda biriken hatalar en son elde edilen değişiklik tespiti haritasının da mutlak doğruluğunun azalmasına neden olmaktadır. Bu çalışmada, yükseklik modelleri için ilinti tabanlı değişimsel bir değişiklik tespit yöntemi önerilmiştir. Özünde bu yöntem, değişimsel bir maliyet fonksiyonunu en küçülterek çözmektedir. Bu maliyet fonksiyonu değişim tespiti yapılan arazinin detaylarını korurken aynı zamanda da elde edilen değişim haritasının pürüzsüz olmasını amaçlar. Bu değişimsel maliyet fonksiyonunda,

yeni bir ilinti tabanlı veri sadakat terimi ve elde edilen değişiklik haritası üzerinde pürüzsüzlüğü sağlayan bir ℓ_1-norm düzenlileştirme terimi kullanılır. Ayrıca, değişim bölgelerinin kenarları ve köşeleri gibi özniteliklerini koruyabildiği için ℓ_1-norm toplam varyasyon (TV) düzenlilik terimi tercih edilir. Önerilen maliyet fonksiyonu, basit matematiksel yaklaşımlar kullanılarak basit be etkili bir algoritma ile yinelenen bir şekilde en küçültülür. Sentetik veri üzerinde yapılan nicel deneyler, önerilen algoritmanın düşük yanlış alarm ile çalışırken, yani 10^{-2} değerinden daha az, yüksek tespit oranı ($\%95$) sağlayabildiğini göstermektedir. Ayrıca, gerçek uzaktan algılama verileri üzerinde yapılan niteliksel deneyler, farklı özelliklere sahip değişiklikler için de önerilen algoritmanın başarısını göstermektedir.

Anahtar Kelimeler: değişiklik tespiti, ilinti, toplam değişim, nokta bulutu, LIDAR, sayısal yüzey modeli

To my lovely family

ACKNOWLEDGMENTS

To start with, I would like to reveal my most sincere gratitude towards my thesis supervisor Prof. Dr. Fatoş Tunay Yarman Vural for her support throughout every step of my graduate studies. In every bit of this journey, she always motivated me to carry on and steered me in the best possible direction. I am deeply glad and honoured that I had the chance to work with her.

I would like to thank Dr. Fatih Nar, for his endless time, effort, support and guidance. He gave attention to my problems and guided me how to proceed whenever I am stuck at some point. I am gratefully indebted to him for his very valuable comments on this thesis.

I would also express my appreciation towards my old colleague Mustafa Andaç Derinpınar who is geomatic engineer, my old project manager Dr. Nigar Şen and the founder of Zibumi, Erdal Yılmaz, for providing me real life data to carry out my experiments.

Sincerest thanks to each member of my family for supporting and believing in me all the way through my academic life.

Last but not least, I would like to thank my beloved friend Murat Özatay for providing me not only help for issues with linguistic related topics but also moral support whenever I need.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
ALS	Aerial Laser Scanner
ANN	Artificial Neural Network
ASCII	American Standard Code for Information Interchange
ASPRS	American Society for Programmetry and Remote Sensing
AUC	Area Under Curve
CC	Cross Correlation
CD	Change Detection
CS	Compressing Sensing
CVA	Change Vector Analysis
CVCD	Correlation-based Variational Change Detection
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
EM	Expectation Maximization
EO	Electro-Optic
FOV	Field of View
GPS	Global Positioning System
GS	Gramm-Schmidt
GT	Ground Truth
IC	Incomplete-Cholesky

IMU	Inertia Measurement Unit
InSAR	Interferometric Synthetic Aperture Radar
IR-MAD	Iteratively Re-weighted Multivariate Alteration Detection
KT	Tasseled Cap (Kauth-Thomas)
LIDAR	Light Detection and Ranging
MAP	Maximum A Posteriori Probability
MLS	Mobile Laser Scanner
MRF	Markov Random Field
NASA	National Aeronautics and Space Administration
NCALM	National Center for Airborne Laser Mapping
PCA	Principal Component Analysis
PCG	Preconditioned Conjugate Gradient
PDF	Probability Density Function
RADAR	Radio Detection and Ranging
RAM	Random Access Memory
RGB	Red Green Blue
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RPE	Reduced Parzen Estimation
SAR	Synthetic Aperture Radar
SDT	Space and Defence Technologies
SLS	Satellite Laser Scanner
SNR	Signal to Noise Ratio
TDMA	Tridiagonal Matrix Algorithm
TIN	Triangular Irregular Network
TLS	Terrestrial Laser Scanner
TV	Total Variation

CHAPTER 1

INTRODUCTION

1.1 Motivation

The planet we live on, the earth, and its natural resources are very important for humankind to continue to exist. As long as humankind continues to think, they will continue to make many contributions to the world with the technology and inventions they have developed since people are constantly curious and inquisitive since their existence because of their nature. In the direction of their curiosity, each day they add something new to the earth. In each second we breathe, the earth is changing due to the consequences of our interventions while we are making a place to stay, warming up, feeding our belly, transporting from one place to another, inventing and producing something new.

With the contributions of humankind to developing technology, their own curiosity is also exponentially increasing. Along with the technology, human beings are informed about the natural disasters such as earthquakes, tsunamis, volcanic explosions, forest fires, and floods and also the effects of natural disasters on the formation of the world as well as on their daily lives. To give a specific example, as one of the most well-known facts, water is indispensable for human life. According to Wired Science, it is estimated that the waters that flow from rivers, lakes and aquifers towards the ocean dry before they reach to the ocean. This is the result of lessened precipitation brought on by deforestation and the construction of man-made dams that occupy water stream in wasteful ways. The reduced water flow influences the climate change and species that live in the wetland. The detecting, understanding and explaining of the relationships between the human and nature is important to make the world better and more viable while managing and using the resources [25]. In accordance with this detailed information, people can make deep inspections about the surface of the earth. Remote sensing is one of the most important technologies that has been developed to satisfy the curiosity of the people. It is the science that combines a large source of information and technology used to make observations, analysis and inferences about the earth and atmospheric events without actually being in contact with the earth [33]. The main source of that knowledge is the measurements and images obtained by collecting and recording reflected or emitted energy from the platforms on aerial and space vehicles [29]. At the end of many years of success in photographic technology and the methods used to observe the world, the first world observation was made with TIROS weather satellite, in 1960. This experience is known as the first use of remote sensing term in history [11].

Remote sensing has been widely used in many areas such as geological, archeological, atmospherical, civil and military applications since it has been developed. Image and video analysis applications in these areas can be exemplified as forest detection, biomass estimate, real-time monitoring and tracking of various disasters like fire, flood, earthquake, eruptions and drought, disaster prevention, urban, port or road mapping, 3D modeling of structures, verification of international treaties, border control, weather forecast, tracking of climate change and air quality, etc [33]. For the past fifty years, remote sensing has also been used in many academic studies and many image processing algorithms. It has a huge importance when the contributions of these applications to the daily lives of people are considered. Within these applications, it is also clear that change detection algorithm is very important in various fields such as disaster monitoring and prevention, surveillance, urban monitoring, agriculture monitoring. In our decade, its popularity is increasing each and every day [47].

One of the most interesting subjects is change analysis in the earth science that investigates the world in the direction of people's curiosity, tries to make the earth better, and evolves with the use of remote sensing systems for this purpose. Change detection is getting even more popular nowadays. To detect changes of earth's surface timely and accurately is extremely important and remote sensing images is widely used for object detection and analysis by monitoring the earth's surface from satellites or airborne platforms with the help of the sensors for many decades. A change analysis is a process that detects how the attributes of an identified region change between two or more time periods. Change detection identifies the pixels of multi-temporal images that belong to same geographic region on the earth, but are taken in different times as changed or unchanged. While determining the change, the most important point is not directly taking the difference of two images' pixel values. In a powerful change detection algorithm, the meaningful changes like construction or destruction of buildings, forestation or deforestation, and residential developments should be determined [52].

The images that are studied for the development of change detection algorithms in academic or industrial researches are very numerous and this diversity in the remotely sensed input images is due to the variety in the systems and methods used to obtain them. The sensor used to obtain the image needs to be on a fixed platform in order to be able to collect and record the energy reflected or emitted from the target or from the surface while observing the earth. The platforms on which the sensor is stabilized can be localized in various places such as a ladder, a tall building, a pier or a crane on the ground or a balloon, an aircraft or a helicopter in the air or a spacecraft or a satellite outside of the world's atmosphere. Film photography, radiometer, spectrometer, hyperspectral radiometer, radio detection and ranging (RADAR), synthetic aperture radar (SAR), and light detection and ranging (LIDAR) can be examples of remote sensors. The choice of sensor for any task has always a trade-off. To illustrate, despite photographic systems have high quality in spatial resolution to provide very detailed information from the observed region, the observed area is very limited and supplied images are weak due to spectral sensitivity beside other sensors. Nevertheless, non-photographic sensors are too complex optically, mechanically, and electrically. This complexity leads to system limitations in terms of power, space and stability causing to high costs [33]. Beside these, there are also various geometric distortions regardless of the type of sensor or platform. In remote sensing, three-dimensional earth's surface is tried to be represented accurately with two-dimensional image which leads to such large geometric distortions [29]. In addition, the monitoring, processing and management of natural resources in very large areas requires a lot of time, workload and storage to collect a lot of data. To avoid such problems, in recent years, LIDAR technology has been the most effective system to represent the earth's surface thanks to ability of storing the data directly in a point cloud form [4, 61].

1.2 Scope and Goal

In this thesis, the proposed study aims to provide a complete, robust and efficient change detection algorithm in remotely sensed elevation models. The purpose of the proposed method is to detect significant variations such as construction or destruction of buildings, forestation or deforestation instead of minimal differences like gaps or sensor noises. In addition, this method handles detecting changes as newly coming and demolished objects, separately. This privilege distinguishes the proposed method from the counterparts in the literature. Although, there exist a vast amount change detection methods, there is not enough data in remote sensing applications. Because of this reason, especially, supervised approaches are not preferred for the change analysis in remote sensing domain. Therefore, an unsupervised change detection algorithm, called as Correlation-based Variational Change Detection (CVCD), is proposed to determine the changes on the elevation maps. In this study, elevation models are preferred as the data since they are less affected from the seasonal changes. Therefore, they can be incorporated to improve the accuracy of the algorithms for detecting the changes. This approach determines the changes while eliminating the effect of image noise due to the nature of remote sensing systems by minimizing correlation based variational cost function. In this variational cost function a novel correlation-based data fidelity term is used with an ℓ_1-norm regularization term which imposes smoothness on obtained change map. Due to the success of the correlation similarity metric, correlation-based data fidelity term is preferred to provide similarity of two associated images within the variational framework. It is employed with ℓ_1-norm total variation (TV) regularization term to deal with subtle differences such as noise. ℓ_1-norm TV regularization term is preferred because it can preserve details such as point-changes, corners and edges of changes. With the help of a numerical approach used in solving of this proposed cost function, geometrically correct and visually attractive change mask is obtained. The most important purpose of this study is to present all qualitative features mentioned above in an effective, fast and robust method. The results of the proposed method have more than $\%95$ accuracy.

1.3 Outline

The outline of the thesis is as follows. First chapter makes an entrance to the specified research area and defines the proposed study. Literature survey for change detection and the employed techniques are presented in the second chapter. This chapter also includes explanations for the theoretic background of the utilized methods in the variational framework. After overviewing the background of important subjects in the first two chapters, third chapter is about digital surface models which are the inputs for the proposed change detection method. The DSM generation types are reviewed and compared. Specifically, information about Lidar sensor such as platforms, components and applications of it and a variational based DSM generation method using Lidar point cloud data is emphasized in this chapter. Fourth chapter focuses on the details of the proposed algorithm. Fifth chapter defines the experiments conducted on different datasets and discusses the obtained results. Lastly, study is concluded in the sixth chapter by wrapping up the key themes and suggesting the possible future work.

CHAPTER 2

LITERATURE SURVEY ON CHANGE DETECTION METHODS AND DATA FOR CHANGE ANALYSIS

In this chapter, it is presented a brief overview on the change detection problem for the remotely sensed data. The first section focuses on change detection problem and employed approaches in the literature. The following section investigates variational methods in image analysis domain since the proposed approach is based on variational methods.

2.1 An Overview of Change Detection Problem and Employed Methods in Literature

The monitoring of natural phenomena around the world and the decision-making in the course of the following natural events in the world are extremely important for solving the problems that may arise. Nowadays, with the use of remote sensing systems more and more, world surface and natural resources can be continuously monitored and real time information can be produced and managed. Furthermore, the identification of natural resources, the extraction of inventories, the planned use of these resources and the protection of ecological balance are important criteria dealt with in the development of the world. The use of remote sensing data is crucial to obtaining accurate, fast and cost-effective data and information in order to identify and update existing assets and potentials of natural resources, and to monitor and update temporal changes in the world. With the help of the widely and easily used remote sensing technology which enables high quality images to be acquired in a serial manner, change analysis has begun to gain an important place among studies in the field of computer vision.

A simple definition of change detection is to compare aerial photographs or satellite images of the same region at different times. In a more formal definition, change analysis is a process that measures how qualitative and quantitative characteristics of a given region have changed between two or more time periods. The point of the change detection is not taking the simple difference of two images, the aimed changes should be meaningful changes like construction or destruction of buildings, man made structures, forestation or deforestation and residential developments.

Since it is possible to acquire multi-temporal images from earth's surface quickly and accurately with the help of developing advanced remote sensing technology, remote sensing applications are developing and playing major role for monitoring, analyzing, understanding and managing the sources of nature. Important change detection applications including land-use and land-cover change detection, forest or vegetation change detection, forest mortality analysis, defoliation and damage assessment, deforestation, wetland change detection, forest fire and fire-affected area detection, urban change detection, environmental change analysis, crop monitoring, sea/ice tracking, and map updating are reviewed in a detailed survey by Lu at 2004 [37].

Although developments in remote sensing technologies providing timely and cost-effectively information about the observation of the earth surface have a significant potential in the image world, change detection may suffer from usage of this technology and may cause some problems that affect the analysis of change. Pacifici et. al. summarized some of the major problems that could be encountered in his research published as follows [44]. The first issue is that data must be processed in very large dimensions in order to detect changes in a small field. The second one is that too many satellite sensor types are used; however, the spectral bands of all these sensors are not identical, so the acquired images are not exactly the same. Next, image noises due to various reasons such as sensor calibration, humidity on the ground, weather and light conditions cause changes that do not really exist between obtained images at different times. Errors that may arise in the course of overlapping of multispectral images or geometric distortion of each image, may cause false detections. Although the preferred sensors are different, the types of acquired images obtained are different, and the usage area of the developed application is different, the proposed methods focus on only one of the mentioned problems or these problems are ignored, basically the goal is always the same: "Identification of altered regions between at least two acquired images at different times from the same region."

In this direction, a good change detection algorithm should involve major requirements. The selection of the change detection method which can be varied according to data and problem. There exists a vast amount of change detection methods for different data types like, SAR, electro-optic (EO), hyper-spectral, 3D medical imaging, and LIDAR in the literature. A vast amount of change detection methods are suggested to provide solutions to the mentioned application areas. In 2004, Radke et. al. dealt with methods of change detection and its application areas in a published research paper [47]. Although the methods in this study have not been developed for remote sensing data, it shows that the methods of change analysis are developed in the same way for other areas in the literature. The change detection methods are introduced in two main categories as simple and advanced. Image differencing, image rationing, image thresholding, change vector analysis (CVA) are among the simple methods while significance and likelihood ratio tests, mixture and predictive models are in the second category. In 1989, Singh published a research article dealing with more recent change detection methods for remotely sensed data [52]. Lu et al. (2004) explained in detail the mostly used algorithms, the accuracy assessment and the effecting factors of the algorithms [37]. The most commonly employed algorithms are frequently taken advantage of in change detection methods can be grouped as seven categories. Some of the methods in the four main groups are as follows: The simplest category is called as algebra based techniques including simple image differencing, image regression, image rationing, and thresholding [38]. The second category is transformation based techniques which are principle component analysis (PCA), change vector analysis, background subtraction, tasseled cap (KT), Gramm-Schmidt, and Chi-Square. The other group includes classification based methods such as post-classification comparison, expectation maximization (EM) and artificial neural networks (ANN) [52]. Advanced approaches such as Markov random field (MRF) [60, 10] graph cut [65], variational level set [8, 12] are the fourth category. Pacifici reviewed all the methods in the literature more generally under two category as supervised and unsupervised [44]. He mentioned about the limitations and advantages of the methods in these two categories. Apart from the methods mentioned earlier, unsupervised approaches such as reduced parzen estimation (RPE), maximum a posteriori probability decision (MAP) and iteratively re-weighted multivariate alteration detection (IR-MAD) are also reviewed in this study. According to him, supervised methods have many advantages over unsupervised methods. The types of changes to be analyzed can be exactly determined. Change detection results can be produced independently from radiometric, atmospheric and light conditions. In addition, change analysis can be performed independently of the sensor from which the image is generated. Unfortunately, a very huge amount of training dataset is needed to generate appropriate results in supervised methods and the acquisition of a proper training set is usually a difficult and costly expensive.

The trending approach nowadays is applying deep learning techniques in change detection since deep learning is a relatively new machine learning method and been paid more and more attention every year. In [66], an innovative change detection algorithm which focuses on analyzing multi temporal SAR images is proposed. This approach uses deep neural networks for handling problems of image change detection. Deep neural networks give the ability to represent images and learn features in an abstract way. Unsupervised feature learning and supervised fine-tuning is featured in proposed deep learning method. This study states that presence of sparkle noise is one of the significant problems in change detection on SAR images, unfortunately the sparkle noise handling method of the proposed algorithm is not clearly mentioned. Another drawback of this paper is not stating explicitly whether the training parameters of the proposed deep learning method stays identical when different datasets are used. In [26], a formal problem statement is proposed which makes effective usage of deep learning methods possible for analyzing time-dependent series obtained from remote sensing images. Also, a new framework is introduced for the development of deep learning models in targeted change detection. This study mentions various advantages of deep learning approaches in change detection methods like being able to extract only the specified classed of changes. Meanwhile it is also stated that amount of the data for training models limits the current solutions in a negative manner. Unfortunately the proposed framework is not implemented in the paper which prevents evaluating the deep learning approach for change detection with tests and results of those tests. In [13], a new difference image creation method which uses deep neural networks is proposed. The most significant modification of the proposed back propagation algorithm is adjusting deep belief network to increase the difference on changed areas meanwhile decreasing the difference on unchanged areas. The proposed method can avoid the radiometric correction procedures for change detection and suppress the noise effectively compared to traditional methods. One significant deficit of this study is labeling the training data manually. Using deep learning in change detection algorithms has both its advantages and disadvantages. However, since the data collection from remote sensing systems are not easily accessible and low cost, it is difficult to train a deep neural network in order to achieve high performance and generic results for remote sensing applications.

These algorithms are suggested to provide partial solutions, to look out for contiguity relationship, and handle feature based segmentation by using difference image. However, a robust change detection method which covers a wide range of application areas is still a challenging problem due to various reasons. A crucial set of problems such as radar illumination, registration errors due to platform motion or moving reflector, sensor noise or subtle changes like leaf movements which affect the quality of the change maps are reported due to the seasonal and/or meteorological changes.

The most decisive part while designing a change detection algorithm is being able to construct a cost function which captures the **desired** changes while it remains invariant under **superfluous** variations. In the proposed method, this crucial step is achieved by using variational methods including various data fidelity terms with ℓ_2-norm or ℓ_1-norm.

2.2 Variational Methods

In this section an overview will be briefly discussed about the variational methods that are widely used in many applications in the signal and image processing community. In particular, the noise reduction method that inspires the Digital Elevation Model (DEM) generation method and the proposed change detection method in this study will be explained in detail.

2.2.1 Variational Calculus

The calculus of variations is a field of mathematics which is used to find the largest or the smallest possible value of an unknown functional [20]. Variational calculus deals with functionals, which can be defined as functions of a function or a set of functions. Function is a mapping from one number to another value. A notation of any function is shown as $y = f(x)$. On the other hand, a functional can be defined as a mapping from a function to a value which means that while it takes a function as input, it generates a value as its output. Functionals are considered as definite integrals that are involving the functions and derivatives of the functions. The representation of a functional can be $J(x, y)$, where J is a functional operator and $y = f(x)$ is the function. In practice, functionals are dependent to function derivatives. The aim of the variational calculus is to find a function that minimizes or maximizes of a functional, subject to certain boundary conditions using the Euler-Lagrange equation of the calculus of variations [17].

Variational methods are used in a wide range of areas including finite element analysis, quantum mechanics, statistical mechanics, and computer vision. The applications of variational methods for each example, a complex problem is converted to a simple problem and it is characterized by the decomposition of the freedom of degree in the original problem [28]. Various scientific and engineering problems can be formulated in variational calculus form and one of the most famous problems in the history is "Brachistochrone Problem". This problem is stated as "*Given two points on a plane at different heights, what is the shape of the wire down which a bead will slide (without friction) under the influence of gravity so as to pass from the upper point to the lower point in the shortest amount of time?*" [50]. Even though the shortest path is a straight line between the given two points, when the curve is under a constraint such as gravity the solution is changing and many possible solutions may occur. Figure 2.1 shows the representation of the brachistochrone problem, there exist many possible routes between two points, A and B; however, the frontmost ball is at the cycloid route from these paths.

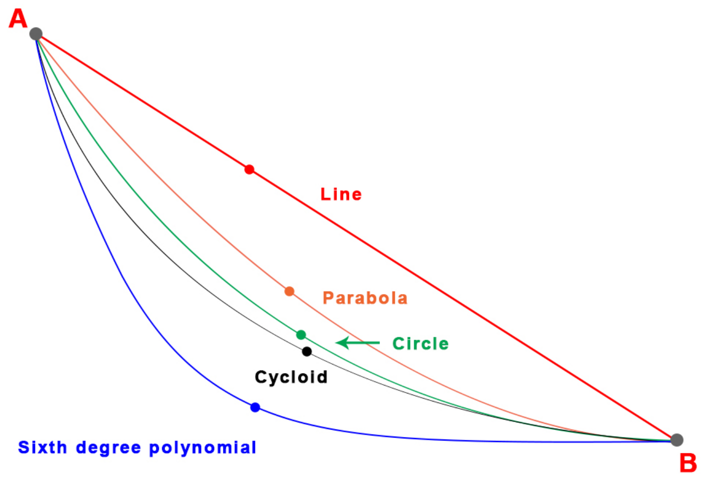


Figure 2.1: Brachistochrone Problem: A ball is traversing through different paths from A to B

In the modern world, images are becoming more and more important and are used to represent and understand the physical world. Since images provide a huge amount of information and the modern computers have the capability of processing more data, computer vision and image processing are amongst the most popular topics. Many problems in these topics can be considered as minimization problems. These continuous minimization problems can be exemplified as image denoising, image deblurring, image inpainting, shape denoising, super-resolution, image segmentation, optical flow and 3D reconstruction. For example, image denoising can be defined as noise removal from an image in image processing community. It can be redefined as finding a smooth approximation to the noisy image in the space of images in mathematic world. While image segmentation problem is partitioning the image into object and background, it can be redefined as finding smooth closed curve between the object and background. As it can be easily seen from the examples given, the aim is to formulate the image problem in the form of a minimization problem with the given constraints and solve the determined unknown function using calculus of variations.

Even though variational methods are perfectly proper for inverse problems, it is not always possible to solve these problems using direct methods. The variational framework can be adopted into image world by minimizing of energy functional, E.

$$f^* = \underset{f}{\operatorname{argmin}}\, E(f) \;, \tag{2.1}$$

where f^* is the aimed solution of f. For instance, a probability information can be attached into the solution for a fundamental image denoising problem using Bayesian inference framework. In this model, the unseen true image is represented with f, g is the observed noisy image and η shows the additive noise, $g = f + \eta$. The joint probability for f and g, called as chain rule, can be written as Equation 2.2:

$$\mathcal{P}(f, g) = \mathcal{P}(f|g)\mathcal{P}(g) = \mathcal{P}(g|f)\mathcal{P}(f) \;. \tag{2.2}$$

According to Bayesian formula the expression can be rewritten as Equation 2.3,

$$\mathcal{P}(f|g) = \frac{\mathcal{P}(g|f)\mathcal{P}(f)}{\mathcal{P}(g)} \;, \tag{2.3}$$

where $\mathcal{P}(f|g)$ is the posterior probability, $\mathcal{P}(f)$ represents the prior probability of hypothesis, the evidence is indicated with $\mathcal{P}(g)$ and finally $\mathcal{P}(g|f)$ shows the likelihood. Here, the posterior probability which is identified as energy functional will be maximized. It is aimed to estimate most likely solution f^* when the observed data and prior belief are given. This approach is called as maximum aposteriori (MAP) estimation and it can be defined at Equation 2.4:

$$f^* = \underset{f}{\operatorname{argmax}}\, \frac{\mathcal{P}(g|f)\mathcal{P}(f)}{\mathcal{P}(g)} \;. \tag{2.4}$$

Since the denominator part of the right hand side of the equation is just used for the normalization, the posterior probability density function (PDF) integrates to unity. Hence, the MAP estimation of Bayesian inference can be rewritten as following Equation 2.5:

$$f^* = \underset{f}{\operatorname{argmax}}\, \mathcal{P}(g|f)\mathcal{P}(f) \; . \tag{2.5}$$

In our problem, the pixels in the images are assumed to be independent and identically distributed according to the probability law. Since all measurements are mutually independent the likelihood is for the entire data is represented and substituted at Equation 2.6:

$$f^* = \underset{f}{\operatorname{argmax}} \prod_{i=1}^{N} \mathcal{P}(g_i|f_i)\mathcal{P}(f_i) \; . \tag{2.6}$$

where i is the pixel index, N represents the total number of pixels in the image. It is considered as the all intensity values in the images are normal distributed and have the Gaussian noise, the likelihood can be defined as the following Equation 2.7:

$$\prod_{i=1}^{N} \mathcal{P}(g_i|f_i) \propto \prod_{i=1}^{N} e^{(-\frac{(f_i-g_i)^2}{2\sigma^2})} \; . \tag{2.7}$$

Since the probability of the f_i, prior probability, is sufficiently characterized by its neighbours that is indicated at Equation 2.8:

$$\mathcal{P}(f) = \mathcal{P}(f_1 ... f_n) = \mathcal{P}(f_1|f_2 ... f_n)\mathcal{P}(f_2 ... f_n) \propto \prod_{i=1}^{N-1} \mathcal{P}(f_i|f_{i+1}) \; . \tag{2.8}$$

When the intensity values of the image are distributed with smoothness prior, this case corresponds to **total variation** (TV) of f and this prior probability is described as Equation 2.9,

$$\mathcal{P}(f) \propto \prod_{i=1}^{N-1} e^{(-\lambda|f_i - f_{i+1}|)} \; , \tag{2.9}$$

where, f_i is the current pixel intensity and f_{i+1} indicates the neighbour pixel. λ is a positive constant value that controls the smoothness factor. Because logarithm function is strictly monotonous function, taking the logarithm of the probabilities preserves the inequalities. So that multiplication of the probabilities can be transformed into addition of log-probabilities to make calculations simple that is redefined at Equation 2.10:

$$f^* = \underset{f}{\operatorname{argmax}} \sum_{i=1}^{N} \{\log \mathcal{P}(g_i|g_i)\} + \log \mathcal{P}(f)\ . \tag{2.10}$$

In addition, instead of maximizing the probability distribution, it is preferable to minimize its negative logarithm. The final energy is drawn up for the defined image denoising problem as the following Equation 2.11:

$$E(f) = \sum_{i=1}^{N} \frac{|g_i - f_i|^2}{2\sigma^2} + \lambda \sum_{i=1}^{N-1} |f_i - f_{i+1}|\ . \tag{2.11}$$

2.2.2 Total Variation Method

In the variational approaches, it is seen that the preferred prior information influences the MAP estimation under the Bayesian framework for a given problem domain, image denoising problem. As a result of the estimation steps, the solution of the problem is transformed to a total variation based energy minimization problem. The total variation concept for one real value is introduced in Camille Jordan by a convergence theorem for Fourier Series of discontinuous periodic functions, in 1881 [27]. Later, some physical concepts are adopted to image processing world; for instance, in 1990, scale-space and edge preserving smoothing using diffusion is suggested by Perona and Malik [45], total variational based image denoising approach is proposed by Rudin, Osher and Fatemi [49], in 1992. Since TV based approaches provide regularizing in inverse problems and preserves sharp discontinuities in the solutions, the solutions are commonly used in signal and image processing communities for many applications such as image denoising [14], [43], change detection [8], [12], and structure extraction [63].

2.2.2.1 ℓ_2-norm TV Denoising on 1D Signal

In this subsection, a numerical algorithm of TV denoising for one-dimensional (1D) signal is presented. The existence of noise in a signal is an important problem for developed applications in the sense of increasing the performance ratio of the algorithm and producing more real results in the engineering and science fields. Of course, since noise is an inevitable problem in the image, it adversely affects any image processing operation and must be removed from the image or at least be reduced. According to Rudin et al. (1992), true signal is usually extracted from the noisy signal using the least square dependent methods and in their work, they aimed to find the denoised signal using variational methods involving ℓ_2-norm because it leads to a set of linear equations [49].

The aim of the suggested method is to find denoised signal, f, efficiently. In order to estimate denoised signal, this proposed cost function, $J(f)$ is solved by minimizing with respect to f, that is described in Equation 2.20;

$$f^* = \underset{f}{\operatorname{argmin}}\, J(f)\ , \tag{2.12}$$

where f^* indicates the aimed denoised signal. The cost function which is proposed for denoising problem for noisy 1D discrete signal is given in equation 2.13;

$$J(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda |(\nabla f)_i|_2^2\ , \tag{2.13}$$

where, $J(f)$ represents the cost function, g is the given noisy signal, $g = (g[1], ..., g[N]) \in \mathbb{R}^N$ of size $N \geq 1$, f indicates the noise removed signal, $f \in \mathbb{R}^N$. i is the index of the signal and N is the total number of elements in the signal. ∇ denotes the gradient operator and since the signal has only one dimension the derivative of the signal is only taken in one direction, x. Therefore, the derivative operator is indicated as ∂_x which is rewritten in Equation 2.14. λ, smoothness parameter, is a positive scalar constant that determines the smoothness level of the obtained signal and controls the balance between the removing the noise from signal and preserving the signal content.

$$J(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda |(\partial_x f)_i|^2 \ . \tag{2.14}$$

In Equation 2.14, the first term is described as **data fidelity term** that is used to make denoised signal is similar to the input signal. The second term, **TV-regularizer**, is added to cost function in order to preserve large-scale edges while penalizing the gradient changes on the signal.

In order to solve this denoising problem and find the optimum denoised signal, it is necessary to take the derivative of the cost function and set it to zero. Since both data fidelity term and TV-regularizer term are quadratic, they are differentiable and strictly convex. Henceforth; since the cost function contains quadratic terms, Equation 2.13, can be written in matrix-vector form, as follows;

$$J(v_f) = (v_f - v_g)^\top (v_f - v_g) + \lambda (v_f \top \mathbf{D_x}^\top \mathbf{D_x} v_f) \ , \tag{2.15}$$

where v_f and v_g are the vector representations of the f and g signals, respectively. $\mathbf{D_x}$ indicates the Toeplitz matrix which is designed for taking the partial derivative of the signal in x direction. In order to illustrate a small version of $\mathbf{D_x}$ for a 1D signal with 10 elements is shown in below;

$$\mathbf{D_x} = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since the objective function, $J(v_f)$ is strictly convex and differentiable, the derivation of $J(v_f)$ with respect to v_f and equalizing it to zero presented in Equation 2.16;

$$\frac{\partial J(v_f)}{\partial v_f} = 0 \; . \tag{2.16}$$

The mathematical steps related to derivation operations are processed and aligned in the following Equation 2.17,

$$2(v_f - v_g) + 2\lambda \mathbf{D_x}^\top \mathbf{D_x} v_f = 0 \; , \tag{2.17}$$

where the terms are simplified and v_f coefficients can be collected as the following Equation 2.18:

$$(\mathbf{I} + \lambda \mathbf{D_x}^\top \mathbf{D_x}) v_f = v_g \; . \tag{2.18}$$

The simplified version of the cost function can be depicted as a linear system as shown in Equation 2.19;

$$\begin{aligned} \mathbf{A} v_f^{(n+1)} &= b \; , \\ \mathbf{A} &= \mathbf{I} + \lambda \mathbf{D_x}^\top \mathbf{D_x} \; , \\ b &= v_g \; , \end{aligned} \tag{2.19}$$

where $\mathbf{I}$ is the identity matrix. Note that, $\mathbf{A}$ and b belongs to n^{th} iteration. In Equation 2.19, $\mathbf{A}$ is sparse, positive definite, 5-point Laplacian matrix. In this solution, since $\mathbf{A}$ is a positive definite and tridiagonal matrix, this linear system can be solved using tridiagonal matrix algorithm (TDMA) in order to estimate the aimed denoised signal. Since taking inverse of the matrix with high dimensions is computationally expensive and TDMA solves this linear system in $O(n)$ complexity, this this approach is preferred.

The experimental signals and results related to ℓ_2-norm based total variatonal denoising on 1D signal are shown in Figure 2.2:

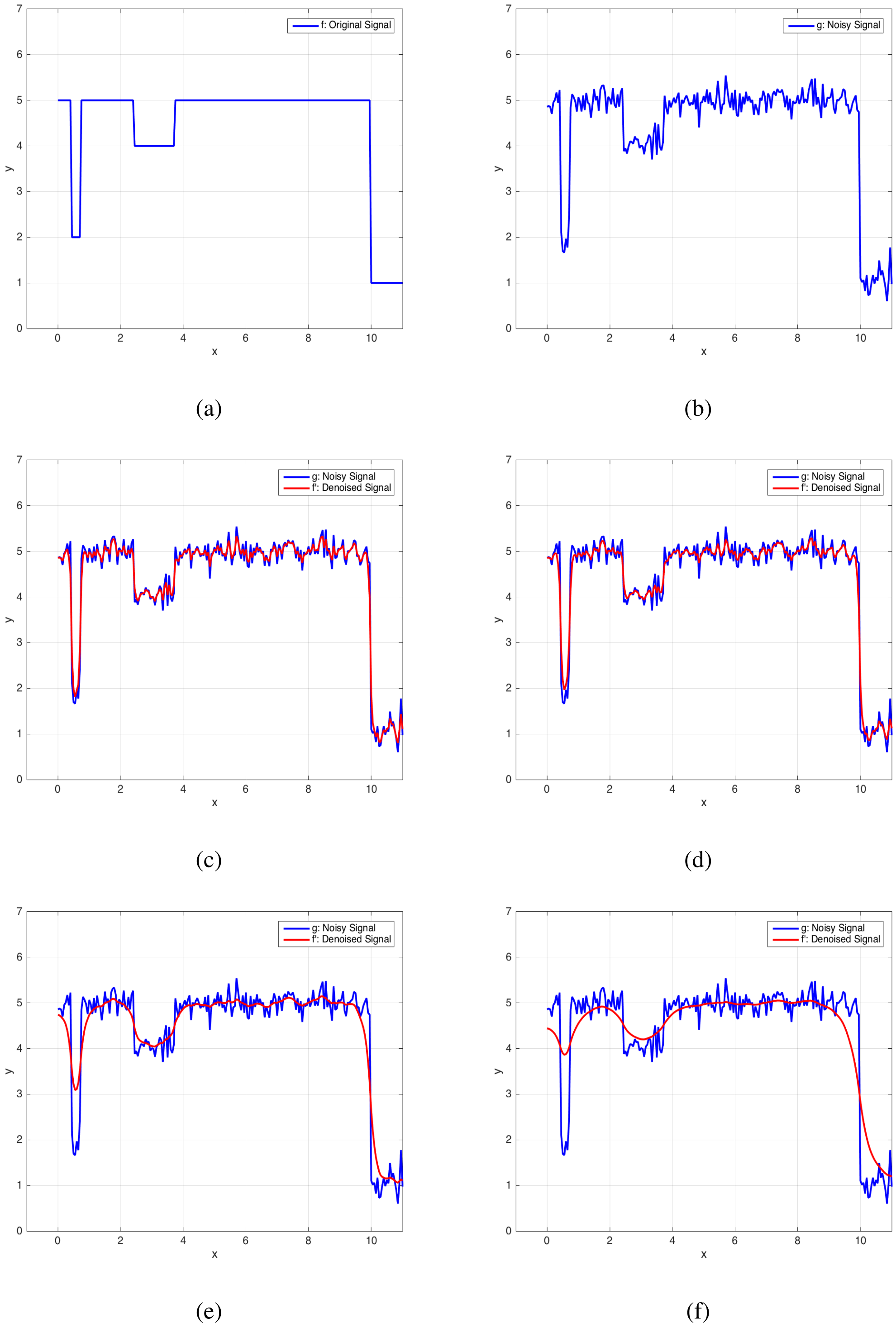


Figure 2.2: ℓ_2-norm denoising results on 1D signal with different λ parameters and constant $\varepsilon = 10^{-5}$ (a) Original signal, f (b) Noisy signal, g (c) ℓ_2-norm denoised signal, f', over the noisy signal, where $\lambda = 0.5$ (d) ℓ_2-norm denoised signal, f' over the noisy signal, where $\lambda = 1$ (e) ℓ_2-norm denoised signal, f', over the noisy signal, where $\lambda = 10$ (f) ℓ_2-norm denoised signal, f', over the noisy signal, where $\lambda = 50$

When ℓ_2-norm cost function is used, the penalty is estimated by multiplication of square of the difference of two numbers and λ value for the minimizer. The purpose of the minimizer is to find the optimum denoised signal by minimizing the cost function. Since the penalty is calculated as a numerically large value because of the quadratic second term, various λ values are tested in order to diminish the penalty. However, when a small λ is chosen, the denoising operation is almost never achieved as it is shown in Figure 2.2c and Figure 2.2d. When the appropriate λ value is given to minimizer, it prefers to be smoother in transitions rather than a great amount penalty by making one big jump. In this case, the edges are not preserved as it can be seen in Figure 2.2e and Figure 2.2f. Although ℓ_2-norm is differentiable, strictly convex, and easily resolvable, it is not edge preserving. Because of this reason, ℓ_1-norm which provides more sparsity, is mostly preferred in the literature.

2.2.2.2 ℓ_1-norm Total Variation Denoising on 1D Signal

In this section, a numerical algorithm of ℓ_1-norm based TV denoising for 1D signal is presented. Rudin et al. (1992) indicates that ℓ_1-norm based TV approaches are more appropriate than ℓ_2-norm based ones. When the same problem is solved with these two different approaches under the same conditions, the result of ℓ_1-norm approximation looks better than the ℓ_2-norm [49].

The aim of the method is to find denoised signal, f, efficiently in terms of speed of the algorithm. In order to estimate denoised signal, this proposed cost function, $J(f)$ is solved by minimizing with respect to f, that is described in Equation 2.20;

$$f^* = \underset{f}{\operatorname{argmin}}\, J(f)\ , \tag{2.20}$$

where f^* indicates the aimed denoised signal. The proposed cost function for denoising problem for noisy 1D discrete signal is given in equation 2.21;

$$J(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda |(\nabla f)_i|_1^1\ , \tag{2.21}$$

where J represents the cost function, g is the estimated noisy signal, $g = (g[1], ..., g[N]) \in \mathbb{R}^N$ of size $N \geq 1$, f indicates the noise removed signal, $f \in \mathbb{R}^N$. i is the index of the signal and N is the total number of elements in the signal. ∇ denotes the gradient operator and since the signal has only one dimension the derivative of the signal is only taken in one direction, x. Therefore, the derivative operator is indicated as ∂_x which is rewritten in Equation 2.22. λ, smoothness parameter, is a positive scalar constant that determines the smoothness level of the obtained signal and controls the balance between the removing the noise from signal and preserving the signal content.

$$J(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda |(\partial_x f)_i| \,. \tag{2.22}$$

In equation 2.22, the first term is described as **data fidelity term** that is used to make denoised signal is similar to the input signal. Because of the quadratic intensity difference, it is not allowed that many pixels drastically change their values so that diminishing salient edges are automatically prevented [39]. The second term, **TV-regularizer**, is added to cost function in order to preserve large-scale edges and corners while penalizing the gradient changes on the signal.

In order to solve this denoising problem and to find the optimum denoised signal, it is necessary to take the derivative of the cost function and equalize it to zero. At this point; although, the ℓ_1-norm is preferred in the denoising problem because it provides a more accurate solution than ℓ_2-norm, it also has its' drawbacks. In the proposed ℓ_1-norm based cost function, the derivative of ℓ_1-norm term cannot be taken directly because of the discontinuity at 0. ℓ_1-norm is known as the least absolute function and the derivative of an absolute function cannot be taken because it is not differentiable at sharp point that is 0. An example graph of absolute function, $y = |x|$, is given at Figure 2.3.

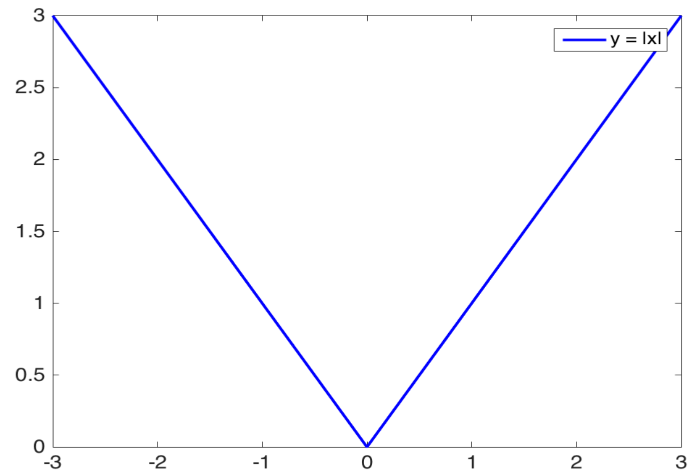


Figure 2.3: The absolute value function

According to definition of the derivative, the left and right hand limits of must be same at a point in order to take the derivative of the function at that point. In other words, the derivative must be a continuous function. However, it is obviously shown that (Figure 2.4), the slope of the absolute function to the left equals to -1 and to the right equals to 1. Since they are different, this absolute function is not differentiable.

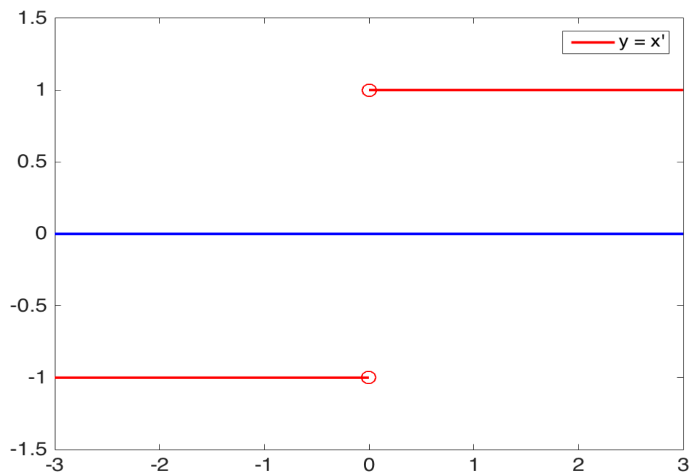


Figure 2.4: The derivative of an absolute value function

An absolute function is approximated using quadratic approximation method which is formulated as Equation 2.23 in order to use convex optimization methods [43, 41].

$$|x| \approx \mathcal{Q}(\hat{x})x^2, \quad \mathcal{Q}(\hat{x}) = (|\hat{x}| + \varepsilon)^{-1} , \tag{2.23}$$

where ε is a small positive constant which is used to avoid indefinite division caused by the zero value of the denominator in the division process. In addition, $\hat{x}$ is a constant proxy for x such that $\hat{x} \leftarrow x$, $\mathcal{Q}(\hat{x})$ is the coefficient of the quadratic approximation of $|x|$. In this approximation method, the absolute function is fitted into a quadratic function with respect to x value. Starting from a point on x domain, the new quadratic function is fitted iteratively for each x value until the global minima point is found. Note that, the newly approximated function is exactly accurate only on both of around evaluated $\hat{x}$ points and 0, and approximation accuracy decreases as x diverges from these points. As it can be observed from the plots on Figure 2.5, in each iteration, the error between the fitted function and the original absolute function is decreasing while the x value comes closer to global minima.

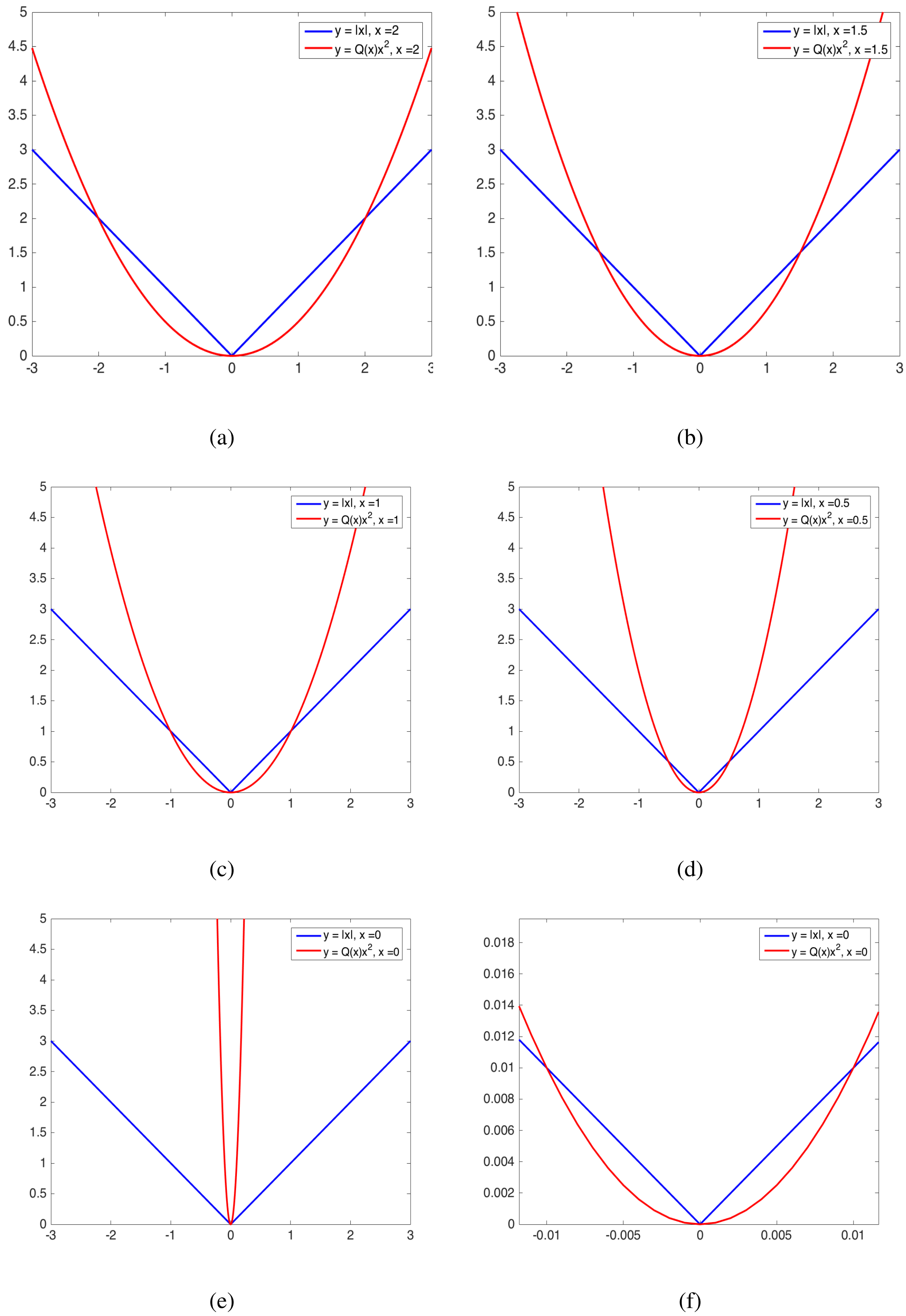


Figure 2.5: Absolute function and quadratic approximations of absolute function for different x values (a) x = 2.0 (b) x = 1.5 (c) x = 1.0 (d) x = 0.5 (e) x = 0 (f) x = 0, zoomed

According to Equation 2.23; in order to minimize the proposed cost function, ℓ_1-norm based regularizer term is quadratically approximated in Equation 2.24;

$$|(\partial_x f)_i| \approx (W_x)_i(\partial_x f)_i^2, \quad (W_x)_i = \mathcal{Q}((\partial_x \hat{f})_i) \ , \tag{2.24}$$

where $(W_x)_i$ is the constant denominator of the quadratic approximation which is evaluated at $\hat{f}$, proxy for f. Moreover, the iteration index (n) is added, since the approximated cost function needs to be solved iteratively and each term belongs to n^{th} iteration unless otherwise specified, for the sake of simplicity. After the quadratic approximation for the ℓ_1-norm is applied to the cost function, it becomes as follows:

$$J^{(n)}(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda(W_x)_i(\partial_x f)_i^2 \ . \tag{2.25}$$

Because the first term is a quadratic, it is differentiable and the second term of the cost function is, also, differentiable. Hence, this differentiable cost function can be written as matrix-vector form which is shown in Equation 2.26;

$$J^{(n)}(v_f) = (v_f - v_g)^\top(v_f - v_g) + \lambda({v_f}^\top {\mathbf{D_x}}^\top \mathbf{W_x}\mathbf{D_x} v_f) \ , \tag{2.26}$$

where v_f and v_g are the vector representations of the f and g signals, respectively. $\mathbf{D_x}$ indicates the Toeplitz matrix which is explained in Section 2.2.2.1 in detail. $\mathbf{W_x}$ denotes the diagonal weight matrix that holds the $(W_x)_i$ coefficients on its diagonal.

Since the objective function, $J^{(n)}(v_f)$ is strictly convex and differentiable, the derivative is taken with respect to v_f, that is presented in Equation 2.27;

$$\frac{\partial J^{(n)}(v_f)}{\partial v_f} = 0 \ . \tag{2.27}$$

The mathematical steps related to derivation operations are proposed and aligned in the following Equation 2.28,

$$2(v_f - v_g) + 2\lambda {\mathbf{D_x}}^\top \mathbf{D_x}\mathbf{D_x} v_f = 0 \ , \tag{2.28}$$

where the terms are simplified and v_f coefficients can be collected as the following Equation 2.29;

$$(\mathbf{I} + \lambda \mathbf{D_x}^\top \mathbf{W_x} \mathbf{D_x}) v_f = v_g \ , \tag{2.29}$$

where $\mathbf{I}$ is the identity matrix. Equation 2.29 is represented as a linear system as shown in Equation 2.30;

$$\begin{aligned} \mathbf{A} v_f^{(n+1)} &= b \ , \\ \mathbf{A} &= \mathbf{I} + \lambda \mathbf{D_x}^\top \mathbf{W_x} \mathbf{D_x} \ , \\ b &= v_g \ . \end{aligned} \tag{2.30}$$

In Equation 2.30, $\mathbf{A}$ is sparse, positive definite, 5-point Laplacian matrix. Note that, $\mathbf{A}$ and b belongs to n^{th} iteration. In this solution, since $\mathbf{A}$ is positive definite matrix, the aimed denoised signal is estimated minimizing the cost function according to Algorithm 1 in an iterative manner due to applied approximations. In each iteration, obtained sparse linear system is solved by using the preconditioned conjugate gradient (PCG) with the incomplete Cholesky (IC) preconditioner to evaluate the $v_f^{(n+1)}$. In Algorithm 1, maximum number of PCG iterations is set to 100, convergence tolerance of the solver is set to 10^{-3}, $n_{max} = 10$, and $C_{tolerance} = 10^{-3}$ as defaults.

Algorithm 1 ℓ_1-norm Total Variation Denoising on 1D Signal

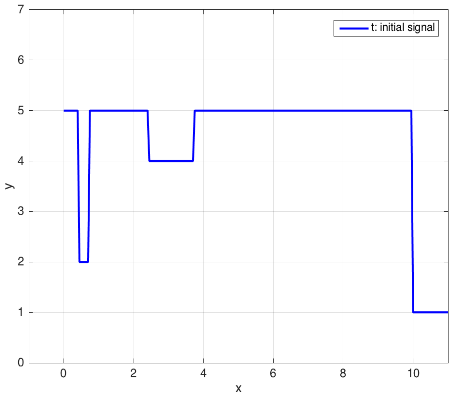
Input: g, λ, ε, n_{max}, $C_{tolerance}$

1: $v_f^{(1)} \leftarrow v_g \leftarrow g$
2: **for** $n = 1 : n_{max}$ **do**
3: $\quad v_{\hat{f}} \leftarrow v_f^{(n)}$
4: $\quad (W_x)_i = \mathcal{Q}((\partial_x \hat{f})_i)$
5: $\quad \mathbf{A}^{(n)} \leftarrow \mathbf{I} + \lambda \mathbf{D_x}^\top \mathbf{W_x} \mathbf{D_x}$
6: $\quad v_b^{(n)} \leftarrow v_g$
7: $\quad$ `Solve` $\mathbf{A}^{(n)} v_{\hat{f}}^{(n+1)} = v_g^{(n)}$
8: $\quad$ if $\|v_{\hat{f}}^{(n+1)} - v_{\hat{f}}\|_\infty < C_{tolerance}$ then break the loop
9: **end for**
10: $f \leftarrow vectorToImage(v_f^{(n+1)})$

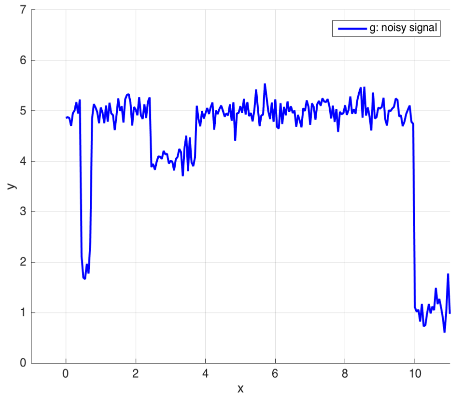
Output: f

In the 8^{th} step of this TV denoising algorithm, the infinite repetition of the algorithm is limited by the maximum number of iterations.

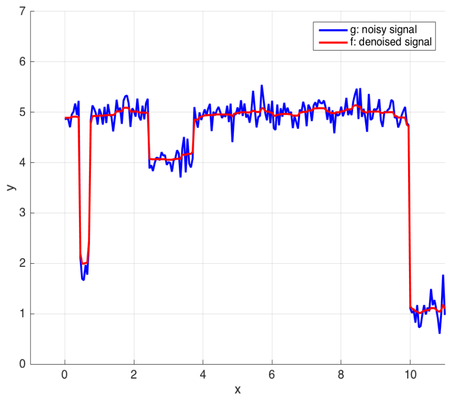
The synthetically geberated signals and results related to ℓ_1-norm based total variational denoising on 1D signal are shown in Figure 2.6:



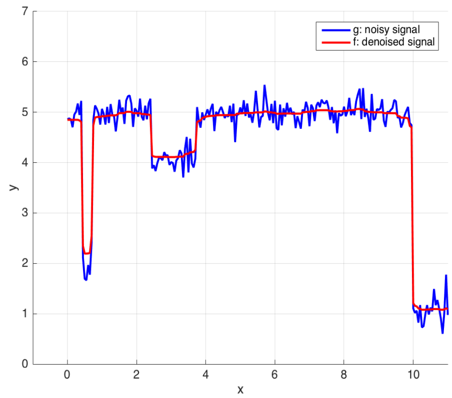
(a)



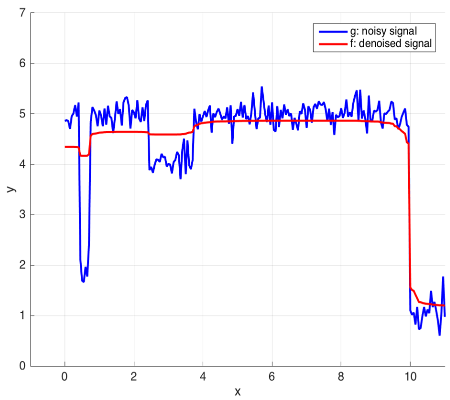
(b)



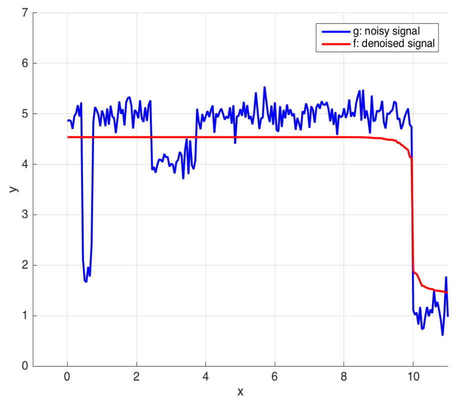
(c)



(d)



(e)



(f)

Figure 2.6: ℓ_1-norm denoising results on 1D signal with different λ parameters and constant $\varepsilon = 10^{-5}$(a) Original signal, f (b) Noisy signal, g (c) ℓ_1-norm denoised signal, f', over the noisy signal, where $\lambda = 0.5$ (d) ℓ_1-norm denoised signal, f' over the noisy signal, where $\lambda = 1$ (e) ℓ_1-norm denoised signal, f', over the noisy signal, where $\lambda = 10$ (f) ℓ_1-norm denoised signal, f', over the noisy signal, where $\lambda = 50$

In the ℓ_1-norm based total variation denoising approach, quadratic approximation compresses the signal amplitudes especially when the λ value is increased as can be seen in Figure 2.6f. Because of this reason is that as the λ value grows, the minimizer takes the real value, f, away from the approximated value, $\hat{f}$, quickly. In order to prevent this situation, slow-step regularization is proposed which prevents the desired solution being too far from the approximated value $\hat{f}$ point. This additional suggestion is explained in detail Section 2.2.2.3.

2.2.2.3 ℓ_1-norm TV Denoising with Slow Step Regularizer on 1D Signal

In this approach, a new term is added to proposed cost function in order not to move away the applied quadratic approximation from the original function while taking the derivative of the ℓ_1-norm based term. By the aid of the added slow step regularization term, the denoised signal $\hat{f}$ does not move away from the original signal f.

In the Section 2.2.2.2, the Equation 2.25 is modified as the following Equation 2.31 by including the small step regularization after the quadratic approximation of ℓ_1-norm based regularizer term in order to minimize the proposed cost function.

$$J^{(n)}(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda (W_x)_i (\partial_x f)_i^2 + \gamma (f_i - \hat{f}_i)^2 \ , \tag{2.31}$$

where g indicates observed noisy signal and f is the denoised signal. i and N denote the pixel index and pixel count of the signal, respectively. Smoothness parameter, λ, and damping parameter, γ, are positive constants. $(W_x)_i$ represents the constant denominators of the quadratic approximation which is evaluated at $\hat{f}$, proxy for f. $(f_p - \hat{f}_i)^2$ is a new regularization term which provides that f_i stays close to $\hat{f}_i$ and that is controlled by damping parameter. Moreover, the iteration index (n) is added, since the approximated cost function needs to be solved iteratively and each term belongs to n^{th} iteration unless otherwise specified, for the sake of simplicity.

Equation 2.31 is strictly convex and differentiable since all terms are quadratic and convex. The cost function is written in matrix-vector form as follows;

$$J^{(n)}(v_f) = (v_f - v_g)^\top (v_f - v_g) + \lambda({v_f}^\top {\mathbf{D_x}}^\top \mathbf{W_x} \mathbf{D_x} v_f) + \gamma(v_f - v_{\hat{f}})^\top (v_f - v_{\hat{f}}) \ , \tag{2.32}$$

where v_f, $v_{\hat{f}}$ and v_g are the vector representations of the f, $\hat{f}$ and g signals, respectively. $\mathbf{D_x}$ indicates the Toeplitz matrix which is explained in Section 2.2.2.1 in detail. $\mathbf{W_x}$ denotes the diagonal matrix form of $(W_x)_i$.

Since the objective function, $J^{(n)}(v_f)$, is strictly convex and differentiable, taking derivation with respect to v_f and equalizing it to zero are presented in Equation 2.33;

$$\frac{\partial J^{(n)}(v_f)}{\partial v_f} = 0 \ . \tag{2.33}$$

The result of the derivation operations are provided as follows,

$$2(v_f - v_g) + 2\lambda({\mathbf{D_x}}^\top \mathbf{W_x} \mathbf{D_x}) v_f + 2\gamma(v_f - v_{\hat{f}}) = 0 \ , \tag{2.34}$$

where the terms are simplified and v_f coefficients can be collected as follows;

$$\big((1+\gamma)\mathbf{I} + \lambda({\mathbf{D_x}}^\top \mathbf{W_x} \mathbf{D_x}) + \big) v_f = v_g + \gamma v_{\hat{f}} \ , \tag{2.35}$$

where $\mathbf{I}$ is the identity matrix. Equation 2.35 is depicted as a linear system as shown in Equation 2.36;

$$\begin{aligned} \mathbf{A} v_f^{(n+1)} &= b \ , \\ \mathbf{A} &= (1+\gamma)\mathbf{I} + \lambda({\mathbf{D_x}}^\top \mathbf{W_x} \mathbf{D_x}) \ , \\ b &= v_g + \gamma v_{\hat{f}} \ . \end{aligned} \tag{2.36}$$

In Equation 2.19, $\mathbf{A}$ is sparse, positive definite, 5-point Laplacian matrix. Note that, $\mathbf{A}$ and b belongs to n^{th} iteration. In this solution, since $\mathbf{A}$ is positive definite matrix, the aimed denoised signal is estimated by minimizing the cost function according to Algorithm 2 in an iterative manner due to applied approximations. In each iteration, a sparse linear system is solved by using the preconditioned conjugate gradient (PCG) with the incomplete Cholesky (IC) preconditioner to evaluate the $v_f^{(n+1)}$. In Algorithm 2, maximum number of PCG iterations is set to 100, convergence tolerance of the solver is set to 10^{-3}, $n_{max} = 10$, and $C_{tolerance} = 10^{-3}$ as defaults.

Algorithm 2 ℓ_1-norm Total Variation Denoising with Small Step Regularization on 1D Signal

---

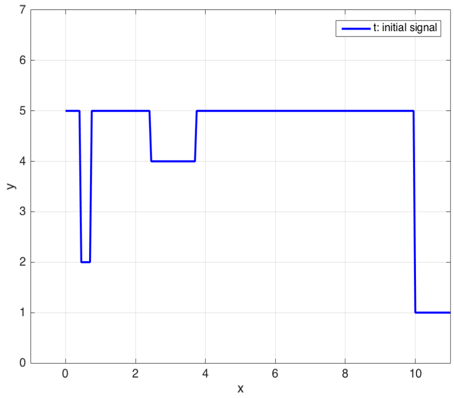
Input: g, λ, ε, n_{max}, $C_{tolerance}$

1: $v_f^{(1)} \leftarrow v_g \leftarrow g$
2: **for** $n = 1 : n_{max}$ **do**
3: $\quad v_{\hat{f}} \leftarrow v_f^{(n)}$
4: $\quad (W_x)_i = \mathcal{Q}((\partial_x \hat{f})_i)$
5: $\quad \mathbf{A}^{(n)} \leftarrow (1+\gamma)\mathbf{I} + \lambda(\mathbf{D_x}^\top \mathbf{W_x} \mathbf{D_x})$
6: $\quad v_b^{(n)} \leftarrow v_g + \gamma v_{\hat{f}}$
7: $\quad$ `Solve` $\mathbf{A}^{(n)} v_{\hat{f}}^{(n+1)} = v_g^{(n)}$
8: $\quad$ if $\|v_{\hat{f}}^{(n+1)} - v_{\hat{f}}\|_\infty < C_{tolerance}$ then break the loop
9: **end for**
10: $f \leftarrow v_f^{(n+1)}$

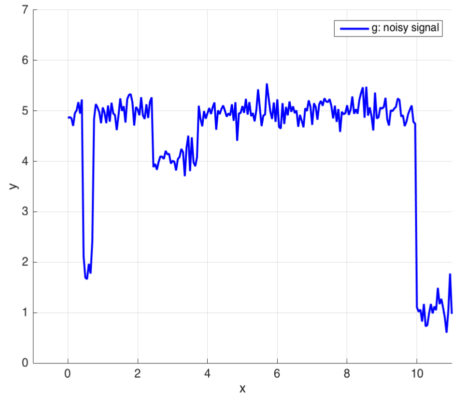
Output: f

In the 8^{th} step of this TV denoising algorithm, the infinite repetition of the algorithm is limited by the maximum number of iterations.

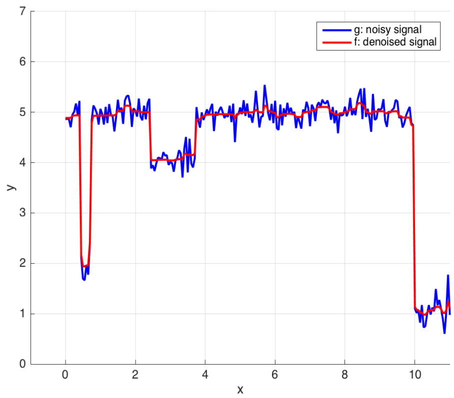
The experimental signals and results related to ℓ_1-norm based total variational denoising with slow step regularizer on 1D signal are shown in Figure 2.7:



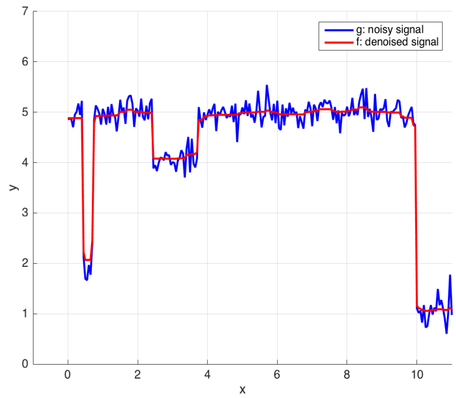
(a)



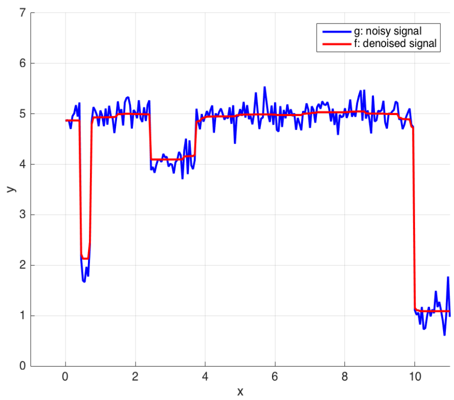
(b)



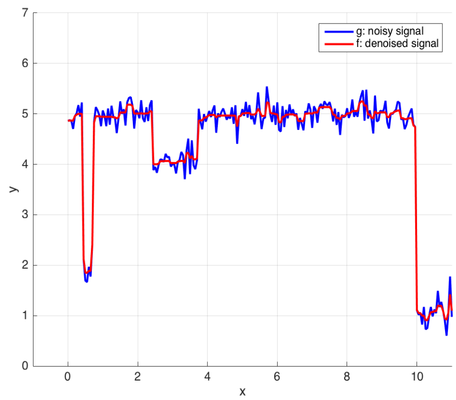
(c)



(d)



(e)



(f)

Figure 2.7: ℓ_1-norm denoising with small step regularizer adaptation results on 1D signal using different smoothing parameters, λ, and constant $\varepsilon = 10^{-5}$ and $\gamma = xxx$ (a) Original signal, f (b) Noisy signal, g (c) ℓ_1-norm denoised signal, $f^{'}$, over the noisy signal, where $\lambda = 0.5$ (d) ℓ_1-norm denoised signal, $f^{'}$ over the noisy signal, where $\lambda = 1$ (e) ℓ_1-norm denoised signal, $f^{'}$, over the noisy signal, where $\lambda = 10$ (f) ℓ_1-norm denoised signal, $f^{'}$, over the noisy signal, where $\lambda = 50$

As it can be seen in the Figure 2.7, the obtained denoised signal by the small step advance is closer to the original noisy signal and it is observed that the accuracy of the proposed method is increased. In addition; although, the smoothness parameter λ is still has an effect on the results, the dependency of the proposed method with this parameter has been reduced.

2.2.2.4 ℓ_1-norm TV Image Denoising

In this section, a total variational based image denoising approach is explained which is commonly discussed in the signal and image processing communities for many times [49, 40, 41]. This approach is stated as an optimization problem of a ℓ_1-norm based cost function minimization using quadratic and linear approximation. In order to estimate despeckled image, f, the minimization of the proposed cost function is defined as follows;

$$f^* = \underset{f}{\operatorname{argmin}}\, J(f) \,, \tag{2.37}$$

where f^* represents the desired despeckled image and $J(f)$ is the proposed variational cost function which is defined as in the following Equation 2.38;

$$J(f) = \sum_{i=1}^{N} (f_i - g_i)^2 + \lambda |(\nabla f)_i| \,, \tag{2.38}$$

where g denotes the original noisy signal. i and N indicate the pixel index and pixel count in the image, respectively. ∇ represents the gradient operator and λ is a positive constant that controls the balance between removing the noise from signal and preserving the signal content and also determines the smoothness level of the denoised signal. In the proposed cost function, the first term is called as data fidelity term which provides that denoised image is similar to the noisy signal given as input. Because of the using quadratic intensity difference, it is not allowed that many pixels drastically change their values so that diminishing salient edges are automatically prevented [39]. The second term, ℓ_1-norm TV regularizer, is added to preserve details such as large scale edges, corners and small sized objects while penalizing the gradient changes on the image. In this term, since the image has two dimensions, ∇ gradient operator is expanded as partial derivatives, ∂_x and ∂_y, of the images with respect to x and y in Equation 2.39 as follows;

$$J(f) = \sum_{i=1}^{N}(f_i - g_i)^2 + \lambda\big(|(\partial_x f)_i| + |(\partial_y f)_i|\big) \ . \tag{2.39}$$

In order to estimate the despeckled image, the cost function given in 2.39 is minimized by taking the derivative of the cost function and equalizing it to zero. However, the cost function is not differentiable because of the ℓ_1-norm based term. ℓ_1-norm is known as the least absolute function and the derivative of an absolute function cannot be taken directly because of discontinuity at 0. As explained at section 2.2.2.2 in detail and shown in Equation 2.23; these ℓ_1-norm based TV regularization terms in Equation 2.39 are quadratically approximated in Equation 2.40;

$$\begin{aligned} |(\partial_x f)_i| &\approx (W_x)_i(\partial_x f)_i^2, \quad (W_x)_i = \mathcal{Q}((\partial_x \hat{f})_i) \ , \\ |(\partial_y f)_i| &\approx (W_y)_i(\partial_y f)_i^2, \quad (W_y)_i = \mathcal{Q}((\partial_y \hat{f})_i) \ , \end{aligned} \tag{2.40}$$

where $\hat{f}$ is a proxy for f. $(W_x)_i$ and $(W_y)_i$ are the constant denominators of the quadratic approximations which are evaluated at $\hat{f}$.

In addition, the slow-step regularization term is added to force the solution to be close to $\hat{f}$, considering the fact that applied approximations are only accurate around this value [40, 41]. Due to the quadratic approximations, the cost function will be solved iteratively so that the iteration index is added into the cost function. After substituting the approximations in Equation 2.40 into Equation 2.39 and expanding the terms, the cost function is described in Equation 2.41;

$$\begin{aligned} J^{(n)}(f) = \sum_{i=1}^{N}&(f_i - g_i)^2 \\ &+\hat{\lambda}\big((W_x)_i(\partial_x f)_i^2\big) + (W_y)_i(\partial_y f)_i^2 \\ &+\gamma(f_i - \hat{f}_i)^2 \ , \end{aligned} \tag{2.41}$$

where γ indicates the damping parameter which controls how $\hat{f}$ is far away from f. n

represents the iteration index and each term belongs to n^{th} iteration unless otherwise specified, for the sake of simplicity.

Equation 2.41 is strictly convex and differentiable since the first term is a quadratic and convex function and the second term is differentiable. This quadratically approximated differentiable equation can be reorganized in matrix-vector form in Equation 2.42;

$$\begin{aligned}
J^{(n)}(v_f) = & (v_f - v_g)^\top (v_f - v_g) \\
& + \lambda({v_f}^\top \mathbf{L} v_f) \\
& + \gamma(v_f - v_{\hat{f}})^\top (v_f - v_{\hat{f}}) \ ,
\end{aligned} \tag{2.42}$$

where v_f, v_g, and $v_{\hat{f}}$ denote the vector forms of f, g, and $\hat{f}$, respectively. $\mathbf{L}$ is defined as $\mathbf{L} = \mathbf{D_x}^\top \mathbf{W_x} \mathbf{D_x} + \mathbf{D_y}^\top \mathbf{W_y} \mathbf{D_y}$ for a compact representation of the Equation 2.42. Here, $\mathbf{W_x}$ and $\mathbf{W_y}$ indicate the diagonal weight matrices that hold the $(W_x)_i$ and $(W_y)_i$ coefficients, respectively. In addition, in order to take the derivative of an image with respect to x, it is calculated as the difference of adjacent pixel values at the same row. In the same way; while taking the derivative of an image with respect to y, it is calculated as the difference of two adjacent pixels at the same column. However, the right and bottom boundaries of the matrix causes the problem while taking the derivatives. At this point, $\mathbf{D_x}$ and $\mathbf{D_y}$ are the Toeplitz matrices consisting of discrete operators, where derivatives are zero at the right and bottom boundaries of the domain. $\mathbf{D_x}$ and $\mathbf{D_y}$ are designed for partial derivative of the 2D image with respect to x and y, respectively.

Illustration of $\mathbf{D_x}$ Toeplitz matrix which belongs to an image with $4x3$ dimensions is shown below;

$$
\mathbf{D_x} = \begin{bmatrix}
\mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Illustration of $\mathbf{D_y}$ Toeplitz matrix which belongs to an image with $4x3$ dimensions is shown below;

$$
\mathbf{D_y} = \begin{bmatrix}
\mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

This matrix-vector form of the cost function, $J^{(n)}(v_f)$, is differentiable. In order to determine the denoised signal, f, the cost function is minimized applying convex optimization methods. This minimization problem is solved by taking the derivative of the $J^{(n)}(v_f)$ with respect to v_f and setting equal it to zero as follows:

$$\frac{\partial J^{(n)}(v_f)}{\partial v_f} = 0 \, . \tag{2.43}$$

This derivation procedure is processed and a substage is shown in Equation 2.44;

$$2(v_f - v_g) + 2\lambda(\mathbf{L}v_f) + 2\gamma(v_f - v_{\hat{f}}) = 0 \, , \tag{2.44}$$

After applying mathematical derivations and alignments into Equation 2.44, a linear system is obtained at the end of the procedures. The obtained linear system is shown in Equation 2.45;

$$\begin{aligned} \mathbf{A}v_f^{(n+1)} &= b \, , \\ \mathbf{A} &= (1+\gamma)\mathbf{I} + \lambda\mathbf{L} \, , \\ b &= v_g + \gamma v_{\hat{f}} \, , \end{aligned} \tag{2.45}$$

where $\mathbf{I}$ is the identity matrix. Note that, $\mathbf{A}$ and b belongs to n^{th} iteration. In Equation 2.45, $\mathbf{A}$ is sparse, positive definite, 5-point Laplacian matrix. Since $\mathbf{A}$ is positive definite, cost function in Equation 2.42 is strictly convex. This linear system can be efficiently solved using an efficient iterative solver according to Algorithm 3. While evaluating the $v_f^{(n+1)}$, preconditioned conjugate gradient (PCG) with an incomplete-Cholesky (IC) preconditioner is used to solve this linear system in each iteration. The inputs of the algorithm is initially given as follows; maximum number of PCG iterations is 10^2, convergence tolerance of PCG is 10^{-3}, $n_{max} = 10$, and $C_{tolerance} = 10^{-3}$.

Algorithm 3 ℓ_1-norm TV Image Denoising

Input: g, λ, ε, n_{max}, $C_{tolerance}$

1: $v_f^{(1)} \leftarrow v_g \leftarrow g$
2: **for** $n = 1 : n_{max}$ **do**
3: $\quad v_{\hat{f}} \leftarrow v_f^{(n)}$
4: $\quad (W_x)_i = \mathcal{Q}((\partial_x \hat{f})_i)$
5: $\quad (W_y)_i = \mathcal{Q}((\partial_y \hat{f})_i)$
6: $\quad \mathbf{A}^{(n)} \leftarrow (1+\gamma)\mathbf{I} + \lambda \mathbf{L}$
7: $\quad v_b^{(n)} \leftarrow v_g + \gamma v_{\hat{f}}$
8: $\quad$ `Solve` $A^{(n)} v_{\hat{f}}^{(n+1)} = v_g^{(n)}$
9: $\quad$ if $\|v_{\hat{f}}^{(n+1)} - v_{\hat{f}}\|_\infty < C_{tolerance}$ then break the loop
10: **end for**
11: $f \leftarrow vectorToImage({v_f}^{(n+1)})$

Output: f

In the 9^{th} step of this TV denoising algorithm, the repetition of the algorithm is limited by the maximum number of iterations.

The first set of experimental images and results related to ℓ_1-norm based total variational image denoising are shown in Figure 2.8. These tests are performed on synthetically generated image. The original image, is shown in Figure 2.8a, is created with $400x400$ dimensions with various geometric figures such as squares, circles, and triangles in different sizes. The noisy image in Figure 2.8b is obtained by adding Gaussian noise with parameters zero mean, $\mu = 0$ and standard deviation, $\sigma = 25$, to the original synthetic image.

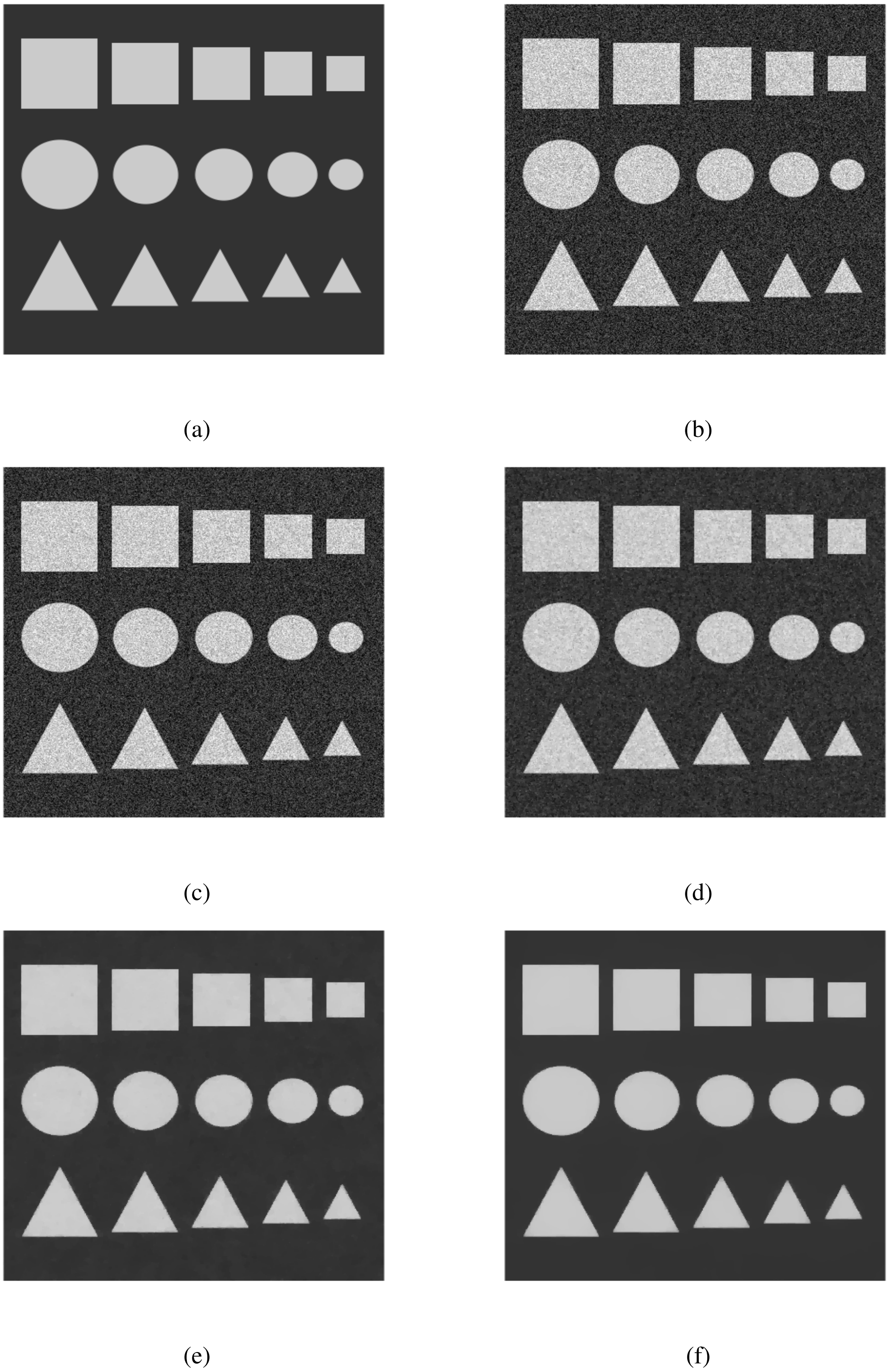


Figure 2.8: ℓ_1-norm total variation based image denoising using different smoothing parameters, λ, and constant $\varepsilon = 10^{-5}$ and $\gamma = 1$ (a) Original synthetic image (b) Noisy synthetic image (c) ℓ_1-norm denoised image, where $\lambda = 1$ (d) ℓ_1-norm denoised image, where $\lambda = 10$ (e) ℓ_1-norm denoised image, where $\lambda = 25$ (f) ℓ_1-norm denoised image, where $\lambda = 50$

As it can be seen in Figure 2.8, when the smoothness level, λ is increased (from Figure 2.8c to Figure 2.8f) the noise amount decreases. This situation can be understood from the quantitative noise ratios; meanwhile, the generated noisy signal has signal to noise ratio (SNR) of 13.30 db, SNR of denoised signal is increasing as 13.87 db, 22.20 db, 28.55 db, and 29.42 db.

The second set of experimental images and results related to ℓ_1-norm based total variational image denoising are shown in Figure 2.9. These tests are performed on Lena image, which is commonly used in image processing area and an image of me and my sister.

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

Figure 2.9: ℓ_1-norm total variation based image denoising using different images with same parameters $\lambda = 25$, $\varepsilon = 10^{-5}$ and $\gamma = 1$ (a) Original Lena image (b) Original my image (c) Noisy Lena ımage (d) Noisy my image (e) ℓ_1-norm denoised Lena image (f) ℓ_1-norm denoised my image

The noisy images in Figure 2.9c and 2.9d are obtained by adding Gauissan noise with parameters, zero mean, $\mu = 0$ and standard deviation, $\sigma = 25$, to the original Lena image and my family image seen in Figure 2.9a and 2.9b, respectively. ℓ_1-norm total variation based image denoising method is applied with same parameters to the noisy images. Figure 2.9e is the denoised image with parameters $\lambda = 25$, $\varepsilon = 10^{-2}$ and $\gamma = 1.0$. The obtained SNR of noisy signal is 14.55 db, meanwhile SNR of denoised signal is 23.50 db in test with this parameters. Figure 2.9f is the denoised image with parameters $\lambda = 25$, $\varepsilon = 10^{-2}$ and $\gamma = 1$. The obtained SNR of noisy signal is 12.98 db, meanwhile SNR of denoised signal is 27.82 db in the second test.

CHAPTER 3

DIGITAL SURFACE MODEL (DSM)

In the first section of this chapter, an overview of digital surface model (DSM) for remotely sensed data is given. This section focuses on DSMs since two multi-temporal registered DSMs are the input data of the proposed change detection method of this study. The following sections explain employed technologies for DSM generation. In the second section, LIDAR systems, a relatively new technology used to acquire 3D elevation data, are described, in detail. It is also mentioned how data is acquired and which applications are developed using this data. Finally, a DSM generation approach based on variational methods is proposed in this section. The third and fourth sections mention photogrammetry and SAR interferometry based DSMs, respectively.

3.1 An Overview of DSM Generation in Literature

3D modeling of the earth's surface has a great significance worldwide. The main reason of this significance is the common use of the height information for many applications in the fields of military operations, remote sensing, geology, mining industry, landscape architecture, agriculture, forest management, urban and regional planning, city modeling, traffic control, and civil engineering [31]. Digital elevation model (DEM) is 3D representation of the surface of the earth corresponding to elevation measurement as a raster layer or a triangular irregular network (TIN) in a digital form. DEM is used as a generic form and has two types which are digital surface model (DSM) and digital terrain model (DTM). DSM is described as the surface of the earth containing topography, all vegetation and man-made objects such as buildings, bridges, and poles. DTM is another kind of DEM that is described as the bare earth's surface including geographical elements and natural features such as rivers, ridge lines but not containing man-made objects and vegetation [59, 22]. The difference between DSM and DTM can be seen in Figure 3.1.

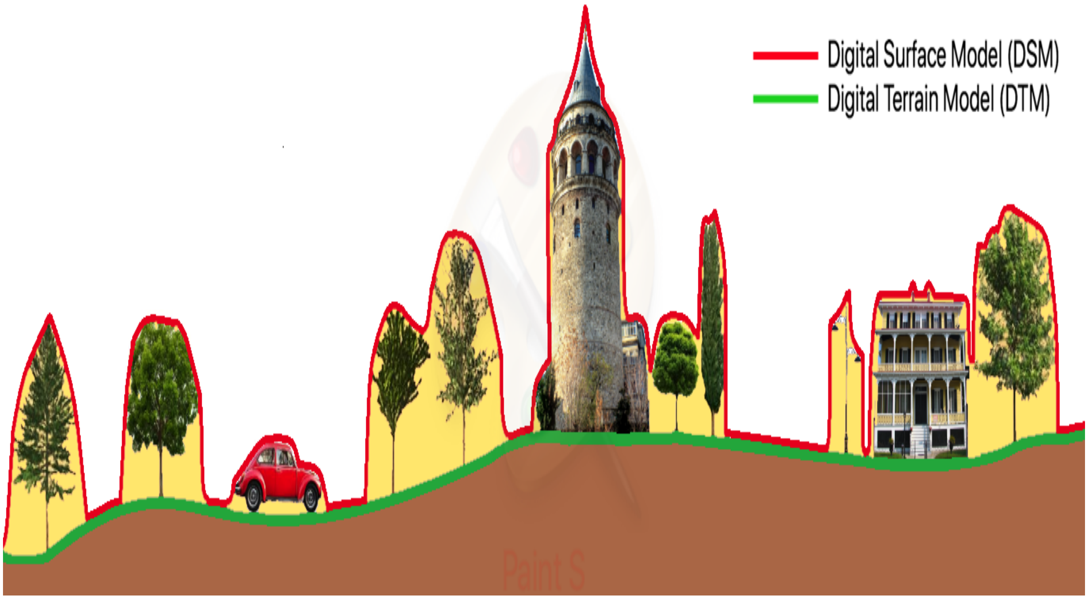


Figure 3.1: A simple representation of DSM and DEM over a region

DSM and DTM data can be generated using both active and passive sensors with the help of ground, aerial or satellite technologies such as LIDAR, stereo photogrammetry, and SAR interferometry [9].

3.2 LIDAR based DSM

LIDAR is one of the most favorable source for collecting terrain data and generating surface models. LIDAR term is acronym for Light Detection and Ranging or Light Imaging, Detection and Ranging and is called as sometimes LADAR, LiDAR, lidar, Lidar, laser scanning, or 3D scanning [16]. LIDAR is an active remote sensor that is used to describe the surface of the earth and that matures and evolves day by day. The active sensor is a remote sensing device that reflects the energy, whose origin is not any natural source, to the earth and receives it back from the earth's surface, according to the definition [58, 29]. In LIDAR technology, sensor transmits laser pulses to the target, which can be described as any object that reflects the energy on the earth surface, too frequently like a hundred thousand times per second. The whole laser pulse or a part of it which illuminates the target is reflected from the target according to the reflectivity and this process triggers the receiver in the sensor. The path of laser pulse between the receiver and the target is demonstrated in Figure 3.2.

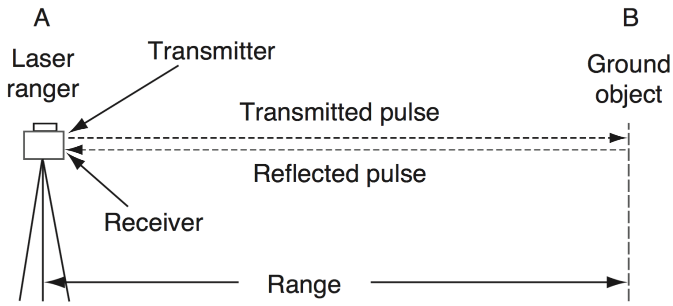


Figure 3.2: The transmission path of the laser pulse between the laser receiver and the target on the ground

By this way, the receiver calculates the elapsed time between the initiation and the return of that pulse. It measures the distance between the target and the sensor using the knowledge of velocity of light in the form of a pulsed laser and calculated elapsed time. This distance can be called as the *range* and is calculated according to following equation 3.1;

$$R = \frac{t_e * v_c}{2} \ , \tag{3.1}$$

where distance is represented as R, t_e is the elapsed time and v_c is the speed of the light which is a known value. The equation must be divided by two to get the actual distance between the sensor and the target. With the combination of other exact information supported by other components of the sensor, the accurate, precise and three-dimensional information is produced about the surface and shape of the earth and its characteristic features [30, 48, 24].

3.2.1 LIDAR Platforms

It is very important to know the platform on which sensor is placed and its' position to ensure that accuracy of any image processing algorithm developed with LIDAR data. LIDAR sensors can be classified to three categories according to mounted platform: space-borne platforms, airborne platforms and ground-based platforms. When the sensor is mounted to an airborne device such as airplane, helicopter or drone, this sensor belongs to Aerial Laser Scanner (ALS) category. In the ground-based category, if it is placed to a static object on the ground like a tripod or mast, it is called as Terrestrial Laser Scanner (TLS) and if placed to a dynamic object like a moving vehicle, it is called as Mobile Laser Scanner (MLS). The last class is called as Satellite Laser Scanner (SLS) because the sensor is placed to a satellite vehicle. Each one has own advantages when compared to the produced data that are used in practice.

3.2.2 Components of LIDAR System

In LIDAR systems, there exist some main components to collect and store the geographically coordinated data of the terrain in an accurate way. In order to achieve the most precise result, each component must individually operate successfully and the combination of these components also must work in a harmony [53, 51]. The system components can be seen in Figure 3.3.

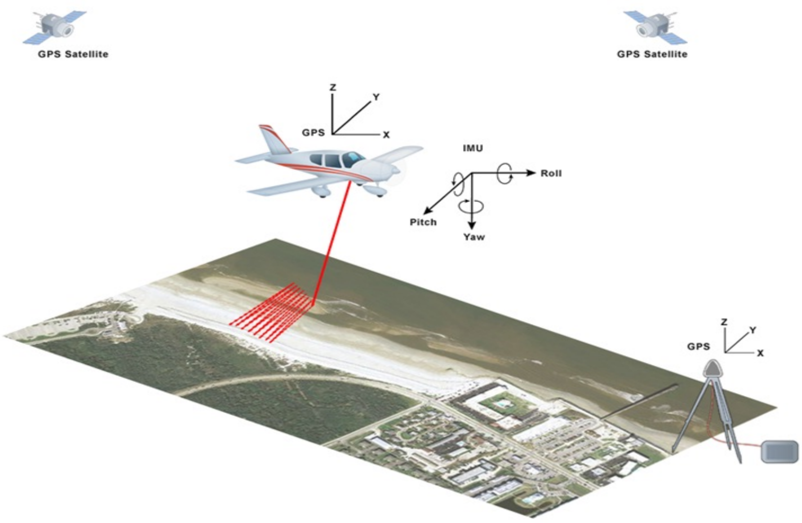


Figure 3.3: The main components of LIDAR system [56]

1. **Laser Source and Laser Detector:** After a consistent stream of laser pulses is generated by laser scanner and fired from the scanning mechanism, a mirror spins or scans them to reflect towards to the surface. The laser pulses reflected back from the target surface is received and recorded by an electro-optical receiver. This process is frequently repeated to obtain highly accurate, dense and automated data.

2. **Timing Electronics:** The highly sensitive timer measures the elapsed time between the separation of the laser pulse from the scanner and the return to the scanner, exactly.

3. **Global Positioning System (GPS):** When the LIDAR data is acquired, it is highly critical that the data should be geographically positioned accurately. In order to provide geo-referenced data, GPS that is now a widely used navigation system is integrated to LIDAR systems.

4. **Inertia Measurement Unit (IMU):** While calculating the accurate range between the surface, the precise angle and location of the scanner is necessary. The inertial navigation system has three components which are accelerometer, gyroscope, and magnetometer. With the help pf these components, the IMU records the pitch, roll, and yaw of the scanner relative to the ground.

5. **Computer Processing Resources:** With the embedded and integrated software tools, mission planning can be provided to determine the route of the vehicle. Collected raw data of the sensor can be displayed on a screen, pre-processed and post-processed like extracting, geo-referencing, classifying and analyzing. In addition to dedicated computer and software tools, a data recording system such as a hard disk is needed to collect and combine all of measured data.

3.2.3 LIDAR Data

Since laser systems use own source of energy to emit laser beams, data collection can be operated in anytime of day like daylight, overcast or night independent from the sun. However; since the LIDAR sensors cannot penetrate the clouds, they can only collect the data when the sensors are below the clouds. Moreover, the sensors are affected negatively from bad weather conditions such as rain, fog, mist, smoke and snowstorm [46]. The collection of returns with three-dimensional coordinates as called as point cloud. Features in the landscape can be represented according to the return pulses. For example, while the first return represents the top of the trees or buildings, generally the last return can be associated with the ground especially in forestry regions. The quality of the LIDAR dataset is determined by resolution which can be defined as number of laser pulses per unit area and it depends on the speed of the aircraft, the flying altitude, laser pulse emission rate and field of view (FOV) of the system. In order to acquire high resolution data which has greater point count, the aircraft flies at low altitude with high frequency rate and narrow FOV. In addition to coordinates of return points, the strength of the return pulses is recorded as intensity. The intensity values are dependent to reflectivity of the object surface which laser pulse hits. Intensity values can be used to extract features from the data and the classification of points. In addition, when a camera is mounted to the LIDAR platform, the red, green and blue image channel values are also kept for each point cloud [54, 51].

The commonly used public file format to store three-dimensional point cloud LIDAR data is LAS which is information specific to nature of the LIDAR data. Even though there exist many LIDAR point data record formats according to used LIDAR system, one of commonly used formats is shown as table 3.1:

Table 3.1: Point Data Record Format

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits (bits 0,1,2)	3 bits	*
Number of Returns (given pulse)	3 bits (bits 3,4,5)	3 bits	*
Scan Direction Flag	1 bit (bit 6)	1 bit	*
Edge of Flight Line	1 bit (bit 7)	1 bit	*
Classification	unsigned char	1 byte	*
Scan Angle Rank	char	1 byte	*
User Data	unsigned char	1 byte	
Point Source ID	unsigned short	2 byte	*
Red	unsigned short	2 byte	*
Green	unsigned short	2 byte	*
Blue	unsigned short	2 byte	*

3.2.4 LIDAR Applications

It is obvious that, LIDAR systems are widely used in many exciting areas like environmental, military, civil, security, historical and cultural areas because it is possible to acquire an accurate, fast and versatile three dimensional data. Some applications are developed using LIDAR systems are listed as follows:

- **Vegetation applications:** Forest monitoring for natural disasters like fire, flood, detection of deforestation, measurement of canopy heights and closure, characterization of canopy density, estimation of biomass volume, [34], estimation of tree height and density, mapping of individual tree and crown, estimation of leaf area index [57], view of agricultural land, crop mapping,
- **Environmental applications:** Determination of coastal change, damage as-

sessment after natural disasters like flood, oil and gas exploration, hurricanes, earthquakes, landslides, flood mapping, measurement of wetlands, measurement of moving glacial regions including snow and ice covered areas, carbon dioxide, sulphur methane, noise and light pollution prediction [62], bird population modeling, land cover classification [34]

- **Urban applications:** Road, building, waterway extraction and mapping, 3D city model for planning, transportation planning, street tree mapping, wireless telecommunication localization, tracking of traffic [62]

- **Other applications:** Weather forecast, determination of clouds, measurement of wind, tracking of climate change and air quality, border control, aiding in the planning of archeological regions, calculation of ore volumes, cellular network planning, recording of accidents and crime, designing, constructing and restoring buildings, navigation systems, detection of speed of vehicles, identification of obstacles in military [3]

3.2.5 DSM Generation using Variational Methods

In this section, a variational based DSM generation method is proposed using point cloud data [5]. With the help of the LIDAR sensors, it is possible to collect high quality three dimensional coordinates of points in terms of accuracy and density, directly. In addition, it is the most preferable way to generate DSM using airborne laser systems because planning a flight and collecting data from difficult and large regions such as forests, vegetation-heavy areas and inclined territories is easy and effective. With these advantages, it is possible to generate highly detailed elevation raster map with high resolution. On the other hand; because of the highly detailed data, a huge volume of data has to be dealt with. Storage, processing and manipulation of this data is difficult and expensive [9, 36]. Even though, there are many methods to generate DSM from LIDAR data in the literature [6, 23, 55, 21]; DSM generation method is suggested in this section and used as preprocessing step for change detection in this thesis.

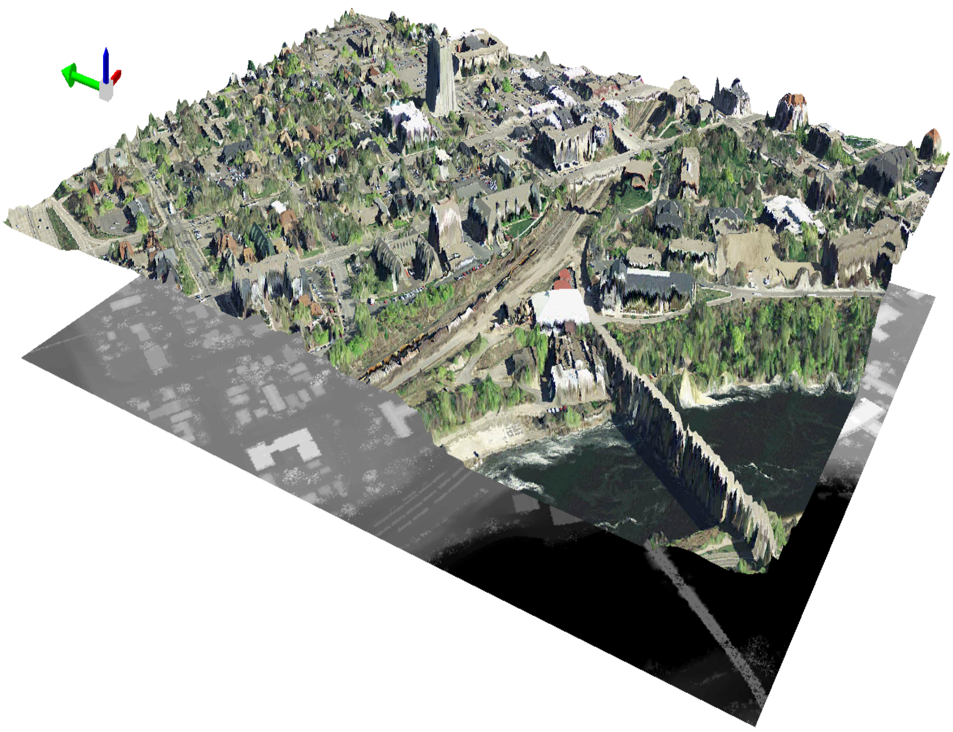


Figure 3.4: An example of generated DSM from LIDAR point cloud

In this proposed method, construction of accurate DSM is defined as an optimization problem and it is solved by minimizing a cost function, which is defined in equation 3.2:

$$H^* = \underset{H}{\operatorname{argmin}}\, J(H) \;, \tag{3.2}$$

where, H indicates the surface model and $J(H)$ represents the proposed cost function which is minimized with respect to H. The variational cost function is formulated in the following equation 3.3:

$$J(H) = \sum_{i}^{N} \mathbf{K}_i (H_i - M_i)^2 + \lambda |(\nabla H)_i|_1^1 \;, \tag{3.3}$$

where, M is a raster image which is generated as a height map by projecting LIDAR point cloud data into 2D grid map according to a resolution. The raster image generation is explained in Algorithm 4. H represents the aimed digital surface model. N indicates the number of pixels in the images. ∇ denotes the gradient operator and since the surface model has two dimensions the derivatives of the map are taken in two directions, x and y. Therefore, the derivative operator is indicated as ∂_x and ∂_y. In order to determine which cell has at least a laser pulse, a 2D indicator matrix, $\mathbf{K}$, is created with the same dimensions as the raster image. i indicates the indices of 2D M, H and and $\mathbf{K}$ matrices. In the raster image, the cells are checked for whether any laser point has fallen into the cell of raster image or not. If the cell has at least one laser pulse projected from point-cloud to 2D map M, then the pixel that has the same x and y indices of the indicator matrix is set to 1, otherwise 0.

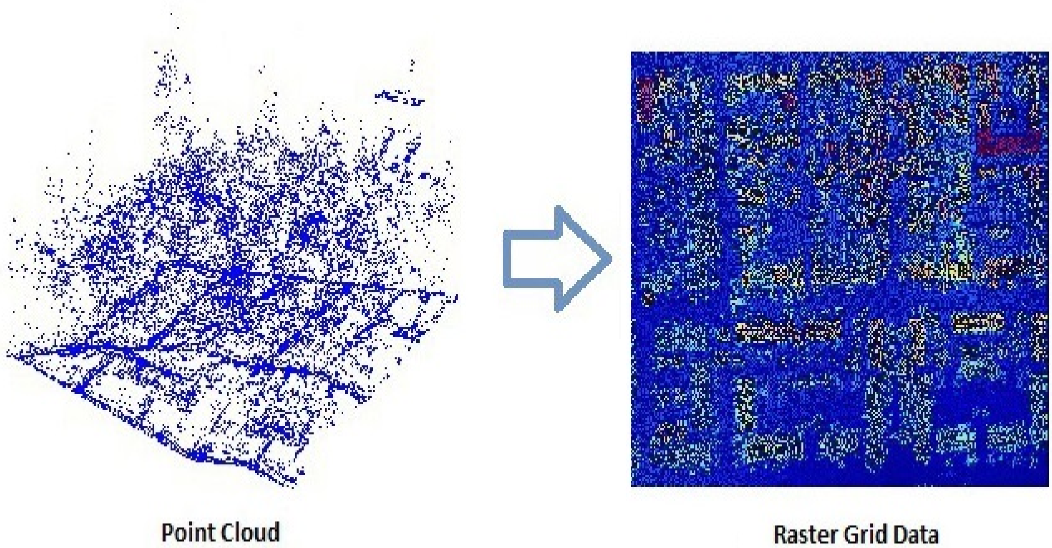


Figure 3.5: The flow diagram of raster grid data construction

Algorithm 4 Raster Image Generation, $las2grid$

Input: Point Cloud P, Grid Size (Resolution) R

```
Find the minimum and maximum coordinates of; P minX,
  maxX, minY, maxY
nRows ← ceil((maxX − minX)/R)
nCols ← ceil((maxY − minY)/R)
Create zero matrix with size estimated number of rows
  and cols, G
for i = 1 : pointSize do
    xIndex ← (P.X(i) − minX)/R
    yIndex ← (P.Y(i) − minY)/R
    if G[yIndex][xIndex] < P.Z(i) then
        G[yIndex][xIndex] = P.Z(i)
    else
        Do nothing
    end if
end for
```

Output: G

In Algorithm 4, the resolution of the raster image can be determined in two different ways. In the first method, the resolution is given as an input by the user as a grid size, R. Secondly, the resolution can be calculated automatically determining an optimum grid size according to an algorithmic approach which is a similar to the algorithms used in compressing sensing (CS), in the literature [19]. It is a signal reconstruction method commonly used in mathematics, electrical engineering, computer science and physics areas. Despite the development of machines with extraordinary computing power for each application to be used in various areas such as image processing, computer vision, medical imaging, remote surveillance and genetics, there exists still a tremendous challenge in handling and processing all kinds of signals. While multi-dimensional data is being processed, the signal is compressed to give the most concise representation of the underlying phenomenon with an acceptable distortion in order to achieve cost reduction and to overcome the computational inefficiency. The CS method, which is one of the most popular methods used to express the signal as more sparse or compressed provides a considerable reduction in sampling and computation cost. The underlying idea of CS is that a finite-dimensional signal with a sparse appearance can be reproduced from a small, linear and nonadaptive set [15].

In this proposed method, though the CS method is not directly applied, the very large point cloud data is compressed to occupy a smaller space before data is started to being processed by being inspired from the main idea of CS method. In the gridding method, it is possible to reconstruct the whole data by interpolating the points falling between the cells. With this method, even if there are no LIDAR points in the cells of the grid, it is enough that about 25% of the whole pixels are being full in order to represent the original data without deteriorating. According to Algorithm 4, the raw LIDAR data is loaded as the input, then x, y, and z values are read and converted to point cloud. Using the specified resolution, the point cloud data is divided into peer cells using the minimum and maximum coordinate information of the points in the data. The elevation value which is the highest value among the laser points falling into each cell is assigned as the value of that pixel. In case of absence of points at a cell, the value of pixel is determined as *no value*. Therefore, using a dense 3D point cloud data, a 2D grid data that is meaningful, sufficient and sparse is produced.

In the proposed cost function, the first term, which is called as **data fidelity** term, allows these H and M maps resemble each other, which means that result surface model, H, is similar to original height map, M. The second term is defined as **regularization term** which is used to penalize the gradient changes on surface map in ℓ_1-norm manner. The effect of the regularizer term is based on λ parameter which is a positive scalar value and the smoothness parameter, which determines the smoothness level. The behaviour of the proposed cost function changes depending on the indicator matrix. In the case $\mathbf{K}_i$ equals to 1, the system performs a smoothing on those cells while preserving the details because of the ℓ_1-norm TV regularization term. In the other case; where cells have no data, it is disabled and the optimizer tries to interpolate empty cells in the M matrix, since data fidelity term is multiplied by 0. This diffusion operation can be called as interpolation [35]. The optimization operation forces a softening, while preserving the details of the produced surface model in the laser point located cells of the indicator matrix, but in the absence of the point, to fill the relevant gaps while smoothing the map [43].

The formulation can be rewritten with the anisotropic expression in 2D as given in equation 3.4:

$$J(H) = \sum_{i}^{N} \mathbf{K}_i (H_i - M_i)^2 + \lambda \big(\mid (\partial_x H)_i \mid + \mid (\partial_y H)_i \mid \big) \; . \tag{3.4}$$

In order to solve this problem and to find the optimum surface model, it is necessary to take the derivative of the cost function and equalize it to zero, since the cost function is convex. However, in this cost function, the derivative of the regularizer term cannot be taken since it is ℓ_1-norm. The derivative of an absolute function cannot be taken because it is not differentiable at sharp point that is 0 which is explained in section 2.2.2.2. According to Equation 2.23; a quadratic approximations of partial derivations of H with respect to x and y are written as in Equation 3.5 [43];

$$\begin{aligned} |(\partial_x H)_i| &\approx (W_x)_i (\partial_x H)_i^2, \quad (W_x)_i = \mathcal{Q}((\partial_x \hat{H})_i) \; , \\ |(\partial_y H)_i| &\approx (W_y)_i (\partial_y H)_i^2, \quad (W_y)_i = \mathcal{Q}((\partial_y \hat{H})_i) \; , \end{aligned} \tag{3.5}$$

where, $(W_x)_i$, and $(W_y)_i$ are the coefficients for the quadratic approximations of the absolute terms which are evaluated at $\hat{H}$, proxy of H.

In addition, the slow-step regularization term is added to force the solution to be close to $\hat{H}$, considering the fact that approximations are only accurate around this value, as explained in 2.2.2.3, in detail. Due to the quadratic approximations, the ℓ_1-norm regularization term of the cost function is approximated numerically. In this mathematical approach, it is aimed to find the best fit value for the objective function and it is performed iteratively by fitting a function to each value of the surface map. After applying the approximations in Equation 3.5 into the Equation 3.4, the cost function is defined in Equation 3.6;

$$\begin{aligned} J^{(n)}(H) = \sum_{i}^{N} & \mathbf{K}_i (H_i - M_i)^2 \\ & + \lambda \big((W_x)_i (\partial_x H)_i^2 + (W_y)_i (\partial_y H)_i^2 \big) \\ & + \gamma (H_i - \hat{H}_i)^2 \; , \end{aligned} \tag{3.6}$$

where, γ denotes the damping parameter and n is the iteration number, which is used to solve iteratively the cost function due to quadratic approximations. In the equation 3.6, in order to keep the expression of the equation simple, the number of iteration is not added to each term of the right side of the equality. Unless indicated otherwise, each H and $\hat{H}$ terms belong to n^{th} iteration. Since $(W_x)_i$ and $(W_y)_i$ are assumed to be constant and all the other terms are already quadratic, the new representation of the model is differentiable. This differentiable cost function can be converted into matrix-vector form which is as in Equation 3.7;

$$\begin{aligned} J^{(n)}(v_h) &= (v_h - v_m)^\top \mathbf{K}(v_h - v_m) \\ &\quad + \lambda(v_h^\top \mathbf{L} v_h) \\ &\quad + \gamma(v_h - v_{\hat{h}})^\top (v_h - v_{\hat{h}}) \,, \end{aligned} \tag{3.7}$$

where, v_h, $v_{\hat{h}}$, and v_m are the vector representations of the H, $\hat{H}$, and M, respectively. $\mathbf{K}$ is a diagonal indicator matrix where all are in size of NxN where N is the number of cells in the grid ($width \times height$). $\mathbf{L}$ is defined as $L = \mathbf{D_x}^\top \mathbf{D_x} \mathbf{D_x} + \mathbf{D_y}^\top \mathbf{W_y} \mathbf{D_y}$, where $\mathbf{W_x}$ and $\mathbf{W_y}$ are the diagonal weight matrices that hold the $(W_x)_i$ and $(W_y)_i$ coefficients on their diagonals, respectively. $\mathbf{D_x}$ and $\mathbf{D_y}$ are the Toeplitz matrices that are designed for partial derivative of the 2D surface map with respect to x and y, respectively. Toeplitz matrices are expressed in detail and exemplified for a sample matrix in Section 2.2.2.1 and Section 2.2.2.4. Although not specified here for the sake of simplicity of the representation, surface map belongs to n^{th} iteration in the vector format.

Since the objective function, $J^{(n)}(v_h)$, is strictly convex and differentiable, the minima can be found by taking the derivative of $J^{(n)}(v_h)$ with respect to v_h. At this point, the derivative of the matrix-vector form of the cost function can be taken with respect to v_h and set it to zero that is shown in Equation 3.8.

$$\frac{\partial J^{(n)}(v_h)}{\partial v_h} = 0 \,. \tag{3.8}$$

All the operations related to taking the derivative and setting it to zero are performed, as given in Equation 3.9:

$$2\mathbf{K}(v_h - v_m) + 2\lambda v_h \mathbf{L} + 2\gamma(v_h - v_{\hat{h}}) = 0 \tag{3.9}$$

Finally, a linear system of equations is acquired as in Equation 3.10;

$$\begin{aligned} \mathbf{A}v_h^{(n+1)} &= b\ , \\ \mathbf{A} &= (1+\gamma)\mathbf{K} + \lambda\mathbf{L}\ , \\ b &= \mathbf{K}v_m + \gamma v_{\hat{h}}\ , \end{aligned} \tag{3.10}$$

where, n represents the iteration number. $\mathbf{A}$ and b belongs to n^{th} iteration. As it can be easily understood from the equations, the corresponding linear system updates the result surface model depending on $\mathbf{A}$ and v_b in each iteration. Since $\mathbf{A}$ is a positive definite, 5-point Laplacian and sparse matrix, this linear system can be efficiently solved using an efficient iterative solver. In order to find the surface model in Equation 3.10, preconditioned conjugate gradient (PCG) with an incomplete-Cholesky (IC) preconditioner is used.

Algorithm 5 DSM Generation

Input: Point Cloud P, Grid Size R, λ, ε, $n_{max} \leftarrow 5$

$G \leftarrow las2grid(P, R)$

$M_i \leftarrow max(G_i)$, where i is the index of matrix

$\mathbf{K}_i = \begin{cases} 0, & \text{if } G_i = \emptyset \\ 1, & \text{if } G_i \neq \emptyset \end{cases}$

$v_h^{(1)} \leftarrow v_m$

for $n = 1 : n_{max}$ **do**

    $(W_x)_i = \mathcal{Q}((\partial_x \hat{H})_i)$

    $(W_y)_i = \mathcal{Q}((\partial_y \hat{H})_i)$

    $\mathbf{A}^{(n)} = (1+\gamma)\mathbf{K} + \lambda\mathbf{L}$

    $v_b^{(n)} = \mathbf{K}v_m + \gamma v_{\hat{h}}$

    Solve Equation 3.10 to find $v_h^{(n+1)}$

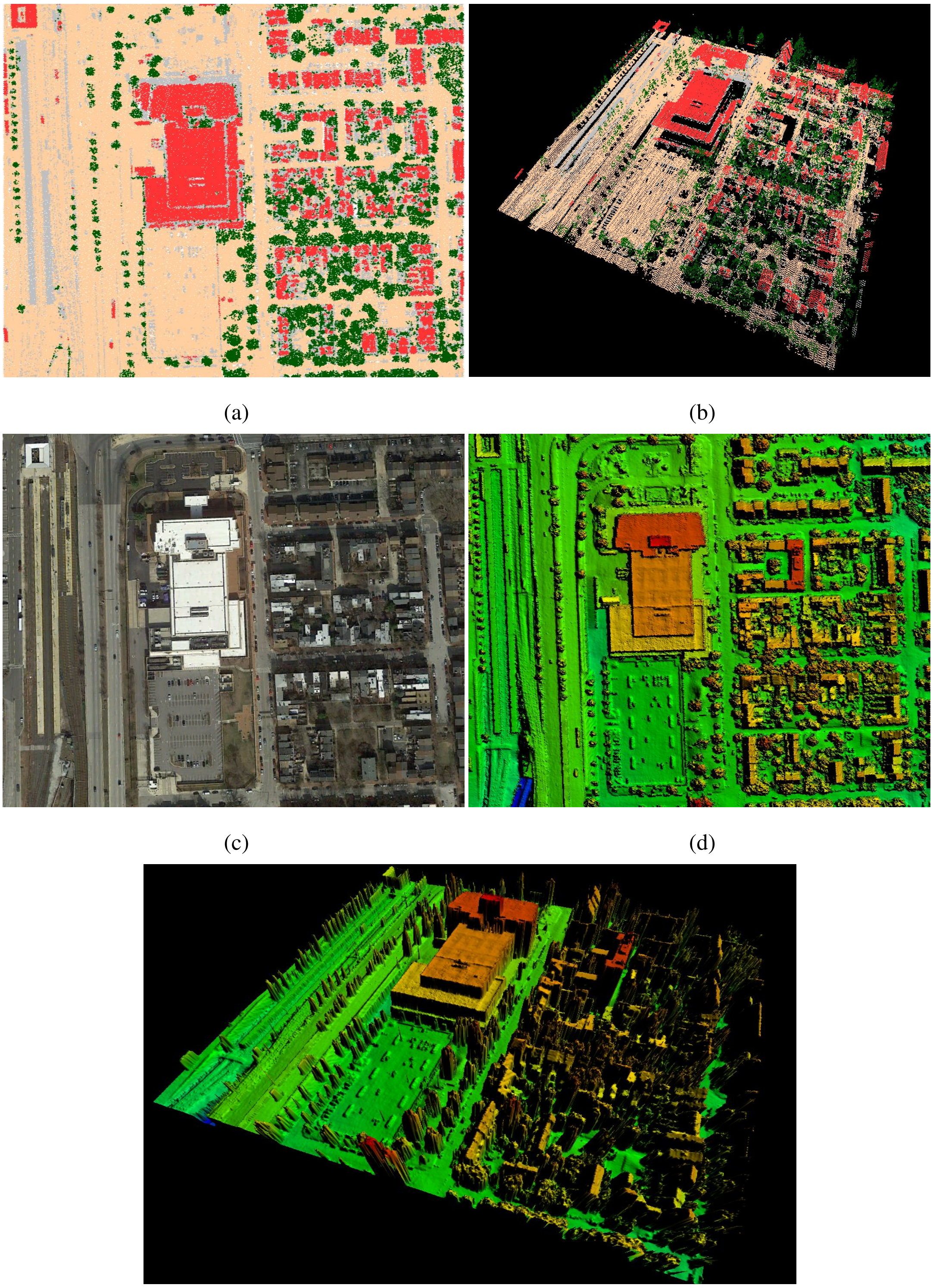
end for

$H \leftarrow vectorToMatrix(v_h^{(n+1)})$

Output: H

3.2.6 Experimental Results of DSM Generation Method

In this section, the experimental results, related to variational method based DSM generation on a sample data, are presented. The sample point cloud, which is collected from the Baltimore region of Maryland, USA. There exist $1.650.677$ points in the whole original data and the total area covered is 1.339 square kilometers. Approximately, the LIDAR point density is equal to 1.57 samples per square meter and LIDAR point spacing is 1 meter. The measured error at the points is 18.5 centimeters and the maximum measured height on the data is 96.67 meters. In addition, this data does not include the classification information of points. In order to visually present the results of the proposed method within the scope of this study, a small patch containing trees, buildings and roads was cropped instead of working with all the data. In this patch, there are 170.318 points and LIDAR point density is 1.64 samples per square meter. The used LIDAR data sample, the electro optic image of the same region and the extracted DSM is shown in Figure 3.6:

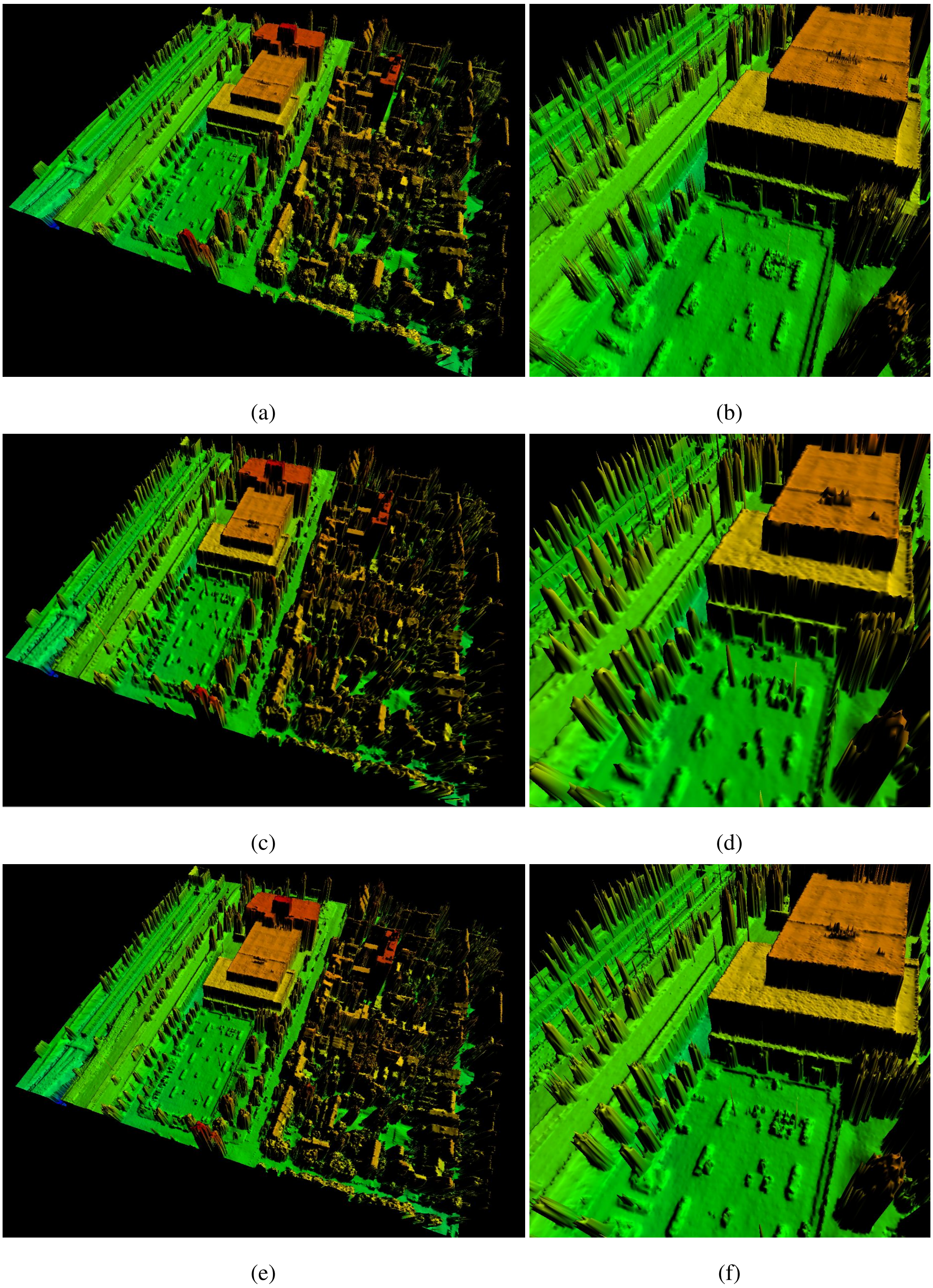


(e)

Figure 3.6: (a) 2D view of the sample point cloud data (b) 3D view of the sample point cloud data (c) Electro-optic image of the same region (d) 2D view of the extracted DSM (e) 3D view of extracted DSM, $R = 0.425$ and $\lambda = 0.01$

In Figure 3.6a and Figure 3.6b, the point cloud data is classified using Global Mapper software tool was used to make the trees and buildings look better visually. The electro-optic image, Figure 3.6c, is taken from Google Earth using the KML file which represents the boundaries of the region. The DSM generation result is stored as an image with float type. The float image is loaded to Global Mapper and visualised as 2D and 3D form in Figure 3.6d and Figure 3.6e.

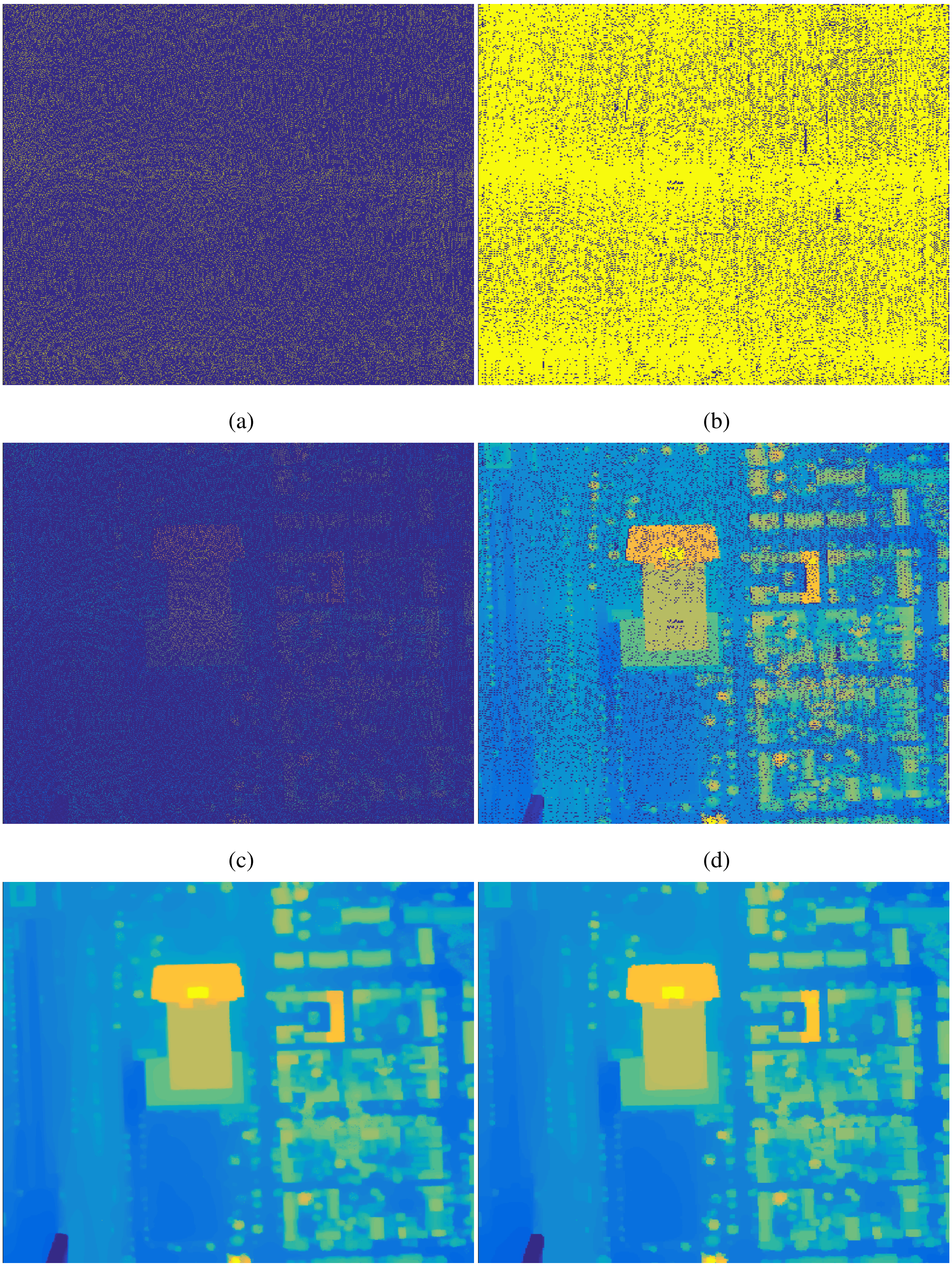
The digital surface models generated by the proposed method depend on two different parameters, namely, the resolution used in the grid structure and the level of smoothing in the variational cost function. Firstly, DSM produced with different resolution values are shown in Figure 3.7 to analyse the effect of resolution on the result of the suggested method.



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

Figure 3.7: The effects of resolution on the 3D DSMs (a) DSM, $R = 0.25$ (b) A region that is zoomed on DSM, $R = 0.25$ (c) DSM, $R = 1.0$ (d) A region that is zoomed on DSM, $R = 1.0$ (e) DSM, $R = 0.425$ that is optimum resolution (f) A region that is zoomed on DSM, $R = 0.425$; $\lambda = 0.01$

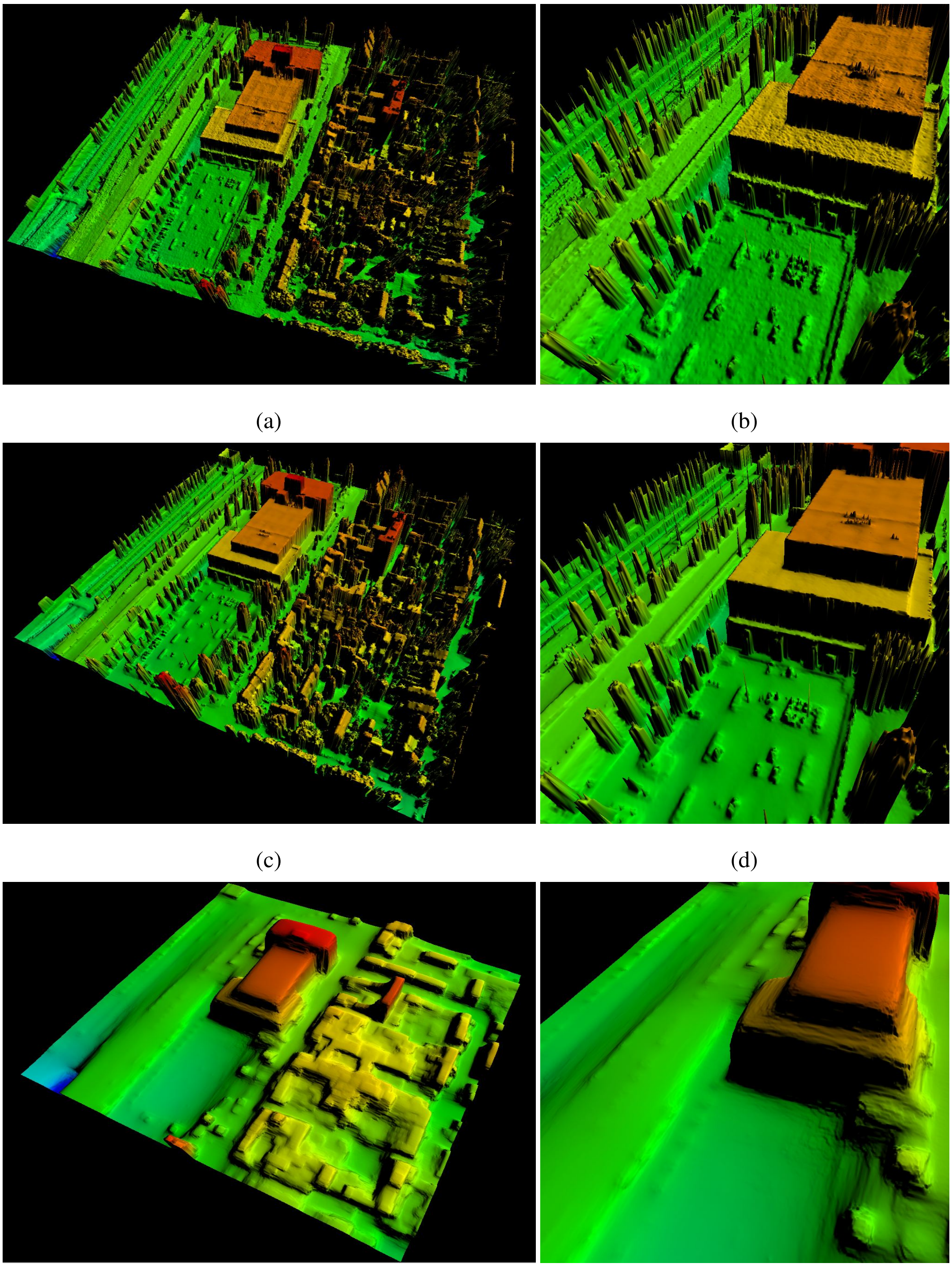
In the first step of the proposed DSM generation method, changing the resolution value of the raster image used as the grid size in the algorithm causes a change in the number of LIDAR points on a cell. When the resolution value is increased which means grid size is enlarged, the area covered by the cell in the reality also grows in the reality. Since the number of points falling on a cell and the possibility of at least one point falling into a cell, the number of cells have "no data" values in the maximum height map is reduced. The effects of the resolution on the number of empty cells in the raster image are shown in the indicator matrices, height maps and final DSMs produced with different resolution parameters in Figure 3.8. As it can easily be seen from the indicator matrices in Figure 3.8a and Figure 3.8b, when the resolution is increased, the number of empty cells is decreasing. When the DSM results produced by two different resolution parameters are compared in Figure 3.7b and Figure 3.7d which are the zoomed regions of the generated surface models with different resolution parameters, since when the value of the resolution is small, the number of empty cells in the height map increases, the amount of cells will be interpolated through the altitude values is also large and the possibility of errors in the produced DSM is high. As the resolution continues to increase, the probability of at least one laser point drops into each cell increases, and the interpolation throughout the elevation values effect decreases, in which case the produced surface model that reflects the real world more accurately. As mentioned earlier, each pixel in the height map is assigned height information with the highest value among the points falling on that cell, and when the grid size is increased, the number of points falling into each cell is increasing. While only points that have the highest values are stored from many points in the cell, the other height values are lost. In that case, the noise points in the data, the loss in height values and the interpolation among the remaining points cause the production of erroneous results again. For this reason, an ideal resolution value must be specified in order to create an accurate and complete surface model. An optimum grid size is estimated to be ensure such that 25% of all the pixels in the elevation map should be filled which means that the ratio of the number of full pixels to the number of all pixels is 25% according to the proposed cost function. In Figure 3.7e and Figure 3.7f, the DSM is extracted with optimum resolution value, $R = 0.425$ which is calculated iteratively with 0.025 steps in a range of defined minimum and maximum grid sizes, respectively 0.05 and 5.00.



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

Figure 3.8: The effects of resolution on the intermediate steps on the algorithm (a) Indicator matrix ($\mathbf{K}$), $R = 0.5$ (b) Indicator matrix ($\mathbf{K}$), $R = 1.0$ (c) Generated height map (M), $R = 0.25$ (d) Generated height map (M), $R = 0.25$ (e) 2D view of extracted DSM (H), $R = 0.25$ (f) 2D view of extracted DSM (H), $R = 1.0$

Another parameter affecting the result in the study is the level of smoothing used in the cost function 3.3. As can be seen in Figure 3.9, the amount of smoothing at the end of the algorithm is increasing in directly proportional to the value of the λ parameter. When the value is kept very small, it is observed that there are undesirable elevations called noise on the model even if the details are preserved on the surface. On the contrary, increasing the value too much, it is observed that while the small elevations on the model disappear, the features of objects such as the corners and edges of buildings are deformed. In Figure 3.9, the surface models are extracted with the estimated optimum resolution value and varied λ parameters. For this sample data, the smoothness parameter is considered as 0.01 to produce the DSM which is preserving the features of the objects while eliminating the noise. Note that, since *no data* cells must be interpolated, λ cannot be set as zero. Hence it must be at least given a small number such as 10^{-8} if minimal smoothing is desired.



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

Figure 3.9: The effects of smoothness on the 3D DSMs (a) DSM, $\lambda = 0.001$ (b) A region that is zoomed on DSM, $\lambda = 0.001$ (c) DSM, $\lambda = 0.1$ (d) A region that is zoomed on DSM, $\lambda = 0.1$ (e) DSM, $\lambda = 10$ (f) A region that is zoomed on DSM, $\lambda = 10$; R = 0.425

3.3 Photogrammetry based DSM

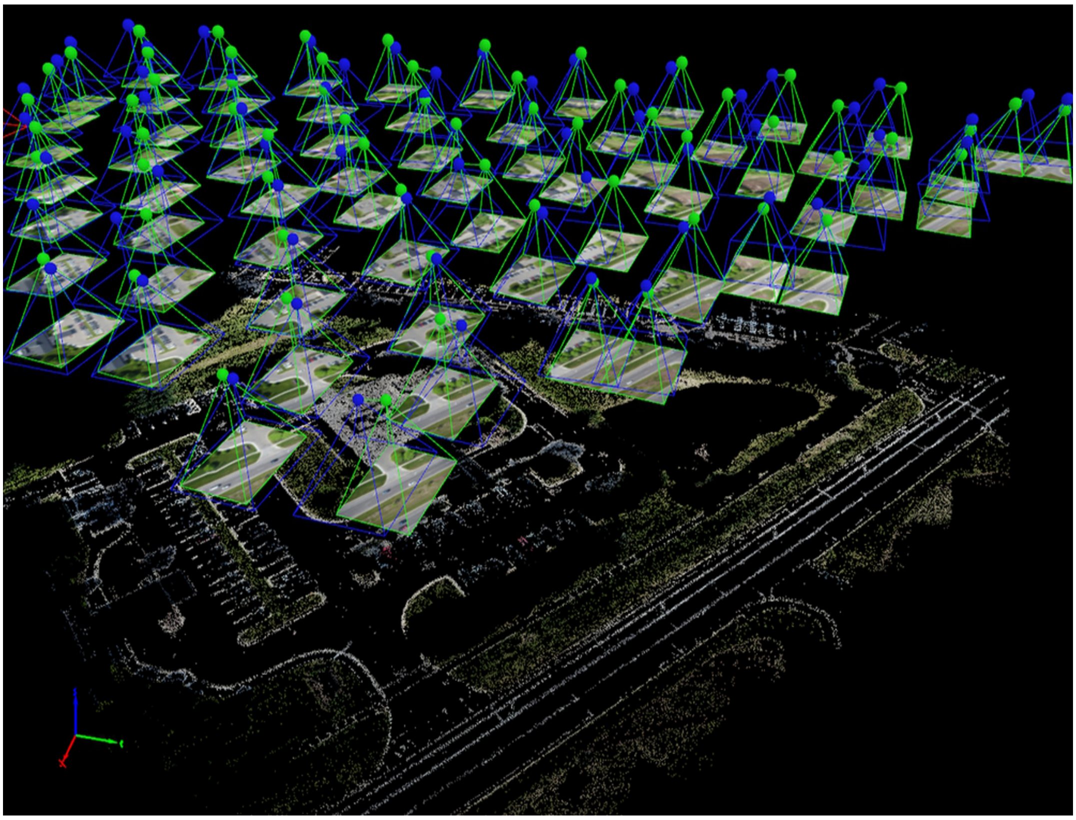


Figure 3.10: An example of photogrammetry based DSM generation

Photogrammetry can be defined as the science of generating reliable information from the multiple 2D photographs of a region that are captured from different positions and angles. Using any kind of digital cameras, at least two stereo image pair is taken in order to produce a map including 3D information about any real-world object and 3D model of the earth's surface

Photogrammetry is an older technology than laser scanning and it was introduced as a robust triangulation method by David Lowe from the University of British Columbia, in 1999 [32]. Whenever the locations of the cameras and the scales of the images are known, the coordinates of any point in 3D space and/or the exact distance between any two points in the scene can be determined. A great number of applications such as 3D elevation model generation like DEM such as DSM and DTM, constitution and visualization of different types of maps like topographic, vegetation and road maps, 2D and 3D reconstruction and classification of objects are highly integrated with the help of this technology [6, 1].

When photogrammetry is compared with LIDAR, both of them has their own advantages and disadvantages. The most crucial strength of photogrammetry is being more cost-effective since the equipments like digital cameras are significantly cheaper than LIDAR components while collecting the data. In addition, the preferred software and hardware used to process a set of 2D photographs are cheaper than those used to process high density 3D point cloud data. Another important advantage is that photogrammetry takes less time and less human effort to acquire the data. Furthermore, it is more portable, versatile, flexible and speedy. Finally, textures of the surfaces are better represented visually. On the other hand; although photogrammetry has high planimetric accuracy, it has a weakness in depth accuracy. However, since laser scanners collect directly 3D information about a surface of any object, it is more reliable in depth accuracy, especially over large areas. Since photographs are not captured automatically and depend on decision of an expert, it causes more human error. The last significant disadvantage of photogrammetry is that errors may occur while 3D modeling of reflective and transparent surfaces [32, 1, 2].

3.4 SAR Interferometry based DSM

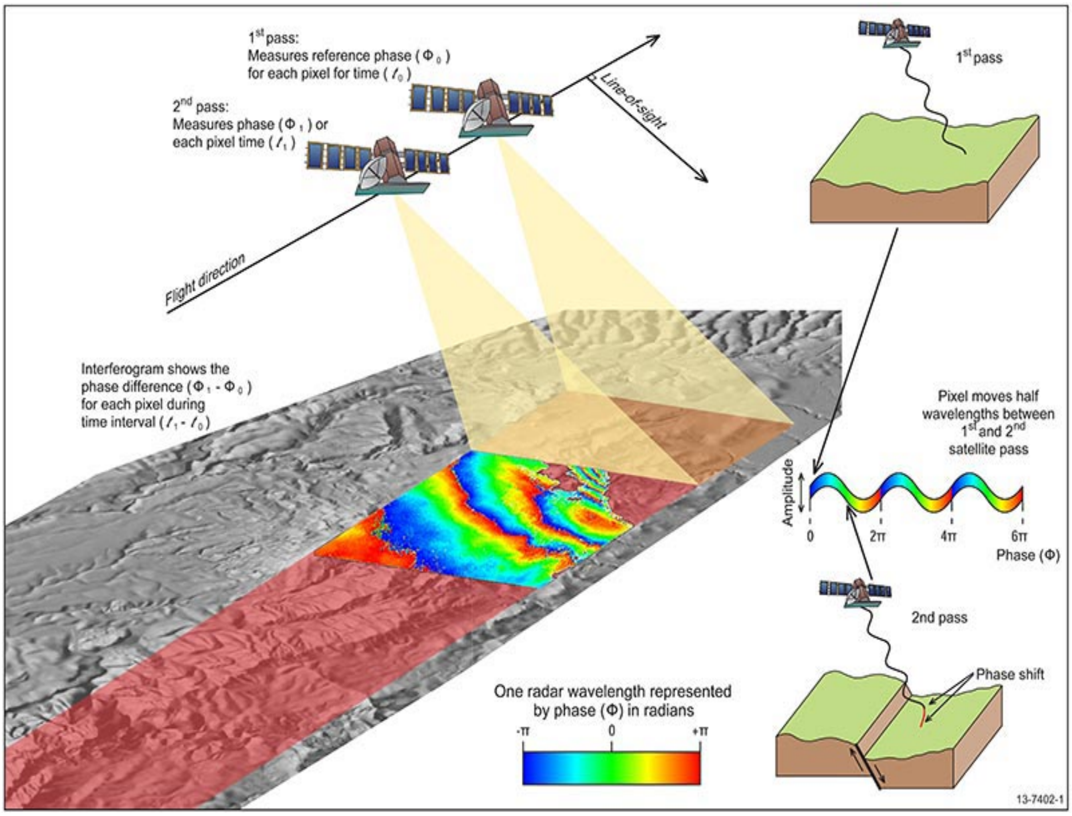


Figure 3.11: An example of photogrammetry based DSM generation

Synthetic aperture radar (SAR) is a type of active sensing and it generates SAR images that are widely used for modeling and mapping the earth's surface because of their being of high-resolution, day and night and weather-independent. These images are generated from the backscattered radiation signals that are spread from an antenna towards the surface of earth. SAR systems are commonly used for monitoring tasks and provide information about the properties of the earth's surface such as topography, morphology, roughness and the characteristics of the reflective layer [7].

Interferometric SAR (InSAR) images are also used for modeling and mapping surface of earth. In order to generate InSAR images, at least two complex SAR images that are captured from the same area but from slightly different line of sight are required. These different SAR images can be acquired in two different ways: The first method is mounting two radars to the distinct platforms. The second one is passing with the same radar at two different times [18]. InSAR images are generated using the phase differences of these multiple complex SAR images and used in a variety of applications such as DEM generation, detection and identification of natural disasters like earthquake, volcanoes, glacier flows, and landslides, analysis of vegetation properties, monitoring of subsidence and structural stability [7].

Generated DEM quality depends on various factors according to preferred InSAR image generation method. To illustrate, the character of terrain like roughness, reflectivity and slope, or the type of vegetation on the observed ground surface may affect the quality of the DEMs. In addition, time interval between two passes has an effect such that increasing this interval decreases the resolution of generated DEM. Finally, the locations, point of views and orientations of the mounted radars have an impact on DEM quality [9].

When compared with photogrammetry based and LIDAR based DEM, SAR interferometry based DEM has advantage in terms of coverage area since wider terrains can be monitored by radar satellites rather than the ones obtained by drones or aircrafts. Another great strength of radar based imaging is the fact it is not affected from neither day and night nor weather conditions such as clouds and fog. However, the weakness of this approach is generating less detailed and less accurate DEM [64, 9].

CHAPTER 4

A NEW CORRELATION BASED VARIATIONAL CHANGE DETECTION METHOD

In this chapter, a new correlation-based variational change detection method is proposed for multi-temporal elevation maps. The first section focuses on the proposed cost function and its data fidelity and regularization term explanations. In the second section, the minimization of the proposed cost function is discussed.

4.1 Correlation-Based Variational Change Detection (CVCD)

In this section, a change detection method is proposed which determines the change map (C) between two elevation maps by minimization of a novel correlation based variational cost function. Cost function $J(C)$ is minimized with respect to C in order to determine the change map as follows:

$$C^* = \underset{C}{\operatorname{argmin}}\, J(C) \; . \tag{4.1}$$

The proposed cost function to be minimized is defined as in the following Equation 4.2:

$$\begin{aligned} J(C) = -\rho(H_1 + C, H_2 - C) + \lambda_z \|C\|_1^1 \\ + \lambda_s \|\nabla C\|_1^1 \; , \end{aligned} \tag{4.2}$$

where, H_1 and H_2 are the registered multi-temporal elevation maps, C is the computed change map as the output of the algorithm, λ_z and λ_s are positive constants, and ∇ is the gradient operator. Here, H_1 and H_2 are constructed as digital surface models in Section 3.2.5 where they are acquired from the same geographical area at time t_1 and t_2. Note that, C is numerically half of the final change map since it is symmetrically influenced by both elevation maps. In this cost function, ρ indicates the cross correlation function that is described in Equation 4.3 for any images, A and B;

$$\rho(A, B) = \sum_{i=1}^{N} (A_i - \boldsymbol{\mu}_A)(B_i - \boldsymbol{\mu}_B) \ , \tag{4.3}$$

where, N is the number of pixels, i is the pixel index, $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ denote the average pixel values for the images A and B.

In Equation 4.2, the first term and second term are together defined as a data fidelity term. ρ measures the degree of similarity between the first and second height map. However, cross correlation metric does not consider shift on elevation in height maps since $\rho(A, B)$ and $\rho(A, B + \Delta z)$ lead to the same correlation value for an arbitrary constant Δz. In the cross correlation function (Equation 4.3), for the shifted B the second multiplier can be described as $(B_i + \Delta z - \boldsymbol{\mu}_{B+\Delta z})$. Adding Δ_z to each pixel value results in increasing average value by Δ_z which is described as $(B_i + \Delta z - (\boldsymbol{\mu}_B + \Delta z))$. This description equals to $(B_i - \boldsymbol{\mu}_B)$. Therefore, it can be deduced that any shift in one of the elevation map leads to the same cost function value; thus, cross correlation based term itself is not a sufficient data fidelity term. Thus, the ℓ_1-norm based term $\|C\|_1^1$ with a small positive weight of λ_z is added into the data fidelity term in order to impose sparsity on change map C, namely cross correlation based regularization. Here, sparse change map means that there are no changes for at least half of the pixels. Therefore, median value of the change map should be close to 0 which is enforced by $\|C\|_1^1$. The proof is as follows:

Let's define S as the sum of desired change map C and undesired shift Δz, such that $S = C + \Delta z$. Desired change map C can be found using $C = S - \Delta z$, where Δz is not known. In Equation 4.2, cross correlation based term and TV regularization term are independent from Δz. Therefore, only $\|C\|_1^1$ in Equation 4.2 needs to be minimized with respect to Δz as follows:

$$C^* = S - \Delta z^* \ , \tag{4.4}$$

where,

$$\Delta z^* = \underset{\Delta z}{\operatorname{argmin}} \|S - \Delta z\|_1^1 \ . \tag{4.5}$$

This equation can be written in summation form as below:

$$\Delta z^* = \underset{\Delta z}{\operatorname{argmin}} \sum_{i=1}^{N} |S_i - \Delta z| \ . \tag{4.6}$$

One should notice that $\frac{\partial |\Delta z|}{\partial \Delta z} = sgn(\Delta z)$, which is the sub-gradient of the non-smooth ℓ_1-norm where $sgn(.)$ is the sign function. Hence, deriving the sum above yields $\sum_{i=1}^{N} sgn(S_i - \Delta z)$. This equals to zero only when the number of positive items equals the number of negative items which happens when Δz is the median of the D. Thus, Δz should be equal (or approximately equal) to $\mathcal{M}(S)$ where $\mathcal{M}(.)$ is the median function. If H_1 and H_2 are noise free, perfectly registered, and there is no shift ($\Delta z = 0$) in pixel values between H_1 and H_2 then change map C is simply equals to $H_2 - H_1$. We assume that C is sparse and at least half of the pixels contain no change so that median of C is approximately zero. Less pixels will be around zero if there is an undesired shift ($\Delta z \neq 0$) between H_1 and H_2. So, as shown in proof, the regularization term will shift the pixel elevation values in C to increase the sparsity such that median of the C will be close to zero (see Figure 4.1).

In Equation 4.2, the third term is called as ℓ_1-norm TV regularization term, which implies a penalty on the changes in image gradients TV regularization term imposes smoothness on the change map by filtering the subtle differences such as noise where λ_s controls the smoothness level. In this term, ∇ is the gradient operator of 2D change map which is expanded as partial derivatives in two directions, ∂_x and ∂_y. In this study, ℓ_1-norm TV regularization term is preferred due its success in preserving details such as the edges of the objects, corners of the objects, and small sized (i.e. point-sized) objects for various image processing.

First, cross correlation based term is expanded as described in Equation 4.3. Second, the the first term is rewritten as the sum of the absolute values according using the ℓ_1-norm definition. Lastly, the gradient operator is defined as the sum of partial derivatives with respect to x and y and rewritten as the sum of the absolute values of these partial derivatives. Expanded cost function is given in Equation 4.7;

$$
\begin{aligned}
J(C) = &- \sum_{i=1}^{N} (H_{1i} + C_i - \boldsymbol{\mu}_1)(H_{2i} - C_i - \boldsymbol{\mu}_2) \\
&+ \lambda_z \sum_{i=1}^{N} |C_i| + \lambda_s \sum_{i=1}^{N} \big(|(\partial_x C)_i| + |(\partial_y C)_i|\big) \ ,
\end{aligned} \tag{4.7}
$$

where, N is the number of pixels in change map and i is the pixel index. Note that, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the mean of $H_1 + C$ and $H_2 - C$, respectively. Using the substitutions $\hat{H}_1 = (H_1 - \boldsymbol{\mu}_1)$ and $\hat{H}_2 = -(H_2 - \boldsymbol{\mu}_2)$ and applying algebraic simplifications Equation 4.7 is transformed into Equation 4.8 as below:

$$
\begin{aligned}
J(C) = &\sum_{i=1}^{N} (C_i + \hat{H}_{1i})(C_i + \hat{H}_{2i}) \\
&+ \lambda_z |C_i| + \lambda_s (|(\partial_x C)_i| + |(\partial_y C)_i|) \ .
\end{aligned} \tag{4.8}
$$

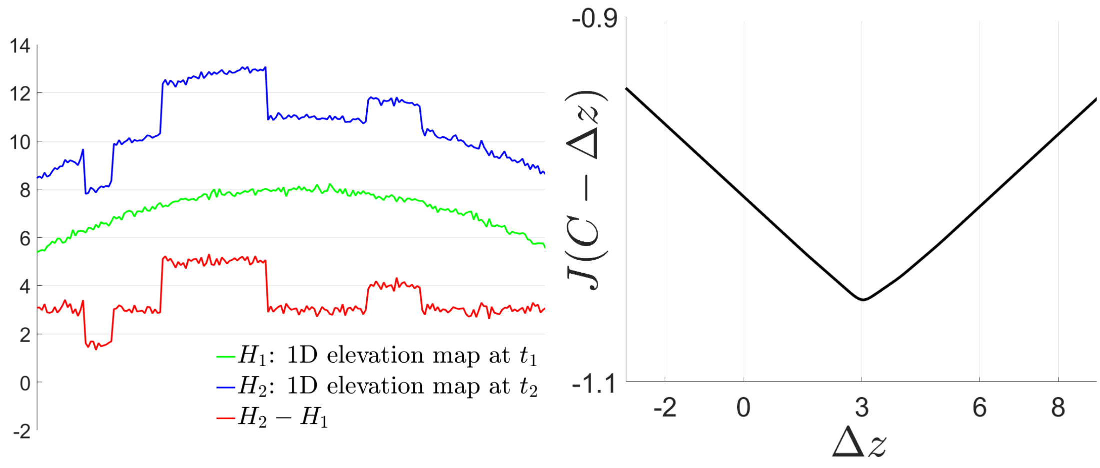
Instead of using constant weight λ_z for each pixel, weights are calculated adaptively for each pixel that is defined in Equation 4.9. Using λ_{z_i} helps $\|C\|_1^1$ term to decreases dependency to the magnitudes of the changes in each pixel within the cost function. Here, λ_{z_i} gets larger as C_i gets closer to median of C and λ_{z_i} gets closer to 0 as C_i deviates from median of C.

$$
\lambda_{z_i} = \lambda_z \big((1 + \varepsilon_z) - \frac{|\tilde{C}_i| + \varepsilon_z}{\|\tilde{C}\|_\infty + \varepsilon_z} \big) \ , \tag{4.9}
$$

where, $\tilde{C} = C - \mathcal{M}(C)$ is median centered form of C, $\|\tilde{C}\|_\infty = \max_j |\tilde{C}_j|$ is maximum of $\tilde{C}$, and ε_z is a small positive constant whose value is assigned as 10^{-5} to guarantee that $\lambda_{z_i} > 0$. Finally, Equation 4.10 is obtained after replacing the fixed λ_z in the Equation 4.8 with the adaptive λ_{z_i} as below:

$$J(C) = \sum_{i=1}^{N} (C_i + \hat{H}_{1i})(C_i + \hat{H}_{2i}) \\ + \lambda_{z_i}|C_i| + \lambda_s(|(\partial_x C)_i| + |(\partial_y C)_i|) \; . \tag{4.10}$$

In Figure 4.1, contribution of the term with adaptive λ_{z_i} is shown for the Equation 4.10 using a $1D$ synthetic example. In this example, $C = H_2 - H_1$, as an estimated change map, $\lambda_z = 0.1$, and $\lambda_s = 0$ to ignore the TV regularization term. Note that, there is a 3 meter difference between H_1 and H_2 so that in this example $\Delta z = 3$. Since cross correlation based term and TV regularization term are independent from the shift in C, optimizer can freely shift C by an arbitrary constant such that regularization term becomes minimum. As seen in Figure 4.1, the regularization term becomes minimum when $\Delta z = 3$.



(a) (b)

Figure 4.1: Synthetically generated 1D signal to prove that necessity of λ_z term: a) Data, b) Cost function w.r.t. Δz

4.2 Minimization of the Cost Function

Change map can be estimated by minimizing the proposed cost function given in Equation 4.10. However, the cost function contains absolute terms that are not differentiable, due to the discontinuity at zero. As explained at section 2.2.2.2 in detail, these absolute terms are approximated using quadratic approximation [41] as given in the Equation 4.11 as follows;

$$|z| \approx \mathcal{Q}(\hat{z})z^2, \quad \mathcal{Q}(\hat{z}) = (|\hat{z}| + \varepsilon)^{-1} , \tag{4.11}$$

where, $\hat{z}$ is a constant proxy for z, such that $\hat{z} \leftarrow z$, $\mathcal{Q}(\hat{z})$ is the coefficient of the quadratic approximation of $|z|$, and ε is a small positive constant to avoid division by zero in $\mathcal{Q}(\hat{z})$. Note that, approximations are only accurate around $\hat{z}$ and accuracy of the approximation increases as ε gets smaller. If ε is less than 0.1, the accuracy of the approximation increases. Nevertheless, when the ε is too small (less than 10^{-5}, errors in numerical operations originating from double precision start to increase. Hence, ε to be used between 0.1 and 10^{-5} provides a sufficient tradeoff. When $\varepsilon = 0.1$, the cost function is solved fast but ℓ_1-norm TV regularization quality drops. If ε is less than 10^{-5}, the system produces results slowly but ℓ_1-norm TV regularization quality is higher.

Absolute terms in cross correlation based regularization term and TV regularization term in Equation 4.10 are approximated using the approximations given as below:

$$\begin{aligned} |C_i| &\approx (W_c)_i C_i^2, & (W_c)_i &= \mathcal{Q}(\hat{C}_i) , \\ |(\partial_x C)_i| &\approx (W_x)_i (\partial_x C)_i^2, & (W_x)_i &= \mathcal{Q}((\partial_x \hat{C})_i) , \\ |(\partial_y C)_i| &\approx (W_y)_i (\partial_y C)_i^2, & (W_y)_i &= \mathcal{Q}((\partial_y \hat{C})_i) , \end{aligned} \tag{4.12}$$

where, $(W_c)_i$, $(W_x)_i$, and $(W_y)_i$ are the coefficients for the quadratic approximations of the absolute terms which are evaluated at $\hat{C}$.

Substituting the approximations in Equation 4.12 into Equation 4.10 and expanding the terms lead to the below differentiable cost function. Moreover, a new regularization term is added to force the solution to be close to $\hat{C}$, considering the fact that employed approximations are only accurate around this value [40, 41]. Finally, in order to minimize the cost function, convex optimization methods are preferred. Due to the employed approximations, the cost function will be solved iteratively, so that the iteration index is added into the cost function. After all placeholders have been placed in the relevant places, the cost function is described as Equation 4.13;

$$
\begin{aligned}
J^{(n)}(C) = \sum_{i=1}^{N} & C_i^2 + (\hat{H}_{1i} + \hat{H}_{2i})C_i + \hat{H}_{1i}\hat{H}_{2i} \\
& + \lambda_{z_i}(W_c)_i C_i^2 \\
& + \lambda_s\big((W_x)_i(\partial_x C)_i^2 + (W_y)_i(\partial_y C)_i^2 \\
& + \quad (C_i - \hat{C}_i)^2\big) \ ,
\end{aligned} \tag{4.13}
$$

where, n indicates the iteration number and for the sake of simplicity n is not added into each term in the right side of the equality. Unless otherwise specified, each term belongs to n^{th} iteration. Note that, $\hat{H}_1$ and $\hat{H}_2$ should be re-evaluated in each iteration.

The reorganized quadratic approximated cost function can be cast into matrix-vector form as below (Equation 4.14);

$$
\begin{aligned}
J^{(n)}(v_c) = & \, {v_c}^\top v_c + (v_{\hat{h}_1} + v_{\hat{h}_2})^\top v_c + v_{\hat{h}_1}^\top v_{\hat{h}_2} \\
& + ({v_c}^\top \mathbf{Z} v_c) \\
& + \lambda_s\big({v_c}^\top \mathbf{L} v_c) + (v_c - v_{\hat{c}})^\top (v_c - v_{\hat{c}})\big) \ ,
\end{aligned} \tag{4.14}
$$

where, $v_{\hat{h}_1}$, $v_{\hat{h}_2}$, and v_c are the vector forms of $\hat{H}_1$, $\hat{H}_2$, and C, respectively. $\mathbf{Z}$ and $\mathbf{L}$ are defined as $\mathbf{Z} = \mathbf{W_z W_c}$ and $\mathbf{L} = \mathbf{D_x}^\top \mathbf{W_x D_x} + \mathbf{D_y}^\top \mathbf{W_y D_y}$ for having a cost function in a compact form. Here, $\mathbf{W_z}$, $\mathbf{W_c}$, $\mathbf{W_x}$, and $\mathbf{W_y}$ denote the diagonal matrices that hold weights $\hat{\lambda}_{z_i}$, $(W_c)_i$, $(W_x)_i$, and $(W_y)_i$ on the diagonals, respectively. Also, $\mathbf{D_x}$ and $\mathbf{D_y}$ represent the Toeplitz matrices that are designed for taking partial derivatives of the 2D change map with respect to x and y as explained in 2.2.2.4. Despite it is not stated for the sake of simplicity of the equation's appearance, change map still belongs to n^{th} iteration in the matrix-vector representation.

Since this matrix-vector form cost function, $J^{(n)}(v_c)$, is differentiable, it enables an optimization procedure and its minimization can be solved by taking its derivative with respect to v_c and equalizing it to zero as below.

$$
\frac{\partial J^{(n)}(v_c)}{\partial v_c} = 0 \ . \tag{4.15}
$$

After the mathematical derivations are processed and aligned, the decomposition of the obtained linear system is given in the following Equation 4.16;

$$
\begin{aligned}
\mathbf{A}v_c^{(n+1)} &= b \ , \\
\mathbf{A} &= (1+\lambda_s)\mathbf{I} + (\mathbf{Z} + \lambda_s \mathbf{L}) \ , \\
b &= \lambda_s v_{\hat{c}} - \frac{1}{2}(v_{\hat{h}_1} + v_{\hat{h}_2}) \ ,
\end{aligned}
\tag{4.16}
$$

where, $\mathbf{I}$ is the identity matrix. Note that, $\mathbf{A}$ and b belongs to n^{th} iteration. In Equation 4.16, $\mathbf{A}$ is sparse, positive definite, 5-point Laplacian matrix. Since $\mathbf{A}$ is positive definite cost function in Equation 4.14 is strictly convex, minimization leads to the global minimum. The proof is as follows:

Let $\mathbf{A}$ in the Equation 4.16 be the second derivative (i.e. Hessian) of the cost function $J^{(n)}(v_c)$ given in the equation 4.14. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{NxN}$ is called positive definite (thus $J^{(n)}(v_c)$ is strictly convex), denoted by $\mathbf{A} \succ 0$, if $x^\intercal \mathbf{A} x > 0$, for every $x \in \mathbb{R}^N$ with $x \neq 0$.

$$
\frac{\partial^2 J^{(n)}(v_c)}{\partial^2 v_c} = (1+\lambda_s)\mathbf{I} + (\mathbf{Z} + \lambda_s \mathbf{L}) = \mathbf{A} \ . \tag{4.17}
$$

Identity matrix $\mathbf{I}$ is positive definite and λ_s is positive. Therefore, we should show that $x^\intercal(\mathbf{Z}+\lambda_s\mathbf{L})x > 0$ to guarantee that $x^\intercal \mathbf{A} x > 0$ for all non-zero x. If we distribute $x^\intercal$ and x from left and right onto $(\mathbf{Z} + \lambda_s \mathbf{L})$:

$$
x^\intercal \mathbf{Z} x + \lambda_s x^\intercal \mathbf{L} x > 0 \ . \tag{4.18}
$$

Which is satisfied when below inequalities are satisfied:

$$
x^\intercal \mathbf{Z} x > 0, \quad x^\intercal \mathbf{L} x > 0 \ , \tag{4.19}
$$

where $\mathbf{Z}$ equals to $\mathbf{W_z W_c}$; therefore, it is a diagonal matrix with positive entries since $\mathbf{W_z}$ and $\mathbf{W_c}$ are also diagonal matrix with positive entries (see Equation 4.9 and Equation 4.12). Thus, $x^\intercal \mathbf{Z} x > 0$ is satisfied.

$x^\intercal \mathbf{L} x > 0$ can be expanded as below two inequalities:

$$
\begin{aligned}
x^{\intercal}(\mathbf{D_x}^{\top}\mathbf{W_x}\mathbf{D_x})x > 0 \ , \\
x^{\intercal}(\mathbf{D_y}^{\top}\mathbf{W_y}\mathbf{D_y})x > 0 \ ,
\end{aligned}
\tag{4.20}
$$

which can be also expressed as:

$$
v_x^{\top}\mathbf{W_x}v_x > 0, \quad v_y^{\top}\mathbf{W_y}v_y > 0 \ , \tag{4.21}
$$

where, $v_x = \mathbf{D_x}x$ and $v_y = \mathbf{D_y}x$. Here, v_x and v_y are non-zero vectors since x is a non-zero vector and $\mathbf{D_x}$ and $\mathbf{D_y}$ are Toeplitz matrices with non-zero diagonal elements. Both inequalities are satisfied, since $\mathbf{W_x}$ and $\mathbf{W_y}$ are all diagonal matrices with positive entries (see and Equation 4.12). Thus, both first order optimality conditions and second order optimality conditions are satisfied which shows that cost function $J^{(n)}(v_c)$ given in equation 4.14 is strictly convex.

The proposed cost function is minimized in Algorithm 6 in an iterative manner due to the employed approximations. In each iteration, a sparse linear system is solved by using the preconditioned conjugate gradient (PCG) with the incomplete Cholesky (IC) preconditioner to evaluate the $v_c^{(n+1)}$. In Algorithm 6, $\varepsilon = 10^{-5}$, maximum number of PCG iterations is set to 10^2, PCG convergence tolerance is set to 10^{-3}, $n_{max} = 10^2$, and $C_{tolerance} = 10^{-3}$ as defaults. At the beginning of the algorithm, the pixel values of the change map v_c^1 is initialized and converted to vector form according to Algorithm 7.

Algorithm 6 Correlation Based Variational Change Detection

Input: H_1, H_2, λ_s, ε, n_{max}, $C_{tolerance}$

$v_c^{(1)} \leftarrow InitializeChangeVector(H_1, H_2)$

$\lambda_z = 0.1$

for $n = 1 : n_{max}$ **do**

  $v_{\hat{c}} \leftarrow v_c^{(n)}$

  $(W_c)_i = \mathcal{Q}(\hat{C}_i)$

  $(W_x)_i = \mathcal{Q}((\partial_x \hat{C})_i)$

  $(W_y)_i = \mathcal{Q}((\partial_y \hat{C})_i)$

  $\mathbf{A}^{(n)} = (1 + \lambda_s)\mathbf{I} + (\mathbf{Z} + \lambda_s \mathbf{L})$

  $v_b^{(n)} = \lambda_s v_{\hat{c}} - \frac{1}{2}(v_{\hat{h}_1} + v_{\hat{h}_2})$

  `Solve` $\mathbf{A}^{(n)} v_c^{(n+1)} = v_b^{(n)}$ `to find` $v_c^{(n+1)}$

  if $\|v_c^{(n+1)} - v_{\hat{c}}\|_\infty < C_{tolerance}$ then break the loop

end for

`where` $C \leftarrow vectorToImage(v_c^{(n+1)})$

Output: $2C$

Algorithm 7 Initialize Change Vector

Input: H_1, H_2

$C_{initial} = (H_2 - H_1)$

$C_{centered} = C_{initial} - \mathcal{M}(C_{initial})$

$v_c \leftarrow imageToVector(\frac{1}{2} C_{centered})$

Output: v_c

CHAPTER 5

EXPERIMENTAL ANALYSIS OF THE CORRELATION-BASED VARIATIONAL CHANGE DETECTION METHOD

In this chapter, results of the proposed change detection method are examined under a collection of datasets with different parameters and/or conditions. Four different sets of experiments are carried out in this chapter where each one constitutes a subsection respectively. In the first set of experiments the convexity of the proposed method is justified by giving different initial change maps to the cost function of the method. In the second set of experiments the effects of the smoothness factor of the proposed method are visualized. For this goal first the effect of the additive noise of the input height maps is tested, then after finding out the optimal noise parameter, the initialization of the cost function is also tested with this noise parameter. In the third set of experiments the effects of the noise level on the data for the proposed methods are demonstrated by changing the smoothness parameter. In the fourth set of tests show that the effects of the smoothness level of the proposed method on the real world datasets. The chapter is organized in five subsections. Addition to the before mentioned four different set of experiments, in the last subsection testing environment is reviewed. Experiments are done with four different datasets each one for the respective subsection. Synthetically generated 1D signal is used for the first set of experiments, meanwhile for the second set of experiments synthetically generated 2D signal is used. For the third set of experiments, synthetically generated real world data is made use of. The real world 3D point cloud data acquired by LIDAR technology is used for the last set of experiments.

In this subsection, proposed algorithm is tested with different parameters and affecting factors using a set of dataset. Since some of datasets include synthetically created data and some of them are real-world digital elevation data, the following sections are divided by the type of dataset used. For each used dataset section, a factor influencing the algorithm result was tested and performance results were pointed out.

5.1 Experiments on Simulated 1D Signal

This set of experiments arranged to prove the convexity of the proposed cost function which is already proved theoretically in Section 4.2. 1D signals are synthetically created to demonstrate the effects of the change results as visually convincing. The following subsections explain how the synthetic data is generated and how the initialization of the change map affects the result of the proposed method.

5.1.0.1 1D Synthetic Signal Generation

To start with, for the first input of the change detection method, an initial signal is generated between a range $[-5, 5]$ as a polynomial form, $p(x) = ax^2 + bx + c$, where $a = -0.1$, $b = 0.01$, and $c = 8.0$. The second signal is adapted from the first signal by adding a constant value, $\Delta_h = 3.0$ to whole signal on y-axes. In order to compose changes between these two signals, different amounts are added or subtracted from the different coordinates of second signal.

Finally, the Gaussian noise with constant noise amplitude, 0.1, and variance, 1.0, are added to these signals. These two generated noisy signals have SNR values as 37.37 db and 40.40 db, respectively. Note that, these noise levels are identical for each run of test meanwhile different for each signal. In this regard, it can be ensured that same test case is performed on each run.

In order to measure the performance of the proposed method and show how close the obtained result to the synthetically added changes, the ground truth (GT) is designated before adding the noise into images. Synthetically generated two signals are shown with blue and red colors and GT is shown with the green color in Figure 5.1;

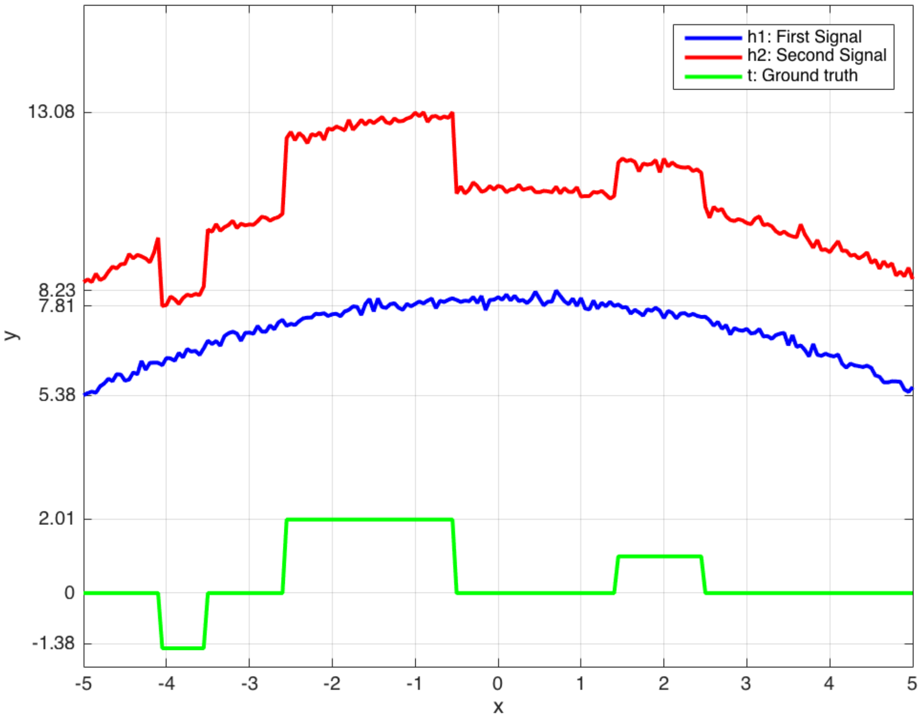
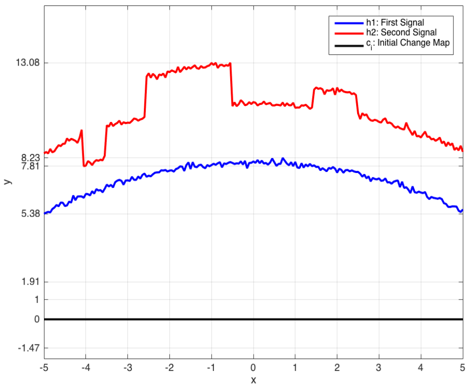


Figure 5.1: Synthetically Generated 1D Signals and Ground Truth

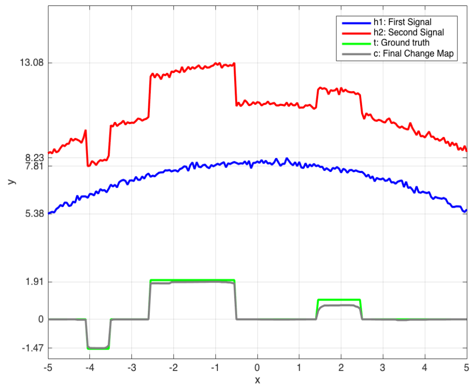
5.1.0.2 Experimental Results on 1D Synthetic Signals

In this section, the set of test cases is just drawn up to experience the effect of the different initial change map given to the cost function to be minimized on the result.

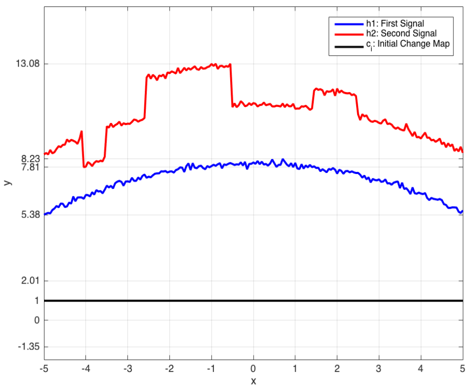
The experiments that indicate the effects of different initializations for the minimization problem are carried out. Because of this reason, the other parameters used in the proposed method are kept same for each test case. These parameters are set as $\lambda_s = 0.15$, $\varepsilon = 10^{-5}$, $C_{tolerance} = 10^{-3}$ which is called as convergence tolerance, and $n_{max} = 10^2$ which is maximum iteration. Firstly the closeness of the obtained result with the ground truth is presented in Figure 5.2 for each initial change map.


h1: First Signal
h2: Second Signal
c_i: Initial Change Map
y
x

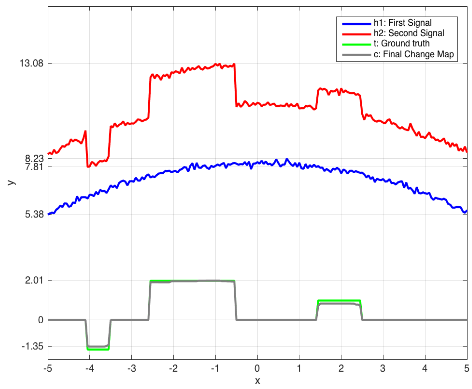
(a)


h1: First Signal
h2: Second Signal
t: Ground truth
c: Final Change Map
y
x

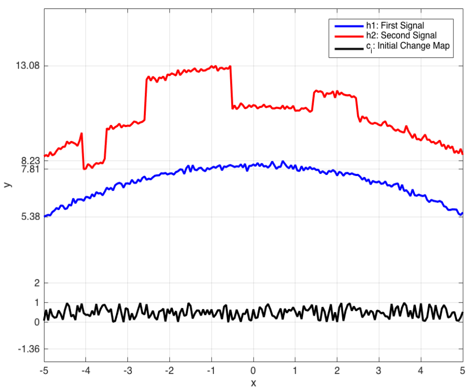
(b)


h1: First Signal
h2: Second Signal
c_i: Initial Change Map
y
x

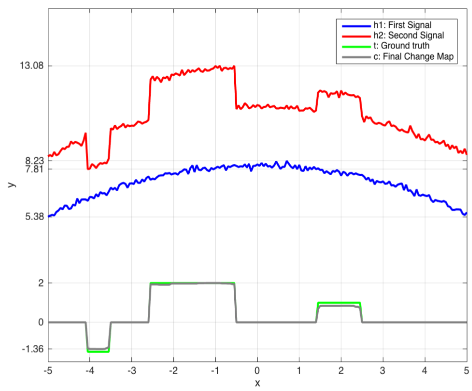
(c)


h1: First Signal
h2: Second Signal
t: Ground truth
c: Final Change Map
y
x

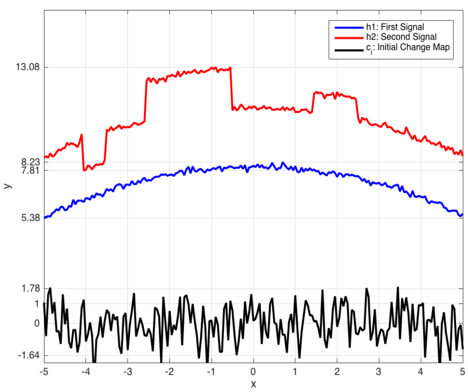
(d)


h1: First Signal
h2: Second Signal
c_i: Initial Change Map
y
x

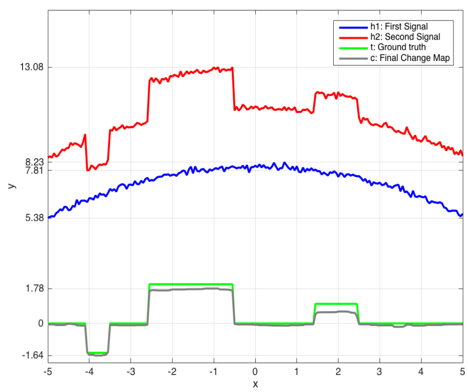
(e)


h1: First Signal
h2: Second Signal
t: Ground truth
c: Final Change Map
y
x

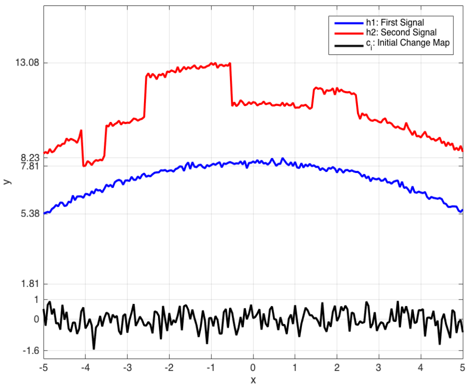
(f)


h1: First Signal
h2: Second Signal
c_i: Initial Change Map
y
x

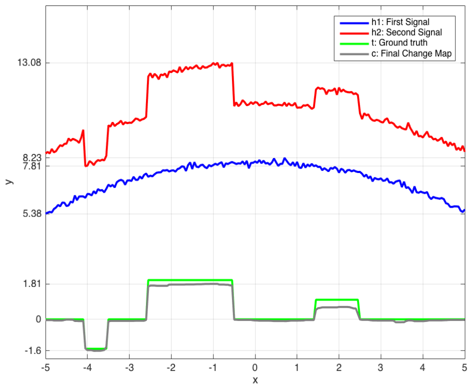
(g)


h1: First Signal
h2: Second Signal
t: Ground truth
c: Final Change Map
y
x

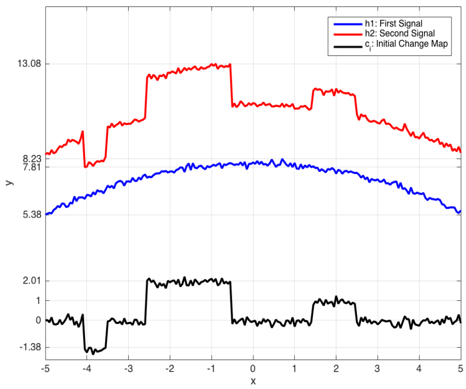
(h)



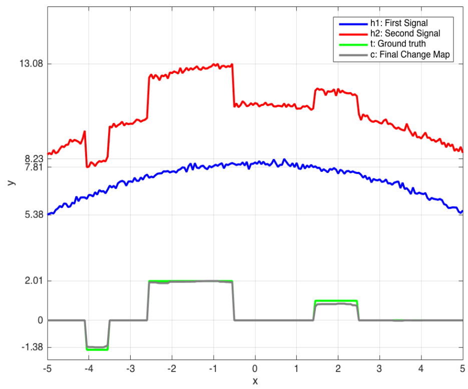
(i)



(j)



(k)



(l)

Figure 5.2: The effects of the initialization of the change map for the proposed cost function minimization (a-b) Change map is initialized with zeros and result change map of that case (c-d) Change map is initialized with ones and result change map of that case (e-f) Change map is initialized with uniformly distributed random numbers and result change map of that case (g-h) Change map is initialized with normalized random noise and result change map of that case (i-j) Change map is initialized with white gaussian noise having and result change map of that case (k-l) Change map is initialized as described in Algorithm 7 and result change map of that case

In Figure 5.2; the first columns demonstrates the synthetically generated 1D input signals and initial change map to be given to proposed cost function. The second columns show that the ground truth and result change map at the same plot. It is clearly seen that the change result map for each case is identical as the GT. In the first test, change map is initialized with zeros which is shown in Figure 5.2a and the result change map is shown in Figure 5.2b. The evaluated root mean square error (RMSE) between the GT and the result is 0.009. For the second test case, change map is initialized with 1 values which is shown in Figure 5.2c and the result change map is shown in Figure 5.2d. RMSE for the ones change initialization is 0.005. Then, change map is initialized with uniformly distributed random numbers in Figure 5.2e and normally distributed random numbers in Figure 5.2g. The result change maps are shown in Figure 5.2f and Figure 5.2h and RMSEs are 0.005 and 0.014, respectively. In the other test case, initial change map is generated with white Gaussian noise having zero mean and power -6 dBW in Figure 5.2i. The result of the proposed approach is shown in Figure 5.2j and RMSE is evaluated as 0.013. Finally, the change map has been initiated as explained in the Section 4.2, in detail and described as pseudo-code in Algorithm 7. The initial and final change maps are shown in Figure 5.2k and in Figure 5.2l, respectively. The RMSE of this test case is evaluated as 0.005. When the calculated RMSE results for each test case are analyzed, it is clearly seen that the values are so close to 0, which means that, the final change map is so close to the GT data.

Secondly; if the cost function is convex, the cost function converges to the same point regardless of the starting point of the minimization method. With these different initiates, the cost function converges to the same point which is shown in the following Figure 5.3.

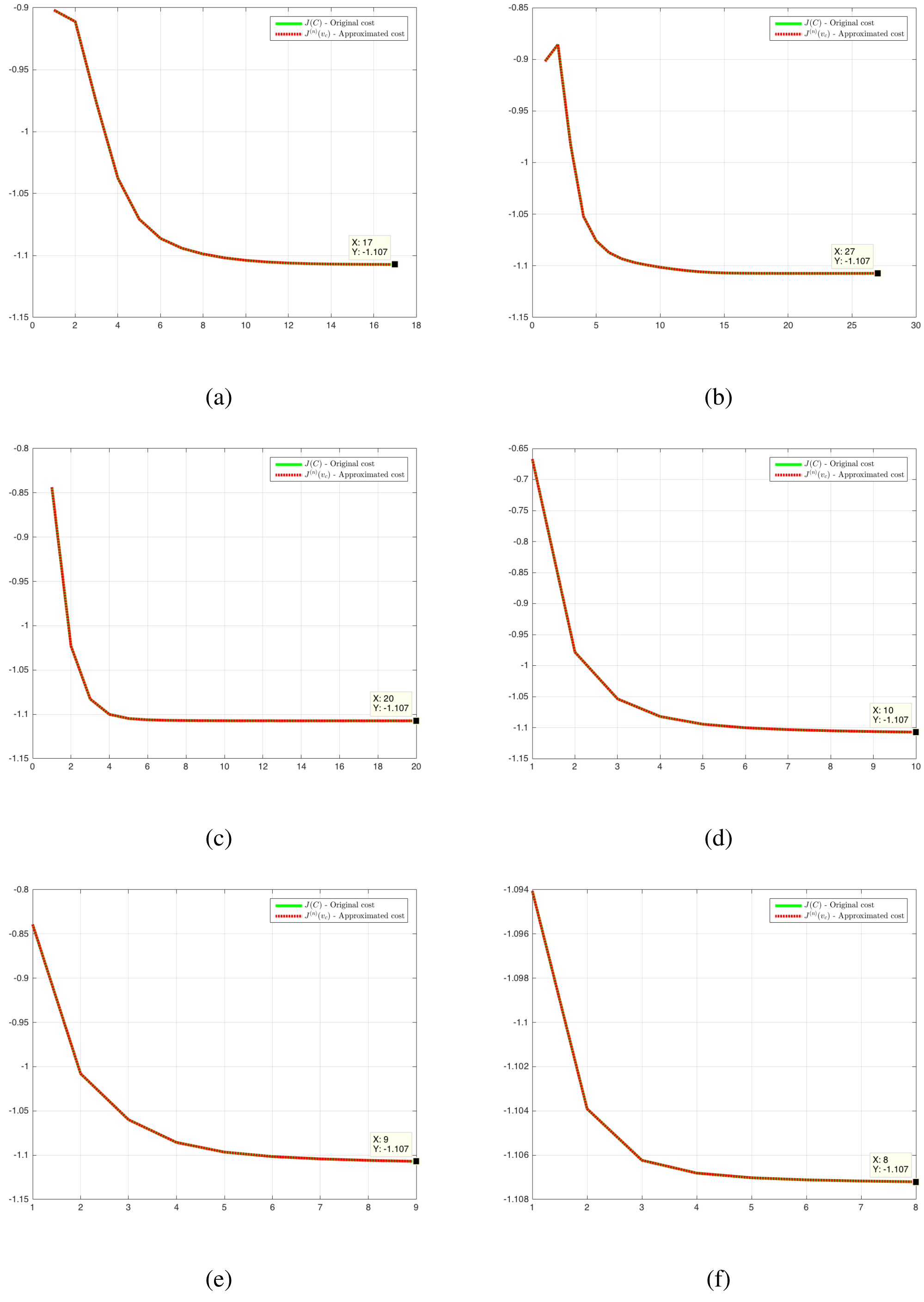


Figure 5.3: The indication that the cost function converges to same value for each initialization (a) Change map is initialized with zeros and result change map of that case (b) Change map is initialized with ones and result change map of that case (c) Change map is initialized with random noise and result change map of that case (d) Change map is initialized with normalized random noise and result change map of that case (e) Change map is initialized with white gaussian noise and result change map of that case (f) Change map is initialized as described in Algorithm 7

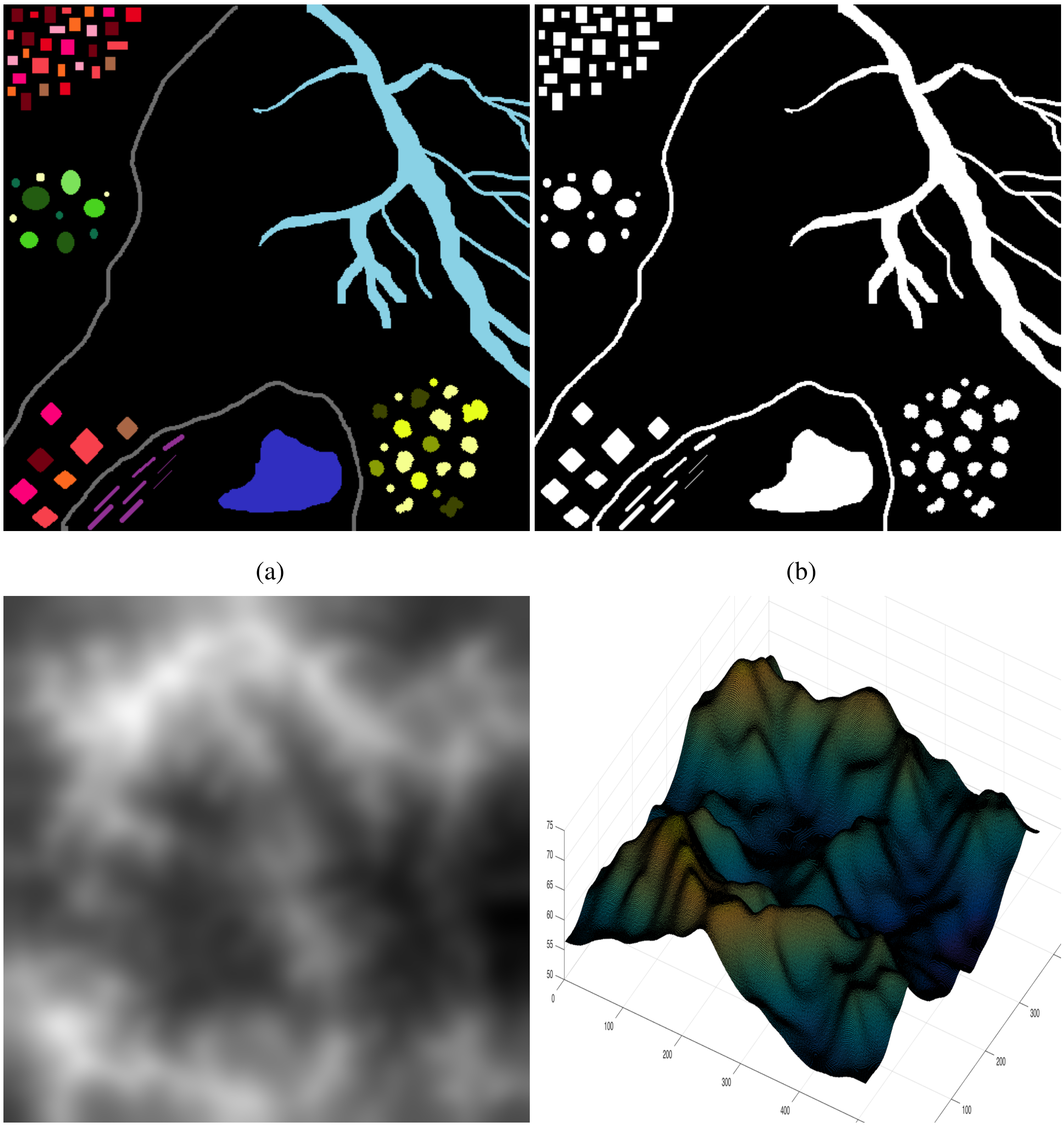
As it can be seen in Figure 5.3, since the proposed method is strictly convex which is proved theoretically in Section 4.2, the objective function converges to exact same global minima point which is -1.107.

5.2 Experiments on Simulated 2D Signal

The goal of this test case is visualizing the effects of the smoothness factor of the proposed change detection method in a better way on the third dimension. Since the proposed method claims that reducing the image noise while preserving the features of the objects such as edges and corners of the buildings, it is important to demonstrate the effectiveness of the algorithm.

5.2.0.3 2D Synthetic Map Generation

In accordance with this purpose, a 2D zeros matrix with size of 512×512 is created, then a variety of geometrical figures with different height values are inserted to represent some real world objects such as buildings, trees, grasses, lake, road and water way which is shown in Figure 5.4a. This generated ground truth is saved to evaluate performance of the proposed method and shown in Figure 5.4b. In order to obtain the synthetic data as a real height map, an example of height map, Figure 5.4c, with size of 2048×2048 from the Netherlands region was downloaded from the internet. Since the terrain elevation of the Netherlands is not very high, simulated changes to be added into the elevation map of this area can be easily seen. Then, it is scaled to size of the created synthetic 2D matrix. The 3D view of the height map can be seen in Figure 5.4d.



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

Figure 5.4: (a) Synthetically generated changes in colorful view (b) Ground truth (c) Initial height map in 2D view (d) Initial height map in 3D view

Two different input height maps of the proposed change detection method are shown in Figure 5.6a and Figure 5.6b designed as follows: One of these two inputs are thought as the first image that is captured at the first time, the other one is the second image that is taken later. It is assumed that; between these two time periods, there are variants such as the cutting of the trees in the area, the construction and destruction of the new buildings, the germination of grasses, the road constructions, the drying of the waterway and the formation of a lake, in course of time. For providing these adaptations, while some geometric figures are added into first image some of them is removed from the first image and some figures are applied to the second image in the same way. When playing with the figures to obtain change, there exists an important point to pay attention. After adding water-based figures into height maps to create changes in water areas, the Gaussian noise with the zero mean and standard deviation parameter as 2.5, is added to height maps in order to simulate the real world. The reason of this approach is that; in the real world, the water level slowly deepens from the ground towards the deepest side. At the end of applied this adaptations, Finally, zero mean Gaussian noise with constant noise amplitude, 1.0, and variance, 1.0, are added to both height maps, which are shown in Figure 5.5 as different colored versions, to adapt the sensor noise problem of the remote sensing data into the synthetic data. The signal noise ratios of these two synthetically generated height maps with these constants are 37.72 db and 37.80 db, respectively.

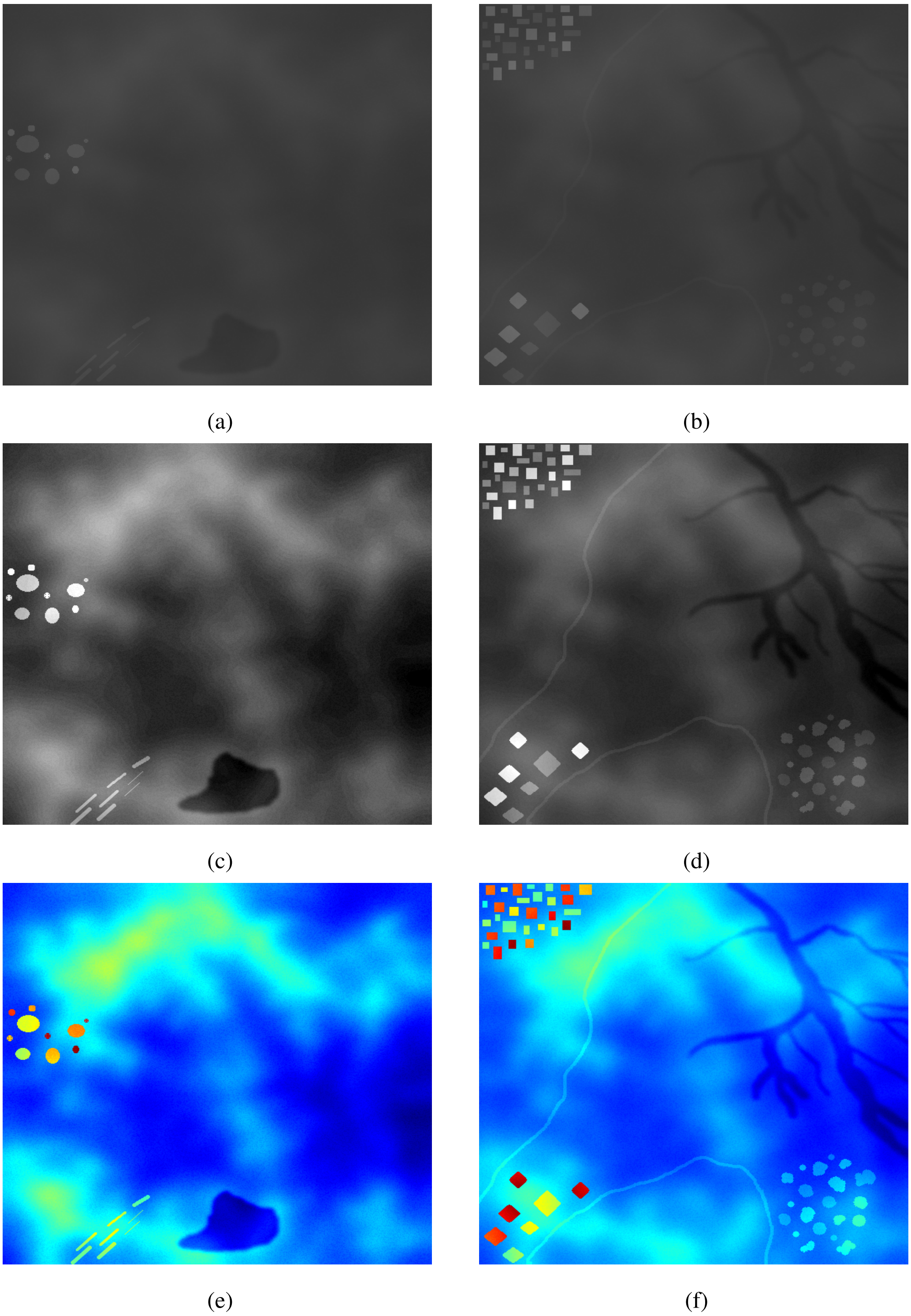
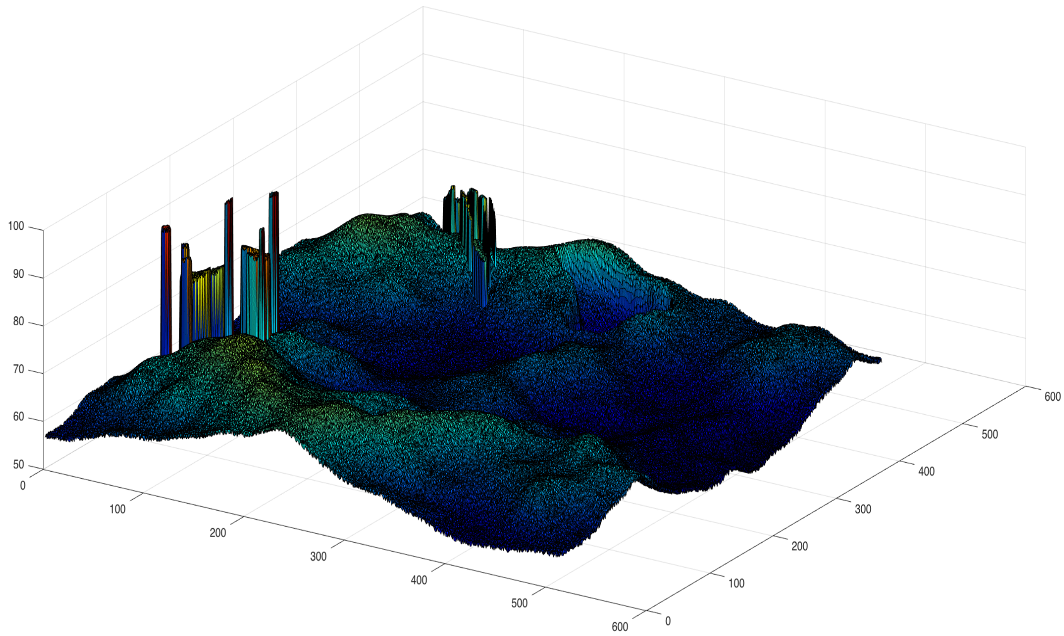
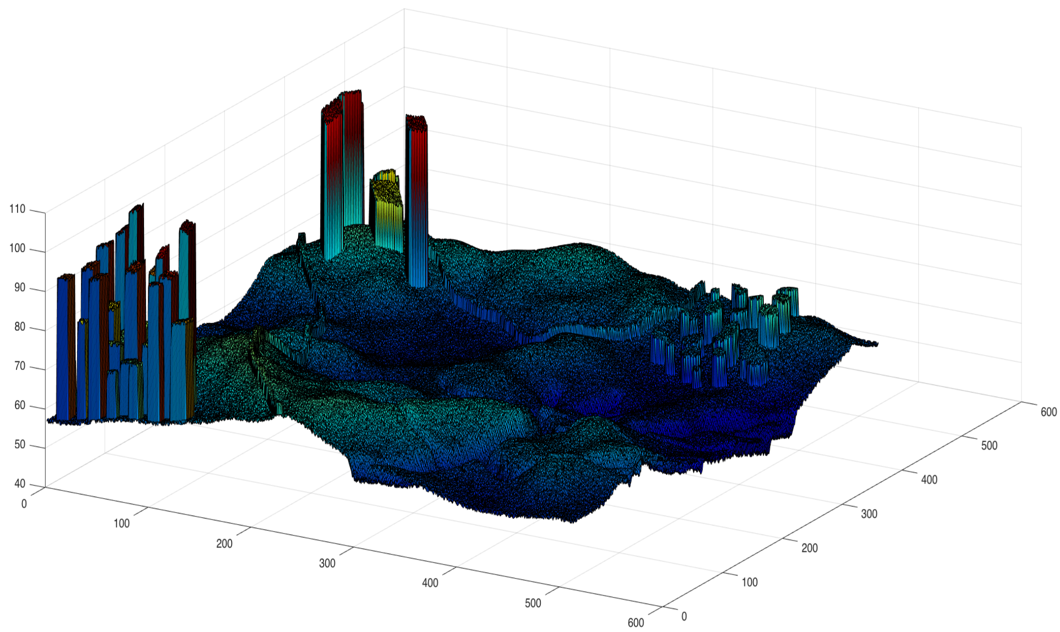


Figure 5.5: 2D views of synthetically generated height maps (a)The first height map as grayscale, at time t_1 (b) The second height map as grayscale, at time t_2 (c) The first height map as auto contrasted, at time t_1 (d) The second height map as auto contrasted, at time t_2 (e) The first height map as colored, at time t_1 (f) The second height map as colored, at time t_2

3D view of the height maps are also shown in Figure 5.6 to visualize the objects how added into the ground or extracted from the ground.



(a)

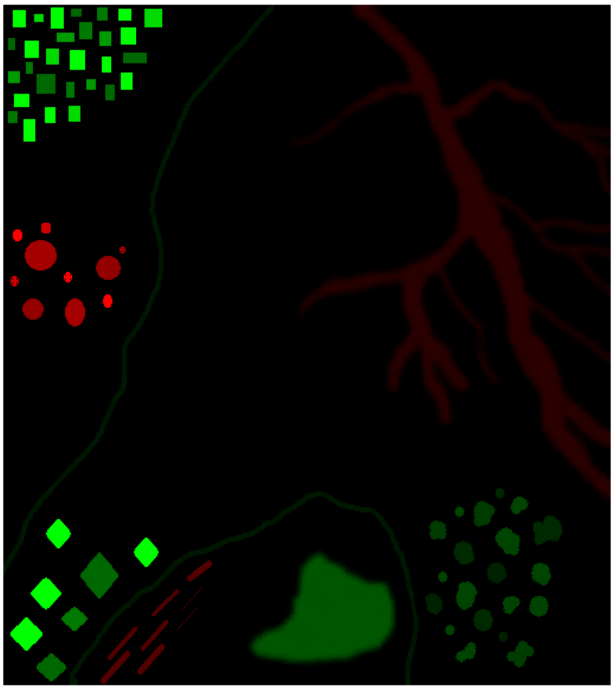


(b)

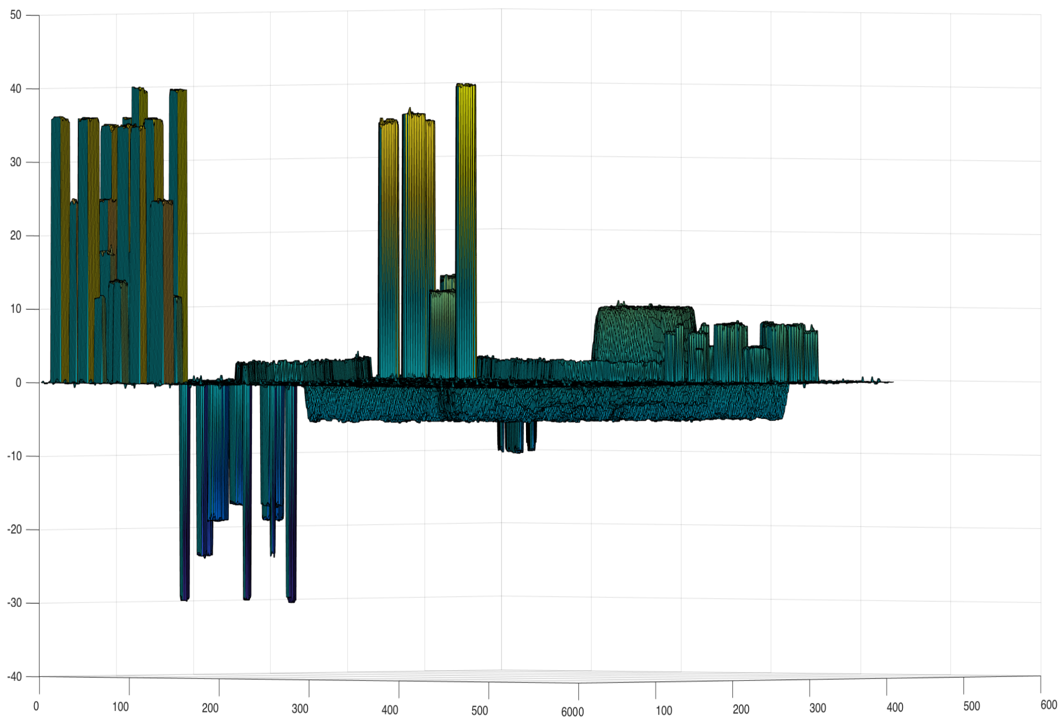
Figure 5.6: 3D views of synthetically generated height maps (a)The first height map, at time t_1 (b) The second height map, at time t_2

5.2.0.4 Experimental Results on 2D Synthetic Maps

In this section, the first constituent to be tested is the effect of the noise of the input height maps on the results of the proposed method. For this test case, in order to generate synthetic height maps having different noise levels, a set of various noise amplitude values but same noise variance, 1.0, are given to the Gaussian noise adding function. The proposed method is applied to height map tuples with different smoothness level, λ_s, for each noise level. The other parameters are set to $\varepsilon = 10^{-5}$, $C_{tolerance} = 10^{-5}$ which is called as convergence tolerance, and $n_{max} = 10^2$ which is maximum iteration.



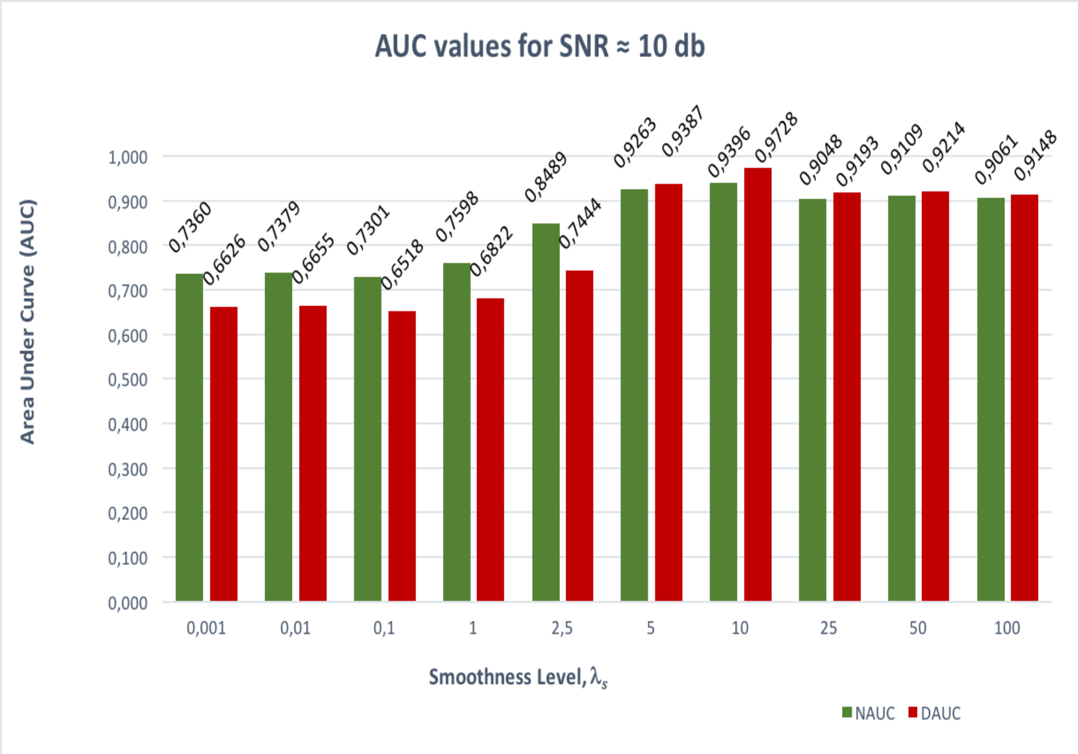
(a)



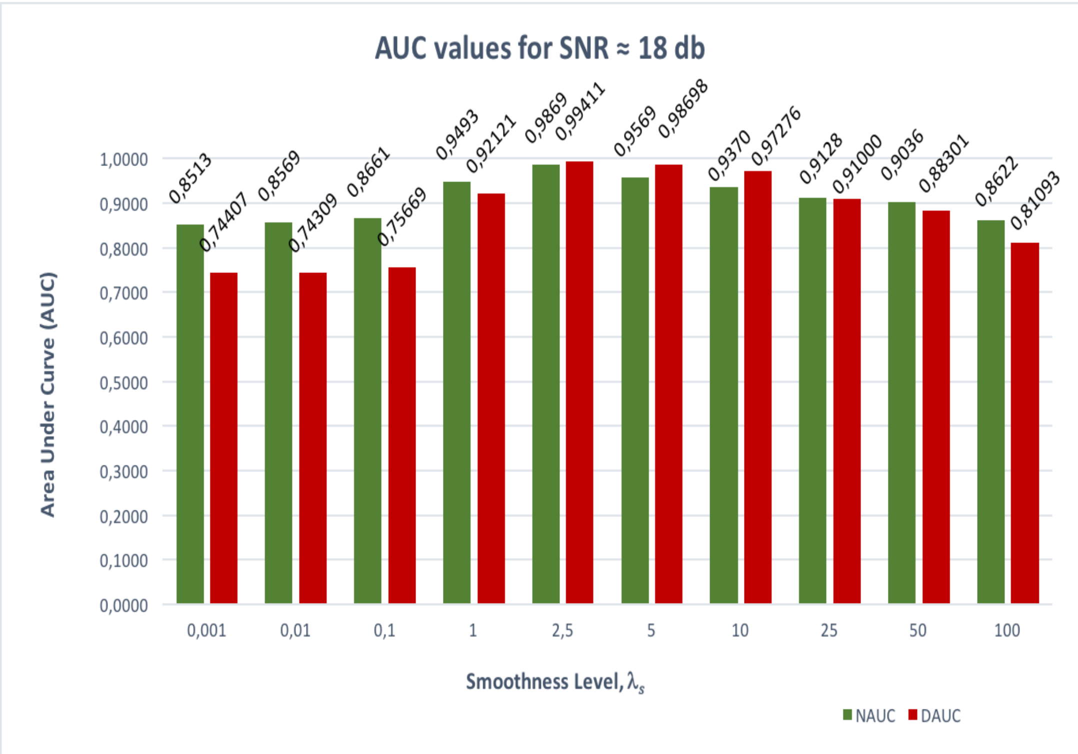
(b)

Figure 5.7: Change results of the proposed algorithm where $\lambda_s = 0.15$, SNR = 37.7 db and change map is initialized as described in Algorithm 7 (a) 2D colorful view of the change detection result (b) 3D view of the change detection result

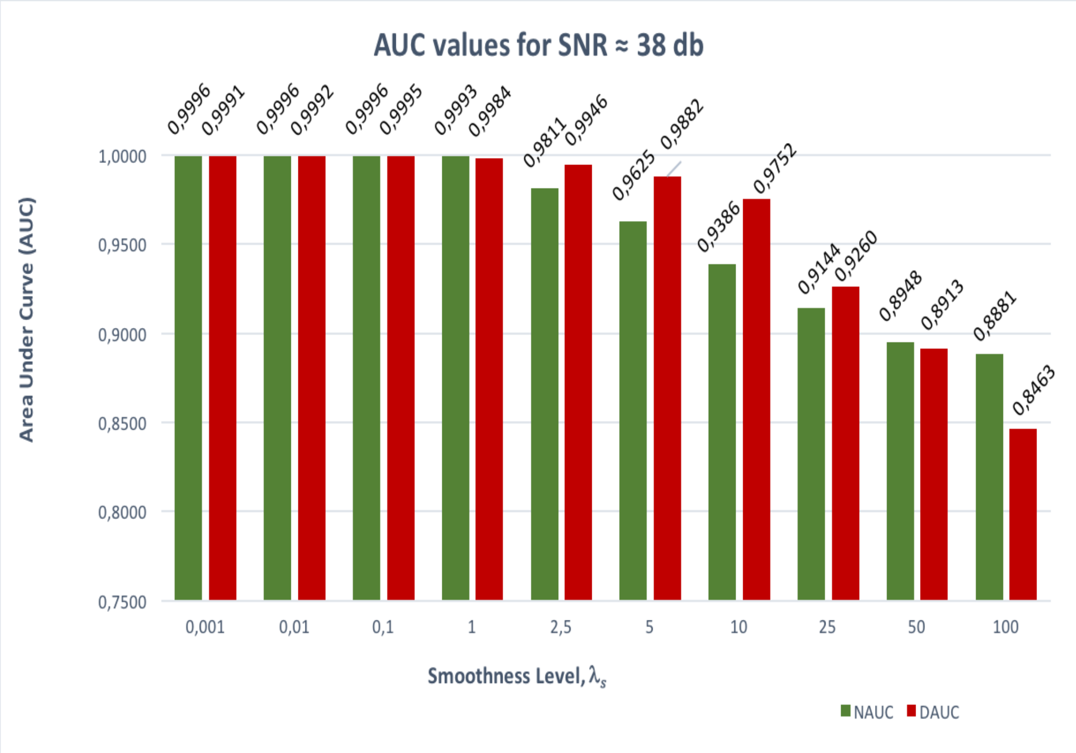
For this test scenario, five different noise levels are tested with the signal noise ratios as 10 db, 18 db, 38 db, 58 db, and 98 db. To achieve this the noise amplitudes are set to 25, 10, 1, 0.1, and 0.001, respectively, during the generation of the noisy maps. Note that, as the SNR increases the amount of noise in the elevation maps decreases. Hence, in test cases with high SNR values, more accurate results are obtained with low smoothness level parameters. Meanwhile; with low SNR values, more accurate results are obtained with high smoothness level parameters. However; when the smoothing parameter is greatly increased denoising operation is performed excessively. Because of this reason, the obtained result gets away from the ground truth and AUC values are decreased. To prove this statement quantitatively, area under curve (AUC) values are evaluated as a performance metric for each noise level and smoothness parameter, which are shown in Figure 5.8.


AUC values for SNR ≈ 10 db
Area Under Curve (AUC)
1,000
0,900
0,800
0,700
0,600
0,500
0,400
0,300
0,200
0,100
0,000
0,7360
0,6626
0,7379
0,6655
0,7301
0,6518
0,7598
0,6822
0,8489
0,7444
0,9263
0,9387
0,9396
0,9728
0,9048
0,9193
0,9109
0,9214
0,9061
0,9148
0,001
0,01
0,1
1
2,5
5
10
25
50
100
Smoothness Level, λ_s
NAUC
DAUC

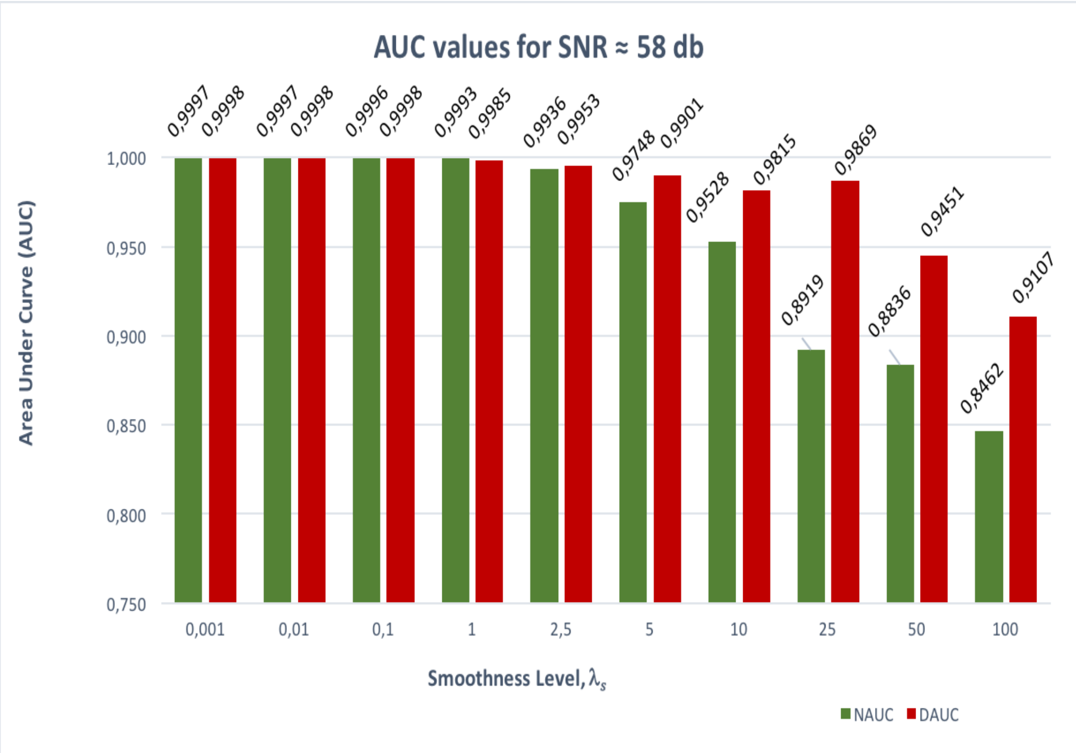
(a)


AUC values for SNR ≈ 18 db
Area Under Curve (AUC)
1,0000
0,9000
0,8000
0,7000
0,6000
0,5000
0,4000
0,3000
0,2000
0,1000
0,0000
0,8513
0,74407
0,8569
0,74309
0,8661
0,75669
0,9493
0,92121
0,9869
0,99411
0,9569
0,98698
0,9370
0,97276
0,9128
0,91000
0,9036
0,88301
0,8622
0,81093
0,001
0,01
0,1
1
2,5
5
10
25
50
100
Smoothness Level, λ_s
NAUC
DAUC

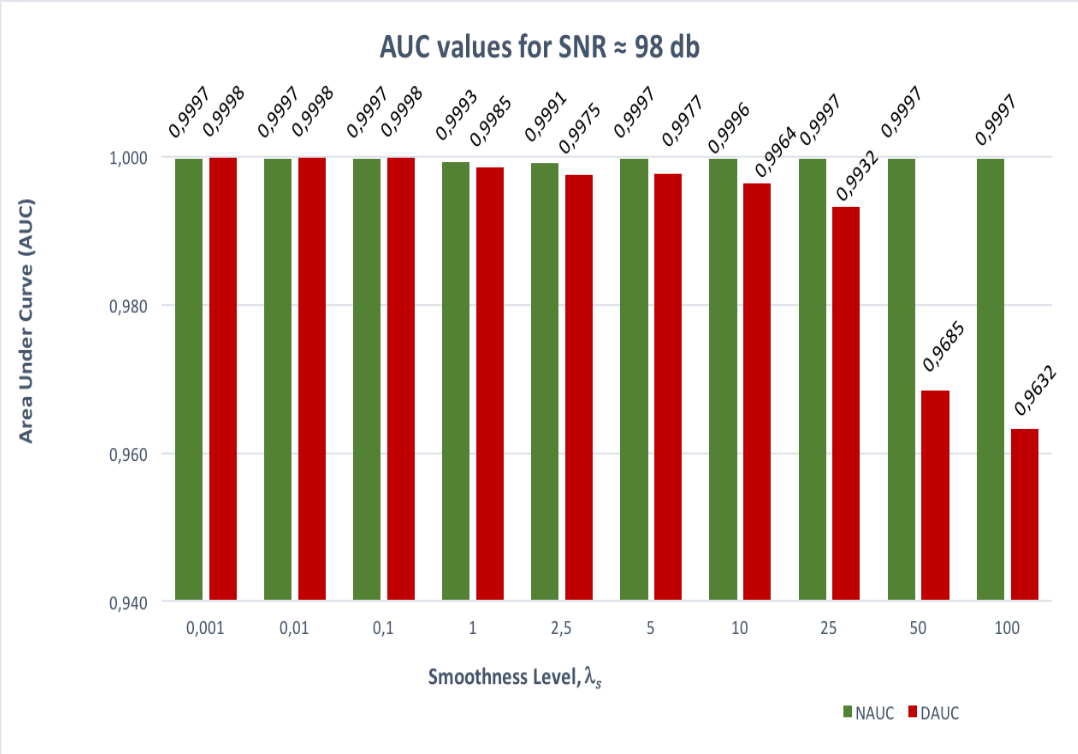
(b)


AUC values for SNR ≈ 38 db
Area Under Curve (AUC)
0,9996
0,9991
0,9996
0,9992
0,9996
0,9995
0,9993
0,9984
0,9811
0,9946
0,9625
0,9882
0,9386
0,9752
0,9144
0,9260
0,8948
0,8913
0,8881
0,8463
1,0000
0,9500
0,9000
0,8500
0,8000
0,7500
0,001
0,01
0,1
1
2,5
5
10
25
50
100
Smoothness Level, λ_s
NAUC
DAUC

(c)


AUC values for SNR ≈ 58 db
Area Under Curve (AUC)
0,9997
0,9998
0,9997
0,9998
0,9996
0,9998
0,9993
0,9985
0,9936
0,9953
0,9748
0,9901
0,9528
0,9815
0,8919
0,9869
0,8836
0,9451
0,8462
0,9107
1,000
0,950
0,900
0,850
0,800
0,750
0,001
0,01
0,1
1
2,5
5
10
25
50
100
Smoothness Level, λ_s
NAUC
DAUC

(d)



(e)

Figure 5.8: The AUC values diagrams of newly added (green) and demolished (red) objects for each smoothness level of the different noise levels (a) SNR = 10 db (b) SNR = 18 db (c) SNR = 38 db (d) SNR = 58 db (e) SNR = 98 db

In Figure 5.8a, since the synthetically created input height maps have too much signal noise, the low smoothness parameter, λ_s, is insufficient to solve problem. While the softening level parameter remained below 5, the performance of the algorithm remained low. As a result of excessively increasing the value of smoothing parameter, the performance of the algorithm may also decrease. As this effect is clearly seen in Figure 5.8b, the AUC values increase until the λ_s sets to 2.5, but after this value of the λ_s the AUC values start to fall. In other noise level tests, the accuracy of the proposed method usually decreases as the value is increased, while the performance results are usually fairly high with small smoothness values, which are shown in Figure 5.8c, Figure 5.8d and Figure 5.8e.

Table 5.1: Optimal Smoothness Parameters for Different Noise Levels

SNR	Noise Amplitude	λ_s	AUC - Newly Build Obj.	AUC - Demolished Obj.
10	25	7.0	0.9553	0.9761
18	10	2.5	0.9869	0.9941
38	1	0.141	0.9996	0.9995
58	0.1	0.005	0.9996	0.9998
98	0.001	0.015	0.9996	0.9998

The ideal elevation map pair which has SNR as 38 db is chosen to test other factors that can change the result of the algorithm from different noise levels. The initialization of the cost function which was tested on 1D signal is also tested on 2D height maps. In order to determine the smoothness level of the algorithm which gives the most optimum result, which has the highest AUC values, for this noise amount; the algorithm is run as Brute-Force with various smoothness level. λ_s is increased by 0.01 between 0.01 and 1 because this range gives the highest AUC values for this noise level as it can be seen in Figure 5.8c. The plotted receiver operating characteristic (ROC) curve which is evaluated AUC values for each λ_s in this range is shown in Figure 5.9.

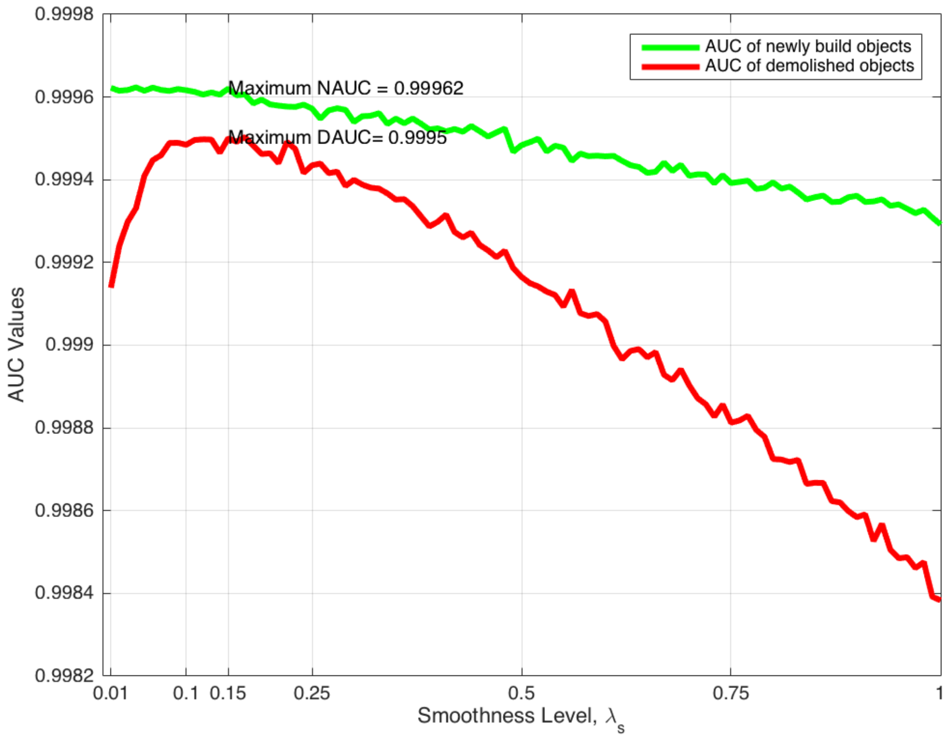
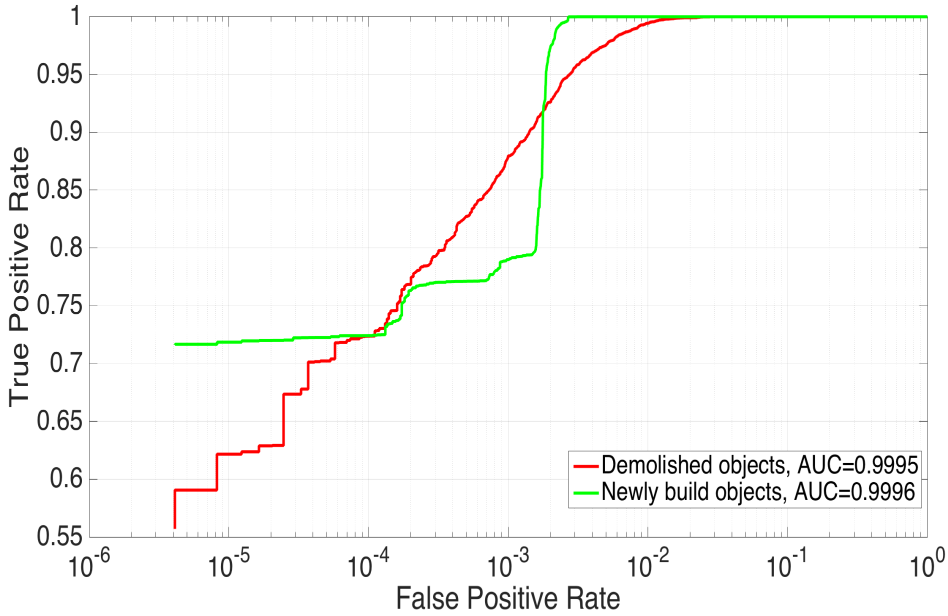


Figure 5.9: The ROC curves of newly build and demolished objects for SNR = 38 db and smoothness level range between 0.01 and 1.0

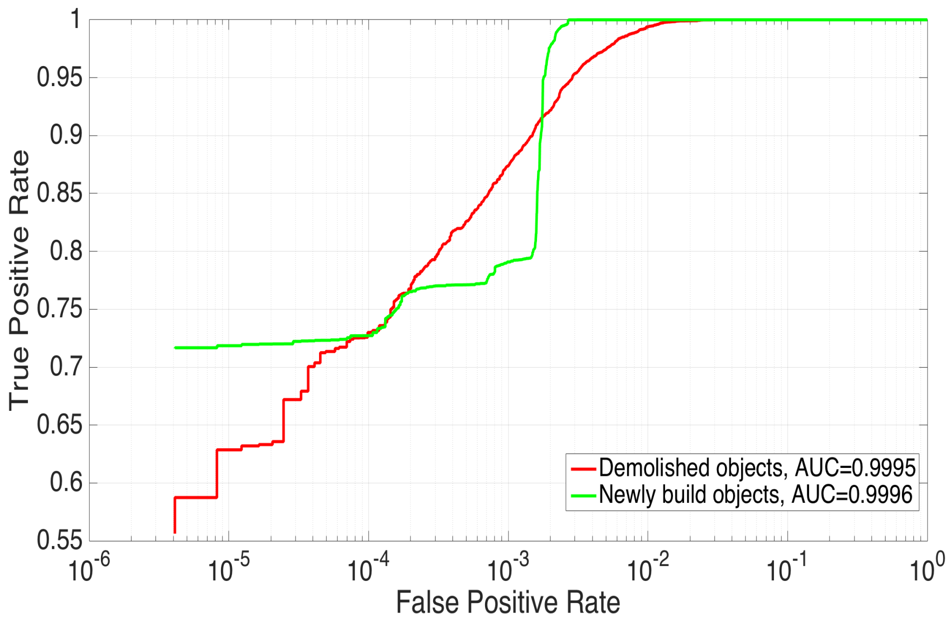
During the generation of synthetic dataset, between the first and second time period, demolished changes are adapted less than newly build objects in terms of height values and regional area. For this reason, in Figure 5.3, the AUC values of the demolished changes which is red ROC curve decreases more rapid than the AUC values of newly coming objects which is green ROC curve.

For selected height maps, the experiments are performed to see the effect of the different initializations of the cost function to be minimized on the result.

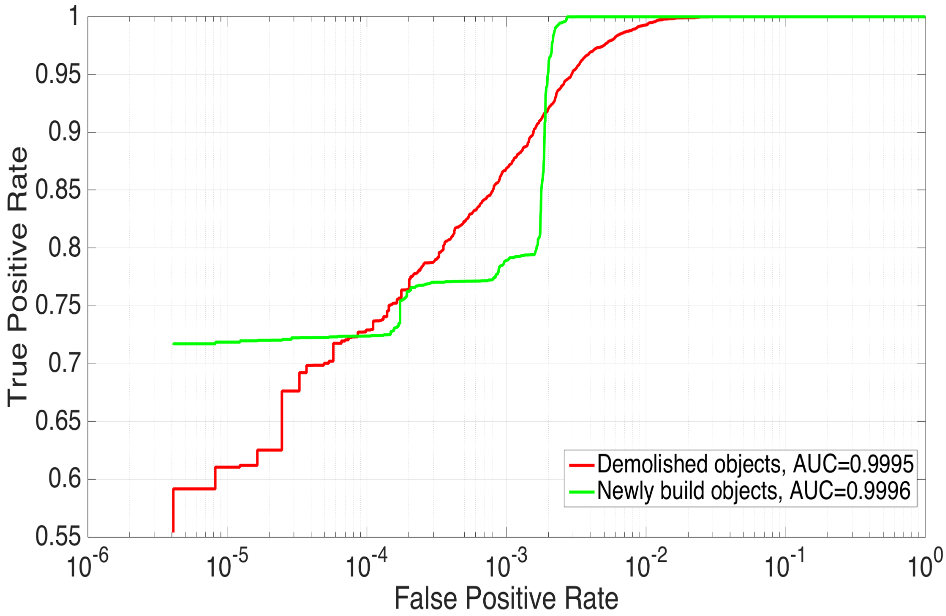
The experiments that indicate the effects of different initializations for the minimization problem are carried out. Because of this reason, the other parameters used in the proposed method are kept same for each test case. These parameters are set as $\lambda_s = 0.15$ which is determined as mentioned above, $\varepsilon = 10^{-5}$, $C_{tolerance} = 10^{-5}$ which is called as convergence tolerance, and $n_{max} = 10^2$ which is maximum iteration.


True Positive Rate
False Positive Rate
Demolished objects, AUC=0.9995
Newly build objects, AUC=0.9996

(a)

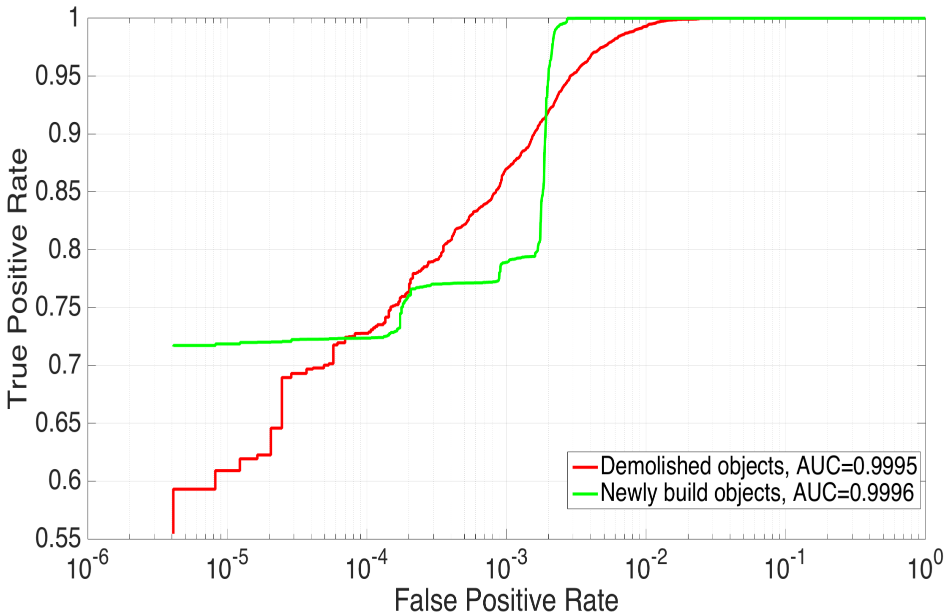

True Positive Rate
False Positive Rate
Demolished objects, AUC=0.9995
Newly build objects, AUC=0.9996

(b)



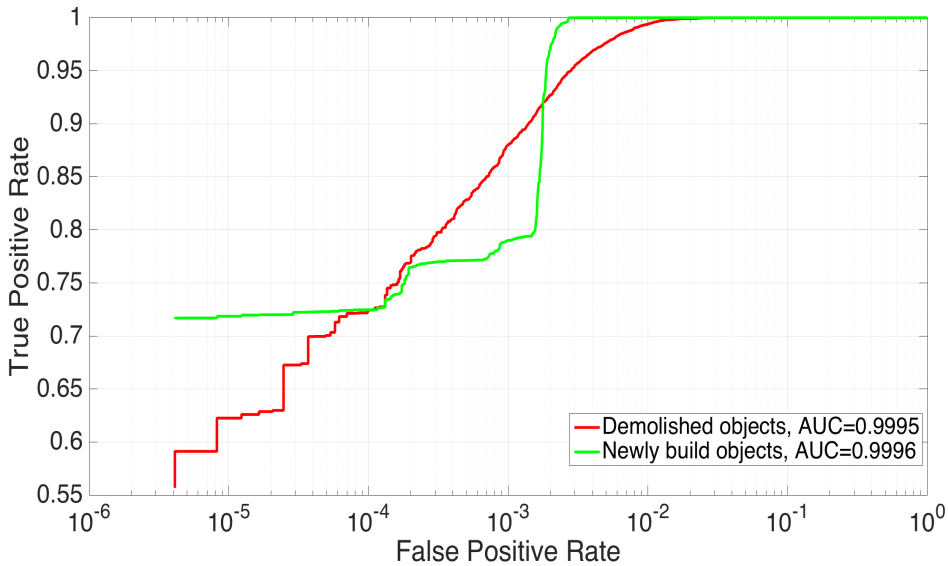
True Positive Rate
False Positive Rate
Demolished objects, AUC=0.9995
Newly build objects, AUC=0.9996

(c)

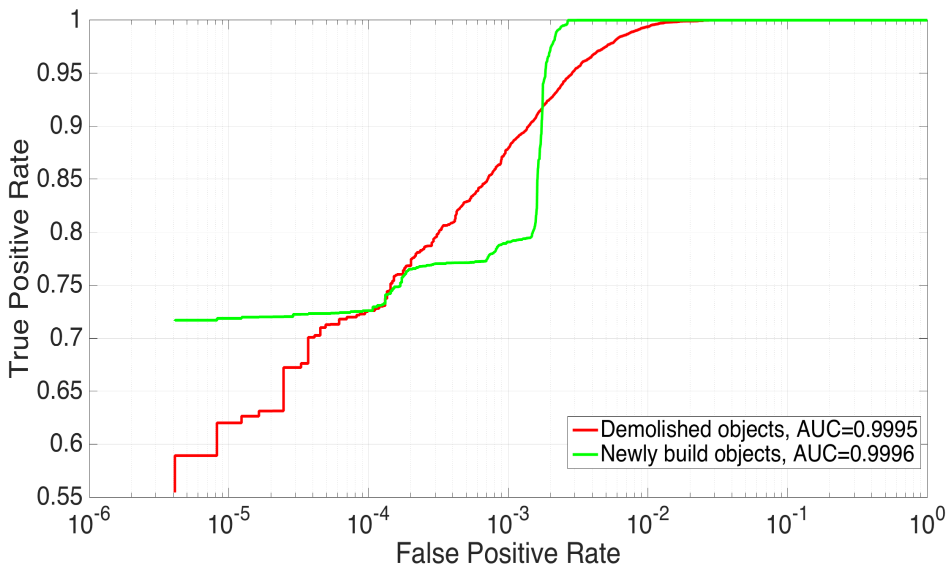


True Positive Rate
False Positive Rate
Demolished objects, AUC=0.9995
Newly build objects, AUC=0.9996

(d)



(e)

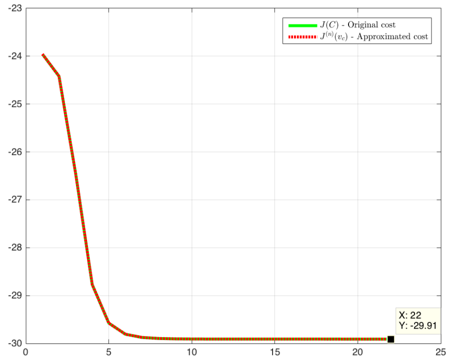


(f)

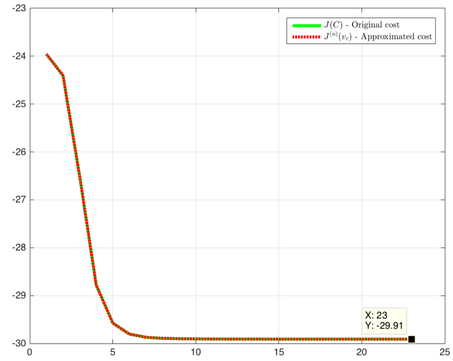
Figure 5.10: The effects on the AUC values of the initialization of the change map for the proposed cost function minimization (a) Change map is initialized with zeros and result change map of that case (b) Change map is initialized with ones and result change map of that case (c) Change map is initialized with uniformly distributed random numbers and result change map of that case (d) Change map is initialized with normalized random noise and result change map of that case (e) Change map is initialized with white gaussian noise having and result change map of that case (f) Change map is initialized as described in Algorithm 7 and result change map of that case

In Figure 5.10; area under curves which is the performance metric of the proposed method for both newly added and demolished objects, separately. As it can be obviously seen, the AUC values are same for each initialization despite the curve of the plots are different.

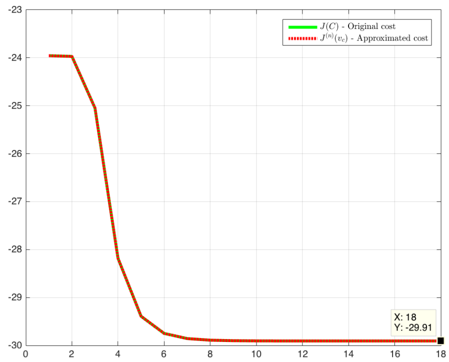
When the cost function is convex, the cost function converges to the same point regardless of the starting point of the minimization method. With these different initializations, the cost function converges to the same point which is shown in the following Figure 5.11.



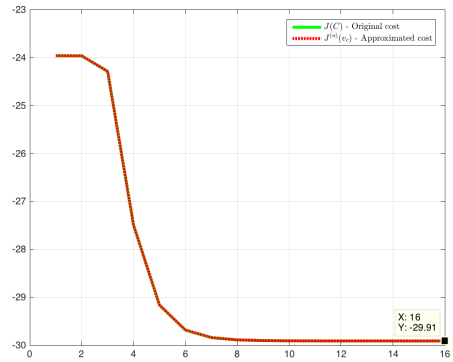
(a)



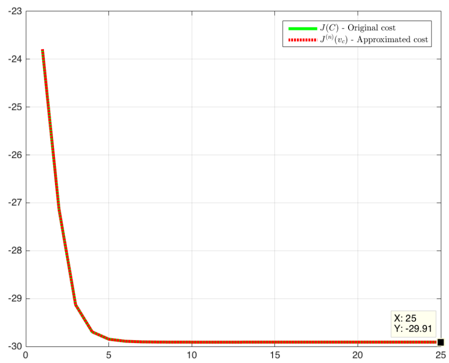
(b)



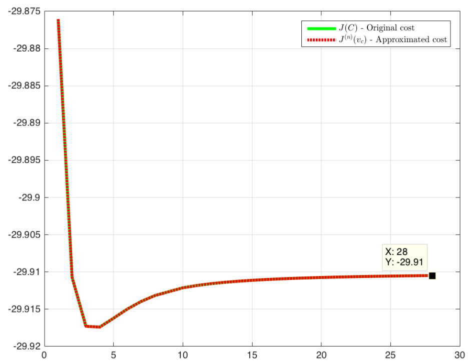
(c)



(d)



(e)



(f)

Figure 5.11: The indication that the cost function converges to same value for each initialization (a) Change map is initialized with zeros and result change map of that case (b) Change map is initialized with ones and result change map of that case (c) Change map is initialized with random noise and result change map of that case (d) Change map is initialized with normalized random noise and result change map of that case (e) Change map is initialized with white gaussian noise and result change map of that case (f) Change map is initialized as described in Algorithm 7

As it can be seen in Figure 5.11, since the proposed method is strictly convex which is proved theoretically in Section 4.2, the objective function converges to same global minima point which is -29.91.

5.3 Experiments on Synthetically Generated Real World Data

5.3.0.5 Synthetically Real World Data Generation

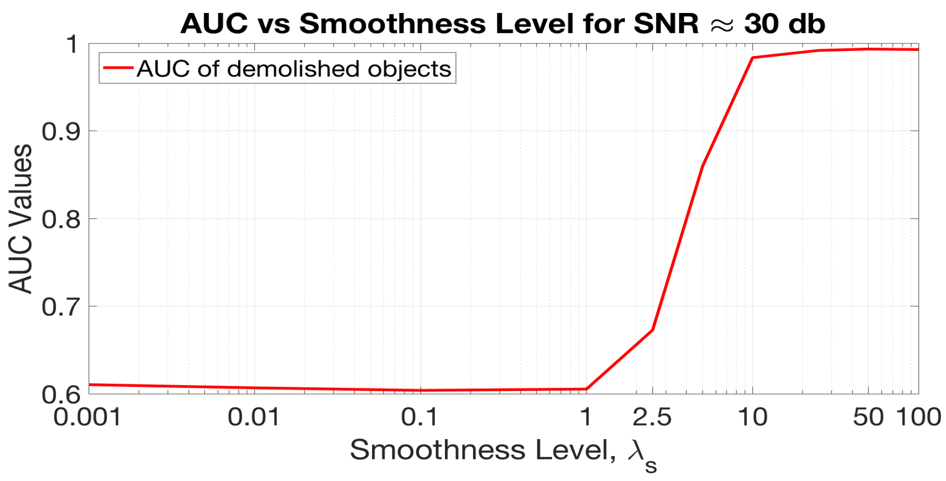
This dataset is prepared to apply the proposed method to see the effects of smoothness level for different noise levels on real world data. The data is collected from Cerkes village belongs to Cankiri city of Turkey via photogrammetry based techniques. The original data covers $11\ km^2$ area with $5\ cm$ pixel resolution. The whole area contains a wide variety of terrestrial shapes and real-world objects such as buildings, poles, cars, trees, bridges, roads, hills, rivers, etc. Nar et al. 2018, generated the DSM and DTM of the original data in his study [42].

For this study, the small patches of the DSM and DTM are specified to work on the proposed method. In order to create a variation between two time intervals, two input height maps of the proposed method is set as DSM patch and DTM patch. When DSM patch is chosen as the first input data and DTM patch is chosen as the second input, the changes that occur in two time intervals are seen as demolished objects. Because, the objects are seen as destructed while passing from surface model to terrain model. On the contrary, the changes are seen as newly build objects when DTM is chosen as the first input and DSM is chosen as the second input. This dataset is created with 5 DSM and 5 DTM patches.

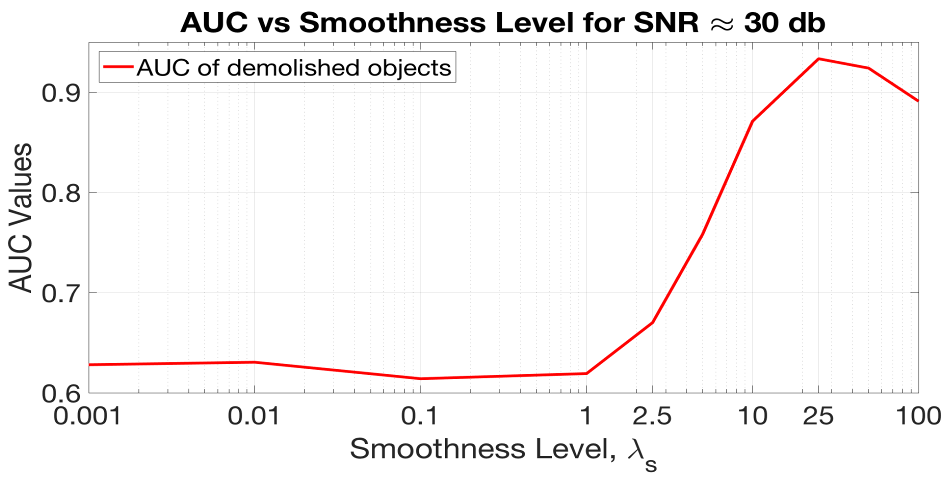
5.3.0.6 Experimental Results on Synthetically Generated Real World Data

In this set of experiments, first of all, 60% of the selected patches (3 DSM and 3 DTM patches) is allocated as training dataset to identify the best parameters of the proposed method and the remaining is allocated as test dataset to see the performance result of the proposed method with the determined parameters. In order to identify the optimum smoothness level for each noise level, zero mean Gaussian noise with different noise amplitude and noise variance parameters are added to train patches. Note that; in this test case, it does not matter the change type whether demolished or newly built because the AUC values are exactly same for each condition. So that, the inputs are set such that the objects disappear in this time interval. Hence, ROC curves are only plotted for demolished objects. The GT is obtained by applying a small threshold value, $\tau = 0.01$, to difference of these elevation maps such that $GT = (DSM - DTM) > \tau$.

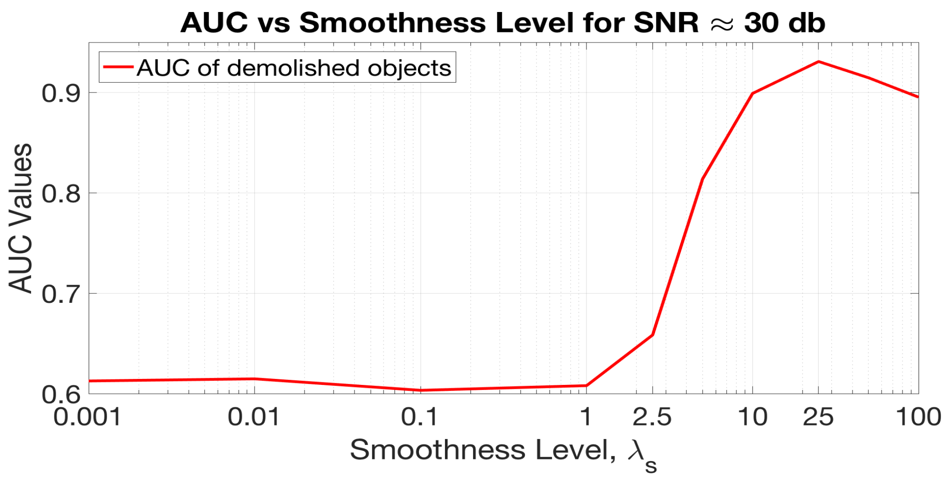
In order to visualize the effects of the smoothness parameter of the proposed method for different noise level on the training data, ROC curves of each case for some noise levels, SNR approximately equals to 29 db, 57 db and 97 db, are shown in Figure 5.25, Figure 5.13 and Figure 5.14, respectively.



(a)

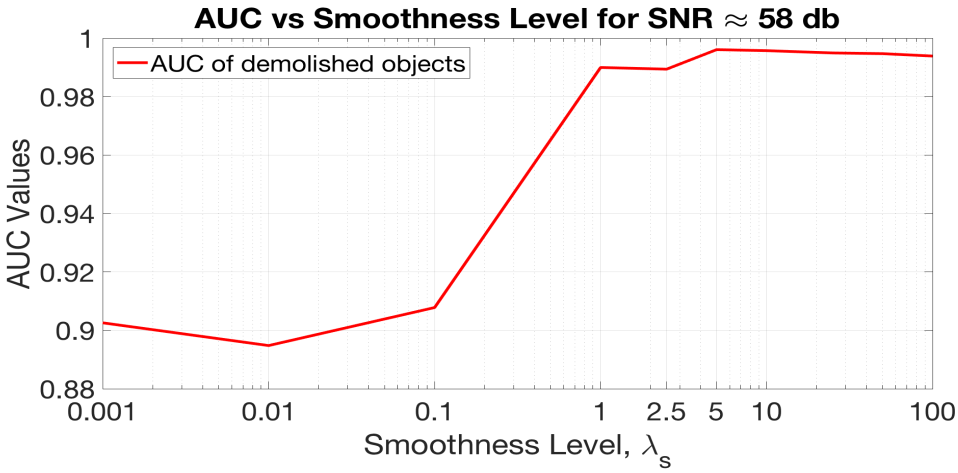


(b)

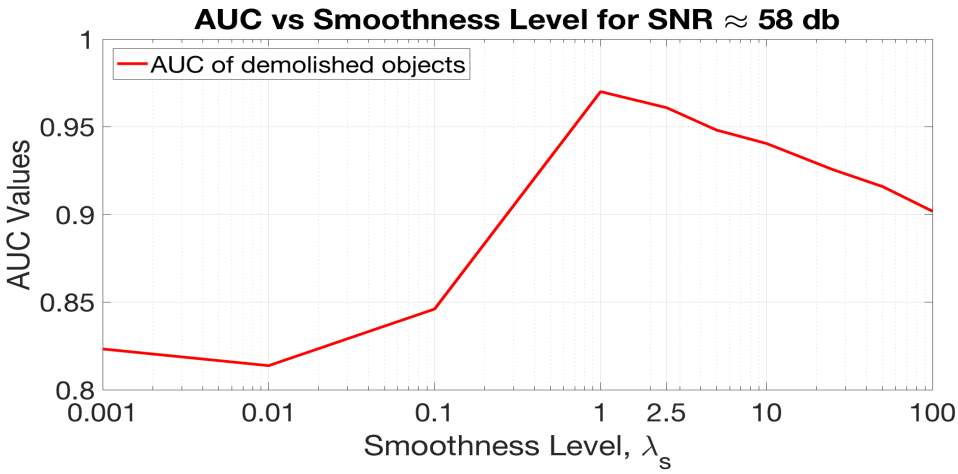


(c)

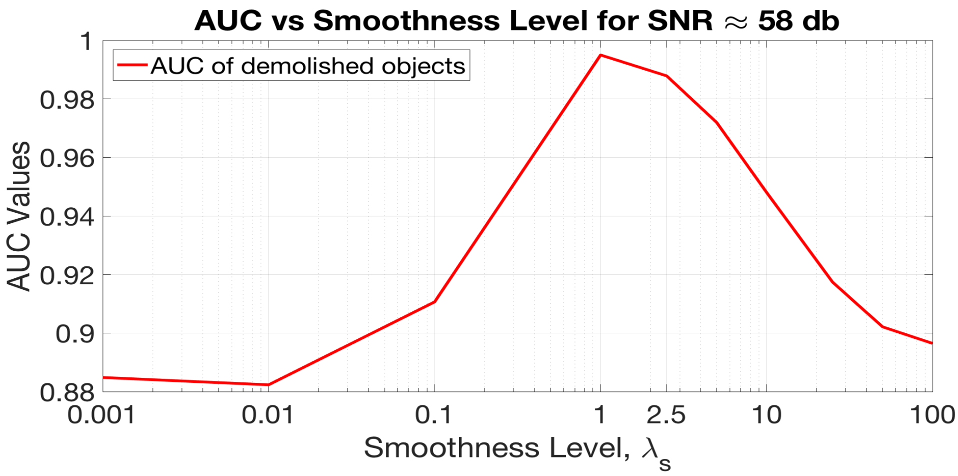
Figure 5.12: The ROC curves of each training samples at identical noise level, SNR $\approx$ 29 db (a) ROC curve of the first training pair (b) ROC curve of the second training pair (c) ROC curve of the third training pair



(a)

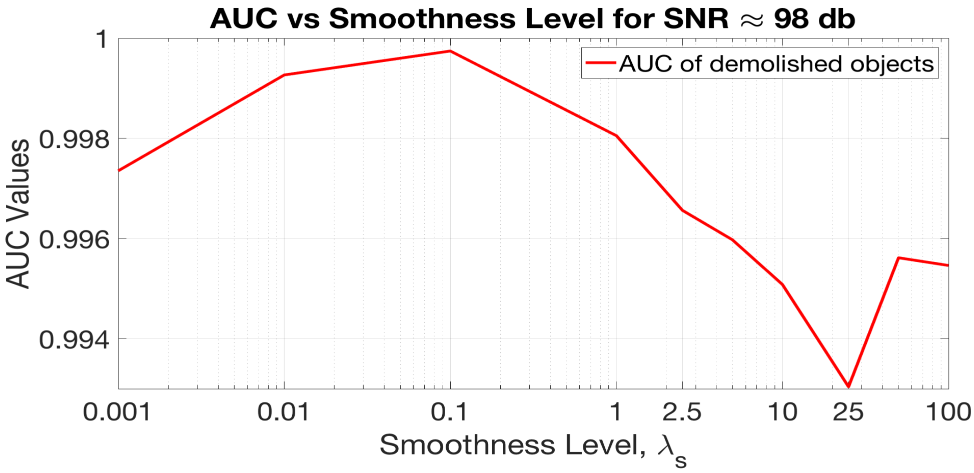


(b)

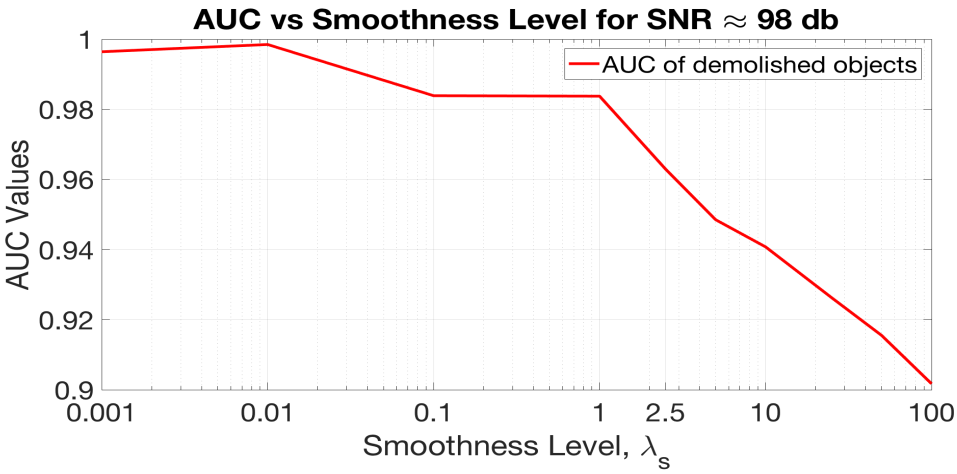


(c)

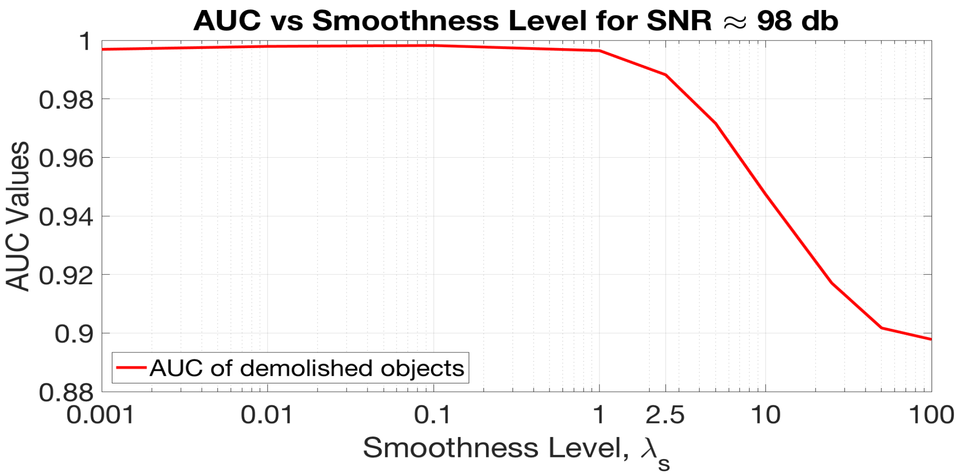
Figure 5.13: The ROC curves of each training samples at identical noise level, SNR ≈ 57 db (a) ROC curve of the first training pair (b) ROC curve of the second training pair (c) ROC curve of the third training pair



(a)



(b)



(c)

Figure 5.14: The ROC curves of each training samples at identical noise level, SNR $\approx$ 97 db (a) ROC curve of the first training pair (b) ROC curve of the second training pair (c) ROC curve of the third training pair

The maximum AUC values for each noise level independent of the level of smoothness are indicated in Table 5.2;

Table 5.2: Maximum AUC Values

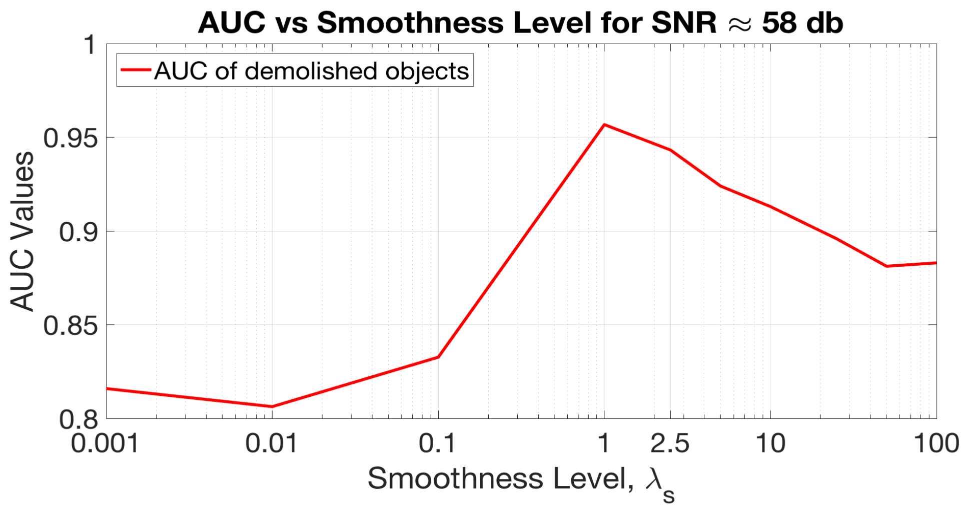
	$\sim$ 29 db	**$\sim$ 37 db**	**$\sim$ 57 db**	**$\sim$ 77 db**	**$\sim$ 97 db**	**$\sim$ 117 db**
Patch1	0.9933	0.9947	0.9961	0.9984	0.9997	1.0000
Patch2	0.9335	0.9267	0.9701	0.9848	0.9985	1.0000
Patch3	0.9308	0.9636	0.9950	0.9963	0.9982	1.0000

The average values of AUC values of three training patches are estimated for each smoothness level at each noise level which is shown in Table 5.3. The best smoothness levels that give the highest average AUC value is determined as the optimum λ_s for each chosen noise level and they are shown with red colors.

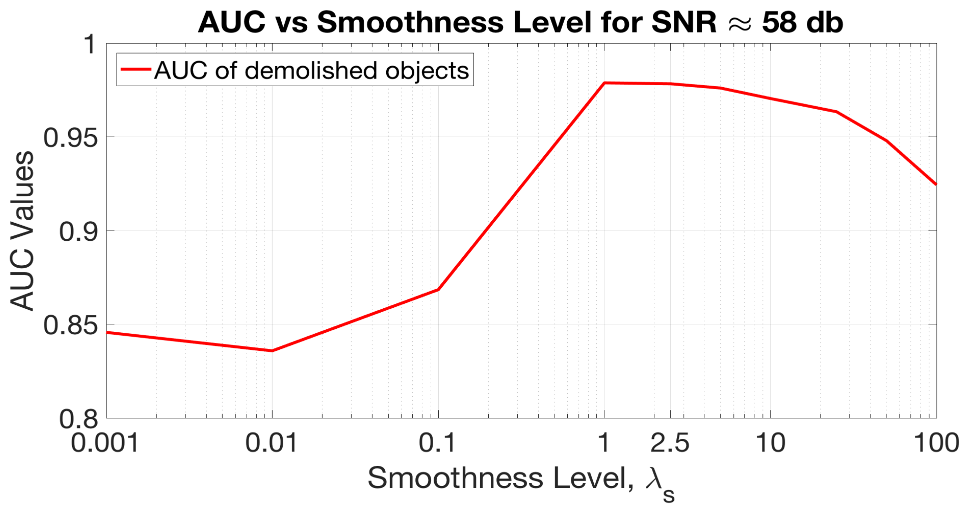
Table 5.3: Average AUC Values

SNR \ λ_s	**0.001**	**0.01**	**0.1**	**1.0**	**2.5**	**5.0**	**10.0**	**25.0**	**50.0**	**100.0**
$\sim$ 29 db	0.6170	0.6173	0.6071	0.6108	0.6671	0.8106	0.9179	**0.9520**	0.9440	0.9264
$\sim$ 37 db	0.6803	0.6798	0.6779	0.7747	0.9066	0.9523	**0.9576**	0.9433	0.9364	0.9306
$\sim$ 57 db	0.8702	0.8636	0.8882	**0.9850**	0.9794	0.9721	0.9614	0.9460	0.9376	0.9307
$\sim$ 77 db	0.9390	0.9465	0.9900	**0.9928**	0.9828	0.9721	0.9611	0.9461	0.9371	0.9326
$\sim$ 97 db	0.9969	**0.9986**	0.9939	0.9928	0.9826	0.9720	0.9611	0.9455	0.9376	0.9317
$\sim$ 117 db	**1.0000**	0.9996	0.9978	0.9926	0.9825	0.9720	0.9610	0.9454	0.9371	0.9304

According to table of average AUC values, a testable noise level, $\approx$ 57 db, is chosen to show the performance results of the proposed method on the test patches. ROC curves of these test samples for different smoothness parameters are shown in Figure 5.15:



(a)



(b)

Figure 5.15: The ROC curves of each test samples at identical noise level, SNR ≈ 57 db (a) ROC curve of the first test pair (b) ROC curve of the second test pair

As it can be seen in Figure 5.15, the proposed algorithm yields the highest AUC value with the smoothness level parameter as $\lambda_s = 1$ for each test patches. The maximum AUC values are 0.95676 and 0.95676 for first test pair and second pair, respectively.

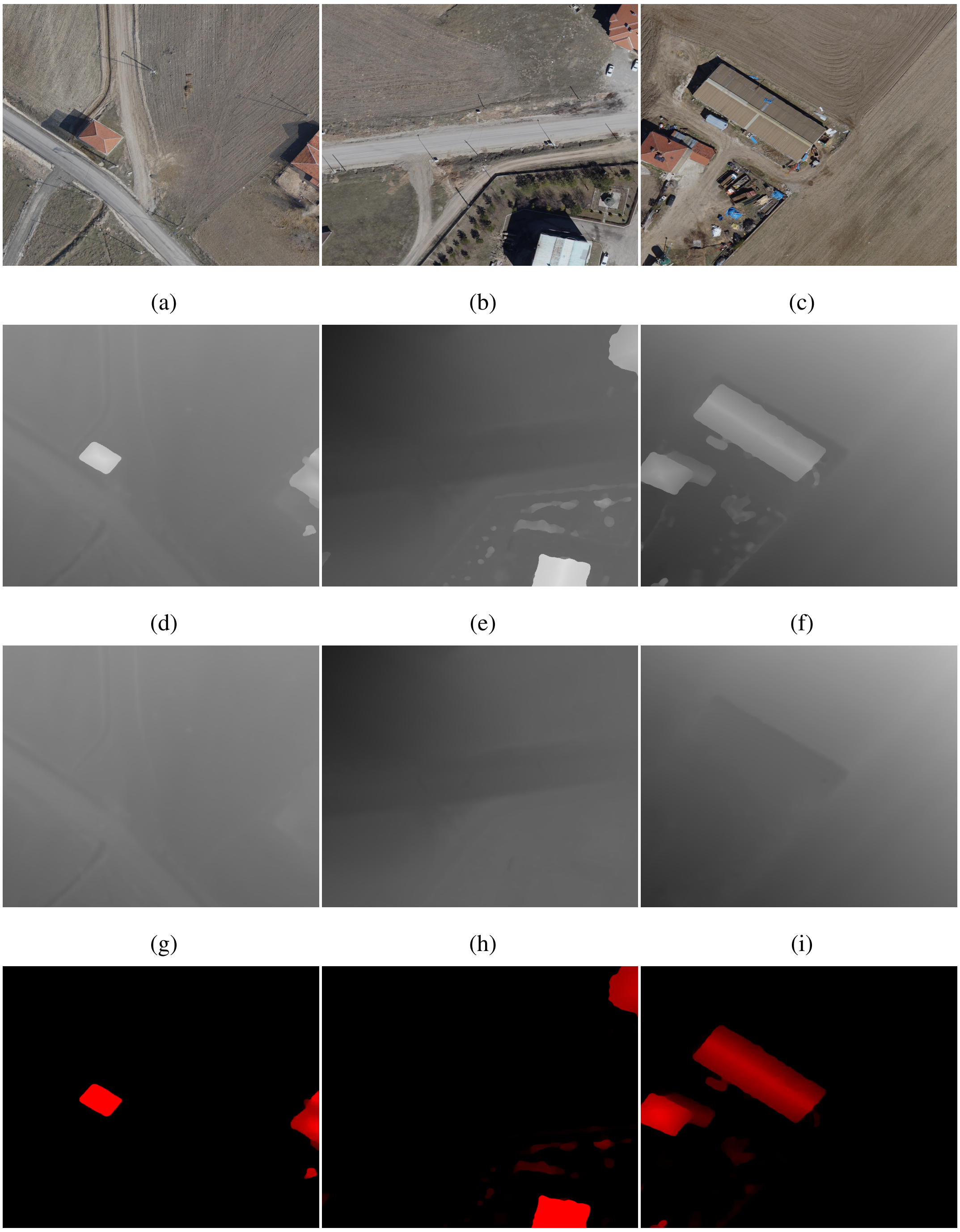
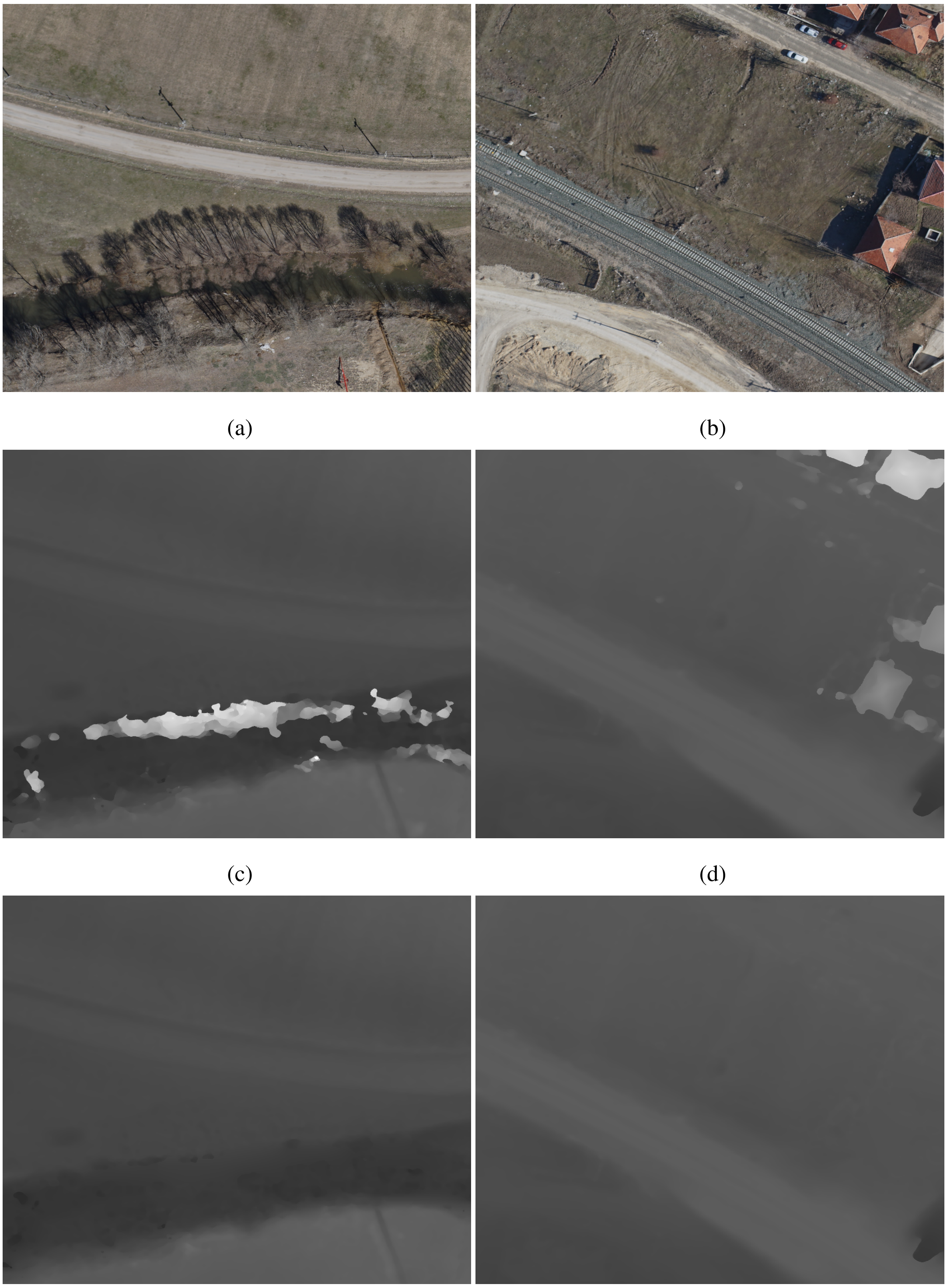

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)
(j) (k) (l)

Figure 5.16: 2D views of three training patches and change results (a-c) Raster images of the training patches (d-f) DSMs of the training patches (g-i) DTMs of the training patches (j-l) Change result maps of the training patches, where SNR ≈ 57 db, $\lambda_s = 1.0$

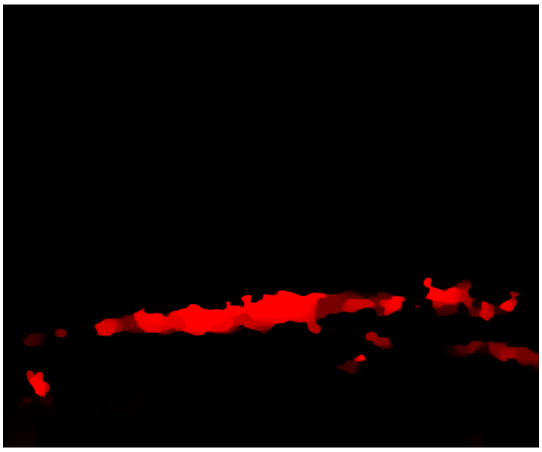
The training input patches of the proposed method and the result change maps of these DSM and DTMs with the determined optimum smoothness parameter at an average noise level are shown in Figure 5.16.



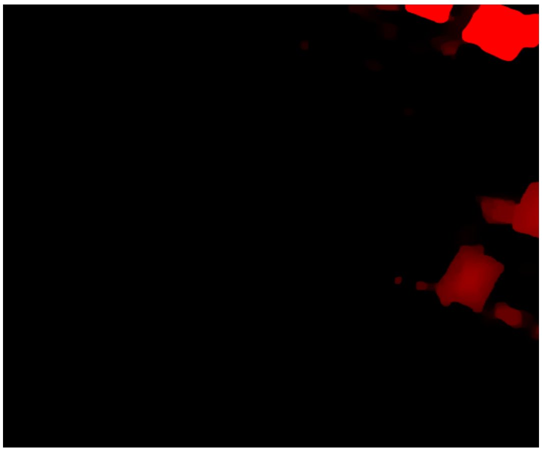
(a) (b)

(c) (d)

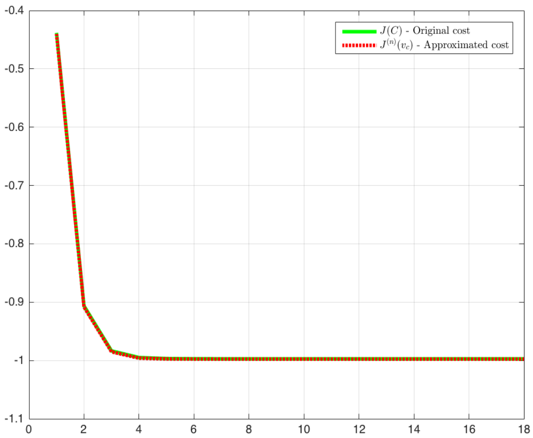
(e) (f)



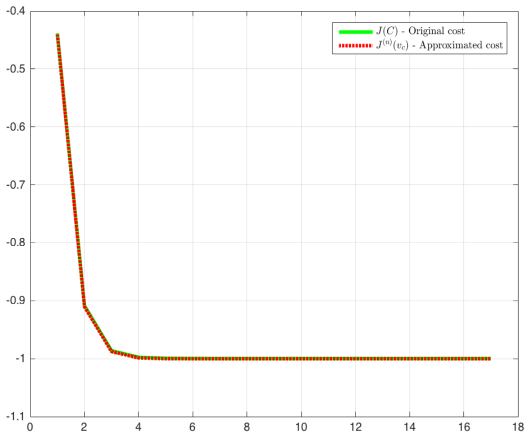
(g)



(h)



(i)



(j)

Figure 5.17: 2D views of three training patches and change results (a-b) Raster images of the test patches (c-d) DSMs of the test patches (e-f) DTMs of the test patches (g-h) Change result maps of the test patches, where SNR $\approx$ 57 db, $\lambda_s = 1.0$ (i-j) Cost functions of the test patches

In Figure 5.18, the ROC curves are plotted of the fourth and fifth test patch with the optimum smoothness level, $\lambda_s = 1.0$ and SNR $\approx$ 57 db, separately.

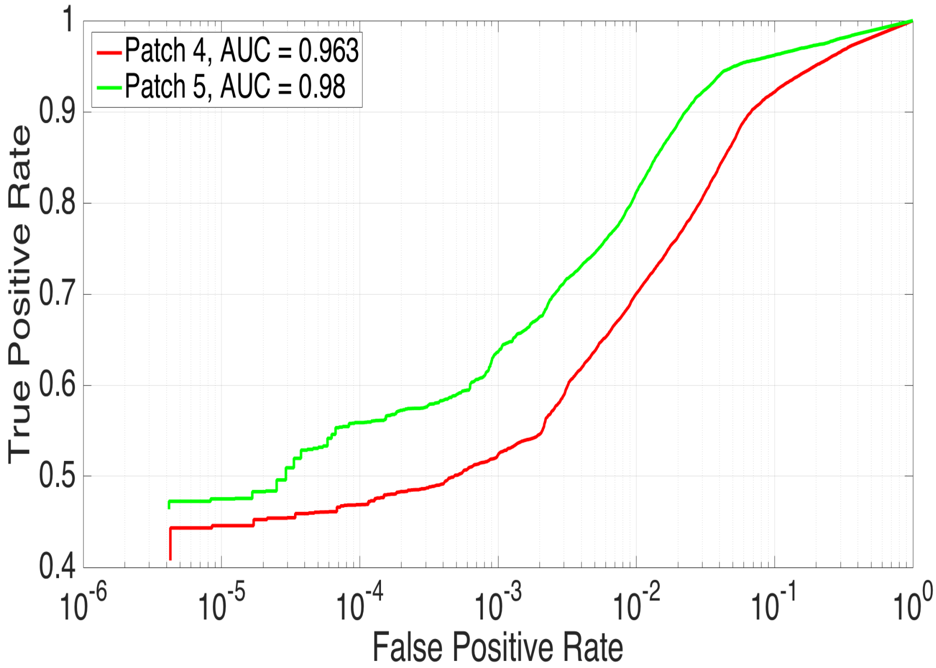


Figure 5.18: ROC curves of the test patches

5.4 Experiments on LIDAR Based DSM

5.4.0.7 Real World Dataset

In this section, the real world 3D point cloud data acquired by Lidar technology is used in order to apply the proposed method and to indicate its performance results. The point cloud samples used in this study are obtained in two different ways. Firstly, they are downloaded from the OpenTopography site, which is available to everyone as an open source. From this open-source site, data that are determined as appropriate for conducting the tests are taken from the same region but from different years are detected. These identified data belong to the Wax Lake region in the state of Louisiana, USA. The acquisition and processing of the data with the Lidar sensor is completed by the National Center for Airborne Laser Mapping (NCALM). Flights were conducted in 2009 and 2013 to obtain data from the region. The second way of obtaining data is purchasing from mapping company. This data are taken from Bilbao region in Spain in 2005 and 2008. The properties of the Lidar dataset such as location, coordinates, number of points, density, year, etc. are stored in their metadata as listed in Table 5.4:

Table 5.4: Lidar Dataset Properties

Region	**Year**	**Upper Left Coordinates**	**Lower Right Coordinates**	**Number of Points**	**Covered Area** (km^2)	**Density** ($samples/m^2$)
Wax Lake, Louisiana, USA	2009	29° 32’ 21.2127" N, 91° 26’ 38.3007" W	29° 32’ 04.1288" N, 91° 25’ 54.8649" W	1169540	0.60	4677
Wax Lake, Louisiana, USA	2013	29° 32’ 21.2127" N, 91° 26’ 38.3007" W	29° 32’ 04.1288" N, 91° 25’ 54.8649" W	3940057	0.56	21
Wax Lake, Louisiana, USA	2009	29° 32’ 21.2127" N, 91° 26’ 38.3007" W	29° 32’ 04.1288" N, 91° 25’ 54.8649" W	750669	0.15	7093
Wax Lake, Louisiana, USA	2013	29° 32’ 21.2127" N, 91° 26’ 38.3007" W	29° 32’ 04.1288" N, 91° 25’ 54.8649" W	2486672	0.15	33287
Bilbao Spain	2005	43° 15’ 31.9471" N, 02° 04’ 32.6882" W	43° 12’ 47.1216" N, 01° 57’ 13.1534" W	7982486	4.00	2302
Bilbao, Spain	2008	43° 15’ 31.9471" N, 02° 04’ 32.6882" W	43° 12’ 47.1216" N, 01° 57’ 13.1534" W	6623064	4.00	2144

5.4.0.8 Experimental Results on Real World Dataset

Since the first set of real world Lidar data is already geolocated, accurately; the point cloud data is directly employed in the preprocessing step, DSM generation, of the proposed method. The intermediate outputs of the DSM generation process is shown in Figure 5.19. The parameters used for the DSM generation method are as follows; grid size, $R = 0.5$ and smoothness level, $\lambda = 2.50$.

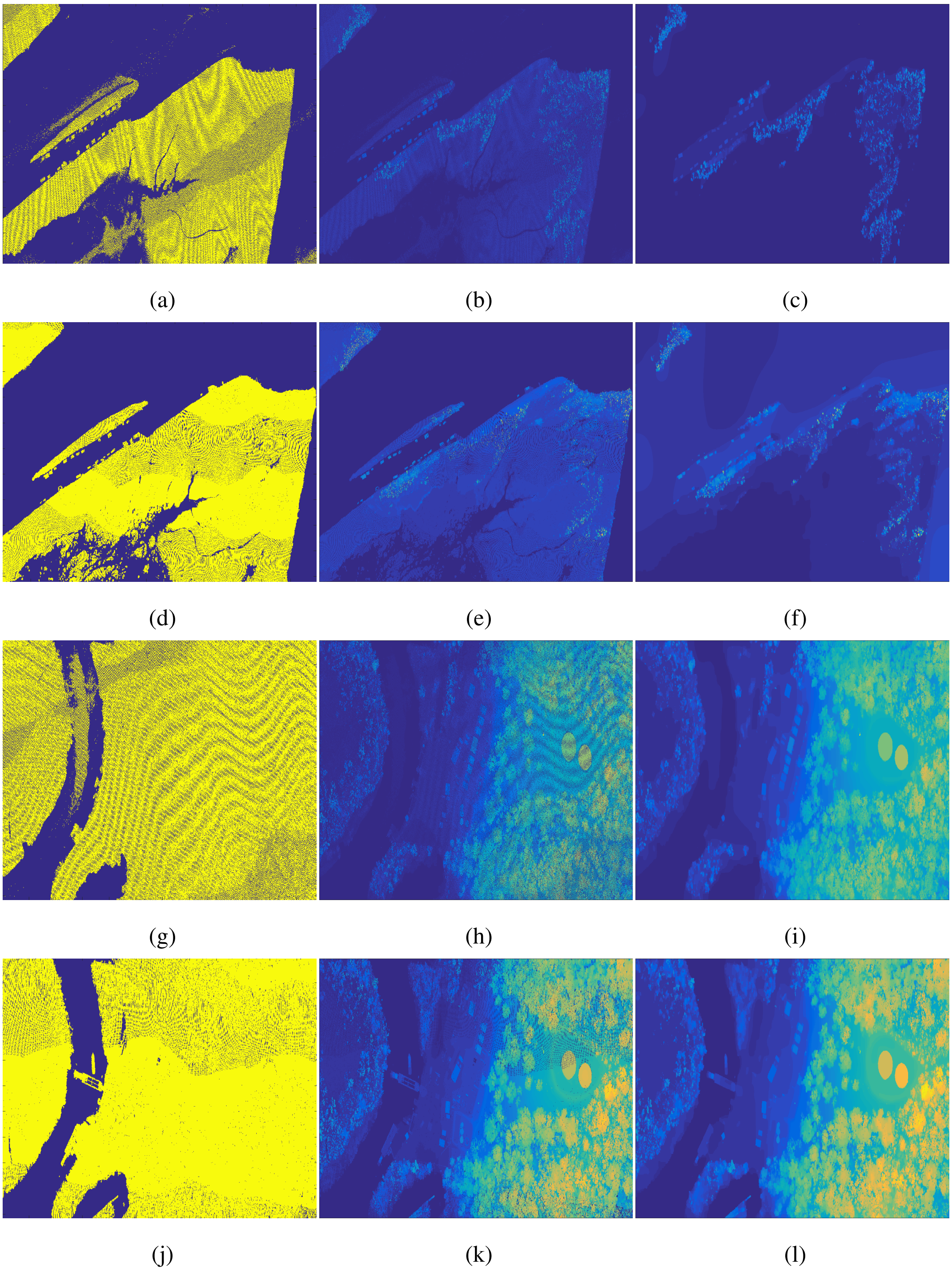
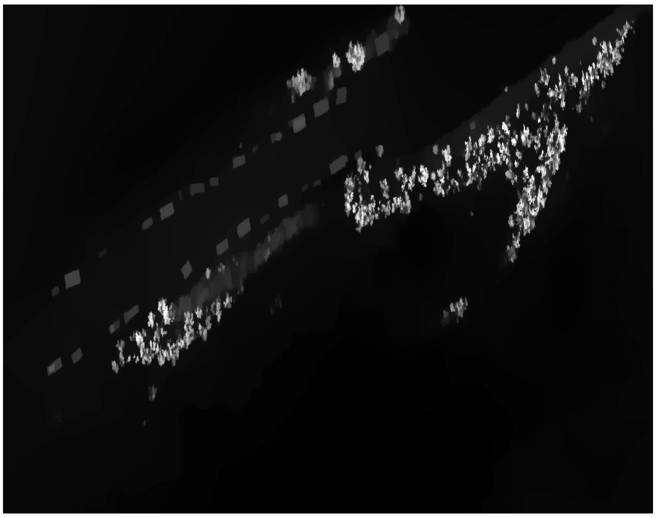


Figure 5.19: The first columns are indicator matrices, the second columns are height maps and the right most columns are digital surface models (a-c) First region in 2009 (d-f) First region in 2013 (g-i) Second region in 2009 (j-l) Second region in 2013

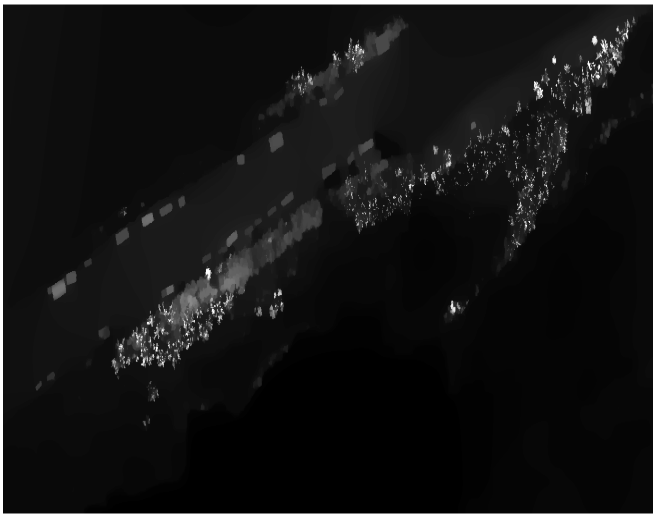
In the obtained surface models, the regions where the changes are clearly evident are determined and cropped from the entire map. The ground truth is generated by a geomatic engineer for demolished and newly built changes in the cropped patches, separately. The raster images of open source data are taken by Google Earth engine, currently.

The following figures indicate the DSMs as inputs, the result change map as output and ROC curves as performance metrics of the proposed method for each data pair.

(a)

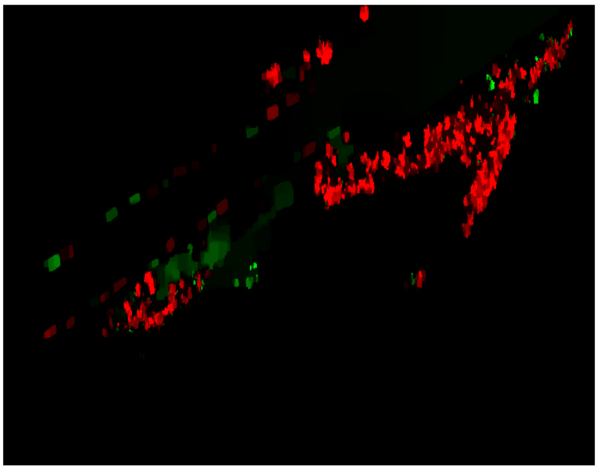


(b)

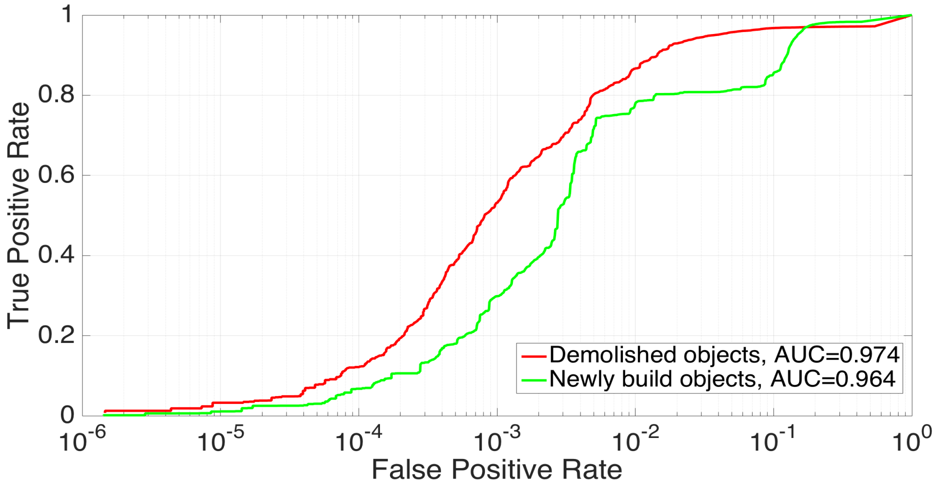


(c)

Figure 5.20: 2D images of Wax Lake region (a) Electro optic raster image (b) DSM, 2009 (c) DSM, 2013

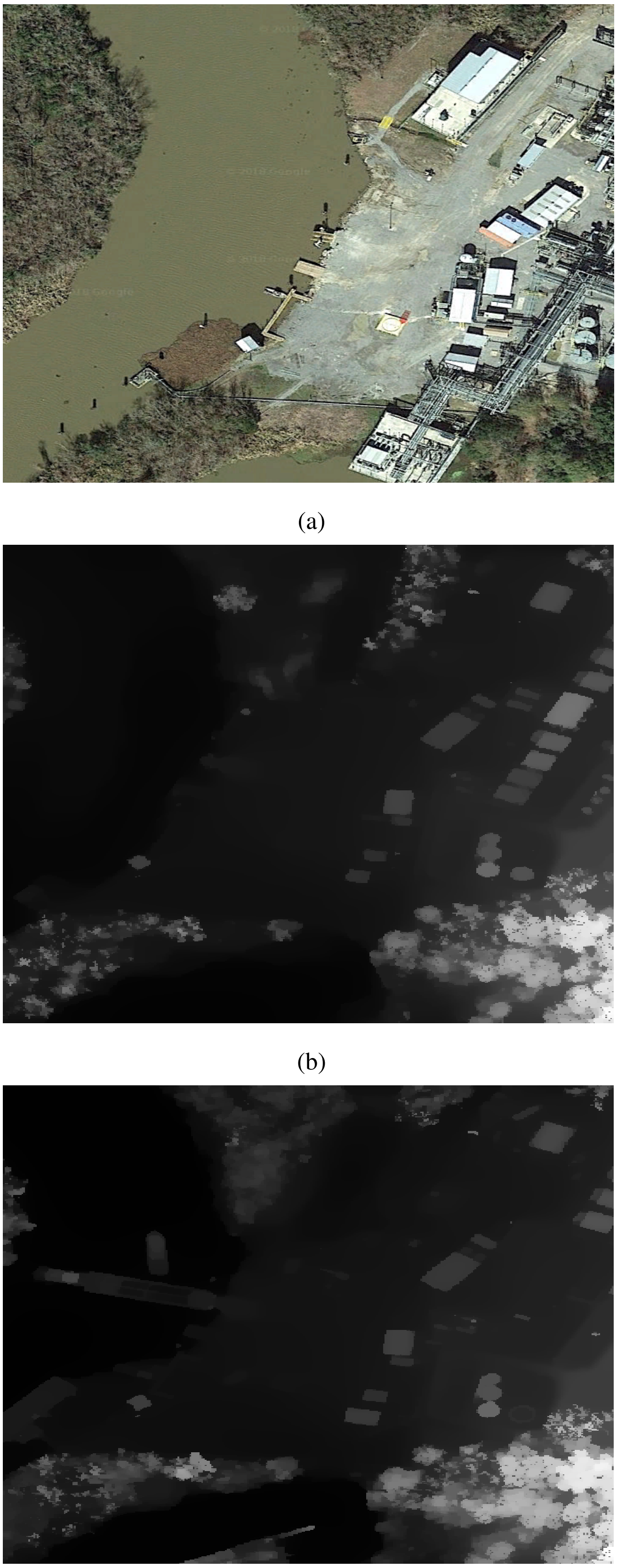


(a)



(b)

Figure 5.21: The result of the proposed method where smoothness level $\lambda_s = 1.0$ (a) RGB change result map (b) ROC curves for newly built and demolished objects

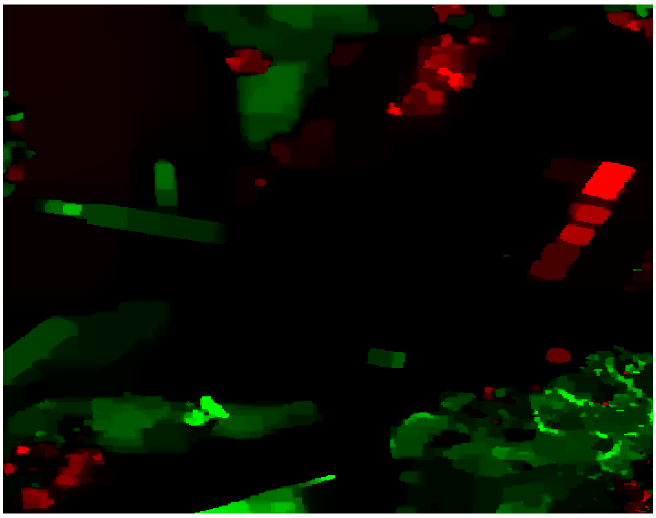


(a)

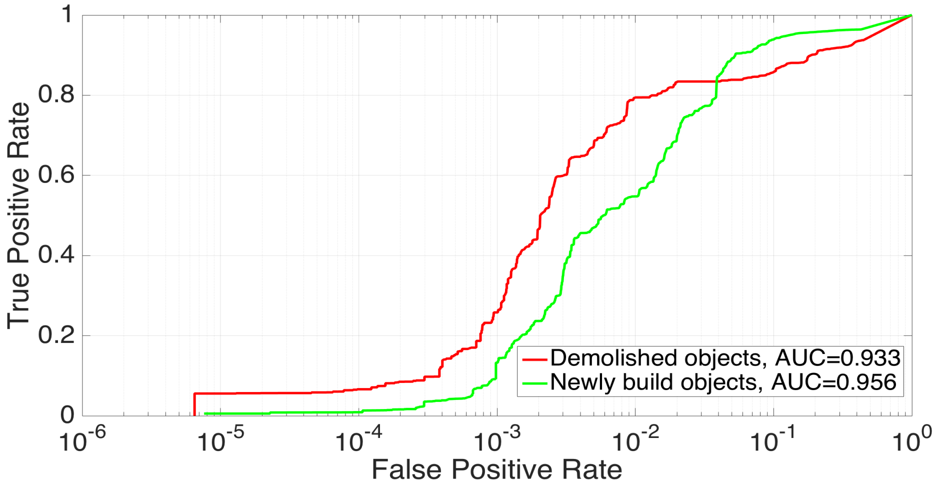
(b)

(c)

Figure 5.22: 2D images of another zone from Wax Lake region (a) Electro optic raster image (b) DSM, 2009 (c) DSM, 2013



(a)



(b)

Figure 5.23: The result of the proposed method where smoothness level $\lambda_s = 1.0$ (a) RGB change result map (b) ROC curves for newly built and demolished objects

The second set of real data does not have accurate location values so that this entire original data is geolocated to coordinates of Bilbao region, Spain. Then, DSM of the data is processed using Global Mapper application and saved as float elevation map with $1m$ resolution. Since the proposed method can use both the elevation map and point cloud data, in this set of tests DSMs are employed as inputs.

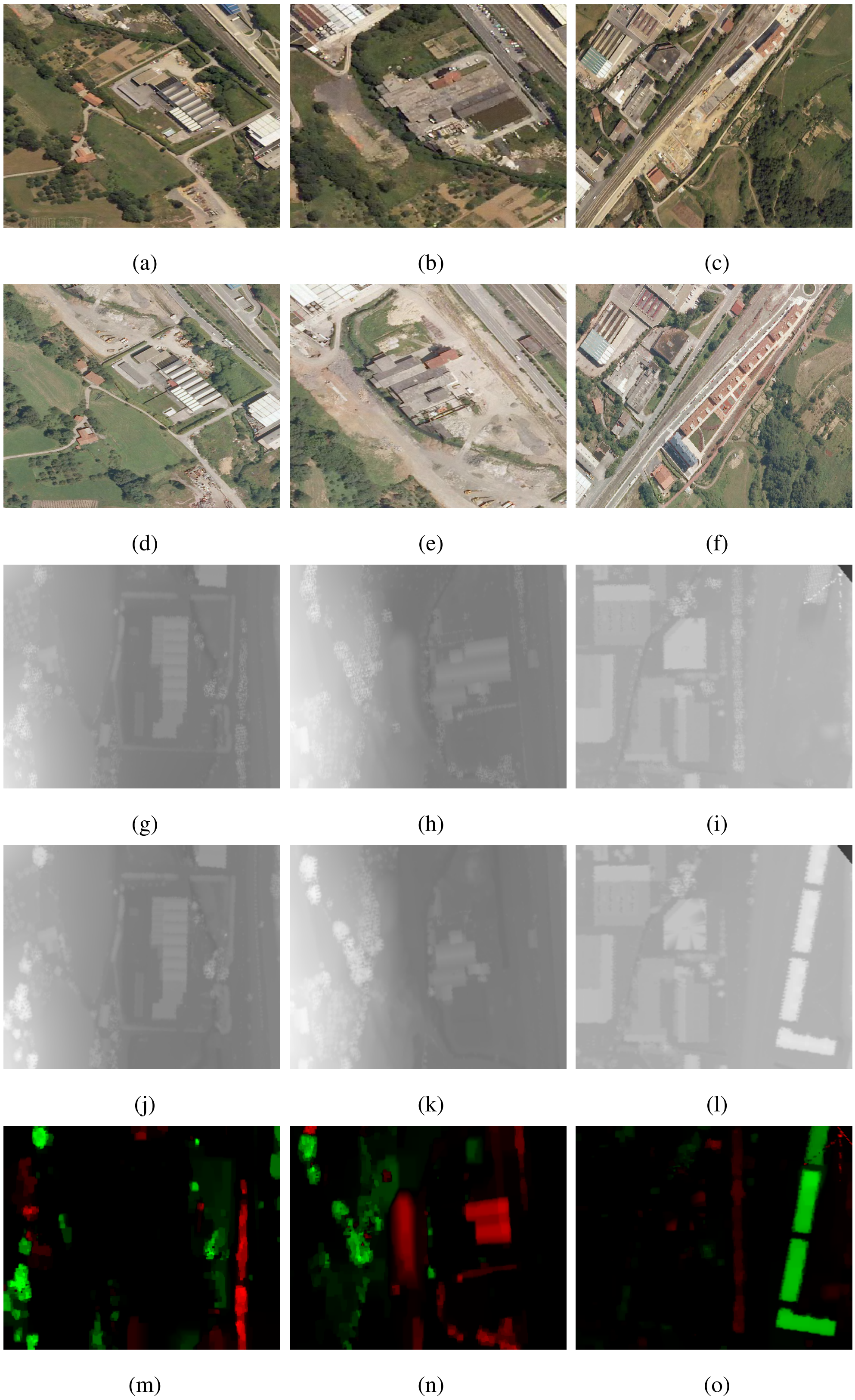
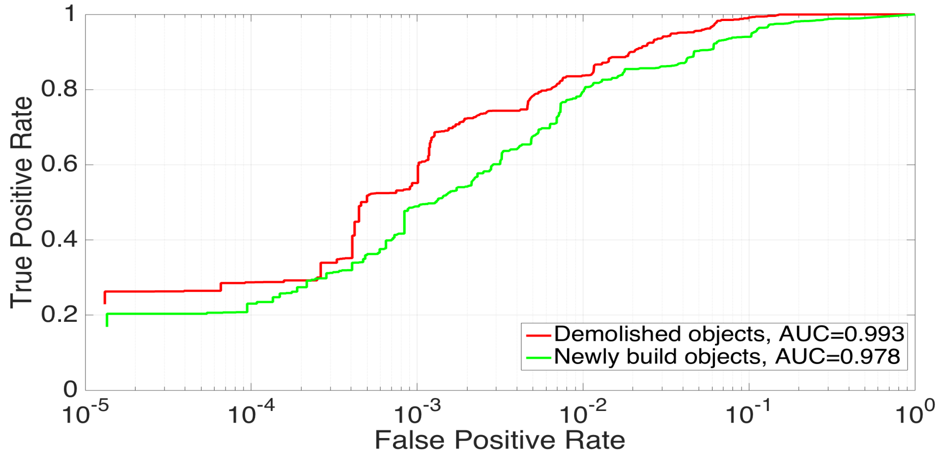
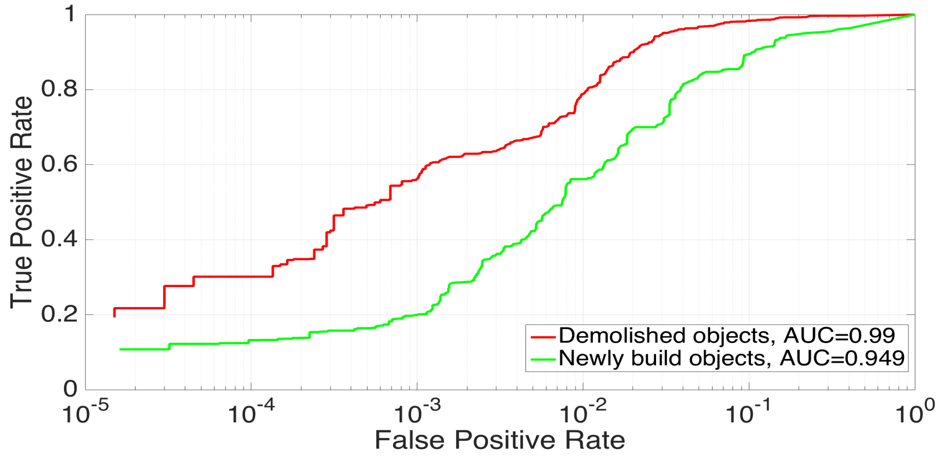


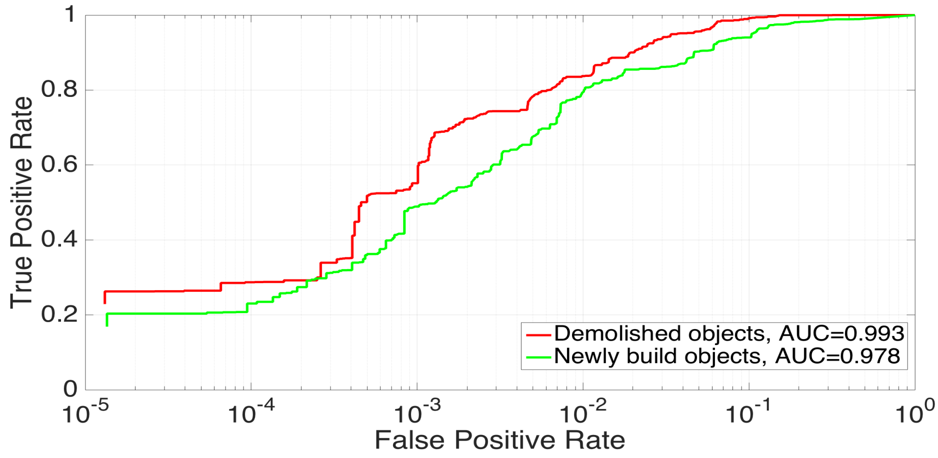
Figure 5.24: 2D images of Bilbao region and change results of the proposed method where $\lambda_s = 1.0$ (a-c) Raster images of 2005 (d-f) Raster images of 2008 (g-i) DSMs of 2005 (j-l) DSMs of 2008 (m-o) Change result maps



(a)



(b)



(c)

Figure 5.25: The ROC curves of the proposed method where smoothness level, $\lambda_s = 1.0$ (a) The first determined region patch (b) The second determined region patch (c) The third determined region patch

5.5 Testing Environment

All the tests are performed on a computer with the following specifications are listed in Table 5.5.

Table 5.5: Computer Specifications

Operating System	Mac OS (High Sierra)
Memory (RAM)	8GB of 1600MHz DDR3L onboard memory
Hard Disk	2.6GHz - 128GB128GB PCIe-based flash storage
Processor	2.6GHz dual-core Intel Core i5 processor
Graphics Card	Intel Iris 1536 MB

All the implementations are executed using Matlab version R2015b. Some results are visualized using Global Mapper toolkit.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 Thesis Sumary

In this study, a new change detection method, correlation-based variational change detection (CVCD), is proposed. This proposed method is constructed as an optimization problem for digital surface models. CVCD extracts the meaningful changes such as construction or destruction of buildings, forestation or deforestation by minimizing a correlation based variational cost function in a complete, robust and efficient way. Since the employed cost function has cross correlation based data fidelity term and ℓ_1-norm TV regularization term, while detecting these significant changes, it smoothes the result change map but preserves the features of changes such as point-changes and corners and edges of the changes. Furthermore; this proposed method deals with detecting changes as newly coming and demolished objects, separately. In order to solve this optimization problem, an efficient numerical minimization approach is applied to the strictly convex function. Hence, fast convergence to global minima is guaranteed. Performance of the method is shown quantitatively, using ROC curves, on synthetically generated 1D signals, 2D signals and a real-world elevation model. The method is, also, analyzed qualitatively, on real-world multi-temporal elevation masks, containing changes with different characteristics. The experiments reveal that proposed method generates satisfactory results.

6.2 Discussion and Future Work

Currently, since the Lidar sensors are very expensive and the planned flights are very costly, the acquisition of Lidar data is very overpriced. Especially since the second flight from a region is lavishness, the elevation map of the same region is not extracted for the second time. In the future, if the use of LIDAR sensors becomes widespread, a huge amount of data can be obtained cheaply. So that, this obtained data can be used for training process of deep learning algorithms. This provides an opportunity to increase the accuracy of the change detection algorithms and decreases the work load of humans.

REFERENCES

[1] Laser scanning vs digital photogrammetry. http://www.amerisurv.com/PDF/LaserScanningVsPhotogrammetry080108.pdf. Accessed: 2018-04-14.

[2] Why use photogrammetry for surveying and mapping? https://info.photomodeler.com/blog/uav-photogrammetry-for-surveying/. Accessed: 2018-04-14.

[3] The uses of lidar what applications are there for lidar systems, 2017.

[4] Abdullah Emin Akay, Hakan Oğuz, Ismail Rakip Karas, and Kazuhiro Aruga. Using lidar technology in forestry activities. *Environmental monitoring and assessment*, 151(1-4):117–125, 2009.

[5] Çağdaş Bak, Mustafa Ergül, Gizem Aktaş, and Fatih Nar. Dsm extraction using variational methods. In *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pages 265–268. IEEE, 2016.

[6] Emmanuel P Baltsavias. A comparison between photogrammetry and laser scanning. *ISPRS Journal of photogrammetry and Remote Sensing*, 54(2-3):83–94, 1999.

[7] Richard Bamler and Philipp Hartl. Synthetic aperture radar interferometry. *Inverse problems*, 14(4):R1, 1998.

[8] Yakoub Bazi and Farid Melgani. A variational level-set method for unsupervised change detection in remote sensing images. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 2, pages II–984. IEEE, 2009.

[9] S Belliss, S McNeill, J Barringer, D Pairman, and H North. Digital terrain modelling for exploration and mining. In *Proceedings on New Zealand Minerals & Mining Conference*, 2000.

[10] Lorenzo Bruzzone and Diego Fernandez Prieto. Automatic analysis of the difference image for unsupervised change detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(3):1171–1182, 2000.

[11] James B Campbell and Randolph H Wynne. *Introduction to remote sensing*. Guilford Press, 2011.

[12] Yin Chen, Armin B Cremers, and Zhiguo Cao. A variational change detection method for multitemporal sar images. *Remote Sensing Letters*, 5(4):342–351, 2014.

[13] Yan Chu, Guo Cao, and Hassan Hayat. Change detection of remote sensing image based on deep neural networks. 2016.

[14] Laurent Condat. A direct algorithm for 1-d total variation denoising. *IEEE Signal Processing Letters*, 20(11):1054–1057, 2013.

[15] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *preprint*, 93(1):2, 2011.

[16] Carol Deering and Jason Stoker. Let's agree on the casing of lidar, 2014.

[17] Carlos Felippa. Variational calculus overview, 2013.

[18] K Fletcher. European space agency, european space research and technology centre, estec (2007) insar principles: guidelines for sar interferometry processing and interpretation.

[19] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.

[20] Charles Fox. *An introduction to the calculus of variations*. Courier Corporation, 1950.

[21] Paolo Gamba and Bijan Houshmand. Digital surface models and building extraction: A comparison of ifsar and lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, 38(4):1959–1968, 2000.

[22] Jan Georgopoulos. Digital elevation models (dem): Terrain (dtm) or surface (dsm) data?, 2015.

[23] Hamid Gharibi. Extraction and improvement of digital surface models from dense point clouds. *Diss., Inst for Photogramm, Univ Stuttg*, 2012.

[24] Nicholas R Goodwin, Nicholas C Coops, and Darius S Culvenor. Assessment of forest structure with airborne lidar and the effects of platform altitude. *Remote Sensing of Environment*, 103(2):140–152, 2006.

[25] Jared Green. Six ways human activity is changing the planet, 2010.

[26] Vladimir Ignatiev, Alexey Trekin, Viktor Lobachev, Georgy Potapov, and Evgeny Burnaev. Targeted change detection in remote sensing images. *arXiv preprint arXiv:1803.05482*, 2018.

[27] Camille Jordan. Sur la série de fourier. *CR Acad. Sci. Paris*, 92(5):228–230, 1881.

[28] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[29] George Joseph. *Fundamentals of remote sensing*. Universities Press, 2005.

[30] Megan Lang, Greg McCarty, Bill Wilen, and Jane Awl. Light detection and ranging.

[31] Hyun S Lee and Nicolas H Younan. Dtm extraction of lidar returns via adaptive processing. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):2063–2069, 2003.

[32] Margarita Leonova. Laser scanning vs. photogrammetry, 2014.

[33] Thomas Lillesand, Ralph W Kiefer, and Jonathan Chipman. *Remote sensing and image interpretation*. John Wiley & Sons, 2014.

[34] Kevin Lim, Paul Treitz, Michael Wulder, Benoît St-Onge, and Martin Flood. Lidar remote sensing of forest structure. *Progress in physical geography*, 27(1):88–106, 2003.

[35] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. *ACM Transactions on Graphics (TOG)*, 25(3):646–653, 2006.

[36] Xiaoye Liu. Airborne lidar for dem generation: some critical issues. *Progress in Physical Geography*, 32(1):31–49, 2008.

[37] Dengsheng Lu, P Mausel, E Brondizio, and Emilio Moran. Change detection techniques. *International journal of remote sensing*, 25(12):2365–2401, 2004.

[38] Gabriele Moser, Sebastiano B Serpico, et al. Generalized minimum-error thresholding for unsupervised change detection from sar amplitude imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2972, 2006.

[39] V Nancharaiah, Sk Azeez, and N Raja Sekhar. Image smoothing by low gradient minimization.

[40] F. Nar, A. Ozgur, and N. Saran. Sparsity-driven change detection in multitemporal SAR images. *IEEE Geoscience and Remote Sensing Letters*, 13(7):1032–1036, 2016.

[41] Fatih Nar. SAR image despeckling using quadratic—linear approximated L1-norm. *Electronics Letters*, 54:387–389(2), March 2018.

[42] Fatih Nar, Erdal Yilmaz, and Gustau Camps-Valls. Sparsity-driven digital terrain model extraction. In *IGARSS'2018*, Accepted.

[43] Caner Ozcan, Baha Sen, and Fatih Nar. Sparsity-driven despeckling for sar images. *IEEE Geoscience and Remote Sensing Letters*, 13(1):115–119, 2016.

[44] Fabio Pacifici. Change detection algorithms: State of the art. *URL: http://www. disp. uniroma2. it/earth_observation/pdf/CD-Algorithms. pdf (Accessed on 4 November, 2011)*, 2007.

[45] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[46] Filip O.L. Popescu C., Dragomir L. Airborne lidar topographical surveying. In *15th International Multidisciplinary Scientific GeoConference SGEM 2015*, volume 1, pages 949–956. SGEM2015 Conference Proceedings, 2015.

[47] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: a systematic survey. *IEEE transactions on image processing*, 14(3):294–307, 2005.

[48] Jerry C Ritchie. Remote sensing applications to hydrology: airborne laser altimeters. *Hydrological Sciences Journal*, 41(4):625–636, 1996.

[49] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[50] Gabriela R. Sanchis. Historical activities for calculus - module 3: Optimization – galileo and the brachistochrone problem, 2014.

[51] Karen Schuckman and Mike Renslow. History of lidar development, 2017.

[52] Ashbindu Singh. Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6):989–1003, 1989.

[53] Andrey Soloviev. Tight coupling of gps, laser scanner, and inertial measurements for navigation in urban environments. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 511–525. IEEE, 2008.

[54] LAS Specification. Version 1.4-r13, 2006.

[55] Cornelis Stal, Frederik Tack, Philippe De Maeyer, Alain De Wulf, and Rudi Goossens. Airborne photogrammetry and lidar for dsm extraction and 3d change detection over an urban area–a comparative study. *International Journal of Remote Sensing*, 34(4):1087–1110, 2013.

[56] Michael J. Starek. Adaptive signal processing laboratory, 2004.

[57] Juan C Suárez, Carlos Ontiveros, Steve Smith, and Stewart Snape. Use of airborne lidar and aerial photography in the estimation of individual tree heights in forestry. *Computers & Geosciences*, 31(2):253–262, 2005.

[58] Fawwaz T Ulaby, Richard K Moore, and Adrian K Fung. Microwave remote sensing active and passive. 2015.

[59] Markus Unger, Thomas Pock, Markus Grabner, Andreas Klaus, and Horst Bischof. A variational approach to semiautomatic generation of digital terrain models. In *International Symposium on Visual Computing*, pages 1119–1130. Springer, 2009.

[60] Maria Vakalopoulou, Konstantinos Karantzalos, Nikos Komodakis, and Nikos Paragios. Simultaneous registration and change detection in multitemporal, very high resolution remote sensing data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 61–69, 2015.

[61] Tuong Thuy Vu, Masashi Matsuoka, and Fumio Yamazaki. Lidar-based change detection of buildings in dense urban areas. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 5, pages 3413–3416. IEEE, 2004.

[62] Aloysius Wehr and Uwe Lohr. Airborne laser scanning—an introduction and overview. *ISPRS Journal of photogrammetry and remote sensing*, 54(2):68–82, 1999.

[63] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM Transactions on Graphics (TOG)*, 31(6):139, 2012.

[64] Xiaowei Yu, Juha Hyyppä, Mika Karjalainen, Kimmo Nurminen, Kirsi Karila, Mikko Vastaranta, Ville Kankare, Harri Kaartinen, Markus Holopainen, Eija Honkavaara, et al. Comparison of laser and stereo optical, sar and insar point clouds from air-and space-borne sources in the retrieval of forest inventory attributes. *Remote Sensing*, 7(12):15933–15954, 2015.

[65] Xiaohua Zhang, Jiawei Chen, and Hongyun Meng. A novel sar image change detection based on graph-cut and generalized gaussian model. *Geoscience and Remote Sensing Letters, IEEE*, 10(1):14–18, 2013.

[66] Jiaojiao Zhao, Maoguo Gong, Jia Liu, and Licheng Jiao. Deep learning to classify difference image for image change detection. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 411–417. IEEE, 2014.